

**DETEKSI HOAKS UNTUK POSTINGAN ARTIKEL TULISAN  
BERBAHASA INDONESIA BERBASIS TOPIK *MODELLING*  
MENGGUNAKAN ALGORITMA *RANDOM FOREST***

**SKRIPSI**

**Oleh :**  
**QONITA NASHIFA ANABILA**  
**NIM. 220605110135**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

**DETEKSI HOAKS UNTUK POSTINGAN ARTIKEL TULISAN  
BERBAHASA INDONESIA BERBASIS TOPIK *MODELLING*  
MENGGUNAKAN ALGORITMA *RANDOM FOREST***

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

**Oleh:**  
**QONITA NASHIFA ANABILA**  
**NIM. 220605110135**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM**  
**MALANG**  
**2025**

## HALAMAN PERSETUJUAN

### DETEKSI HOAKS UNTUK POSTINGAN ARTIKEL TULISAN BERBAHASA INDONESIA BERBASIS TOPIK *MODELLING* MENGGUNAKAN ALGORITMA *RANDOM FOREST*

## SKRIPSI

Oleh :  
**QONITA NASHIFA ANABILA**  
**NIM. 220605110135**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 8 Desember 2025

Pembimbing I,



Okta Qomaruddin Aziz, M.Kom  
NIP. 199110192019031013

Pembimbing II,



Dr. Cahyo Crisdian, M.Cs  
NIM. 197404242009011008

Mengetahui

Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom

NIP. 19841010 201903 1 012

## HALAMAN PENGESAHAN

### DETEKSI HOAKS UNTUK POSTINGAN ARTIKEL TULISAN BERBAHASA INDONESIA BERBASIS TOPIK *MODELLING* MENGGUNAKAN ALGORITMA *RANDOM FOREST*

#### SKRIPSI

Oleh :  
QONITA NASHIFA ANABILA  
NIM. 220605110135

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal: 19 Desember 2025

#### Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. Irwan Santoso, M.Kom</u> NIP. 19770103 201101 1 004
Anggota Penguji I	: <u>Nurizal Dwi Priandani, M.Kom</u> NIP. 199208302022031001
Anggota Penguji II	: <u>Okta Qomaruddin Aziz, M.Kom</u> NIP. 199110192019031013
Anggota Penguji III	: <u>Dr. Cahyo Crysdiyan, M.Cs</u> NIP. 19740424 200901 1 008

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika

Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom

NIP. 19841010 201903 1 012

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Qonita Nashifa Anabila  
NIM : 220605110135  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Deteksi Hoaks Untuk Postingan Artikel Tulisan Berbahasa Indonesia Berbasis Topik *Modelling* Menggunakan Algoritma *Random Forest*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 23 Desember 2025  
Yang membuat pernyataan,



Qonita Nashifa Anabila  
NIM.220605110135

## MOTTO

*“Toman tabhâlâccar sakalèyan bhâkal tao carana ongghâ ka sè lebbih tengghi,  
Kađhâng pajhât ambhu lopot sopajâ ajhâr dhâddhi orèng pènter.”*

إِنَّ مَعَ الْعُسْرِ يُسْرًا

*“Sesungguhnya setelah kesulitan itu ada kemudahan.”*

*“Finish what you’ve started.”*

## HALAMAN PERSEMBAHAN

Dengan rasa syukur yang mendalam kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-nya, serta taburan cinta dan kasih sayang-nya yang telah memberikan kekuatan, membekali penulis dengan ilmu. Atas karunia dan kemudahan yang diberikan, akhirnya skripsi ini dapat terselesaikan. Sholawat dan salam senantiasa terlimpahkan kepada Rasulullah SAW.

Penulis persembahkan karya ini sebagai tanda bakti, hormat, dan rasa terima kasih yang tiada terhingga kepada orang yang sangat berharga dalam hidup penulis, yaitu papa tercinta dan tersayang (Samsul), ibu tercinta dan tersayang (Rohaniyah), serta nenek tercinta dan tersayang (Buyani dan Raode) yang senantiasa memberikan motivasi dan kasih sayang yang tidak mungkin dapat dibalas hanya dengan selembar kertas yang bertuliskan kata cinta dan persembahan. Kepada papa, mama dan nenek yang selalu memotivasi, selalu mendo'akan, selalu menasehati agar menjadi pribadi yang lebih baik, serta dengan penuh kesabaran membimbing penulis hingga saat ini.

Tak lupa, untuk adik tercinta, Riyanda Baroki yang selalu mengalah, tulus memberi dukungan, dan menjadi penyemangat dalam setiap langkah penulis. Semoga Allah SWT senantiasa melimpahkan kemudahan, melapangkan jalan pendidikannya, serta mengabulkan doa-doa terbaik agar kelak dapat menempuh bangku perkuliahan dan menyelesaikan setiap proses kehidupan dengan penuh keberkahan dan kesuksesan.

## KATA PENGANTAR

*Bismillahirrahmaanirrahiim, Assalamu'alaikum wr. Wb*

*Alhamdulillahi Rabbil 'Alamin*, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan inayah-nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Deteksi Hoaks untuk Postingan Artikel Tulisan Berbahasa Indonesia Berbasis Topik *Modelling* Menggunakan Algoritma *Random Forest*.” Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang penuh ilmu pengetahuan serta cahaya *ad-din-al-islam*.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Dalam proses penyusunannya, penulis banyak menerima dukungan, doa, semangat, serta bantuan dari berbagai pihak yang sangat berarti. Oleh karena itu, dengan penuh rasa hormat dan ketulusan, penulis menyampaikan ucapan terima kasih kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Okta Qomaruddin Aziz, M.Kom., selaku dosen pembimbing I, yang telah meluangkan waktu dengan penuh kesabaran dalam membimbing penulis

selama proses penyusunan skripsi ini sehingga penulis dapat menyelesaiannya dengan baik. Di setiap keterbatasan, kebingungan, dan keraguan yang dialami penulis, beliau senantiasa memberikan arahan, menenangkan, serta membuka jalan melalui ilmu dan pengalaman yang dimiliki. Waktu dan perhatian yang dicurahkan di tengah kesibukan akademik merupakan bentuk pengabdian yang sangat berarti bagi penulis. Setiap masukan dan koreksi yang diberikan tidak hanya membantu menyempurnakan skripsi ini, tetapi juga menjadi pelajaran berharga yang akan penulis ingat dan bawa sepanjang perjalanan hidup. Semoga segala kebaikan dan ilmu yang telah diberikan menjadi amal jariyah yang terus mengalir.

5. Dr. Cahyo Crysdiyan, M.Cs., selaku dosen pembimbing II, yang telah memberikan bimbingan, saran, dan pandangan akademik yang sangat berharga dalam proses penyusunan skripsi ini. Melalui arahan, masukan, dan sudut pandang yang diberikan, penulis memperoleh pemahaman yang lebih mendalam dalam menyempurnakan penelitian ini, baik dari segi konsep, metode, maupun alur penulisan. Setiap saran yang disampaikan menjadi landasan perbaikan yang membantu penulis berpikir lebih kritis dan sistematis sehingga skripsi ini dapat diselesaikan dengan baik.
6. Dr. Irwan Budi Santoso, M.Kom., selaku ketua penguji dan Nurizal Dwi Priandani, M.Kom., dosen penguji I yang telah menguji dan memberikan saran dalam menyelesaikan skripsi ini dengan baik.

7. Nia Faricha S, Si., selaku admin Program Studi Teknik Informatika yang dengan penuh kesabaran selalu memberikan informasi, membantu proses administrasi, serta memberikan arahan selama masa perkuliahan hingga penyusunan skripsi ini.
8. Seluruh Dosen, Laboran dan jajaran Staf Program Studi Teknik Informatika yang telah memberikan banyak bantuan selama studi ini.
9. Cinta pertama bagi penulis yaitu papa Samsul yang selalu memberikan doa, dukungan, serta motivasi tanpa henti kepada penulis. Di tengah jarak dan keterbatasan selama penulis menjalani kehidupan sebagai anak rantaui, papa selalu hadir melalui doa yang menguatkan langkah penulis. Kesabaran, nasihat, dan ketulusan yang diberikan menjadi kekuatan besar bagi penulis dalam menjalani proses perkuliahan hingga akhirnya mampu menyelesaikan skripsi ini. Setiap perjuangan dan pengorbanan papa menjadi pengingat bagi penulis untuk terus berusaha, bertanggung jawab, dan tidak mudah menyerah dalam menghadapi setiap proses kehidupan.
10. Mama tercinta yaitu Rohaniyah yang tanpa lelah mendampingi penulis melalui doa, nasihat, dan kasih sayang dalam setiap tahap perkuliahan. Ketulusan dan kesabaran yang diberikan menjadi kekuatan yang menenangkan dan menguatkan penulis dalam menghadapi berbagai tantangan hingga akhirnya mampu menyelesaikan skripsi dengan sebaik-baiknya.
11. Nenek tersayang yaitu Buyani dan Raode yang senantiasa mengiringi langkah penulis dengan doa, perhatian, dan kasih sayang yang tulus. Sejak

masa kecil hingga penulis menempuh pendidikan di bangku perkuliahan, kehadiran nenek menjadi pengingat bagi penulis untuk selalu bersyukur, sabar, dan terus berusaha dalam setiap proses kehidupan. Nasihat yang sederhana namun penuh makna, serta doa-doa yang tidak pernah terputus, menjadi penopang bagi penulis dalam menghadapi berbagai tantangan hingga akhirnya skripsi ini dapat diselesaikan dengan baik.

12. Adikku tersayang Riyanda Baroki, yang dengan ketulusan hati selalu mengalah, menunjukkan kesabaran, dan menjadi sosok pengingat bagi penulis dalam menjaga nilai-nilai agama. Sikapnya yang tenang, sabar, dan penuh pengertian sering kali menjadi penguat bagi penulis dalam menghadapi berbagai proses selama perkuliahan. Nasihat sederhana yang disampaikan dengan ketulusan hati menjadi pengingat bagi penulis untuk tetap melangkah di jalan yang baik. Penulis senantiasa mendoakan semoga Allah SWT melimpahkan kemudahan, kelancaran, serta kesuksesan dalam setiap langkah hidupnya, dan membimbingnya untuk meraih masa depan yang baik dan penuh keberkahan.
13. Almarhum kakek Tahir dan Asnawi, yang menjadi bagian penting dalam perjalanan hidup penulis. Kasih sayang, nasihat, dan keteladanan yang pernah diberikan akan selalu penulis kenang dan jadikan pegangan dalam menjalani kehidupan, termasuk dalam menyelesaikan pendidikan hingga tahap akhir ini. Penulis senantiasa memanjatkan doa semoga Allah SWT mengampuni dosa-dosanya, menerima seluruh amal ibadahnya, dan memberikan tempat terbaik di sisi-nya.

14. Keluarga besar penulis, yang dengan penuh ketulusan selalu mengiringi perjalanan penulis melalui doa dan dukungan yang tak pernah putus. Perhatian dan kebersamaan yang diberikan menjadi penguat hati dan penyemangat bagi penulis dalam menghadapi berbagai tantangan selama perkuliahan hingga akhirnya dapat menyelesaikan skripsi ini. Semoga Allah SWT senantiasa melimpahkan kebaikan, keberkahan, dan kebahagiaan kepada seluruh keluarga besar penulis.
15. Teman-teman satu kamar penulis di Ma'had “*Cutie 20*”, yaitu Mirza, Fadila, Firdha, Fatima, Dini, Latifa, Via, Melsa, dan Azza, yang tidak hanya menjadi rekan tinggal, tetapi juga telah penulis anggap sebagai saudara selama berada di Ma'had. Kebersamaan yang terjalin melalui berbagai cerita, tawa, serta saling menguatkan dalam keseharian menjadi kenangan berharga bagi penulis. Di tempat inilah penulis belajar tentang arti kebersamaan, kesabaran, dan saling memahami, yang turut membentuk pribadi penulis hingga saat ini. Semoga kebersamaan dan kebaikan yang telah terjalin senantiasa membawa keberkahan bagi kita semua.
16. Dita Suci Yofana, sosok sahabat bagi penulis yang bukan sekadar teman, melainkan saudara dan keluarga yang Allah hadirkan selama masa perantauan dan perkuliahan. Di tengah jauhnya penulis dari rumah, kehadiranmu dengan ketulusan, kepedulian, dan kebaikan hati menjadi tempat pulang, tempat berbagi cerita, serta penguat di saat penulis berada pada titik lelah dan hampir menyerah. Bantuan, perhatian, serta dukungan yang diberikan selama proses penyusunan skripsi ini merupakan bentuk

kebaikan yang tidak mungkin dapat penulis balas dengan apa pun selain doa dan rasa syukur yang mendalam. Berkat kehadiran dan peranmu, penulis mampu bertahan hingga akhirnya menyelesaikan skripsi ini. Semoga Allah SWT membalas setiap kebaikanmu dengan balasan terbaik, melapangkan jalan hidupmu, serta mengiringi setiap langkahmu dengan keberkahan dan kesuksesan.

17. Firna, dengan panggilan tersayangnya yaitu Pino, gadis Papua sahabat penulis yang selalu hadir dengan kebaikan hati dan keceriaannya. Di tengah lelah dan tekanan selama masa perkuliahan, kehadiranmu dengan sikap yang kocak namun penuh kepedulian menjadi penyemangat tersendiri bagi penulis. Kehadiran, bantuan, dan kepedulianmu menjadi kekuatan besar bagi penulis untuk terus bertahan di tengah proses yang tidak selalu mudah. Semoga Allah SWT membalas setiap kebaikanmu dengan kemudahan, kebahagiaan, dan kesuksesan dalam setiap langkah kehidupan.
18. Aisyah Nur Fitriyah, sahabat penulis dengan kebaikan hati dan kepeduliannya. Sikap suka membantu serta kepedulian yang tulus menjadi bagian penting dalam perjalanan perkuliahan penulis. Kehadiran dan dukungan yang diberikan, meskipun sering hadir dalam bentuk perhatian sederhana, memberikan kekuatan tersendiri bagi penulis untuk terus bertahan dan melangkah. Semoga Allah SWT membalas setiap kebaikanmu dengan keberkahan, kemudahan dalam setiap urusan, serta kebahagiaan dan kesuksesan di masa yang akan datang.

19. Radifan Roihanul Firqi, sahabat penulis yang hadir di waktu yang tepat dan menjadi bagian penting dalam proses penyelesaian skripsi ini. Kesediaanmu untuk selalu hadir, membantu dengan penuh kesabaran pada setiap proses penyusunan skripsi ini menjadi kekuatan yang tidak ternilai bagi penulis. Di tengah perjalanan yang panjang dan melelahkan, ketulusanmu membantu penulis untuk tetap bertahan untuk menyelesaikan skripsi ini hingga akhir. Semoga Allah SWT membalas setiap kebaikanmu dengan keberkahan ilmu, dan kemudahan dalam setiap langkah dan urusan.
20. Moh. Musa Al Kadzim, sahabat penulis yang dikenal sebagai juragan *sparepart* yang selalu hadir membawa energi positif bagi penulis dan orang-orang di sekitarnya. Meski dihadapkan pada berbagai kesibukan dan tanggung jawab, kehadirannya dengan canda, tawa, dan sikap ringan mampu mencairkan suasana. Dari kebersamaan tersebut, penulis memahami bahwa kekuatan tidak selalu lahir dari sikap keras, tetapi juga dari kemampuan untuk terus menghadirkan senyum dan keceriaan yang memberi kehangatan bagi sekitar.
21. Mirza Aulia Al-Furqany dan Fadilah Mutammima, sahabat penulis yang selalu hadir dengan kebaikan dan ketulusan selama masa perantauan. Terima kasih telah menjadi tempat pulang sementara bagi penulis, dengan lapang hati selalu membuka pintu, menyediakan tempat untuk beristirahat, serta menerima penulis untuk menginap dan berbagi kebersamaan. Kamar yang menjadi ruang kebersamaan tersebut menjadi saksi berbagai cerita, tawa, lelah, dan obrolan panjang yang kerap menjadi penguat di tengah

proses perkuliahan yang tidak selalu mudah. Dari kebersamaan ini, penulis merasakan arti keluarga yang tidak selalu terikat oleh hubungan darah, melainkan tumbuh dari ketulusan, kepedulian, dan rasa saling menguatkan. Semoga Allah SWT membalas setiap kebaikan yang telah diberikan dengan keberkahan hidup, kemudahan dalam setiap urusan, serta kebahagiaan yang senantiasa menyertai langkah ke depan.

22. Muhammad Abel Firmansyah Pratama, sosok yang telah penulis anggap sebagai saudara. Kehadirannya dengan berbagai masukan, saran, serta motivasi yang diberikan dengan tulus menjadi penguat bagi penulis dalam menjalani masa perkuliahan. Teladan yang ditunjukkan melalui sikap, pemikiran, dan cara bersikap menjadi contoh yang baik serta memberikan pelajaran berharga bagi penulis. Dukungan dan arahan yang diberikan membantu penulis untuk terus berkembang, berpikir lebih baik, dan melangkah dengan keyakinan dalam setiap proses. Semoga setiap kebaikan yang telah diberikan mendapatkan balasan terbaik, dilapangkan langkahnya, serta selalu diberikan kemudahan dan kebaikan dalam setiap jalan yang ditempuh.
23. Teman-teman seperjuangan dengan dosen pembimbing yang sama, yaitu Atsila, Anissa, Yoza, Gavrila, Kila, dan Iim, yang telah membersamai penulis dalam setiap proses bimbingan. Kebersamaan, saling berbagi cerita, serta dukungan yang diberikan satu sama lain menghadirkan suasana yang hangat dan penuh semangat, khususnya selama berada di laboratorium. Kehadiran rekan-rekan seperjuangan ini menjadikan setiap proses yang

dilalui terasa lebih ringan karena dijalani bersama dengan rasa saling menguatkan.

24. Teman-teman “Infinity” Teknik Informatika angkatan 2022 yang telah membersamai penulis sepanjang masa studi, menghadirkan kebersamaan, dukungan, dan semangat, serta meninggalkan kenangan berharga yang akan selalu dikenang dalam perjalanan akademik ini.
25. Seluruh pihak yang telah memberikan bantuan, perhatian, dan kebaikan dalam berbagai bentuk selama proses penyusunan skripsi ini, yang tidak dapat disebutkan satu per satu, penulis menyampaikan terima kasih yang sebesar-besarnya.
26. Untuk diri sendiri, terima kasih telah bertahan hingga sejauh ini. Terima kasih karena tidak menyerah meskipun sering merasa lelah, ragu, dan berada di titik terendah selama proses perkuliahan dan penyusunan skripsi. Setiap langkah yang dilalui, setiap air mata, dan setiap usaha yang dilakukan menjadi bukti bahwa penulis telah berjuang dengan sebaik mungkin. Semoga pencapaian ini menjadi pengingat bahwa segala proses, seberat apa pun, akan selalu menemukan akhirnya. Semoga ke depan penulis senantiasa diberikan kekuatan, keikhlasan, dan keberanian untuk terus belajar, bertumbuh, dan melangkah menuju masa depan yang lebih baik dengan penuh rasa syukur.

Malang, 19 Desember 2025

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PENGAJUAN.....</b>	<b>ii</b>
<b>HALAMAN PERSETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN.....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>HALAMAN PERSEMBERAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI .....</b>	<b>xvii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xix</b>
<b>DAFTAR TABEL .....</b>	<b>xxiii</b>
<b>ABSTRAK.....</b>	<b>xxvii</b>
<b>ABSTRACT.....</b>	<b>xxviii</b>
<b>مستخلص البحث.....</b>	<b>xxix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Pernyataan Masalah .....	8
1.3 Batasan Masalah .....	8
1.4 Tujuan Penelitian .....	8
1.5 Manfaat Penelitian .....	8
<b>BAB II STUDI PUSTAKA.....</b>	<b>10</b>
2.1 Deteksi <i>Hoax</i> .....	10
2.2 Deteksi Hoaks Berbasis Aspek .....	18
2.2.1 Aspek Kesehatan .....	18
2.2.2 Aspek Politik .....	20
2.2.3 Aspek Bencana Alam .....	21
2.2.4 Aspek Keagamaan .....	23
2.2.5 Aspek Pendidikan.....	24
2.3 Ekstraksi Fitur <i>Bag-Of-Words</i> .....	27
2.4 Algoritma <i>Random Forest</i> .....	27
<b>BAB III DESAIN DAN IMPLEMENTASI SISTEM.....</b>	<b>33</b>
3.1 Pengumpulan Data .....	33
3.2 Desain Sistem .....	37
3.3 <i>Preprocessing</i> .....	38
3.3.1 <i>Cleaning Data</i> .....	38
3.3.2 <i>Case Folding</i> .....	39
3.3.3 Normalisasi.....	40
3.3.4 <i>Tokenizing</i> .....	40
3.3.5 <i>Stopwords Removal</i> .....	41
3.3.6 <i>Stemming</i> .....	42
3.4 <i>Bag Of Words</i> .....	43
3.5 Algoritma <i>Random Forest</i> .....	47
<b>BAB IV UJI COBA DAN PEMBAHASAN.....</b>	<b>58</b>
4.1 Skenario Pengujian .....	58
4.1.1 <i>Split Data</i> .....	59

4.1.2 Menghitung Kinerja Sistem .....	60
4.1.3 <i>K-Fold Cross Validation</i> .....	64
4.2 Hasil Training .....	65
4.3 Hasil Pengujian .....	66
4.3.1 Hasil Skenario Uji 1 .....	67
4.3.2 Hasil Skenario Uji 2 .....	89
4.4 Pembahasan .....	148
4.4.1 Rangkuman Hasil Evaluasi Model .....	149
4.4.2 Pengaruh Jumlah <i>n_estimator</i> pada Metode <i>Random Forest</i> .....	152
4.4.3 Pengaruh <i>Max_depth</i> pada <i>Random Forest</i> .....	154
4.5 Perbandingan <i>SingleStage</i> dan <i>Multistage Classification</i> .....	157
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>161</b>
5.1 Kesimpulan .....	161
5.2 Saran .....	162

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## DAFTAR GAMBAR

Gambar 2. 1	Alur Metode <i>Random Forest</i> .....	31
Gambar 3. 1	Desain Sistem .....	37
Gambar 3. 2	<i>Flowchart Bag Of Word</i> .....	44
Gambar 3. 3	Desain Algoritma <i>Random Forest</i> .....	48
Gambar 4. 1	<i>Source Code Split Data</i> .....	60
Gambar 4. 2	Ilustrasi <i>5-Fold Cross Validation</i> .....	64
Gambar 4. 3	Visualiasi Salah Satu Pohon Keputusan <i>Random Forest</i> .....	65
Gambar 4. 4	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 50-10 .....	67
Gambar 4. 5	Boxplot Performa <i>Singlestage Hyperparameter</i> 50-10.....	68
Gambar 4. 6	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 50-20 .....	70
Gambar 4. 7	Boxplot Performa <i>Singlestage Hyperparameter</i> 50-20.....	71
Gambar 4. 8	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 50-30 .....	72
Gambar 4. 9	Boxplot Performa <i>Singlestage Hyperparameter</i> 50-30.....	73
Gambar 4. 10	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 75-10 .....	75
Gambar 4. 11	Boxplot Performa <i>Singlestage Hyperparameter</i> 75-10.....	76
Gambar 4. 12	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 75-20 .....	77
Gambar 4. 13	Boxplot Performa <i>Singlestage Hyperparameter</i> 75-20.....	78
Gambar 4. 14	Perbandingan Akurasi <i>Singlestage Kombinasi</i> 75-30 .....	80
Gambar 4. 15	Boxplot Performa <i>Singlestage Hyperparameter</i> 75-30.....	81
Gambar 4. 16	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 100-10 .....	82
Gambar 4. 17	Boxplot Performa <i>Singlestage Hyperparameter</i> 100-10.....	83
Gambar 4. 18	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 100-20 .....	84
Gambar 4. 19	Boxplot Performa <i>Singlestage Hyperparameter</i> 100-20.....	86
Gambar 4. 20	Perbandingan Akurasi <i>Singlestage Hyperparameter</i> 100-30 .....	87
Gambar 4. 21	Boxplot Performa <i>Singlestage Hyperparameter</i> 100-30.....	88
Gambar 4. 22	Perbandingan Akurasi <i>Multistage Stage 1 Hyperparameter</i> 50-10 .....	90
Gambar 4. 23	Boxplot Performa <i>Multistage Stage 1 Hyperparameter</i> 50-10 .....	91
Gambar 4. 24	Perbandingan Akurasi <i>Multistage Stage 1 Hyperparameter</i> 50-20 .....	92
Gambar 4. 25	Boxplot Performa <i>Multistage Stage 1 Hyperparameter</i> 50-20 .....	93
Gambar 4. 26	Perbandingan Akurasi <i>Multistage Stage 1 Hyperparameter</i> 50-30 .....	94
Gambar 4. 27	Boxplot Performa <i>Multistage Stage 1 Hyperparameter</i> 50-30 .....	95
Gambar 4. 28	Perbandingan Akurasi <i>Multistage Stage 1 Hyperparameter</i> 75-10 .....	96
Gambar 4. 29	Boxplot Performa <i>Multistage Stage 1 Hyperparameter</i> 75-10 .....	97
Gambar 4. 30	Perbandingan Akurasi <i>Multistage Stage 1 Hyperparameter</i> 75-20 .....	98

Gambar 4. 31 Boxplot Performa <i>Multistage Stage 1</i> <i>Hyperparameter 75-20</i> .....	99
Gambar 4. 32 Perbandingan Akurasi <i>Multistage Stage 1</i> <i>Hyperparameter 75-30</i> .....	101
Gambar 4. 33 Boxplot Performa <i>Multistage Stage 1</i> <i>Hyperparameter 75-30</i> .....	102
Gambar 4. 34 Perbandingan Akurasi <i>Multistage Stage 1</i> <i>Hyperparameter 100-10</i> .....	103
Gambar 4. 35 Boxplot Performa <i>Multistage Stage 1</i> <i>Hyperparameter 100-10</i> .....	104
Gambar 4. 36 Perbandingan Akurasi <i>Multistage Stage 1</i> <i>Hyperparameter 100-20</i> .....	105
Gambar 4. 37 Boxplot Performa <i>Multistage Stage 1</i> <i>Hyperparameter 100-20</i> .....	106
Gambar 4. 38 Perbandingan Akurasi <i>Multistage Stage 1</i> <i>Hyperparameter 100-30</i> .....	107
Gambar 4. 39 Boxplot Performa <i>Multistage Stage 1</i> <i>Hyperparameter 100-30</i> .....	108
Gambar 4. 40 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 50-10</i> .....	109
Gambar 4. 41 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 50-10</i> .....	110
Gambar 4. 42 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 50-20</i> .....	111
Gambar 4. 43 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 50-20</i> .....	112
Gambar 4. 44 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 50-30</i> .....	114
Gambar 4. 45 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 50-30</i> .....	115
Gambar 4. 46 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 75-10</i> .....	116
Gambar 4. 47 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 75-10</i> .....	116
Gambar 4. 48 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 75-20</i> .....	118
Gambar 4. 49 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 75-20</i> .....	119
Gambar 4. 50 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 75-30</i> .....	120
Gambar 4. 51 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 75-30</i> .....	121
Gambar 4. 52 Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 100-10</i> .....	122
Gambar 4. 53 Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 100-10</i> .....	123

Gambar 4. 54	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 100-20</i> .....	124
Gambar 4. 55	Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 100-20</i> .....	125
Gambar 4. 56	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 100-30</i> .....	126
Gambar 4. 57	Boxplot Performa <i>Multistage Stage 2</i> Aspek Pendidikan <i>Hyperparameter 100-30</i> .....	127
Gambar 4. 58	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 50-10</i> .....	129
Gambar 4. 59	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 50-10</i> .....	130
Gambar 4. 60	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 50-20</i> .....	131
Gambar 4. 61	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 50-20</i> .....	132
Gambar 4. 62	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 50-30</i> .....	133
Gambar 4. 63	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 50-30</i> .....	134
Gambar 4. 64	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 75-10</i> .....	135
Gambar 4. 65	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 75-10</i> .....	136
Gambar 4. 66	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 75-20</i> .....	138
Gambar 4. 67	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 75-20</i> .....	139
Gambar 4. 68	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 75-30</i> .....	140
Gambar 4. 69	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 75-30</i> .....	141
Gambar 4. 70	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 100-10</i> .....	142
Gambar 4. 71	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 100-10</i> .....	143
Gambar 4. 72	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 100-20</i> .....	144
Gambar 4. 73	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 100-20</i> .....	145
Gambar 4. 74	Perbandingan Akurasi <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 100-30</i> .....	147
Gambar 4. 75	Boxplot Performa <i>Multistage Stage 2</i> Aspek Politik <i>Hyperparameter 100-30</i> .....	148
Gambar 4. 76	Hasil Rata-Rata Evaluasi <i>Singlestage</i> Gambar 4. 77 Hasil Rata-Rata Evaluasi <i>Multistage</i> .....	150

Gambar 4. 78 Grafik Pengaruh <i>n_estimator Singlestage</i> Gambar 4. 79 Grafik Pengaruh <i>n_estimator Multistage</i> .....	152
Gambar 4. 80 Grafik Pengaruh <i>Max_Depth Singlestage</i> Gambar 4. 81 Grafik Pengaruh <i>Max_Depth Multistage</i> .....	155
Gambar 4. 82 Grafik Perbandingan <i>Singlestage</i> dan <i>Multistage</i> .....	158

## DAFTAR TABEL

Tabel 2.1	Penelitian-Penilitian Deteksi Hoaks .....	12
Tabel 2.2	Penelitian-Penelitian Deteksi Hoaks Berbasis Aspek.....	26
Tabel 2.3	Penelitian-Penelitian Algoritma <i>Random Forest</i> .....	30
Tabel 3.1	Penjelasan Fitur.....	33
Tabel 3.2	Contoh Dataset Postingan Hoaks.....	34
Tabel 3.3	Data Hoaks Berbasis Aspek.....	35
Tabel 3.4	<i>One Hot Encoding</i> Label Aspek .....	36
Tabel 3.5	Proses Data <i>Cleaning</i> .....	38
Tabel 3.6	Proses Data <i>Case Folding</i> .....	39
Tabel 3.7	Proses Data Normalisasi .....	40
Tabel 3.8	Proses <i>Tokenizing</i> .....	41
Tabel 3.9	Proses <i>Stopword Removal</i> .....	42
Tabel 3.10	Proses <i>Stemming</i> .....	43
Tabel 3.11	Contoh Data Postingan .....	45
Tabel 3.12	Contoh Kata Unik .....	46
Tabel 3.13	Contoh Data Vektor Frekuensi .....	46
Tabel 4. 1	Skenario Pengujian .....	58
Tabel 4. 2	<i>Confusion Matrix Singlestage</i> $n\_estimator = 50$ dan $max\_depth = 10$ .....	67
Tabel 4. 3	Hasil Evaluasi <i>5-Fold Cross-Validation Singlestage</i> <i>Hyperparameter</i> 50-10.....	68
Tabel 4. 4	<i>Confusion Matrix Singlestage</i> $n\_estimator = 50$ dan $max\_depth = 20$ .....	69
Tabel 4. 5	Hasil Evaluasi <i>5-Fold Cross-Validation Singlestage</i> <i>Hyperparameter</i> 50-20.....	71
Tabel 4. 6	<i>Confusion Matrix Singlestage</i> $n\_estimator = 50$ dan $max\_depth = 30$ .....	72
Tabel 4. 7	Hasil Evaluasi <i>5-Fold Cross-Validation Singlestage</i> <i>Hyperparameter</i> 50-30.....	73
Tabel 4. 8	<i>Confusion Matrix Singlestage</i> $n\_estimator = 75$ dan $max\_depth = 10$ .....	74
Tabel 4. 9	Hasil Evaluasi <i>5-Fold Cross-Validation Singlestage</i> <i>Hyperparameter</i> 75-10.....	76
Tabel 4. 10	<i>Confusion Matrix Singlestage</i> $n\_estimator = 75$ dan $max\_depth = 20$ .....	77
Tabel 4. 11	Hasil Evaluasi <i>5-Fold Cross-Validation Singlestage</i> <i>Hyperparameter</i> 75-20.....	78
Tabel 4. 12	<i>Confusion Matrix Singlestage</i> $n\_estimator = 75$ dan $max\_depth = 30$ .....	79
Tabel 4. 13	Hasil Evaluasi <i>5-Fold Cross-Validation Singlestage</i> <i>Hyperparameter</i> 75-30.....	81
Tabel 4. 14	<i>Confusion Matrix Singlestage</i> $n\_estimator = 100$ dan $max\_depth = 10$ .....	82

Tabel 4. 15 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 100-10.....	83
Tabel 4. 16 Confusion Matrix Singlestage $n\_estimator = 100$ dan $max\_depth = 20$ .....	84
Tabel 4. 17 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 100-20.....	85
Tabel 4. 18 Confusion Matrix Singlestage $n\_estimator = 100$ dan $max\_depth = 30$ .....	87
Tabel 4. 19 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 100-30.....	88
Tabel 4. 20 Confusion Matrix Multistage Stage 1 $n\_estimator = 50$ dan $max\_depth = 10$ .....	89
Tabel 4. 21 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 50-10.....	90
Tabel 4. 22 Confusion Matrix Multistage Stage 1 $n\_estimator = 50$ dan $max\_depth = 20$ .....	92
Tabel 4. 23 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 50-20.....	93
Tabel 4. 24 Confusion Matrix Multistage Stage 1 $n\_estimator = 50$ dan $max\_depth = 30$ .....	94
Tabel 4. 25 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 50-30.....	95
Tabel 4. 26 Confusion Matrix Multistage Stage 1 $n\_estimator = 75$ dan $max\_depth = 10$ .....	96
Tabel 4. 27 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 75-10.....	97
Tabel 4. 28 Confusion Matrix Multistage Stage 1 $n\_estimator = 75$ dan $max\_depth = 20$ .....	98
Tabel 4. 29 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 75-20.....	99
Tabel 4. 30 Confusion Matrix Multistage Stage 1 $n\_estimator = 75$ dan $max\_depth = 30$ .....	100
Tabel 4. 31 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 75-30.....	101
Tabel 4. 32 Confusion Matrix Multistage Stage 1 $n\_estimator = 100$ dan $max\_depth = 10$ .....	103
Tabel 4. 33 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 100-10.....	104
Tabel 4. 34 Confusion Matrix Multistage Stage 1 $n\_estimator = 100$ dan $max\_depth = 20$ .....	105
Tabel 4. 35 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 100-20.....	106
Tabel 4. 36 Confusion Matrix Multistage Stage 1 $n\_estimator = 100$ dan $max\_depth = 30$ .....	107
Tabel 4. 37 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 100-30.....	108

Tabel 4. 38 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 50 dan <i>max_depth</i> = 10.....	109
Tabel 4. 39 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 50-10.....	110
Tabel 4. 40 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 50 dan <i>max_depth</i> = 20.....	111
Tabel 4. 41 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 50-20.....	112
Tabel 4. 42 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 50 dan <i>max_depth</i> = 30.....	113
Tabel 4. 43 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 50-30.....	114
Tabel 4. 44 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 75 dan <i>max_depth</i> = 10.....	115
Tabel 4. 45 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 75-10.....	116
Tabel 4. 46 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 75 dan <i>max_depth</i> = 20.....	117
Tabel 4. 47 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 75-20.....	118
Tabel 4. 48 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 75 dan <i>max_depth</i> = 30.....	119
Tabel 4. 49 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 75-30.....	121
Tabel 4. 50 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 100 dan <i>max_depth</i> = 10.....	122
Tabel 4. 51 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 100-10.....	123
Tabel 4. 52 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 100 dan <i>max_depth</i> = 20.....	124
Tabel 4. 53 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 100-20.....	125
Tabel 4. 54 <i>Confusion Matrix Multistage Stage 2 Aspek Pendidikan</i> <i>n_estimator</i> = 100 dan <i>max_depth</i> = 30.....	126
Tabel 4. 55 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan <i>Hyperparameter</i> 100-30.....	127
Tabel 4. 56 <i>Confusion Matrix Multistage Stage 2 Aspek Politik</i> <i>n_estimator</i> = 50 dan <i>max_depth</i> = 10.....	129
Tabel 4. 57 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik <i>Hyperparameter</i> 50-10 .....	130
Tabel 4. 58 <i>Confusion Matrix Multistage Stage 2 Aspek Politik</i> <i>n_estimator</i> = 50 dan <i>max_depth</i> = 20.....	131
Tabel 4. 59 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik <i>Hyperparameter</i> 50-20 .....	132
Tabel 4. 60 <i>Confusion Matrix Multistage Stage 2 Aspek Politik</i> <i>n_estimator</i> = 50 dan <i>max_depth</i> = 30.....	133

Tabel 4. 61 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 50-30 .....	134
Tabel 4. 62 Confusion Matrix Multistage Stage 2 Aspek Politik $n\_estimator = 75$ dan $max\_depth = 10$ .....	135
Tabel 4. 63 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 75-10 .....	136
Tabel 4. 64 Confusion Matrix Multistage Stage 2 Aspek Politik $n\_estimator = 75$ dan $max\_depth = 20$ .....	137
Tabel 4. 65 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 75-20 .....	138
Tabel 4. 66 Confusion Matrix Multistage Stage 2 Aspek Politik $n\_estimator = 75$ dan $max\_depth = 30$ .....	140
Tabel 4. 67 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 75-30 .....	141
Tabel 4. 68 Confusion Matrix Multistage Stage 2 Aspek Politik $n\_estimator = 100$ dan $max\_depth = 10$ .....	142
Tabel 4. 69 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 100-10 .....	143
Tabel 4. 70 Confusion Matrix Multistage Stage 2 Aspek Politik $n\_estimator = 100$ dan $max\_depth = 20$ .....	144
Tabel 4. 71 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 100-20 .....	145
Tabel 4. 72 Confusion Matrix Multistage Stage 2 Aspek Politik $n\_estimator = 100$ dan $max\_depth = 30$ .....	146
Tabel 4. 73 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 100-30 .....	147
Tabel 4. 74 Hasil Evaluasi Model Random Forest .....	149
Tabel 4. 75 Pengaruh $n\_estimator$ Singlestage dan Multistage .....	152
Tabel 4. 76 Pengaruh $max\_depth$ Singlestage dan Multistage .....	155
Tabel 4. 77 Perbandingan Singlestage dan Multistage .....	157

## ABSTRAK

Anabila, Qonita Nashifa. 2025. **Deteksi Hoaks Untuk Postingan Artikel Tulisan Berbahasa Indonesia Berbasis Topik *Modelling* Menggunakan Algoritma *Random Forest*.** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Okta Qomaruddin Aziz, M.Kom (II) Dr. Cahyo Crysdiyan, M.Cs.

**Kata Kunci:** Deteksi Hoaks, *Random Forest*, *Topic Modelling*, *Multistage Classification*, *Bag-of-Words*

Penyebaran hoaks di media sosial Indonesia mengancam stabilitas informasi publik. Penelitian ini mengembangkan sistem deteksi hoaks pada artikel berbahasa Indonesia menggunakan algoritma *Random Forest* dengan pendekatan topik *modelling*. Dataset berjumlah 733 postingan diklasifikasikan berdasarkan dua aspek: pendidikan, politik. Penelitian membandingkan pendekatan *singlestage* dan *multistage classification* dengan data seimbang. Hasil menunjukkan *singlestage classification* mencapai akurasi 70,75%, sedangkan *multistage* menghasilkan performa *end-to-end* 46,77% akibat propagasi kesalahan antar tahap. Kombinasi ekstraksi fitur *Bag-of-Words* dengan parameter *n\_estimators* 50-100 dan *max\_depth* 10-30 memberikan performa optimal. Penelitian ini berkontribusi pada pengembangan sistem deteksi hoaks yang adaptif terhadap karakteristik bahasa Indonesia.

## ABSTRACT

Anabila, Qonita Nashifa. 2025. Hoax Detection for Indonesian-Language Article Posts Based on Topic Modelling Using Random Forest Algorithm. Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisors: (I) Okta Qomaruddin Aziz, M.Kom (II) Dr. Cahyo Crysdiyan, M.Cs.

**Keywords:** *Hoax Detection, Random Forest, Topic Modelling, Multistage Classification, Bag-of-Words*

The spread of hoaxes on Indonesian social media threatens public information stability. This research develops a hoax detection system for Indonesian-language articles using Random Forest algorithm with topic modelling approach. A dataset of 733 posts was classified based on two aspects: education and politics. The study compared single-stage and multistage classification approaches with balanced data. Results show singlestage classification achieved 70.75% accuracy, while multistage yielded 46.77% end-to-end performance due to error propagation between stages. The combination of Bag-of-Words feature extraction with `n_estimators` parameters of 50-100 and `max_depth` of 10-30 provided optimal performance. This research contributes to developing hoax detection systems adaptive to Indonesian language characteristics.

## مستخلص البحث

أناپلا، قونیتا ناشفة. ٢٠٢٥. الكشف عن الأخبار الزائفة لمنشورات المقالات المكتوبة باللغة الإندونيسية بناءً على نمذجة الموضوعات باستخدام خوارزمية الغابة العشوائية. بحث جامعي. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفان: (١) أوكتا قمر الدين عزيز، الماجستير في علوم الحاسوب (٢) د. تشاهيو كريسيديان، الماجستير في علوم الحاسوب.

**الكلمات المفتاحية:** كشف الأخبار الزائفة، الغابة العشوائية، نمذجة الموضوعات، التصنيف متعدد المراحل، كيس الكلمات

يهدف انتشار الأخبار الزائفة على وسائل التواصل الاجتماعي الإندونيسية استقرار المعلومات العامة. يطور هذا البحث نظام كشف الأخبار الزائفة للمقالات باللغة الإندونيسية باستخدام خوارزمية الغابة العشوائية مع نجح نمذجة الموضوعات. صُنفت مجموعة بيانات مكونة من ٧٣٣ منشوراً بناءً على جانبين: التعليم والسياسة. قارن البحث بين نجح التصنيف أحادي المرحلة ومتعدد المراحل مع بيانات متوازنة. أظهرت النتائج أن التصنيف أحادي المرحلة حقق دقة ٧٥٪، بينما حقق التصنيف متعدد المراحل أداءً شاملًا ٤٦,٧٪ بسبب انتشار الأخطاء بين المراحل. وفر المجمع بين استخراج ميزات كيس الكلمات مع معاملات  $n\_estimators$  من ٣٠-١٠ أداءً مثاليًا. يساهم هذا البحث في تطوير أنظمة كشف الأخبار من  $100-50$  و  $max\_depth$ .

الزائفة التي تكيف مع خصائص اللغة الإندونيسية

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan teknologi dan informasi telah membawa perubahan yang signifikan dalam kehidupan masyarakat. Akses terhadap informasi kini semakin mudah dan cepat melalui berbagai platform, khususnya media sosial. Kemajuan ini memungkinkan setiap individu untuk menyebarkan informasi secara instan, salah satunya melalui unggahan atau *postingan*. Namun, kemudahan ini sering disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab untuk menyebarkan informasi palsu atau hoaks sehingga dapat menimbulkan kecemasan publik dan mengancam stabilitas sosial (Yopita, 2024).

Penyebaran hoaks tidak hanya terjadi di media sosial, tetapi juga melalui postingan artikel yang tersebar di berbagai situs daring. Postingan artikel hoaks umumnya disusun dengan gaya penulisan yang menyerupai berita asli, lengkap dengan judul yang menarik, narasi informatif, dan kutipan tokoh tertentu. Pola penyajian semacam ini membuat artikel tampak seperti sumber informasi resmi, padahal isi yang disampaikan sering kali telah dimanipulasi atau dilebih-lebihkan untuk membentuk opini tertentu.

Oleh karena itu, internet kini menjadi sumber utama dalam memperoleh berbagai informasi. Di satu sisi, internet membuka peluang untuk berkarya dan menghasilkan keuntungan. Di sisi lain, jika tidak digunakan secara bijak, masyarakat rentan terhadap paparan informasi yang tidak valid.

Penelitian (Arintasari *et al.*, 2023) mencatat bahwa pengguna internet di Indonesia pada tahun 2021 mencapai 202,6 juta dan mengalami peningkatan 11% dari tahun sebelumnya. Hal ini menjadikan Indonesia sebagai salah satu negara dengan pengguna internet terbanyak. Namun, dari tingginya konsumsi informasi tersebut, hanya sekitar 73% informasi yang telah melalui proses validasi, sehingga menunjukkan adanya celah penyebaran hoaks yang cukup besar.

Hoaks merupakan informasi yang sengaja direkayasa untuk menyesatkan atau membalikkan fakta (Maulidya *et al.*, 2023). Dalam penelitian (Batoebara & Hasugian, 2023) menyatakan bahwa hoaks menurut KBBI adalah informasi palsu yang dibuat oleh pelaku iseng dengan mengumpulkan suatu informasi penerus yang penyebarannya melalui beberapa platform digital, terutama media sosial seperti *Facebook* dan *Twitter* yang kini disebut dengan aplikasi *X* sehingga menjadi tantangan serius karena dapat menciptakan kebencian dan krisis kepercayaan masyarakat terhadap media sosial (Rahmadhany *et al.*, 2021). Penelitian (Tambunan *et al.*, 2023) menunjukkan bahwa per November 2019, terdapat 122 juta pengguna *Facebook* dan 78 pengguna *X* di Indonesia yang menjadi kanal potensial penyebaran hoaks terutama dalam bentuk teks. Menurut (Juliswara & Muryanto, 2022), literasi digital yang praktis seperti mengenali judul, alamat situs, dan kata-kata tertentu dalam teks dapat membantu mengidentifikasi hoaks. Namun, literasi saja belum cukup untuk mengimbangi arus informasi yang masif.

(Saptoyo & Galih, 2024) menyatakan bahwa kominfo telah menemukan 12.547 konten hoax 5 tahun terakhir pada Agustus 2018 hingga Desember 2023 dengan kategori terbanyak meliputi kesehatan (2.357 konten), politik (1.628

konten), serta penipuan daring seperti *phishing*, akun palsu, dan pemberitaan palsu. Dengan besarnya jumlah konten hoaks yang tersebar menunjukkan informasi hoaks sulit dibedakan dari informasi yang valid. Fenomena tersebut menggambarkan bahwa masalah utama tidak hanya terletak pada banyaknya konten hoaks yang beredar, tetapi juga pada kesulitan dalam membedakan hoaks dari informasi yang valid. Oleh karena itu, dari berbagai jenis hoaks yang beredar seperti dalam bentuk postingan artikel tulisan menjadi salah satu yang paling sulit dikenali.

Maka dari itu, sulitnya membedakan antara postingan artikel valid dan postingan artikel hoaks menjadi persoalan utama dalam penyebaran informasi di era digital, terutama karena keduanya memiliki struktur dan gaya penulisan yang serupa. Postingan artikel hoaks dikemas secara sistematis dengan bahasa formal dan narasi yang tampak informatif, sehingga tidak mudah dikenali hanya melalui pengamatan manusia. Di sisi lain, proses verifikasi manual seperti literasi digital dan pemeriksaan fakta (*fact-checking*) masih bersifat reaktif, membutuhkan waktu yang lama, dan tidak mampu mengimbangi kecepatan penyebaran informasi di internet yang sangat masif. Dalam Penelitian (Amaly & Armiah, 2021) selama 24 jam 7 hari Kominfo melakukan monitoring *cyber crime* di media sosial terkait informasi hoaks, dan hampir setiap hari menemukan informasi salah di media sosial.

Kondisi tersebut menunjukkan upaya deteksi hoaks yang dilakukan dengan cara verifikasi manual masih belum efektif untuk menandingi kecepatan dan keragaman penyebaran informasi di media sosial. Maka, dibutuhkan model yang mampu memahami konteks isi dan pola topiknya secara menyeluruh agar hasil

klasifikasi menjadi lebih akurat untuk postingan artikel tulisan berbahasa indonesia yang mengandung variasi struktur dan tema yang lebih luas.

Oleh karena itu, diperlukan solusi berbasis teknologi untuk mendeteksi dan mengklasifikasikan postingan hoaks secara otomatis. Salah satunya dengan menggunakan pendekatan yang efektif melalui algoritma *machine learning* yaitu *Random Forest*. Algoritma ini dipilih karena memiliki keunggulan yang mampu menangani data dalam jumlah besar, mampu bekerja dengan baik pada data yang tidak seimbang sehingga tetap memberikan hasil klasifikasi yang akurat, serta mampu menghasilkan akurasi yang tinggi (Yoga Religia *et al.*, 2021) . Selain itu, *Random Forest* lebih stabil dan tahan terhadap overfitting jika dibandingkan dengan algoritma lain seperti *Decision Tree*, *Naïve Bayes*, maupun *K-Nearest Neighbors* (KNN), karena merupakan metode *ensamble learning* yang digunakan untuk menggabungkan banyak pohon keputusan dalam menghasilkan prediksi yang lebih kuat (Nugroho, 2025). Dengan demikian, penelitian ini menerapkan pendekatan klasifikasi *multistage* dengan algoritma *Random Forest* yang dapat membagi proses dengan bertahap sehingga model dapat mengenali pola teks yang lebih terstruktur.

Dalam penelitian (Aryanto, 2021) klasifikasi *multistage* didefinisikan sebagai metode klasifikasi yang digunakan untuk penentuan tahapan penyelesaian dengan proses data yang dibagi menjadi kelompok pada beberapa tahapan atau yang disebut dengan *stage*. Klasifikasi *multistage* dapat bekerja dengan cara membagi masalah klasifikasi ke dalam beberapa tahap. Setiap tahap, model dirancang untuk menyelesaikan sub-masalah tertentu dengan lebih sederhana sebelum melanjutkan

ke tahap selanjutnya. Maka dari itu, hasil dari satu tahap klasifikasi tersebut menjadi input untuk tahap selanjutnya hingga label akhir. Klasifikasi *multistage* memiliki keunggulan yaitu dapat meningkatkan akurasi, menganalisis kinerja setiap tahap secara berpisah, membagi tugas klasifikasi menjadi beberapa tahapan yang lebih sederhana (Hassan, 2021). Untuk memastikan klasifikasi *multistage* dengan algoritma *Random Forest* yang dapat bekerja secara konsisten dalam setiap tahap yang dimulai dari tahap awal hingga akhir agar mengetahui atau menilai seberapa baik model *Random Forest* dengan pendekatan klasifikasi *multistage* dalam menyelesaikan sub-masalah sebelum mencapai tahap akhir.

Dengan demikian, untuk mendukung proses klasifikasi algoritma *Random Forest* dengan klasifikasi *multistage*, diperlukan teknik *Natural Language Processing* (NLP) untuk data yang berupa teks, sehingga pemrosesan bahasa alami digunakan karena untuk mengetahui struktur dan makna dari informasi yang ada dalam setiap postingan. Salah satu metode yang digunakan dalam ekstraksi fitur teks adalah *Bag-Of-Words*, yang merupakan teknik dari tahapan NLP. Metode ini dapat merepresentasikan frekuensi kata untuk mengubah teks menjadi vektor numerik yang digunakan sebagai input untuk membedakan antara postingan hoaks dan non-hoaks dengan lebih akurat. Dalam penelitian ini, performa model berdasarkan hasil klasifikasi yang di evaluasi melalui metrik akurasi, *precision*, *recall*, *f1-Score* yang dapat diperoleh dari *confusion matrix* sehingga dapat memberikan gambaran yang lebih ekstensif mengenai kemampuan *Random Forest* dalam mendeteksi postingan hoaks berbahasa Indonesia secara bertahap, serta

menilai keseimbangan antara *precision* dan *recall* dalam hasil klasifikasi (Fadli & Saputra, 2023).

Dengan algoritma *Random forest* yang menggunakan pendekatan klasifikasi *multistage* dan *Natural Language Processing*, penelitian ini membagun deteksi hoaks berbasis Bahasa Indonesia yang diharapkan dapat memberikan kontribusi nyata dalam mengurangi penyebaran hoaks di media sosial melalui pendekatan teknologi yang andal dan adaptif terhadap karakteristik bahasa lokal. Oleh karena itu, dapat menghasilkan klasifikasi yang tidak hanya mampu membedakan kategori postingan hoaks, tetapi juga konsisten dalam mengenali berbagai pola penulisan dan susunan informasi pada postingan hoaks di media sosial.

Penggunaan algoritma *Random forest* dengan klasifikasi *multistage* dan ekstraksi fitur *Bag-Of-Word* juga sejalan dengan prinsip-prinsip Islam, terutama dalam hal pentingnya kejujuran. Islam mengajarkan umat muslim untuk berhati-hati dalam ucapan maupun perbuatan. Dalam Al-Qur'an surat Al-Hujurat ayat 6 memaparkan mengenai peringatan untuk melakukan tabayyun pada suatu informasi, sebagaimana pada penelitian ini yang mengkaji postingan hoaks. Berikut merupakan bunyi dari Q.S Al-Hujurat yang terdapat pada ayat 6.

يَأَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَيْنِ فَتَبَيَّنُوا أَنْ تُصِيبُونَا فَوْمًا هُنْجَاهَلٌ فَتُصْبِخُونَا عَلَىٰ مَا فَعَلْنَا  
نَدِيمُنَّ

“Wahai orang-orang yang beriman, jika seorang fasik datang kepadamu membawa berita penting, maka telitilah kebenarannya agar kamu tidak mencelakakan suatu kaum karena ketidaktahanan(-mu) yang berakibat kamu menyesali perbuatanmu itu.” (Q.S Al-Hujurat: 6).

Seperti yang kita ketahui tafsir dari ayat tersebut adalah (wahai orang-orang beriman, jika seseorang fasik datang) maksud dari seseorang fasik ialah orang yang menyimpang dari ketaatan kepada Allah dan Rasul-nya. Kemudian dijelaskan pada ayat setelahnya yang artinya (membawa berita penting, maka telitilah kebenarannya) merupakan peringatan bahwa berita yang diberikan oleh (orang fasik) tersebut perlu menerapkan prinsip tabayyun. Prinsip tabayyun merupakan cara untuk memeriksa kebenaran berita sebelum seseorang menerima dan memercayai berita tersebut. Dengan demikian, dijelaskan juga pada ayat tersebut yakni (agar kamu tidak mencelakakan suatu kaum karena ketidaktahuan(mu) yang berakibat kamu menyesali perbuatanmu itu) bahwa orang yang menerapkan prinsip tersebut tidak akan menyesali perbuatan setelah seseorang menerima dan memercayai berita tersebut. Pada penelitian ini tabayyun sangat di perlukan agar tidak mudah memercayai informasi yang belum jelas.

Maka dari itu, dengan membangun klasifikasi deteksi hoaks penelitian ini bertujuan untuk membantu masyarakat dalam mengatasi permasalahan hoaks agar tidak mudah terjerumus pada postingan informasi kurang valid yang ada di Indonesia. Sistem ini mengintegrasikan informasi dari berbagai postingan yang akan diolah menggunakan ekstraksi fitur *Bag-Of-Word* dan mengklasifikasikan menggunakan algoritma *Random Forest* agar bisa membedakan postingan hoaks dan non-hoaks.

Dengan demikian, penelitian ini berperan penting dalam membantu masyarakat mengatasi permasalahan informasi melalui postingan di media sosial yang ada di Indonesia.

## 1.2 Pernyataan Masalah

Seberapa besar performa algoritma *Random Forest* dengan pendekatan klasifikasi *multistage* untuk mengklasifikasikan postingan berbahasa Indonesia ke dalam kelas hoaks dan non-hoaks.

## 1.3 Batasan Masalah

1. Penelitian ini menggunakan dataset *Indonesia False News* yang diperoleh dari situs resmi “*Kaggle*” (Muhamar, 2020).
2. Dataset ini berisi postingan berbahasa Indonesia yang terdiri dari dua label kategori, yaitu hoaks dan non-hoaks, serta memiliki tiga atribut utama, yakni label, judul, dan narasi.

## 1.4 Tujuan Penelitian

Untuk Mengetahui dan mengukur performa algoritma *Random Forest* dengan pendekatan klasifikasi *multistage* dalam mengklasifikasikan postingan berbahasa Indonesia menjadi hoaks dan non-hoaks.

## 1.5 Manfaat Penelitian

1. Bagi Kementerian Komunikasi dan Informatika (KOMINFO), penelitian ini dapat menjadi dasar pengembangan sistem pemantauan otomatis untuk mengidentifikasi dan mengklasifikasikan postingan hoaks sehingga dapat membantu dalam menangani disinformasi.
2. Bagi dunia jurnalistik, penelitian ini dapat menjadi acuan dalam memahami pola linguistik pada berita hoaks berbahasa Indonesia serta membantu

meningkatkan kemampuan analisis redaksi dalam membedakan informasi faktual dan hoaks.

## **BAB II**

### **STUDI PUSTAKA**

#### **2.1 Deteksi *Hoax***

Deteksi hoaks sering didefinisikan sebagai proses untuk menangani atau mengindikasi sebuah informasi dengan mengklasifikasikan ke dalam kategori hoaks dan non-hoaks sehingga sistem mampu mendeteksi hoaks dengan mengenali pola bahasa dan isi informasi secara mendalam. Dengan demikian, deteksi otomatis ini sangat diperlukan sebagai bentuk perlindungan masyarakat dari misinformasi (Fardhina *et al.*, 2025).

Hoaks merupakan informasi yang disebarluaskan tanpa mencari tahu kejadiannya secara nyata. Hoaks dapat didefinisikan sebagai ucapan yang bohong, tipuan, ataupun menyesatkan sehingga membuat orang lain tersesat akan informasi yang diterima. Informasi ini dapat merugikan masyarakat karena mudah tersebar dengan luas. Fenomena hoaks sering terjadi di Indonesia ketika masa kampanye politik. Hal ini masyarakat sering salahpaham atau saling menyalahkan satu sama lain ketika kampanye politik berlangsung seperti pemilihan presiden Indonesia atau pemilihan kepala daerah (Irhamdhika, 2022).

Pertama kali ada kata hoaks yaitu pada tahun 1808 yang berasal dari bahasa *Inggris* dengan kata “*hoax*” dan hoaks juga dikenal dengan istilah “*hocus*” yang bermakna “*hocus pocus*” pertama kali di lakukan oleh pesulap. Dalam penelitian ini, menyebutkan bahwa *website* Oposisi.net telah menyebarkan informasi hoaks yang tidak sesuai dengan faktanya sehingga dapat menggiring opini kepada

pembaca dari judul yang provokatif dan isi informasi tersebut memanipulasi elemen informasi yang berkaitan dengan pernyataan tetapi tidak sesuai dengan peristiwa yang disampaikan. Elemen-elemen yang sering digunakan dalam menyampaikan informasi yaitu 5W + 1H. Maka dari itu, banyak *website* yang tersedia untuk menyampaikan informasi, namun dari banyaknya *website* yang ada tidak semua resmi atau fakta dalam menyampaikan informasi salah satu contoh dengan maraknya penyebaran hoaks yang membuat masyarakat panik dan cemas yaitu bersumber dari postingan di media sosial, tetapi setelah di telusuri lebih lanjut ternyata hoaks tersebut berasal dari situs yang tidak resmi. Masyarakat yang lebih banyak membaca dan rentan percaya terhadap informasi yang di sampaikan di media sosial adalah generasi milenial (Oktaviana & Hasfi, 2020).

Dalam hal ini terdapat salah satu pola dalam penyebaran hoaks yang dilakukan secara berulang seperti informasi hoaks yang dibuat kemudian disebar luaskan. Pola penyebaran hoaks ini terjadi karena informasi palsu yang di rancang lalu tersebar ke dalam kelompok atau aplikasi media sosial, kemudian informasi tersebut kembali ke orang yang pertama kali merancang informasi hoaks. Oleh karena itu, penyebaran hoaks ini tidak bisa di pantau bahkan orang yang pertama kali membuat informasi hoaks tidak dapat mengetahui seberapa banyak orang yang sudah menerima informasi hoaks tersebut (Dulkiah & Setia, 2020).

Penggunaan bahasa dalam hoaks banyak yang bersifat memanipulasi ataupun provokatif sehingga dari setiap kata yang disusun membuat masyarakat panik dan emosional. Strategi utama dalam membentuk opini publik salah satu unsur informasi hoaks yaitu mengandung propaganda. Dengan demikian, pola

bahasa dalam hoaks dapat menyampaikan informasi yang tidak valid, tetapi bukan hanya itu hoaks ini juga menjadi alat untuk mengendalikan persepsi masyarakat (Jamilah & Missouri, 2024).

Hoaks memberikan dampak yang merugikan kepada masyarakat karena dapat menimbulkan kebingungan dan ketidakpercayaan. Oleh karena itu, masyarakat harus meneliti kebenaran informasi sebelum membagikan atau mengambil tindakan berdasarkan informasi yang diterima (Salsabila, 2023).

Tabel 2.1 Penelitian-Penelitian Deteksi Hoaks

No.	Judul Penelitian	Peneliti	Metode yang digunakan	Hasil
1.	<i>Comparison of K-NN, SVM, and Random Forest Algorithm for Detecting Hoax on Indonesian Election 2024</i>	(Indra <i>et al.</i> , 2024).	<i>K-Nearest Neighbor (KNN), Support Vector Machine (SVM), dan Random Forest</i>	<i>Random Forest</i> menghasilkan akurasi sebesar 81,82%, dengan <i>precision</i> 80% dan <i>recall</i> 83%.
2.	<i>Fake News Detection using the Random Forest Algorithm</i>	(Setyadin <i>et al.</i> , 2025)	<i>Random Forest</i>	<i>Random Forest</i> memiliki performa yang sangat baik dalam mendekripsi berita hoaks, dengan nilai akurasi sebesar 0,989 atau 98,9%, <i>precision</i> sebesar 0,990 (99,0%), <i>recall</i> sebesar 1,000 (100%), dan <i>F1-score</i> sebesar 0,995 (99,5%)
3.	<i>Fake News Detection in Indonesian Popular News Portal Using Machine Learning For Visual Impairment</i>	(Triyono <i>et al.</i> , 2023)	<i>Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree Classifier (DTC), Gradient Boosting Classifier (GBC), dan Random Forest</i>	SVM memperoleh akurasi tertinggi yaitu 83,55%, sedangkan DTC memiliki akurasi terendah, yakni 75,33%.
4.	<i>Hoax Analyzer for Indonesian News Using Deep Learning Models</i>	(Nayoga <i>et al.</i> , 2021)	Long Short-Term Memory (LSTM) dan Bidirectional LSTM (BiLSTM)	<i>Bidirectional LSTM</i> memberikan performa terbaik dibandingkan model lainnya dengan nilai akurasi mencapai 94,52%, <i>precision</i>

No.	Judul Penelitian	Peneliti	Metode yang digunakan	Hasil
				95,09%, <i>recall</i> 94,37%, dan F1-score 94,73%.
5.	<i>Implementation of CNN Algorithm for Indonesian Hoax News Detection on Online News Portals</i>	(Hati & Sulistiani, 2025).	Convolutional Neural Network (CNN)	CNN mampu mencapai performa yang sangat baik, dengan akurasi pelatihan sebesar 95,77% dan akurasi validasi sebesar 95,18%.

Tabel 2.1 mengenai beberapa refensi penelitian yang menganalisis terkait sistem deteksi hoaks yang menunjukkan beraneka ragam metode yang digunakan dengan hasil yang berbeda-beda.

Penelitian ini menerapkan tiga algoritma machine learning yaitu *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), dan *Random Forest* dalam mendeteksi berita hoaks pada *tweet* berbahasa Indonesia yang berkaitan dengan pemilu 2024. Tujuan penelitian ini adalah untuk membandingkan performa ketiga algoritma tersebut dalam mengklasifikasikan data hoaks dan non-hoaks berdasarkan teks dari media sosial *Twitter*. Dataset dikumpulkan dengan menggunakan pustaka *Tweepy* untuk melakukan *crawling* terhadap *tweet* yang memuat kata kunci terkait pemilu 2024. Data yang terkumpul divalidasi menggunakan situs pemeriksa fakta terpercaya seperti turnbackhoax.id, cekfakta,tempo.co, dan cekfakta.kompas.com, kemudian dilabeli secara manual menjadi dua kelas, yaitu “hoaks” dan “non-hoaks”. Setelah melalui tahap preprocessing (yang mencakup *case folding*, *cleansing*, *slangword normalization*, *stopword removal*, dan *stemming*), dilakukan pembobotan fitur menggunakan metode *Term Frequency Inverse Document Frequency* (TF-IDF) sebelum data diproses ke tahap klasifikasi. Pengujian dilakukan dengan membandingkan nilai

*accuracy, precision, dan recall* menggunakan *Confusion Matrix*. Berdasarkan hasil eksperimen, algoritma SVM dan K-NN dengan pengukuran jarak *Euclidean Distance* menghasilkan akurasi tertinggi sebesar 86,36%, sedangkan algoritma *Random Forest* menghasilkan akurasi sebesar 81,82%, dengan *precision* 80% dan *recall* 83%. Dalam penelitian ini, *Random Forest* juga menunjukkan konsistensi kinerja pada data teks berdimensi tinggi, menegaskan efektivitasnya sebagai algoritma klasifikasi yang tangguh dalam konteks data sosial media (Indra *et al.*, 2024).

Dalam penelitian ini yang berfokus pada pengembangan sistem deteksi berita hoaks menggunakan algoritma *Random Forest* dengan pendekatan *machine learning* ini, tujuan utamanya adalah membangun model klasifikasi otomatis yang mampu membedakan antara berita hoaks dan non-hoaks secara akurat berdasarkan konten teks. Dataset yang digunakan berasal dari website resmi Kominfo, yang berisi data berita terverifikasi sebanyak 12.000 artikel, terdiri atas 6.000 berita hoaks dan 6.000 berita non-hoaks. Atribut yang digunakan meliputi judul, deskripsi, penulis, tanggal, kategori, URL berita, dan URL gambar. Proses *preprocessing* teks meliputi *lowercase*, penghapusan karakter khusus, *stopword removal*, *stemming*, serta pembobotan kata menggunakan *Term Frequency Inverse Document Frequency* (TF-IDF) untuk mengubah teks menjadi representasi numerik. Hasil pengujian menunjukkan bahwa algoritma *Random Forest* memiliki performa yang sangat baik dalam mendekripsi berita hoaks, dengan nilai akurasi sebesar 0,989 atau 98,9%, *precision* sebesar 0,990 (99,0%), *recall* sebesar 1,000 (100%), dan *F1-score* sebesar 0,995 (99,5%). Nilai akurasi ini menunjukkan bahwa

model mampu mengklasifikasikan hampir seluruh berita secara benar, baik hoaks maupun non-hoaks, dengan tingkat kesalahan yang sangat rendah. Oleh karena itu, penelitian ini membuktikan bahwa algoritma *Random Forest* sangat efektif dalam mendeteksi berita hoaks berbahasa Indonesia dengan tingkat akurasi mencapai 98,9%, serta mampu memberikan keseimbangan yang baik antara ketepatan (*precision*) dan kemampuan mendeteksi seluruh data positif (*recall*). Selain itu, model ini juga terbukti stabil dan konsisten berdasarkan hasil uji validasi silang yang menunjukkan variasi performa yang sangat kecil (Setyadin *et al.*, 2025)

Penelitian ini membahas penerapan algoritma *machine learning* untuk mendeteksi hoaks pada media sosial, khususnya berita hoaks berbahasa Indonesia. Tujuan dari penelitian ini untuk membandingkan performa beberapa algoritma machine learning yaitu dengan *Support Vector Machine* (SVM), *Logistic Regression* (LR), *Decision Tree Classifier* (DTC), *Gradient Boosting Classifier* (GBC), dan *Random Forest* dalam mengklasifikasikan berita hoaks menggunakan dataset dari *Kaggle Indonesia False News*. Lima algoritma tersebut diuji dengan mengukur akurasi, sensitivitas, dan spesifitas untuk menentukan algoritma mana yang paling efektif dalam mendeteksi berita hoaks. algoritma SVM memperoleh akurasi tertinggi yaitu 83,55%, sedangkan DTC memiliki akurasi terendah, yakni 75,33%. Selain itu, untuk area di bawah kurva *Receiver Operating Characteristic* (AUC), SVM memiliki AUC sebesar 71,61% yang menunjukkan kemampuan klasifikasinya yang sangat baik. Di sisi lain, DTC menunjukkan AUC terendah, yakni 54,89% meskipun memiliki sensitivitas tinggi, namun metode ini kurang efektif dalam mengidentifikasi kelas negatif atau berita yang valid. Dalam hal

sensitivitas, yang mengukur sejauh mana algoritma dapat mendeteksi berita hoaks dengan benar, SVM dan GBC menunjukkan performa yang baik dengan nilai sensitivitas berita hoaks yang benar, dengan sensitivitas 83,57% dan spesifisitas yang rendah yaitu 28,03% (Triyono *et al.*, 2023)

Pengembangan sistem deteksi hoaks pada berita berbahasa Indonesia menggunakan pendekatan *deep learning*. Tujuan penelitian ini adalah untuk membangun model klasifikasi yang mampu membedakan antara berita hoaks dan non-hoaks secara otomatis dengan tingkat akurasi tinggi. Dalam implementasinya, penulis menggunakan beberapa model *deep learning* seperti *Long Short-Term Memory* (LSTM) dan *Bidirectional LSTM* (BiLSTM) untuk memproses teks berita yang diambil dari berbagai sumber daring. Data yang digunakan melalui tahap praproses meliputi *tokenization*, *stopword removal*, dan *word embedding* menggunakan Word2Vec untuk mengubah teks menjadi representasi numerik yang dapat dipahami oleh model. Hasil penelitian menunjukkan bahwa model *Bidirectional LSTM* memberikan performa terbaik dibandingkan model lainnya dengan nilai akurasi mencapai 94,52%, *precision* 95,09%, *recall* 94,37%, dan F1-score 94,73%. Hal ini menunjukkan bahwa model BiLSTM lebih efektif dalam menangkap konteks dua arah dari kalimat berita, sehingga lebih mampu mengenali pola-pola linguistik yang menunjukkan indikasi hoaks. Sementara itu, model LSTM standar memperoleh akurasi sedikit lebih rendah, yaitu sekitar 92,87% (Nayoga *et al.*, 2021).

Penelitian ini berfokus pada penerapan algoritma *Convolutional Neural Network (CNN)* dalam mendeteksi berita hoaks berbahasa Indonesia pada portal

berita daring. Tujuannya adalah mengembangkan model *deep learning* yang mampu melakukan klasifikasi teks secara otomatis untuk membedakan antara berita valid dan berita hoaks. Dataset yang digunakan diperoleh dari *Kaggle*, dengan sumber berita berasal dari portal CNN Indonesia dan Tempo untuk kategori valid news, serta situs Turnbackhoax untuk kategori hoaks news. Jumlah total data yang digunakan sebanyak 26.976 artikel dengan rentang waktu dari Juli 2015 hingga Maret 2023. Proses praproses data *meliputi stopword removal, tokenization, word embedding* menggunakan *FastText*, serta penerapan teknik *Synthetic Minority Oversampling Technique* (SMOTE) untuk mengatasi ketidakseimbangan kelas. Arsitektur CNN yang digunakan terdiri atas tiga blok konvolusional yang dilengkapi dengan *Batch Normalization, MaxPooling1D*, dan *Dropout* untuk mencegah *overfitting*. Hasil dari ketiga blok konvolusional ini kemudian diproses menggunakan *GlobalMaxPooling1D* dan dense layer sebelum menuju ke output layer dengan fungsi aktivasi sigmoid untuk klasifikasi biner. Model dilatih menggunakan *Adam optimizer* dengan *binary crossentropy loss function*, serta dilakukan pembagian data sebesar 80% untuk pelatihan dan 20% untuk pengujian. Hasil evaluasi menunjukkan bahwa model CNN mampu mencapai performa yang sangat baik, dengan akurasi pelatihan sebesar 95,77% dan akurasi validasi sebesar 95,18%. Nilai loss pada pelatihan dan validasi masing-masing sebesar 0,1146 dan 0,1210, sedangkan nilai F1-score mencapai 0,95 untuk kedua kelas (valid dan hoaks). Evaluasi juga menunjukkan nilai *Area Under Curve* (AUC) mendekati 1,00, menandakan kemampuan model yang sangat baik dalam membedakan dua kelas tersebut. Selain itu, penerapan 5-Fold *Stratified Cross Validation*

menunjukkan hasil yang konsisten, menandakan bahwa model mampu melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya. Berdasarkan hasil tersebut, penelitian ini menyimpulkan bahwa kombinasi algoritma CNN dengan *FastText embedding*, *Batch Normalization*, dan SMOTE mampu meningkatkan efektivitas dan stabilitas model dalam mendeteksi berita hoaks berbasis teks (Hati & Sulistiani, 2025).

## 2.2 Deteksi Hoaks Berbasis Aspek

Hoaks dapat muncul dalam berbagai aspek kehidupan, sehingga diperlukan pengelompokan aspek untuk memudahkan analisis dan pembahasan deteksi hoaks secara lebih terarah. Dalam penelitian ini, aspek yang dianalisis meliputi:

### 2.2.1 Aspek Kesehatan

Pada aspek kesehatan banyak berita hoaks yang beredar terutama di masa COVID-19. Di era COVID-19 banyak hoaks beredar mengenai isu informasi vaksin sehingga dapat mengurangi kepercayaan masyarakat terhadap program vaksinasi sehingga sebagian orang menolak dan ragu untuk vaksin. Hal ini membuat kondisi dalam aspek kesehatan rentan untuk menyebarkan informasi hoaks karena layanan medis secara langsung berkaitan dengan keselamatan dan kesehatan masyarakat (Roshinta *et al.*, 2023).

Dalam penelitian ini memaparkan masalah informasi hoaks dalam aspek kesehatan karena sejak Februari 2016 sampai Februari 2017 banyak informasi hoaks beredar tentang kesehatan. Oleh karena itu, untuk mendeteksi informasi hoaks tersebut dengan menerapkan algoritma *Support Vector Machine* untuk

mengembangkan dan menganalisis informasi hoaks terkait Covid-19. Penelitian ini memperoleh dataset dengan menggunakan teknik *web scraping* dengan jumlah data yang akan diproses yaitu 8.172 terdiri dari 837 data hoaks dan 7.335 data non-hoaks. Teknik pelabelan data dilakukan dengan cara manual, data disebut hoaks karena dapat mengandung kata atau isi berita tidak sesuai dengan kebenarannya atau berasal dari situs *turnbackhoax.id*, sedangkan data disebut non-hoaks karena mengandung isi berita fakta. Hasil pengujian dalam penelitian ini menunjukkan 111 data hoaks diprediksi benar sebagai hoaks, sementara 61 data hoaks salah diprediksi sebagai non-hoaks. Untuk data non-hoaks, terdapat 55 yang keliru diprediksi sebagai hoaks dan 1408 teridentifikasi dengan benar sebagai non-hoaks. Algoritma *Support Vector Machine* (SVM) dengan dua fungsi kernel menunjukkan variasi akurasi pada setiap skenario. Kernel linear memperoleh akurasi tertinggi sebesar 97,06% pada skenario 1 (60:40), sedangkan akurasi terendahnya muncul pada skenario 4 (90:10) dengan nilai 95,87%. Pada skenario 3 (80:20), kernel linear juga memberikan hasil cukup baik dengan akurasi 92,90%, karena mampu mengklasifikasikan berita hoaks dan non-hoaks secara seimbang. Sementara itu, kernel RBF menghasilkan akurasi terendah pada skenario 4 (90:10) yaitu 90,46%, sehingga performanya kurang optimal. Pada skenario tersebut, terdapat 28 data hoaks yang berhasil diprediksi benar, namun 75 data hoaks salah diklasifikasikan sebagai non-hoaks. Untuk kategori non-hoaks, sebanyak 3 data keliru diprediksi sebagai hoaks, sementara 712 data lainnya berhasil dikenali dengan benar. (Ropikoh et al., 2021).

### 2.2.2 Aspek Politik

Dalam aspek politik, hoaks seringkali menyebar luas untuk menggiring opini publik serta membentuk persepsi yang dilakukan untuk menyerang lawan politik. Penelitian ini membahas kampanye politik yang sering dimainkan di media sosial dan hoaks sering dibuat seolah-olah informasi yang salah disebut benar. Dari informasi yang disangka benar akan tersebar berulang-ulang kali bahkan lebih dari sekali. Kebohongan tersebut jika sudah dikatakan berulang-ulang kali akan menjadi benar. Maka dari itu, masyarakat akan percaya apapun tanpa konsekuensi logis jika informasi tersebut tersebar berulang-ulang kali. Oleh karena itu dalam aspek politik, informasi sering kali menjadikan alasan untuk menyerang lawan atau kandidat di era kampaye. Di media sosial hubungan antar individu membuat kecepatan informasi dari pada kebenarannya, sehingga hoaks sering menjadi topik informasi publik meskipun itu tidak valid (Doddy Suryadi *et al.*, 2024).

Penelitian ini menganalisis deteksi berita hoaks dalam aspek politik yaitu pemilu pada tahun 2024 menggunakan algoritma *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM). Tujuan dari penelitian ini untuk mengembangkan sistem deteksi hoaks dengan menggabungkan dua algoritma KNN dan SVM. Oleh karena itu, penelitian ini menerapkan algoritma KNN untuk menganalisis frekuensi kemunculan kata dalam kalimat, sedangkan SVM berfungsi untuk mengklasifikasikan suatu jenis data. Dataset yang digunakan dalam penelitian ini diperoleh dengan *web scrapping* dari situs *turnbackhoax.id* untuk data hoaks dan data non-hoaks diperoleh dari situs *kumparan.com*, *detik.com*, *liputan6.com*, dan *cnn Indonesia*, dengan data berjumlah 4283 terdiri dari 3040 berita non-hoaks dan

1234 berita hoaks. Dataset dengan label hoaks dikategorikan sebagai informasi palsu atau hoaks, sedangkan non-hoaks untuk kategori data yang berisi informasi benar. Algoritma *K-Nearest Neighbors* (KNN) diawali dengan menentukan nilai  $k$  sebagai jumlah tetangga terdekat, proses klasifikasi yang digunakan untuk menghitung jarak antar data menggunakan *Eulidean Distance* dan setiap kelas data di prediksi dengan memilih tetangga terdekat sebagai mayoritas kelas. Algoritma *Support Vector Machine* (SVM) diawali dengan memilih kernel (kernel, *polynomial*, atau *radial basis function*) serta memisahkan kelas dengan membangun *hyperplane*. Dalam penelitian ini, hasil pengujian dari algoritma KNN yaitu menghasilkan akurasi 90.73%, presisi 0.90, *recall*, 0.91, dan *F1-Score* 0.90. sedangkan algoritma SVM menghasilkan akurasi 91.98%, presisi 0.92, *recall* 0.92, dan *F1-Score* 0.91 (DickiPrabowo et al., 2025).

### **2.2.3 Aspek Bencana Alam**

Arus informasi sering disalahgunakan oleh masyarakat salah satunya informasi hoaks yang bebas dirancang tanpa mengetahui kebenarannya. Salah satu contoh beredarnya hoaks dalam aspek bencana alam seperti gempa bumi. Dalam penelitian ini memaparkan fenomena gempa bumi yang terjadi di Kabupaten Cianjur pada tanggal 21 November 2022 dengan kekuatan 5,6 magnitudo. Dengan situasi darurat, banyak pihak yang memanfaatkan media sosial dan sengaja merancang serta menyebarkan informasi hoaks sehingga masalah utamanya dapat menambah beban psikologis korban gempa . Namun, hoaks dalam aspek bencana alam lebih berdampak pada orang tua karena membuat kekhawatiran dan ketakutan misalnya “adanya gempa susulan dan tsunami”, contoh lainnya “gunung gede akan

meletus". Oleh karena itu, agar tidak mudah terkecoh oleh informasi yang belum tentu fakta maka harus memiliki penalaran kritis dan sikap skeptis karena seseorang yang cermat tidak akan mudah percaya pada informasi yang beredar di media sosial (Agustinah *et al.*, 2024).

Informasi hoaks bencana alam yang sering beredar dan menimbulkan ketakutan pada masyarakat. Tujuan dari penelitian ini untuk mengatasi informasi bencana alam di Indonesia dengan menerapkan algoritma *Long Short-Term Memory* (LSTM) dan model *word embedding*. Dataset yang digunakan dari situs *turnbackhoaks* dan *detik.com* dengan total data 4.032 yang terdiri dari 2.016 data hoaks dan 2.016 data non-hoaks. Penelitian ini menerapkan tiga metode *word embedding* untuk merepresentasikan teks ke dalam bentuk vektor numerik, yaitu *Word2Vec*, *FastText*, dan *GloVe*. Pada *Word2Vec* terdapat dua pendekatan, yakni CBOW (*Continuous Bag of Words*) yang memprediksi kata target berdasarkan kata di sekitarnya, serta Skip-Gram yang menggunakan kata target untuk memperkirakan kata konteksnya, sedangkan *FastText* meningkatkan performa dengan memasukkan informasi sub-kata, sehingga lebih efektif dalam mengolah kosakata yang jarang muncul maupun kata baru dalam korpus. Adapun *GloVe* memanfaatkan statistik *co-occurrence* global untuk menghasilkan representasi vektor yang mampu menangkap hubungan semantik antar kata secara lebih mendalam. Hasil pengujian penelitian ini mendapatkan beraneka ragam nilai akurasi berdasarkan metode *word embedding* dengan algoritma LSTM yang menunjukkan bahwa ketiga model *word embedding* yang dikombinasikan dengan LSTM memiliki performa yang baik, meskipun dengan tingkat akurasi yang

berbeda. *Word2Vec* pada tahap awal mencatat akurasi 0,95, namun setelah evaluasi lebih lanjut performanya meningkat dengan akurasi 0,99 dan *F1-score* 0,99, disertai keseimbangan antara presisi dan *recall*. Model *FastText* menampilkan hasil paling unggul, dengan akurasi awal 0,98 dan hasil akhir mencapai akurasi 0,99 serta *F1-score* 0,99. *FastText* juga memperoleh presisi sempurna (1,00) pada kelas non-hoaks, sehingga mampu mengidentifikasi berita yang valid secara akurat. Sementara itu, *GloVe* menunjukkan performa lebih rendah dibandingkan dua model lainnya, dengan akurasi awal 0,93 dan meningkat menjadi 0,98 pada pengujian akhir dengan *F1-score* 0,98, meskipun *recall* untuk kelas hoaks sedikit lebih rendah yaitu 0,97. Secara keseluruhan, baik *Word2Vec* maupun *FastText* terbukti lebih unggul dibanding *GloVe* dalam klasifikasi berita hoaks dan non-hoaks (Pratama & Basuki, 2025).

#### **2.2.4 Aspek Keagamaan**

Literasi digital sangat dibutuhkan dan tidak hanya mencakup kemampuan teknik tetapi juga membutuhkan kemampuan menilai kebenaran dalam informasi karena banyak isu yang membuat masyarakat sensitif seperti informasi agama. Dalam aspek agama, banyak antar individu yang rentan percaya terhadap terpaparnya hoaks di media sosial karena kurang memperhatikan dan pemahaman dalam menggunakan media sosial. Hoaks dalam aspek keagamaan adalah informasi palsu atau bohong yang disebarluaskan dengan memanfaatkan sentimen yang menyebabkan kesalahpahaman terhadap masyarakat bahkan radikalisasi pemikiran, sehingga membuat masyarakat kebingungan dalam memahami agama yang benar

serta dapat mudah terpengaruh dalam hal negatif seperti dalam toleransi dan prasangka (Arentania *et al.*, 2025).

Penelitian ini berfokus pada aspek keagamaan yaitu mengenai radikalisme menjadi hal penting untuk mengenali karakteristik konten bermuatan radikal di internet, khususnya di Indonesia. Radikalisme tidak hanya muncul dalam tindakan fisik tetapi juga tersebar melalui media sosial sehingga dapat memengaruhi opini publik. Penelitian ini menggunakan metode *text mining* untuk mencari pola kata dalam kalimat informasi. Algoritma yang digunakan yaitu *K-Nearest Neighbor* (KNN) dengan parameter  $k$  untuk menentukan jumlah nilai tetangga terdekat, umumnya menggunakan bilangan ganjil seperti 1, 3, atau 5. Proses klasifikasi menghasilkan label (Merah, Kuning, Hijau, atau Putih) yang ditetapkan berdasarkan hasil voting dari tetangga terdekat tersebut. Data yang diperoleh dengan teknik *scraping* dan penentuan label dilakukan secara manual dan jumlah dataset yaitu 116 sampel. Dalam penelitian ini, hasil pengujian menggunakan algoritma KNN yang diperoleh pada nilai  $k = 7$  akurasi sebesar 66,37% (Nuwairah *et al.*, 2024).

### 2.2.5 Aspek Pendidikan

Teknologi sangat berfungsi terutama di kalangan pelajar karena dapat memberikan dampak positif dalam dunia pendidikan. Dalam aspek pendidikan, hoaks sengaja dirancang untuk mengakali agar remaja percaya terhadap informasi tersebut sehingga dapat memicu kesalahpahaman dan menimbulkan perilaku negatif di kalangan remaja. Oleh karena itu, karakter pendidikan sangat penting dalam menanggulangi informasi hoaks yaitu dengan cara melakukan literasi digital

(Sumantri *et al.*, 2024). Contoh informasi hoaks dalam aspek pendidikan yang di sebarkan di media sosial yaitu mengenai libur sekolah penuh selama bulan Ramadhan 2025, informasi ini tersebar dalam bentuk yang meyakinkan sehingga masyarakat percaya karena informasi disebar dengan tangkapan layar surat edaran dari Kementerian Pendidikan dan Kebudayaan serta lengkap dengan kop surat dan tanda tangan. Pola umum dalam penyebaran hoaks informasi digital konten yang sering memanipulasi dengan desain tampak resmi dan sudah beredar di lingkungan tertutup seperti grup keluarga, komunitas sekolah, dan forum keagamaan (Khasanah *et al.*, 2025).

Analisis efektivitas beberapa algoritma *machine learning* dalam mendeteksi berita hoaks berbahasa Indonesia. Dataset yang digunakan berupa artikel berita online yang telah melalui tahapan *preprocessing* teks, seperti *tokenizing*, *case folding*, *filtering*, *stopword removal*, *stemming*, serta pembobotan menggunakan metode TF-IDF dengan kombinasi unigram dan bigram. Algoritma yang diuji meliputi *Multilayer Perceptron* (MLP), *Naive Bayes* (NB), *Support Vector Machine* (SVM), dan *Random Forest* (RF). Hasil pengujian menunjukkan bahwa SVM memberikan performa terbaik dengan akurasi, presisi, *recall*, dan *F1-score* mencapai 99,91%, diikuti oleh MLP dengan nilai rata-rata 99,88%. Algoritma *Random Forest* juga memperlihatkan kinerja yang stabil dengan skor rata-rata 99,70%, sedangkan *Naive Bayes* memperoleh hasil lebih rendah yakni 97,68%, meskipun tetap unggul dari sisi efisiensi komputasi. Secara keseluruhan, penelitian ini menyimpulkan bahwa SVM merupakan algoritma paling efektif, dengan MLP

sebagai alternatif yang juga sangat baik untuk klasifikasi teks berita hoaks (Desriansyah *et al.*, 2025).

Tabel 2.2 mengenai beberapa refensi penelitian yang menganalisis terkait sistem deteksi hoaks dengan berbagai aspek yang menunjukkan beraneka ragam metode yang digunakan dengan hasil yang berbeda-beda.

Tabel 2.2 Penelitian-Penelitian Deteksi Hoaks Berbasis Aspek

No.	Judul Penelitian	Peneliti	Metode yang digunakan	Hasil
1.	Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma K-Nearest Neighbor (knn) Dan Support Vector Machine (svm)	(DickiPrabowo <i>et al.</i> , 2025)	<i>K-Nearest Neighbor, Support Vector Machine</i>	KNN akurasi 90,73%. SVM lebih baik: akurasi 91,98%, presisi & recall 0,92, <i>F1-score</i> 0,91.
2.	Penerapan Algoritma <i>Support Vector Machine</i> (SVM) untuk Klasifikasi Berita Hoax Covid-19	(Ropikoh <i>et al.</i> , 2021)	<i>Support Vector Machine</i> (SVM)	Kernel linear terbaik: akurasi 97,06% (60:40). Kernel RBF lebih rendah, min. 90,46%.
3.	Klasifikasi Konten Pada Situs Web Dakwah Indonesia Menggunakan Text Mining Untuk Deteksi Paham Islam Radikal	(Nuwairah <i>et al.</i> , 2024b)	<i>K-Nearest Neighbor</i> (KNN)	akurasi 66,37% yang diperoleh pada nilai k = 7
4.	Klasifikasi Hoax Vs Non-Hoax Pada Berita Bencana Alam Berbahasa Indonesia Menggunakan Word Embedding	(Pratama & Basuki, 2025b)	<i>Long-Short-Term Memory</i> (LSTM) dengan metode <i>Word Embedding</i> Word2Vec, FastText, dan GloVe	<i>FastText-LSTM</i> terbaik: akurasi 0,99, <i>F1-score</i> 0,99, presisi 1,00 (kelas non-hoaks).
5.	Analisis Efektivitas Algoritma <i>Machine Learning</i> dalam Deteksi Hoaks: Pada Berita Digital Berbahasa Indonesia	(Desriansyah <i>et al.</i> , 2025)	<i>Multilayer Perceptron</i> (MLP), <i>Naïve Bayes</i> (NB), <i>Support Vector Machine</i> (SVM), dan <i>Random Forest</i> (RF)	SVM terbaik: akurasi 99,91%. MLP 99,88%, RF 99,70%, NB 97,68% (efisien).

### 2.3 Ekstraksi Fitur *Bag-Of-Words*

*Bag-Of-Word* adalah metode yang digunakan untuk mengubah sebuah kalimat ke dalam bentuk vektor berdasarkan kata yang paling muncul pada setiap kalimat tersebut, karena metode ini merupakan bentuk representasi kata untuk mengubah data numerik (Darmawan *et al.*, 2023).

*Bag of Words* merepresentasikan setiap dokumen dengan menghitung frekuensi kemunculan kata. Dalam prosesnya, BoW tidak memperhatikan urutan kata, maupun struktur sintaksis dokumen dan kalimat. Penelitian ini memaparkan bahwa pada *Scikit-learn python*, ekstraksi fitur *Bag of Words* dapat dilakukan melalui modul *CountVectorizer*. Modul ini membentuk matriks yang merepresentasikan dokumen beserta token-token yang terkandung di dalamnya. *CountVectorizer* melakukan proses tokenisasi sekaligus menghitung frekuensi kemunculan setiap token dalam suatu dokumen (Rosanti *et al.*, 2025).

Frekuensi kata dimanfaatkan sebagai pelatihan kalimat, sedangkan proses vektorisasi mengubah data teks menjadi representasi numerik dengan menghitung kemunculan token sehingga terbentuk vektor, dan *Bag of Words* memberikan nilai pada setiap atribut berdasarkan frekuensi tersebut (Ridwan *et al.*, 2024).

### 2.4 Algoritma *Random Forest*

*Random Forest* merupakan salah satu algoritma ensemble learning yang terdiri atas sejumlah pohon keputusan (*desicion tree*) yang dibangun dengan proses pengambilan sampel acak (*random sample selection*) dan pemilihan fitur acak (*random feature selection*). Setiap pohon dilatih menggunakan subset data dan subset fitur yang berbeda, kemudian hasil prediksi dari seluruh pohon digabungkan

menggunakan *majority voting* untuk menghasilkan keputusan akhir klasifikasi. Algoritma ini dapat berfungsi secara efektif meskipun terdapat data yang hilang, karena tersusun atas banyak pohon keputusan sederhana yang berperan sebagai *weak classifiers* namun bekerja secara kolektif (*voting mayoritas*) untuk meningkatkan stabilitas model. *Random Forest* memiliki kemampuan anti-interferensi yang kuat, dapat beroperasi secara efisien pada kumpulan data berukuran besar tanpa memerlukan reduksi dimensi, serta cocok untuk data berdimensi tinggi dan variabel yang saling berinteraksi (Zhang *et al.*, 2021).

Penelitian yang dilakukan oleh Israt Jahan *et al.* (2024) membahas penerapan berbagai algoritma *machine learning* dalam mendeteksi berita palsu (*fake news*) dengan memanfaatkan data teks yang telah melalui tahap *preprocessing* dan ekstraksi fitur menggunakan metode *Term Frequency–Inverse Document Frequency*(*TF-IDF*). Peneliti membandingkan lima algoritma, yaitu *Random Forest*, *Support Vector Machine* (SVM), *Neural Network*, *Logistic Regression*, dan *Naïve Bayes*. Hasil penelitian menunjukkan bahwa algoritma *Random Forest* memberikan performa paling tinggi dengan akurasi sebesar 99,95%, presisi 99,94%, dan *recall* 99,96%, melampaui model lain seperti SVM (99,55%) dan *Neural Network* (99,34%). Hal ini membuktikan bahwa *Random Forest* unggul dalam mengelola data teks berdimensi tinggi, tahan terhadap *overfitting*, dan mampu menghasilkan klasifikasi hoaks dengan akurasi tinggi, sehingga mendukung efektivitas algoritma ini untuk deteksi hoaks otomatis.

Penelitian yang dilakukan oleh Afuan. (2024) berfokus pada pengembangan sistem deteksi jatuh (*fall detection*) menggunakan data sensor dari perangkat

*wearable* seperti *smartwatch*. Penelitian ini mengoptimalkan algoritma *Random Forest* melalui proses *feature engineering* dan *hyperparameter tuning* menggunakan *Grid Search* serta *Randomized Search*, dengan tujuan meningkatkan akurasi dan efisiensi komputasi. Hasil pengujian menunjukkan bahwa model *Random Forest* yang dioptimalkan mencapai akurasi sebesar 92%, presisi 91%, *recall* 89%, dan *F1-score* 90%, sehingga terbukti mampu memberikan hasil prediksi yang cepat dan akurat untuk aplikasi *real-time*.

Penelitian yang dilakukan oleh Almuayqil *et al.* (2022) mengusulkan sebuah *framework* untuk analisis sentimen yang ditingkatkan melalui penerapan *Random Forest Classifier* dengan teknik *random minority oversampling* guna mengatasi masalah ketidakseimbangan kelas pada data *tweet* pengguna. Studi ini menerapkan tahapan *preprocessing* yang sistematis meliputi pembersihan teks, *tokenization*, normalisasi, dan *feature extraction* menggunakan *TF-IDF* sebelum melakukan klasifikasi dengan beberapa algoritma *machine learning*, di antaranya *Random Forest*, *Naïve Bayes*, *Support Vector Machine*, *Gradient Boosting*, dan *XGBoost*. Hasil pengujian menunjukkan bahwa *Random Forest* menghasilkan akurasi tertinggi sebesar 98,3 % dan F1-score 0,98, melampaui model lain seperti SVM yang menghasilkan akurasi sebesar 97,8%. Keunggulan *Random Forest* dalam menangani data teks berdimensi tinggi serta kemampuannya mempertahankan kinerja tinggi meski menghadapi *class imbalance*.

Penelitian Adiguna *et al.* (2024) berfokus pada peningkatan performa analisis sentimen ulasan film menggunakan algoritma *Random Forest* dengan seleksi fitur *Information Gain*. Dataset yang digunakan terdiri dari 500 ulasan film

berbahasa Indonesia yang dibagi menjadi 250 ulasan positif dan 250 ulasan negatif. Setelah melalui tahapan preprocessing seperti *case folding*, *tokenizing*, *stopword removal*, dan pembobotan kata menggunakan TF-IDF, dilakukan seleksi fitur dengan metode *Information Gain* untuk menentukan kata-kata paling relevan terhadap polaritas sentimen. Hasil penelitian menunjukkan bahwa *Random Forest* dengan parameter *Information Gain* menghasilkan akurasi tertinggi sebesar 95,20%, mengungguli *Support Vector Machine* (91%) dan *Naïve Bayes* (69%). Model ini terbukti efektif dalam menangani kompleksitas bahasa dan mampu menghasilkan klasifikasi sentimen yang lebih akurat serta stabil dibandingkan algoritma lain. Oleh karena itu, kombinasi *Random Forest* dengan *feature selection* *Information Gain* dapat meningkatkan kinerja klasifikasi teks secara signifikan.

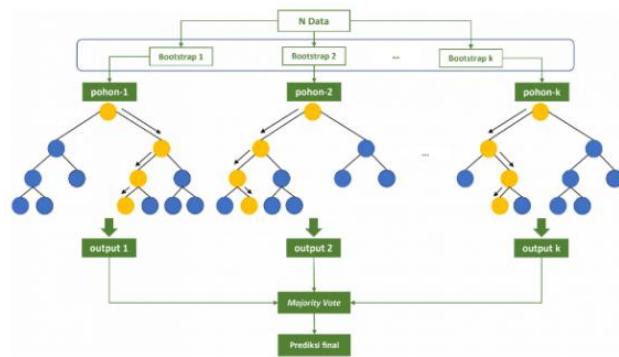
Tabel 2.3 Penelitian-Penelitian Algoritma Random Forest

No.	Judul Penelitian	Peneliti	Metode yang digunakan	Hasil
1.	<i>Advanced Machine Learning Techniques for Fake News Detection: A Comprehensive Analysis</i>	Israt Jahan <i>et al.</i> (2024)	TF-IDF, <i>Random Forest</i> , SVM, <i>Neural Network</i> , <i>Logistic Regression</i> , <i>Naïve Bayes</i>	<i>Random Forest</i> memperoleh akurasi tertinggi 99,95%, unggul dalam menangani data teks berdimensi tinggi dan tahan terhadap <i>overfitting</i> .
2.	<i>Enhanced Fall Detection using Optimized Random Forest Classifier on Wearable Sensor Data</i>	Lasmedi Afuan & R. Rizal Isnanto (2025)	<i>Random Forest</i> , SVM, KNN	<i>Random Forest</i> mencapai akurasi 92%, presisi 91%, dan F1-score 90%; menunjukkan kinerja efisien, stabil, dan akurat setelah optimasi parameter.
3.	<i>Framework for Improved Sentiment Analysis via Random Forest Classifier</i>	Almuayqil <i>et al.</i> (2022)	TF-IDF, <i>Random Forest</i> , SVM, <i>Naïve Bayes</i> , <i>Gradient Boosting</i> , <i>XGBoost</i>	<i>Random Forest</i> menghasilkan akurasi 98,3% dan F1-score 0,98; unggul dalam menghadapi <i>class imbalance</i> dan menjaga stabilitas performa.
4.	<i>Improving Performance Sentiment Analysis</i>	Adiguna <i>et al.</i> (2024)	TF-IDF, <i>Information Gain</i> ,	<i>Random Forest</i> memberikan akurasi

No.	Judul Penelitian	Peneliti	Metode yang digunakan	Hasil
	<i>Movie Review Film using Random Forest with Feature Selection Information Gain</i>		<i>Random Forest, SVM, Naïve Bayes</i>	95,20%, lebih tinggi dari SVM (91%) dan NB (69%), serta efektif menangani kompleksitas bahasa.

Tabel 2.3 dapat disimpulkan bahwa algoritma *Random Forest* terbukti menghasilkan akurasi tinggi pada berbagai penelitian. Oleh karena itu, penelitian ini mengisi kesenjangan tersebut dengan menerapkan *Random Forest* untuk deteksi hoaks pada postingan berbahasa Indonesia berbasis topik.

Berdasarkan keunggulan tersebut, selanjutnya alur kerja algoritma *Random Forest* sebagaimana ditunjukkan pada Gambar 2.1 yaitu tahapan utama dalam algoritma *Random Forest*, yang menggambarkan proses pembentukan model klasifikasi berdasarkan kumpulan data pelatihan (*training data*) hingga menghasilkan prediksi akhir.



Gambar 2.1 Alur Metode Random Forest

Sumber: <https://sainsdata.id/machine-learning/893/random-forest-untuk-model-klasifikasi-menggunakan-scikitlearn-python/>

Proses alur dasar dari algoritma *Random Forest* seperti pada gambar di atas yaitu bekerja berdasarkan konsep *ensemble learning* dengan menggabungkan sejumlah pohon keputusan (*decision tree*) untuk membentuk satu model klasifikasi

yang lebih akurat dan stabil. Setiap pohon tidak dibangun dari keseluruhan data, melainkan dari subset data yang diambil secara acak melalui teknik *bootstrap sampling (sampling with replacement)*, sehingga setiap pohon dilatih menggunakan kombinasi data latih yang berbeda dan memiliki karakteristik unik. Selain itu, pada setiap node pembentukan pohon, algoritma ini juga memilih sejumlah fitur secara acak dan menentukan fitur terbaik sebagai pemisah berdasarkan kriteria seperti *Gini Index* atau *Entropy*, sehingga struktur tiap pohon berbeda dan risiko *overfitting* dapat diminimalkan.

Menurut Thelagathoti *et al.* (2025), mekanisme tersebut memungkinkan *Random Forest* mengevaluasi pentingnya setiap fitur berdasarkan kontribusinya terhadap akurasi model menggunakan metrik seperti *Mean Decrease Gini (MDG)* atau *Mean Decrease Accuracy (MDA)*, yang menjadikan model lebih stabil dan tahan terhadap *overfitting*. Setelah seluruh pohon terbentuk, hasil klasifikasi dari masing-masing pohon digabungkan melalui mekanisme *majority voting* untuk klasifikasi atau *averaging* untuk regresi, di mana hasil akhir diambil dari kelas dengan jumlah suara terbanyak atau nilai rata-rata. Dengan mekanisme tersebut, *Random Forest* memperoleh kekuatan prediksinya dari keragaman pohon yang berasal dari berbagai subset data dan fitur acak, sehingga mampu memberikan hasil yang stabil, akurat, dan tahan terhadap *overfitting*.

## **BAB III**

### **DESAIN DAN IMPLEMENTASI SISTEM**

#### **3.1 Pengumpulan Data**

Pada tahapan ini, pengumpulan data dilakukan melalui penggunaan data sekunder. Dalam penelitian ini, data yang digunakan yaitu dataset “*Indonesia False News(Hoax) Dataset*” yang sudah terverifikasi dari *website turnbackhoax* dan merupakan *dataset open acces* dari situs *Kaggle* dataset ini mencakup 733 data postingan yang berbahasa Indonesia dengan 6 fitur dimana 3 atributnya yaitu tanggal, id, judul, dan nama file gambar tidak digunakan karena tidak mempengaruhi hasil penelitian ini (Muharam, 2020).

Atribut-atribut tersebut menyediakan informasi penting yang diperlukan dalam proses analisis klasifikasi postingan hoaks.

Tabel 3.1 Penjelasan Fitur

Fitur	Keterangan
Id	Identifikasi unik untuk setiap data atau postingan
Label	Kategori data yang menunjukkan apakah postingan termasuk hoaks atau non-hoaks (variable target/output)
Tanggal	Informasi waktu publish postingan
Judul	Judul postingan
Narasi	Isi dari postingan
Nama File Gambar	Nama file gambar terkait postingan

Tabel 3.1 menjelaskan fitur *label* yang digunakan sebagai variabel target yang menunjukkan kategori postingan hoaks dan non-hoaks, sedangkan fitur *narasi* dimanfaatkan sebagai input utama dalam proses klasifikasi karena berisi informasi teks yang dianalisis. Sementara itu, fitur *id*, *tanggal*, *judul*, dan *nama file gambar* tidak digunakan dalam pemodelan karena tidak berpengaruh langsung terhadap

proses klasifikasi. Pemilihan fitur ini bertujuan untuk memfokuskan model pada atribut yang relevan sehingga proses deteksi hoaks dapat dilakukan secara lebih efektif.

Tabel 3.2 Contoh Dataset Postingan Hoaks

<b>Id</b>	<b>Label</b>	<b>Tanggal</b>	<b>Judul</b>	<b>Narasi</b>	<b>Nama File Gambar</b>
71	1	17/8/2020	Pemakaian Masker Menyebabkan Penyakit Legionnaires	A caller to a radio talk show recently shared that his wife was hospitalized n told she had COVID n only a couple of days left to live	71.jpg
461	1	17/7/2020	Instruksi Gubernur Jateng tentang penilangan bagi yg tidak bermasker di muka umum Rp.150.000 menggunakan e-tilang Via apps PIKOBAR	Yth.Seluruh Anggota Grup Sesuai Instruksi Gubernur Jawa Tengah Hasil Rapat Tim Gugus Tugas Covid 19 Jateng sbb:	461.png
495	1	13/7/2020	Foto Jim Rohn: Jokowi adalah presiden terbaik dlm sejarah bangsa Indonesia	Jokowi adalah presiden terbaik dlm sejarah bangsa Indonesia. Jim Rohn, motivator terbaik dunia	495.png
...	...	...	...	...	...
733	0	25/10/2017	Sandi Sarankan Warga Ngadu ke Kelurahan Saja	Astaga! Sandiaga Sarankan Warga Ngadu ke Kelurahan Saja Tak Perlu ke Balai Kota	6775.jpg

Tabel 3.2 menampilkan contoh data postingan yang digunakan dalam penelitian ini, yang terdiri dari atribut *id*, *label*, *tanggal*, *judul*, *narasi*, dan *nama file gambar*. Pada tabel tersebut, nilai *label* menunjukkan kategori postingan, yaitu hoaks (1) dan non-hoaks (0). Contoh data ini menggambarkan variasi isi teks pada kolom *narasi* yang menjadi fokus utama dalam proses klasifikasi, karena memuat informasi yang dianalisis untuk membedakan postingan hoaks dan non-hoaks.

Pelabelan untuk data tiap aspek akan dilakukan dengan sistem berdasarkan kata kunci. Aspek yang digunakan yaitu pendidikan, politik, dan bencana alam.

Pada aspek pendidikan menggunakan kata kunci 'sekolah', 'kampus', 'universitas', 'guru', 'dosen', 'mahasiswa', 'pendidikan', 'kurikulum', 'beasiswa', 'ujian', 'belajar', 'kelas', 'murid', 'akreditasi', 'pelajar', 'edukasi', 'siswa', 'paud', 'sd', 'smp', 'sma', 'smk', 'kuliah', 'skripsi', 'tesis' (Bulut *et al.*, 2023). Pada aspek politik menggunakan kata kunci 'presiden', 'wakil', 'pemerintah', 'politik', 'partai', 'menteri', 'dpr', 'parlemen', 'pilkada', 'pemilu', 'kampanye', 'kabinet', 'legislatif', 'caleg', 'birokrat', 'kandidat', 'pemerintahan', 'dewan', 'politik uang', 'koalisi', 'oposisi', 'peraturan', 'undang', 'uu', 'capres' (Chandra & Yusuf, 2024). Sedangkan, pada aspek bencana alam menggunakan kata kunci 'gempa', 'anjir', 'longsor', 'kebakaran', 'tsunami', 'letusan', 'gunung', 'evakuasi', 'korban', 'bencana', 'cuaca ekstrem', 'topan', 'angin', 'puting beliung', 'darurat', 'siaga', 'banjir bandang', 'tanah longsor', 'gelombang tinggi', 'abrasi', 'hujan deras' (Algiriyage *et al.*, 2021).

Berdasarkan kata kunci pada masing-masing kategori aspek tersebut, dilakukan proses pelabelan terhadap data hoaks sehingga setiap data memiliki label aspek yang sesuai. Adapun hasil dari proses pelabelan aspek dapat dilihat pada

Tabel 3.3 berikut:

Tabel 3.3 Data Hoaks Berbasis Aspek

Label	Judul	Narasi	Aspek	Label Aspek
1	mahasiswa usir para penumpang gelap ini	Ayo teman-teman mahasiswa usir para penumpang gelap ini dari barisan kalian!	Pendidikan	0
1	Hasil Scanning Port Bolong	KPU harusnya punya server sendiri. Ini bawa2 ITB, Moratel.. Gimana pemilu berikutnya...	Politik	1
1	Gambar Salib Pada Barang Bantuan Bencana	hati hati, dan Harus teliti menerima segala bentuk bantuan korban gempa. Kami mendapatkan gambar salib di handuk bantuan yg di edarkan	Bencana Alam	2

Tabel 3.3 menunjukkan data hoaks yang telah diberi label berdasarkan aspek masing-masing kategori, di mana nilai 0 merepresentasikan aspek Pendidikan, nilai 1 menunjukkan aspek Politik, dan nilai 2 menunjukkan aspek Bencana Alam.

Hasil pelabelan aspek pada Tabel 3.3, label numerik yang dihasilkan kemudian diubah ke dalam bentuk representasi vektor agar dapat digunakan pada tahap pelatihan model. Selanjutnya, hasil pelabelan numerik tersebut dikonversi menjadi bentuk representasi vektor biner menggunakan teknik *One Hot Encoding*. Teknik ini digunakan untuk mengubah label kategorikal menjadi format numerik yang dapat diproses oleh algoritma *Random Forest*. Dalam penelitian ini terdapat tiga kategori utama, sehingga setiap label direpresentasikan ke dalam vektor biner sepanjang tiga elemen, di mana nilai 1 menunjukkan kelas yang aktif dan nilai 0 menunjukkan kelas lainnya. Misalnya, aspek pendidikan dengan label numerik 0 direpresentasikan sebagai [1, 0, 0], aspek politik dengan label 1 direpresentasikan sebagai [0, 1, 0], dan aspek bencana alam dengan label 2 direpresentasikan sebagai [0, 0, 1]. Nilai tersebut dihasilkan dengan menempatkan angka 1 pada posisi indeks yang sesuai dengan kelas aktif, sedangkan posisi lainnya diisi dengan nilai 0. Adapun hasil konversi label aspek ke dalam bentuk *One Hot Encoding* dapat dilihat pada Tabel 3.4 berikut.

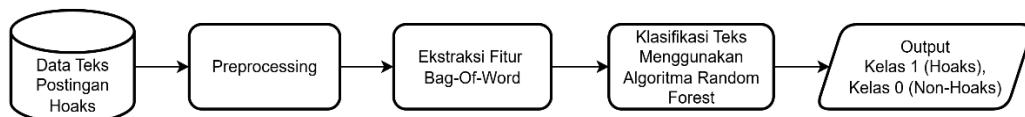
Tabel 3.4 *One Hot Encoding* Label Aspek

Label	Aspek	Label Aspek	One Hot Encoding
1	Pendidikan	0	[1, 0, 0]
1	Politik	1	[0, 1, 0]
1	Bencana Alam	2	[0, 0, 1]

Tabel 3.4 menunjukkan representasi hasil transformasi label numerik menjadi format *One Hot Encoding*. Setiap kategori aspek memiliki identitas numerik yang unik tanpa menunjukkan urutan atau hubungan hierarkis antar kelas, sehingga model dapat mengenali setiap label secara independen selama proses pelatihan dan klasifikasi.

### 3.2 Desain Sistem

Desain sistem dalam penelitian ini merupakan rancangan alur kerja yang menjelaskan langkah-langkah yang dilakukan oleh sistem. Tahapan dimulai dari input data postingan hoaks hingga tahap akhir, yaitu menghasilkan output. Secara umum, alur penelitian ditunjukkan pada Gambar 3.2.



Gambar 3.1 Desain Sistem

Gambar 3.1 merupakan tahapan penelitian ini diawali dengan input data postingan hoaks. Setelah data diinputkan, maka tahap selanjutnya adalah *preprocessing* yang meliputi *cleaning data*, *case folding*, *tokenizing*, *stopword removal*, normalisasi, dan *stemming*. *Preprocessing* ini dilakukan karena mencakup pembersihan dan persiapan data untuk di analisis lebih lanjut. Tahap selanjutnya yaitu ekstraksi fitur *Bag-of-Word* yang digunakan untuk mengubah data teks menjadi vektor numerik dengan menghitung frekuensi kemunculan kata dalam teks, yang digunakan untuk melatih model. Tahap terakhir adalah implementasi algoritma *Random Forest* yang digunakan untuk melakukan klasifikasi dengan

membangun sejumlah pohon keputusan atau *decision tree*. Melalui tahapan ini model akan mengklasifikasi teks berupa kelas 1 (hoaks) dan kelas 0 (non-hoaks).

### 3.3 *Preprocessing*

Data *preprocessing* adalah rangkaian tahapan yang dilakukan pada data mentah untuk mempersiapkan sebelum dianalisis lebih lanjut oleh model. *Preprocessing* ini bertujuan untuk memperbaiki kualitas data dan meminimalkan kekurangan yang terdapat dalam data mentah (Sari *et al.*, 2025). Adapun tahapan *preprocessing* sebagai berikut:

#### 3.3.1 *Cleaning Data*

Tahap *cleaning* merupakan bagian dari langkah awal yang dilakukan dalam *preprocessing*. Tahap *Cleaning* ini digunakan untuk membersihkan teks dari karakter atau elemen yang tidak penting seperti tanda baca, angka, maupun simbol sehingga data menjadi lebih terstruktur untuk dianalisis lebih lanjut. Proses ini menggunakan *library Regular Expression* (re) pada python dengan memanfaatkan ekspresi reguler untuk mengenali serta menghapus elemen yang tidak diperlukan.

Tabel 3.5 Proses Data Cleaning

Sebelum Data Cleaning	Setelah Data Cleaning
selamat siang teman teman fb ku semuanya 😊,cuman mau memberi informasi dan rezeky gak kemana ya 😊, kemarin saya mengikuti event promo smartphone JNE 2020 spesial di bulan juni ini dan saya mesan smartphone VIVO V15 PRO cuman seharga Rp.900.000,-,gak nyangka cuma seharga Rp.900.000 aku bisa mendapat hp VIVO V15 PRO 😊,bagi teman teman fb ku yg penasaran atau mau ikutan INBOX AJA WA ADMINNYA 0823-7142-9***	selamat siang teman teman fb ku semuanya cuman mau memberi informasi dan rezeky gak kemana ya kemarin saya mengikuti event promo smartphone JNE spesial di bulan juni ini dan saya mesan smartphone VIVO V PRO cuman seharga Rp gak nyangka cuma seharga Rp aku bisa mendapat hp VIVO V PRO bagi teman teman fb ku yg penasaran atau mau ikutan INBOX AJA WA ADMINNYA
Jokowi adalah presiden terbaik dlm sejarah bangsa Indonesia. Jim Rohn, motivator terbaik dunia	Jokowi adalah presiden terbaik dlm sejarah bangsa Indonesia Jim Rohn motivator terbaik dunia

Tabel 3.5 menunjukkan perbandingan data teks sebelum dan sesudah dilakukan proses cleaning. Pada tahap ini, karakter yang tidak relevan seperti simbol, tanda baca, dan elemen non-teksual dihilangkan sehingga teks menjadi lebih bersih dan mudah diproses. Hasil cleaning menghasilkan teks yang lebih terstruktur dan siap digunakan pada tahap preprocessing selanjutnya.

### 3.3.2 *Case Folding*

Tahap *case folding* dalam *preprocessing* teks merupakan proses mengonversi seluruh karakter huruf menjadi bentuk huruf kecil. Tujuan *case folding* adalah mengurangi perbedaan yang disebabkan oleh penggunaan huruf kapital maupun huruf kecil, sehingga mempermudah pengolahan teks. *Case folding* menggunakan *method lower()* untuk mengubah setiap huruf dalam string menjadi huruf kecil tanpa memengaruhi karakter lain seperti angka maupun tanda baca.

Tabel 3.6 Proses Data Case Folding

Sebelum Data <i>Case Folding</i>	Setelah Data <i>Case Folding</i>
Jokowi adalah presiden terbaik dlm sejarah bangsa Indonesia Jim Rohn motivator terbaik dunia	jokowi adalah presiden terbaik dlm sejarah bangsa indonesia jim rohn motivator terbaik dunia
YthSeluruh Anggota Grup Sesuai Instruksi Gubernur Jawa Tengah Hasil Rapat Tim Gugus Tugas Covid Jateng sbb	ythseluruh anggota grup sesuai instruksi gubernur jawa tengah hasil rapat tim gugus tugas covid jateng sbb

Tabel 3.6 menunjukkan perubahan teks sebelum dan sesudah dilakukan proses *case folding*. Pada tahap ini, seluruh huruf pada teks dikonversi menjadi huruf kecil sehingga perbedaan penulisan akibat penggunaan huruf kapital dapat diminimalkan. Proses ini bertujuan untuk menyamakan representasi kata agar pengolahan teks pada tahap selanjutnya dapat dilakukan secara lebih konsisten.

### 3.3.3 Normalisasi

Normalisasi dalam *preprocessing* teks adalah tahap mengubah kata-kata tidak baku menjadi bentuk baku sesuai aturan bahasa. Tujuan normalisasi yaitu menjaga konsistensi data agar lebih mudah dianalisis. Proses ini umumnya dilakukan dengan mencocokkan kata tidak baku pada teks dengan daftar kata baku yang tersedia dalam file referensi.

Tabel 3.7 Proses Data Normalisasi

Sebelum Data Normalisasi	Setelah Data Normalisasi
polisi goblok bukanya cpet d tolongin malah d liati aja	polisi goblok bukan cepat di tolongin malah di liati aja
polisi china telah menganiaya wanita uighur lehernya di cekik pake kaki melek tu yg bilang tidak ada penganiayaan dan ham diam seribu bahasa seolah polisi itu berkata liat nanti giliran anda sama orang di sekitarnya biadab	polisi china telah menganiaya wanita uighur lehernya di cekik pakai kaki melek itu yang bilang tidak ada penganiayaan dan ham diam seribu bahasa seolah polisi itu berkata liat nanti giliran anda sama orang di sekitarnya biadab

Tabel 3.7 memperlihatkan perbandingan teks sebelum dan sesudah dilakukan proses normalisasi. Pada tahap ini, kata-kata tidak baku diubah menjadi bentuk baku sesuai kaidah bahasa, sehingga konsistensi penulisan dapat terjaga. Proses normalisasi ini membantu mengurangi variasi kata yang memiliki makna sama, sehingga mempermudah analisis teks pada tahap selanjutnya.

### 3.3.4 Tokenizing

*Tokenizing* adalah tahap memecah teks menjadi bagian-bagian kecil seperti kata, frasa, atau kalimat yang disebut token. Tujuan dari *tokenizing* yaitu untuk memudahkan analisis dengan mengenali elemen-elemen dasar dalam teks. Proses ini membantu memahami struktur teks serta mempersiapkannya untuk tahap pemrosesan berikutnya.

Tabel 3.8 Proses Tokenizing

Sebelum Data <i>Tokenizing</i>	Setelah Data <i>Case Folding</i>
jokowi adalah presiden terbaik dalam sejarah bangsa indonesia jim rohn motivator terbaik dunia	“Jokowi”, “adalah”, “presiden”, “terbaik”, “dalam”, “sejarah”, “bangsa”, “indonesia”, “jim”, “rohn”, “motivator”, “terbaik”, “dunia”
turut berduka cita atas meninggalnya pejuang angkatan bapak cosmas batubara hari sabtu juli pukul wib semoga yang maha kuasa mengampuni segala dosanya dan mendapat tempat yang layak disisi nya aamiin	“turut”, “berduka”, “cita”, “atas”, “meninggalnya”, “pejuang”, “angkatan”, “bapak”, “cosmas”, “batubara”, “hari”, “sabtu”, “juli”, “pukul”, “wib”, “semoga”, “yang”, “maha”, “kuasa”, “mengampuni”, “segala”, “dosanya”, “dan”, “mendapat”, “tempat”, “yang”, “layak”, “disisi”, “nya”, “aamiin”

Tabel 3.8 menunjukkan hasil proses *tokenizing*, yaitu pemecahan teks menjadi satuan kata atau token. Pada tahap ini, kalimat yang sebelumnya masih berbentuk teks utuh diubah menjadi daftar kata yang terpisah. Proses *tokenizing* bertujuan untuk mempermudah pengolahan teks pada tahap selanjutnya, seperti penghapusan kata tidak penting dan pembobotan kata.

### 3.3.5 *Stopwords Removal*

*Stopword removal* merupakan tahap menghapus kata-kata umum yang tidak memberikan informasi penting, seperti 'dan', 'di', atau 'yang', dari teks. Langkah ini bertujuan meningkatkan efisiensi analisis dengan menyisakan kata yang lebih relevan terhadap konteks. Dalam penerapannya, dapat digunakan *library* seperti Sastrawi untuk memperoleh daftar *stopword* yang bisa disesuaikan dengan kebutuhan, sehingga proses pembersihan teks lebih optimal dan fokus pada kata-kata bermakna dalam analisis. Setiap hasil token dari proses *tokenizing* akan diperiksa apakah masuk ke dalam daftar *stopword*. Apabila token tersebut tercantum dalam daftar *stopword*, maka akan dihapus dan tidak disertakan dalam

proses selanjutnya. Sebaliknya, jika token tidak terdapat dalam daftar tersebut, maka token akan diteruskan ke proses lanjutan untuk dianalisis lebih mendalam.

Tabel 3.9 Proses Stopword Removal

Sebelum <i>Stopword removal</i>	Setelah <i>Stopword removal</i>
“fadlizon”, “semua”, “warga”, “indonesia”, “yang”, “tidak”, “pilih”, “Prabowo”, “di”, “pilpres”, “akan”, “kamu”, “usir”, “dari”, “indonesia”, “serius”, “ini”, “si”, “zonk”	“fadlizon”, “semua”, “warga”, “indonesia”, “pilih”, “prabowo”, “pilpres”, “kamu”, “usir”, “indonesia”, “serius”, “si”, “zonk”
“pada”, “april”, “lalu”, “menteri”, “perdagangan”, “rachmat”, “gobel”, “memberlakukan”, “kebijakan”, “larangan”, “penjualan”, “minuman”, “beralkohol”, “alias”, “minuman”, “keras”, “di”, “minimarket”, “seluruh”, “indonesia”, “namun”, “larangan”, “yang”, “tertuang”, “dalam”, “peraturan”, “menteri”, “perdagangan”, “permendag”, “nomor”, “mdagper”, “tentang”, “pengendalian”, “dan”, “pengawasan”, “terhadap”, “pengadaan”, “peredaran”, “dan”, “penjualan”, “minuman”, “beralkohol”, “itu”, “kini”, “akan”, “diubah”	“april”, “lalu”, “menteri”, “perdagangan”, “rachmat”, “gobel”, “memberlakukan”, “kebijakan”, “larangan”, “penjualan”, “minuman”, “beralkohol”, “alias”, “minuman”, “keras”, “minimarket”, “seluruh”, “indonesia”, “larangan”, “tertuang”, “peraturan”, “menteri”, “perdagangan”, “permendag”, “nomor”, “mdagper”, “pengendalian”, “pengawasan”, “pengadaan”, “peredaran”, “penjualan”, “minuman”, “beralkohol”, “kini”, “diubah”

Tabel 3.9 menunjukkan hasil proses *stopword removal*, yaitu penghapusan kata-kata umum yang tidak memiliki kontribusi signifikan terhadap makna teks. Pada tahap ini, kata seperti kata hubung dan kata ganti dihilangkan sehingga hanya tersisa kata-kata yang lebih informatif. Proses ini bertujuan untuk mengurangi noise dan meningkatkan kualitas fitur teks yang digunakan pada tahap klasifikasi.

### 3.3.6 *Stemming*

*Stemming* adalah tahap yang bertujuan mengubah kata-kata dalam teks menjadi bentuk dasar (*root word*). Proses ini biasanya dilakukan dengan memanfaatkan modul *StemmerFactory* dari *library Sastrawi* yang menyediakan fitur *stemming*. Dalam tahap ini, setiap token yang diperoleh dari proses sebelumnya akan diperiksa terhadap kamus kata dasar. Apabila token tersebut

terdapat dalam kamus kata dasar, maka kata itu dianggap sudah berbentuk dasar dan tidak perlu diubah lebih lanjut. sebaliknya, apabila token tidak ditemukan dalam kamus kata dasar, token tersebut dikategorikan sebagai kata berimbuhan, sehingga dilakukan proses *stemming* dengan menghapus *prefix* (awalan), *infix* (sisipan), dan *suffix* (akhiran). Langkah ini berfungsi menyederhanakan analisis teks dengan menghapus variasi kata yang memiliki arti sama, sehingga mempermudah proses pengelompokan serta pemahaman konteks dari data teks.

Tabel 3.10 Proses Stemming

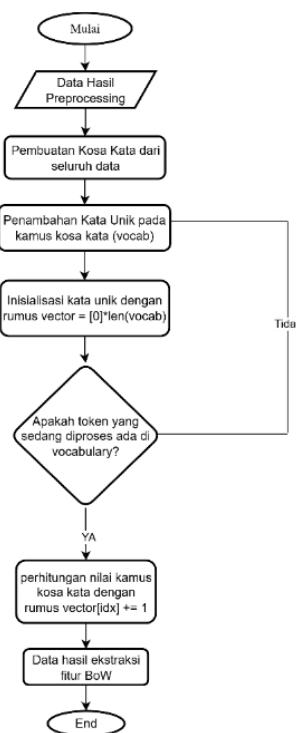
Sebelum Stemming	Setelah Stemming
“Jokowi”, “presiden”, “terbaik”, “Sejarah”, “bangsa”, “indonesia”, “jim”, “rohn”, “motivator”, “terbaik”, “dunia”	“jokowi”, “presiden”, “baik”, “Sejarah”, “bangsa”, “indonesia”, “jim”, “rohn”, “motivator”, “baik”, “dunia”
“begal”, “mulai”, “merambat”, “wilayah”, “cianjur”, “wilayah”, “rewan”, “begal”, “harapkan”, “waspada”, “penuh”, “terutama”, “malam”, “hari”, “jalur”, “gekbrong”, “arah”, “sukabmi”, “cibeber”, “arah”, “campaka”, “sampe”, “cilaku”	“begal”, “mulai”, “rambat”, “wilayah”, “cianjur”, “wilayah”, “rewan”, “begal”, “harap”, “waspada”, “penuh”, “utama”, “malam”, “hari”, “jalur”, “gekbrong”, “arah”, “sukabmi”, “cibeber”, “arah”, “campaka”, “sampe”, “cilaku”

Tabel 3.10 menunjukkan hasil proses *stemming*, yaitu pengubahan kata berimbuhan menjadi bentuk kata dasar. Pada tahap ini, variasi kata yang memiliki makna sama disederhanakan sehingga jumlah fitur dapat dikurangi. Proses *stemming* bertujuan untuk meningkatkan konsistensi representasi kata dan membantu model dalam menangkap makna utama teks pada tahap klasifikasi.

### 3.4 Bag Of Words

Setelah dilakukan *preprocessing*, maka akan dihasilkan kumpulan kata yang akan dijadikan aturan untuk proses selanjutnya yaitu mengubah data teks menjadi vektor numerik berdasarkan frekuensi kemunculan pada setiap kata tanpa mempertimbangkan tata bahasa dan urutan kata yaitu dengan menggunakan

ekstraksi fitur *Bag Of Word*. Tahap ini dilakukan dengan menggabungkan setiap kata unik yang muncul dalam korpus ke suatu indeks, kemudian menghitung frekuensi kemunculan kata tersebut pada setiap dokumen. Pada representasi BoW, setiap dokumen dianggap sebagai kumpulan kata tanpa mempertimbangkan urutannya serta direpresentasikan dalam bentuk vektor pada ruang dengan dimensi sebanyak jumlah kata berbeda dalam seluruh korpus. Sebagai contoh, apabila korpus memiliki seribu kata unik, maka tiap dokumen digambarkan sebagai vektor berdimensi seribu, dimana setiap elemen vektor merepresentasikan frekuensi kemunculan kata tertentu dalam dokumen tersebut, berdasarkan index yang diberikan kepada setiap kata unik dalam daftar kosakata yang digunakan untuk implementasi BoW. Alur ekstraksi fitur *Bag-Of-Word* dapat dilihat pada Gambar 3.2 sebagai berikut.



Gambar 3.2 Flowchart *Bag Of Word*

Gambar 3.2 alur ekstraksi fitur *Bag Of Word* yang dilakukan setelah tahap *preprocessing*. Proses diawali dengan pembuatan Kamus Kosa Kata (KKK) untuk menampung kata unik yang ada pada data. kemudian tahap kedua setiap kata unik yang ditemukan ke dalam kamus kosa kata sehingga membentuk kumpulan kata yang akan digunakan untuk representasi. Tahap ketiga yaitu inisialisasi kata unik dengan nilai 0, kemudian dilakukan looping atau perulangan untuk memastikan apakah token terdapat dalam kamus kosa kata, jika token tersebut sesuai dengan kamus kosa kata maka nilai pada indeks kata yang bersangkutan akan ditambahkan 1, sebaliknya apabila token tidak sesuai dengan kamus kosa kata maka akan di proses kembali ke tahapan penambahan kata unik kamus kosa kata. Hal ini akan terus dilakukan sampai dokumen selesai diproses. Hasil akhir dari alur ini adalah data representasi BoW berupa vektor frekuensi kata yang menggambarkan seberapa sering setiap kata unik muncul dalam dokumen, sehingga dokumen teks dapat diubah menjadi bentuk numerik.

Tabel 3.11 Contoh Data Postingan

Narasi	Label
Instruksi gubernur wajib masker di jalan umum tanpa dasar resmi	1
Rapat resmi pemerintah bahas vaksinasi covid di kantor gubernur	0
Program vaksin covid dilaksanakan resmi oleh dinas kesehatan	0

Tabel 3.11 menampilkan contoh data postingan yang telah melalui tahap *preprocessing* dan siap digunakan dalam proses ekstraksi fitur. Pada tabel ini, kolom *narasi* berisi teks hasil pengolahan, sedangkan kolom *label* menunjukkan kelas postingan, yaitu hoaks (1) dan non-hoaks (0). Data ini menjadi input awal sebelum dilakukan transformasi teks ke dalam bentuk vektor numerik.

Dari data tersebut akan diubah menjadi vektor numerik dengan menggunakan ekstraksi fitur BoW. Cara ini dilakukan dengan menghitung frekuensi kemunculan kata dalam data. Dari data diatas akan dilakukan penambahan kosa kata berdasarkan kata unik.

Tabel 3.12 Contoh Kata Unik

Kata Unik	Jumlah
["instruksi", "gubernur", "masker", "resmi", "vaksin", "covid"]	6 fitur

Tabel 3.12 menunjukkan daftar kata unik yang diperoleh dari seluruh data narasi setelah proses *preprocessing*. Kata unik ini membentuk kosakata (*vocabulary*) yang digunakan dalam metode *Bag-of-Words*. Jumlah kata unik menentukan jumlah fitur yang akan direpresentasikan dalam bentuk vektor numerik pada tahap selanjutnya.

Dari kata unik yang sudah ditentukan yaitu berjumlah 10 kata unik dari data tersebut. Setelah menentukan kata unik, maka representasikan setiap dokumen ke dalam vektor frekuensi.

Tabel 3.13 Contoh Data Vektor Frekuensi

Doc	Instruksi	Gubernur	Masker	Resmi	Vaksin	Covid	Label
Doc 1	1	1	1	1	0	0	1 (hoaks)
Doc 2	0	1	0	1	1	1	0 (non-hoaks)
Doc 3	0	0	0	1	1	1	0 (non-hoaks)

Berdasarkan Tabel 3.13 setiap narasi teks telah dikonversi menjadi representasi numerik dengan ekstraksi fitur *Bag Of Word*, dimana kosa kata unik yang telah ditentukan pada Tabel 3.12 dijadikan sebagai fitur. Setiap dokumen digambarkan melalui jumlah kemunculan kata tersebut. Pada Tabel 3.13 terdapat 3 data, dokumen 1 terdapat kata unik yaitu instruksi, gubernur, dan maker masing-

masing kata muncul 1 kali dengan label hoaks, dokumen 2 terdapat kata unik gubernur, resmi, covid dengan label non-hoaks, dan dokumen 3 terdapat kata resmi, vaksin, dan covid dengan label non-hoaks. Data yang awalnya berupa kalimat telah di konversi menjadi vektor numerik sehingga dapat di proses lebih lanjut oleh algoritma *Random Forest* untuk melakukan klasifikasi.

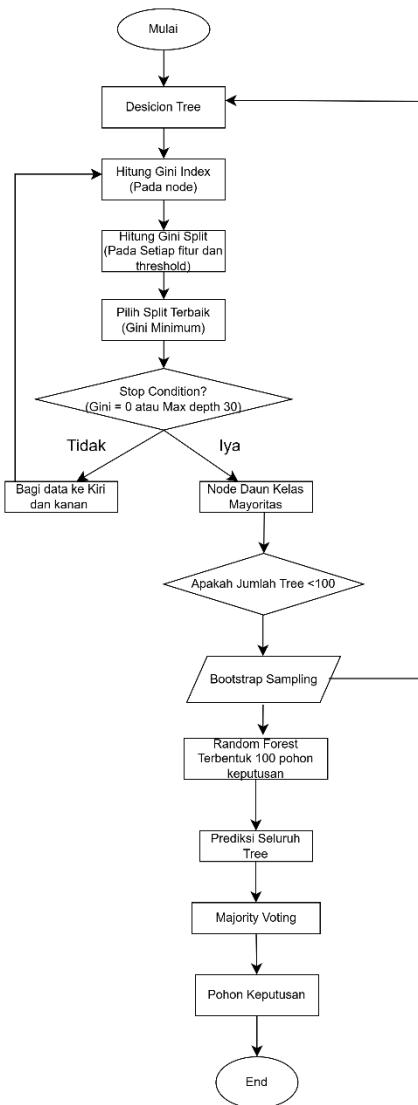
### 3.5 Algoritma *Random Forest*

Berdasarkan dataset yang digunakan dengan 2 fitur yaitu label 0 sebagai non-hoaks dan label 1 sebagai hoaks. Pada penelitian ini, menerapkan algoritma *Random Forest* untuk membangun sejumlah *decision tree* dan hasilnya akan di prediksi berdasarkan voting dari setiap pohon.

*Random Forest* adalah algoritma *machine learning* yang mempunyai banyak pohon keputusan dengan menggabungkan metode *bagging (bootstrap aggregating)* dengan pemilihan fitur acak (*subspaces*). Menurut Breiman, seorang ilmuan yang pertama kali mengemukakan konsep *Random Forest* yaitu dengan membentuk sejumlah pohon keputusan yang dibangun secara acak dari sebagian fitur data (Momole, 2022). Pohon keputusan merupakan kumpulan pertanyaan yang disusun secara terstruktur yang dimana setiap pertanyaan menghasilkan cabang berdasarkan nilai dari suatu atribut dan berhenti pada daun pohon yang menunjukkan hasil prediksi dari kelas variabel.

Dalam algoritma *Random forest* terdapat tiga aspek utama yaitu penggunaan sampel individu pada setiap atribut yang dipilih secara acak untuk membangun pohon keputusan. Setiap jumlah pohon keputusan yang diinisialkan sebagai  $n$  melakukan prediksi dan memberikan keputusan sesuai dengan model prediktor

yang terbentuk. Selanjutnya, *Random Forest* menerapkan proses *majority voting* untuk menentukan hasil akhir prediksi berdasarkan kombinasi prediksi dari seluruh jumlah ( $n$ ) pohon keputusan yang dihasilkan (Ferdita Nugraha *et al.*, 2022). Berikut gambaran alur dari algoritma *Random Forest*:



Gambar 3.3 Desain Algoritma Random Forest

Gambar 3.3 Proses klasifikasi menggunakan algoritma *Random Forest* diawali dengan pembentukan pohon keputusan (*Decision Tree*) sebagai model dasar. Pada setiap node, sistem menghitung nilai *Gini Index* untuk mengukur

tingkat ketidakmurnian data, kemudian mengevaluasi seluruh kemungkinan *split* pada setiap fitur dan nilai ambang (*threshold*). Split terbaik dipilih berdasarkan nilai *Gini Index* paling rendah karena mampu memisahkan kelas secara optimal.

Proses pembentukan node akan dihentikan apabila nilai Gini mencapai nol atau kedalaman maksimum pohon (*max depth*) telah tercapai. Jika kondisi penghentian belum terpenuhi, data akan dibagi ke cabang kiri dan kanan, sedangkan jika terpenuhi, node akan menjadi node daun dengan kelas ditentukan berdasarkan kelas mayoritas.

Setiap pohon dibangun menggunakan teknik *bootstrap sampling*, dan proses ini diulang hingga jumlah pohon mencapai 100 ( $n\_estimators = 100$ ). Setelah seluruh pohon terbentuk, masing-masing pohon menghasilkan prediksi terhadap data uji. Hasil prediksi akhir ditentukan menggunakan mekanisme *majority voting*, yaitu kelas yang paling banyak dipilih oleh seluruh pohon keputusan ditetapkan sebagai hasil klasifikasi.

Tahap terakhir adalah evaluasi performa model, yang dilakukan dengan menghitung metrik evaluasi seperti *accuracy* dan metrik klasifikasi lainnya. Evaluasi ini bertujuan untuk mengukur tingkat keberhasilan sistem dalam mendeteksi hoaks pada artikel berita berdasarkan topik yang dianalisis. Setelah proses evaluasi selesai, alur sistem dinyatakan berakhir.

Menurut (Barreñada *et al.*, 2024), mekanisme kerja *Random Forest* dalam melakukan estimasi probabilitas dapat dijelaskan melalui beberapa tahapan berikut:

- Proses pembentukan *Random Forest* diawali dengan pembuatan sejumlah pohon keputusan (*ntree*). Setiap pohon dilatih menggunakan subset data yang

diambil secara acak dari dataset pelatihan menggunakan metode *bootstrap sampling (sampling with replacement)*. Teknik ini memungkinkan setiap pohon memiliki kombinasi data latih yang berbeda, sehingga meningkatkan keragaman model dan mengurangi korelasi antar pohon.

Setiap pohon dibangun dari subset data acak yang diambil dengan pengembalian.

$$D_b = \{(x_i, y_i)\}_{i=1}^{n_b} \sim D \quad (3.1)$$

Keterangan:

$D$  : dataset pelatihan penuh  
 $D_b$  : subset *bootstrap* untuk pohon ke- $b$   
 $x_i$  : vektor fitur  
 $y_i$  : label kelas  
 $n_b$  : jumlah sampel dalam subset *bootstrap*

Persamaan 3.1 menunjukkan proses pembentukan subset data *bootstrap* yang digunakan untuk membangun setiap pohon keputusan dalam algoritma *Random Forest*. Subset  $D_b$  dibentuk dengan mengambil sejumlah sampel  $n_b$  dari dataset pelatihan  $D$  secara acak dengan pengembalian (*sampling with replacement*). Proses ini memungkinkan setiap pohon dilatih menggunakan kombinasi data yang berbeda, sehingga meningkatkan keragaman model dan mengurangi korelasi antar pohon.

- b. Setelah data *bootstrap* terbentuk, setiap pohon dibangun melalui proses pembagian (*splitting*) data secara rekursif. Pada setiap node, algoritma memilih sejumlah fitur secara acak sebanyak nilai parameter  $mtry$ , kemudian menentukan fitur terbaik yang menghasilkan pemisahan paling optimal berdasarkan kriteria seperti *Gini Index* atau *Entropy*.

- c. Pemilihan fitur dilakukan secara acak dengan jumlah fitur yang dipertimbangkan ditentukan berdasarkan akar dari jumlah total fitur dalam dataset, yang dihitung sebagai berikut:

$$mtry = \sqrt{p} \quad (3.2)$$

Persamaan 3.2 menunjukkan jumlah fitur yang dipilih secara acak pada setiap proses pemisahan node dalam algoritma *Random Forest*. Nilai *mtry* ditentukan sebagai akar dari jumlah total fitur *p*, sehingga hanya sebagian fitur yang dipertimbangkan pada setiap node. Strategi ini bertujuan untuk meningkatkan keragaman antar pohon keputusan dan mengurangi korelasi, sehingga kinerja model secara keseluruhan menjadi lebih stabil.

- d. Proses pembentukan pohon dilakukan terus-menerus hingga setiap *child node* memenuhi syarat minimum jumlah observasi yang ditetapkan oleh parameter *min.node.size*. Apabila syarat tersebut terpenuhi, node akan menjadi simpul akhir (*leaf node*). Pada simpul akhir ini, model menyimpan proporsi jumlah data dari masing-masing kelas yang digunakan untuk menentukan hasil prediksi.
- e. Setelah seluruh pohon terbentuk, proses prediksi dilakukan dengan cara menentukan posisi data baru pada setiap *leaf node* di seluruh pohon yang telah dibuat. Hasil prediksi dari semua pohon kemudian digabungkan melalui dua mekanisme, yaitu *majority voting* untuk menentukan kelas dengan jumlah suara terbanyak pada kasus klasifikasi.
- f. Kombinasi antara pengambilan sampel acak dan pemilihan fitur acak membuat *Random Forest* mampu menghasilkan model yang stabil, memiliki akurasi

tinggi, serta tahan terhadap *overfitting*. Algoritma ini juga efektif digunakan untuk data berukuran besar dan mampu menangani variabel dengan korelasi tinggi maupun data yang mengandung *noise*.

- g. Dalam proses pembentukan pohon, *Random Forest* menentukan fitur terbaik pada setiap node menggunakan ukuran ketidakmurnian data (*impurity measure*), yaitu *Gini Index* dan *Gini Split*. *Gini Index* digunakan untuk mengukur homogenitas suatu node dan dirumuskan sebagai berikut (Suci Amaliah *et al.*, 2022).

$$Gini (s_i) = 1 - \sum_{i=0}^{c-1} p_i^2 \quad (3.3)$$

Keterangan:

- $Gini (s_i)$  : nilai impurity (ketidakmurnian) pada node  $s_i$   
 $s_i$  : subset data pada node  
 $p_i$  : probabilitas data pada kelas ke- $i$  di dalam node  $s_i$   
 $c$  : jumlah kelas  
 $\sum_{i=1}^c p_i^2$  : jumlah kuadrat dari proporsi tiap kelas.

Persamaan 3.3 digunakan untuk menghitung nilai *Gini impurity* pada suatu node  $s_i$  dalam pohon keputusan. Nilai *Gini* mengukur tingkat ketidakmurnian data pada node tersebut berdasarkan distribusi kelas di dalamnya. Semakin kecil nilai *Gini*, maka data pada node semakin homogen, sehingga pemisahan yang dihasilkan semakin baik dalam membedakan kelas.

Jika sebuah fitur dengan nilai ambang yang digunakan untuk memisahkan node, maka data dibagi menjadi dua subset (Abedinia & Seydi, 2024):

- Subset Kiri (Left) =  $\{x \in S \mid A(x) \leq t\}$
- Subset Kanan (Right) =  $\{x \in S \mid A(x) > t\}$

Keterangan:

Subset Kiri (Left)

$x$  adalah data atau titik yang sedang di proses

$S$  adalah himpunan data yang ada di node sebelum pemisahan

$A(x)$  adalah nilai fitur A pada data  $x$

$t$  adalah ambang batas (threshold) untuk pemisahan data.

| berarti “yang memenuhi kondisi”

Jadi, subset kiri berisi semua data  $x$  dalam himpunan  $S$  yang memenuhi kondisi  $A(x) \leq t$  (artinya fitur  $A(x)$  lebih kecil atau sama dengan dari threshold  $t$ ).

Subset Kanan (Right)

subset kanan berisi semua data  $x$  dalam himpunan  $S$  yang memenuhi kondisi  $A(x) > t$  (artinya fitur  $A(x)$  lebih besar dari threshold  $t$ ).

Masing-masing subset memiliki nilai impurity:

$$Gini_{Left} = 1 - \sum_{i=0}^{c-1} (P_i, Left)^2 \quad (3.4)$$

$$Gini_{Right} = 1 - \sum_{i=0}^{c-1} (P_i, Right)^2 \quad (3.5)$$

Keterangan:

$P_i, Left$  : proporsi data kelas  $i$  pada subset kiri

$P_i, Right$  : proporsi data kelas  $i$  pada subset kanan

Persamaan 3.4 dan 3.5 digunakan untuk menghitung nilai *Gini impurity* pada subset data hasil pemisahan, yaitu node kiri (*Left*) dan node kanan (*Right*). Nilai *Gini* pada masing-masing subset menunjukkan tingkat ketidakmurnian distribusi kelas setelah proses split dilakukan. Perhitungan ini menjadi dasar dalam mengevaluasi kualitas pemisahan, di mana pemisahan yang baik akan menghasilkan nilai *Gini* yang lebih rendah pada kedua subset.

Nilai *Gini Index* pada kedua subset tersebut kemudian dibandingkan untuk mengetahui tingkat ketidakmurnian data pada masing-masing cabang hasil pemisahan. Semakin kecil nilai *Gini*, maka semakin homogen data pada subset tersebut. Maka, setelah fitur dipilih secara acak, titik pemisahan terbaik ditentukan menggunakan nilai *Gini Split*, yang dihitung sebagai rata-rata tertimbang dari tingkat *impurity* pada dua subset yang dihasilkan dari pemisahan tersebut.

$$Gini_{split} = \sum_{i=0}^{c-1} \left( \frac{n_i}{n} \right) \times Gini_{(s_i)} \quad (3.6)$$

Keterangan :

$c$	: jumlah kelas
$i$	: indeks subset (mulai dari 1 sampai $c$ )
$n$	: total seluruh data pada node sebelum split
$n_i$	: jumlah data pada subset ke- $i$
$\left( \frac{n_i}{n} \right)$	: probabilitas data subset ke- $i$
$Gini_{(s_i)}$	: nilai gini pada subset ke- $i$
$Gini_{split}$	: impurity gabungan (rata-rata tertimbang) setelah split
$\sum$	: penjumlahan untuk semua subset ( $i = 1$ sampai $c$ )

Persamaan 3.6 digunakan untuk menghitung nilai *Gini split*, yaitu tingkat ketidakmurnian gabungan setelah suatu node dibagi menjadi beberapa subset. Nilai *Gini split* diperoleh dari rata-rata tertimbang nilai *Gini* pada setiap subset berdasarkan proporsi jumlah data di masing-masing subset. Pemisahan terbaik dipilih berdasarkan nilai *Gini split* yang paling kecil, karena menunjukkan distribusi kelas yang lebih homogen setelah proses split.

Karena penelitian ini menggunakan proses pemisahan secara biner (*binary split*), maka perhitungan *Gini Split* pada Persamaan 3.6 dapat disederhanakan menjadi dua cabang, yaitu subset kiri (*Left*) dan subset kanan (*Right*), dengan rumus sebagai berikut (Carrasco *et al.*, 2025).

$$Gini_{Split} = \frac{N_{Left}}{N} Gini_{left} + \frac{N_{Right}}{N} Gini_{Right} \quad (3.7)$$

Pada Persamaan 3.7,  $N_{Left}$  dan  $N_{Right}$  masing-masing merepresentasikan jumlah data pada subset kiri dan kanan, sedangkan  $N$  menunjukkan total jumlah data pada node sebelum pemisahan. Nilai  $Gini_{left}$  dan  $Gini_{Right}$  merupakan tingkat *impurity* (ketidakmurnian) dari masing-masing subset. Hasil perhitungan  $Gini_{Split}$  ini digunakan untuk menentukan titik pemisahan terbaik, di mana nilai  $Gini$  yang paling kecil menunjukkan pemisahan data paling optimal.

Setelah  $Gini_{Split}$  dihitung, dipilih pemisahan dengan nilai Gini terkecil lalu proses ini diulang pada setiap node sampai pohon selesai atau tidak bisa di split lagi. Setelah seluruh pohon dalam *Random Forest* selesai dibangun, tahap akhir adalah menentukan hasil klasifikasi berdasarkan gabungan prediksi dari seluruh pohon tersebut.

Oleh karena itu, menyebabkan *Random Forest* memiliki banyak pohon yang tumbuh, membentuk sebuah *forest* (hutan) yang akan di analisis dalam proses klasifikasi. Semakin banyak pohon yang digunakan dalam *Random Forest*, maka akurasi klasifikasi akan semakin baik. Untuk melakukan prediksi pada sampel baru, sampel tersebut dimasukkan ke dalam setiap pohon keputusan yang sudah terbentuk, dan kelasnya ditentukan oleh hasil *voting* dari semua pohon keputusan tersebut. Proses ini diulang untuk seluruh pohon keputusan dalam *Random Forest*. Hasil dari *voting* tersebut akan menjadi klasifikasi akhir dari sampel baru dalam proses klasifikasi (Huda *et al.*, 2023).

Berdasarkan data yang terdapat pada Tabel 3.13 yang telah dikonversi menjadi data numerik dengan total data 3 dokumen dan terdapat 2 kelas yaitu 1 (hoaks) dan 0 (non-hoaks). Oleh karena itu, tahapan perhitungan yang dilakukan dalam penerapan algoritma *Random Forest* menggunakan persamaan 3.2 untuk menghitung jumlah fitur, persamaan 3.3 untuk menghitung gini index dan 3.4 untuk gini split. Maka dari itu, sebagai contoh membentuk 3 pohon ( $n\_tree = 3$ ) dengan jumlah data ( $n$ ) = 3. Proses dimulai dengan *bootstrap sampling* untuk memilih data secara acak dengan pengembalian, dan dihitung tingkat ketidakmurniannya dengan gini index. Hasil Gini Index digunakan untuk menentukan pemisahan terbaik, yang kemudian dihitung menggunakan *Gini Split*.

Setelah dilakukan pemisahan data pada setiap pohon, langkah selanjutnya adalah melakukan voting mayoritas untuk menentukan hasil klasifikasi akhir. Dengan hasil voting mayoritas yang telah dilakukan, algoritma *Random Forest* menunjukkan keunggulannya dalam klasifikasi yang akurat. Hal ini didukung oleh pemilihan fitur secara acak pada setiap pohon yang mengurangi risiko *overfitting*, sehingga membuat model lebih efisien dan efektif.

Oleh karena itu, berdasarkan karakteristik hasil ekstraksi fitur dan struktur data yang digunakan pada penelitian ini, algoritma *Random Forest* dipilih karena memiliki kemampuan yang sesuai dengan kebutuhan sistem deteksi hoaks berbasis teks berbahasa Indonesia. Dataset yang digunakan memiliki jumlah data 733 dengan fitur berdimensi tinggi hasil dari ekstraksi *Bag-of-Words*, serta distribusi kelas yang tidak seimbang antara kategori hoaks dan non-hoaks. Proses ekstraksi fitur *Bag-of-Words* menghasilkan total 1.864 fitur unik, sehingga sesuai dengan

prinsip pemilihan fitur acak (*random feature selection*), jumlah fitur yang digunakan pada setiap node dalam *Random Forest* ditentukan dengan mengambil akar dari total fitur tersebut, yaitu  $\sqrt{1.864} = 43$ . Dengan demikian, setiap pohon hanya menggunakan 43 fitur acak dari total 1.864 fitur, yang membuat model menjadi lebih efisien dan mampu mengurangi risiko *overfitting*.

Selain itu, algoritma *Random Forest* memiliki mekanisme *bagging* dan *majority voting* yang menggabungkan hasil prediksi dari seluruh pohon keputusan, sehingga menghasilkan klasifikasi yang lebih stabil dan akurat. Berdasarkan hasil penelitian terdahulu yang telah dijelaskan pada Tabel 2.3, *Random Forest* juga terbukti unggul dalam menangani data berdimensi tinggi dan ketidakseimbangan kelas, dengan tingkat akurasi mencapai 92%, 99% (Israt Jahan *et al.*, 2024; Lassmedi Afdan & Rizal Isnanto, 2025; Almuayayi *et al.*, 2022; Adiguna *et al.*, 2024). Hal ini sejalan dengan karakteristik dataset yang digunakan dalam penelitian ini, yaitu data teks hasil ekstraksi *Bag-of-Words* yang menghasilkan ribuan fitur dan distribusi kelas yang tidak seimbang antara kategori hoaks dan non-hoaks. Oleh karena itu, algoritma *Random Forest* dipilih dalam penelitian ini karena mampu memberikan keseimbangan antara akurasi, efisiensi komputasi, dan ketahanan terhadap *overfitting*, serta sesuai dengan karakteristik data yang digunakan.

## **BAB IV**

### **UJI COBA DAN PEMBAHASAN**

#### **4.1 Skenario Pengujian**

Skenario pengujian yang bertujuan untuk mengevaluasi performa sistem dalam melakukan klasifikasi. Dalam penelitian ini, tahap pengujian dirancang untuk mengevaluasi performa model dalam mengklasifikasikan postingan artikel tulisan hoaks dan non-hoaks berdasarkan pendekatan *singlestage classification* dan *multistage classification*. Setiap skenario disusun untuk melihat bagaimana variasi proses klasifikasi memengaruhi akurasi, *precision*, *recall*, dan *F1-score* model *Random Forest*. Skenario ini juga dapat membantu mengamati sejauh mana pendekatan *multistage* memberikan kontribusi dalam meningkatkan kemampuan model mendeteksi hoaks pada tiap aspek. Pada Tabel 4.1 rancangan skenario pengujian yang digunakan dalam penelitian ini.

Tabel 4. 1 Skenario Pengujian

<b>Uji Coba</b>	<b>Type of Stage Classification</b>	<b>N_estimator</b>	<b>Max_depth</b>	<b>Data Train</b>	<b>Data Test</b>	<b>Distribusi Data</b>
1	<i>SingleStage Classification</i>	50	10	80%	20%	<i>Balance</i>
2		50	20			
3		50	30			
4		75	10			
5		75	20			
6		75	30			
7		100	10			
8		100	20			
9		100	30			
10	<i>MultiStage Classification</i>	50	10			
11		50	20			
12		50	30			
13		75	10			
14		75	20			
15		75	30			
16		100	10			
17		100	20			
18		100	30			

Tabel 4.1, penelitian ini melakukan 18 skenario pengujian untuk mengevaluasi kinerja algoritma *Random Forest* menggunakan dua pendekatan, yaitu *singlestage classification* dan *multistage classification*. Seluruh skenario menggunakan pembagian data yang sama, yakni 80% data latih dan 20% data uji, dengan distribusi data latih yang telah diseimbangkan (*balanced*), sehingga hasil pengujian dapat dibandingkan secara objektif. Pendekatan *singlestage classification* diuji melalui 9 skenario dengan variasi jumlah pohon (*n\_estimators*) sebesar 50, 75, dan 100 serta kedalaman maksimum pohon (*max\_depth*) sebesar 10, 20, dan 30.

Pendekatan *multistage classification* juga diuji menggunakan 9 skenario dengan kombinasi parameter yang identik, sehingga perbedaan performa yang dihasilkan mencerminkan pengaruh pendekatan klasifikasi yang digunakan. Perancangan skenario ini bertujuan untuk menganalisis pengaruh variasi parameter dan pendekatan klasifikasi terhadap performa model *Random Forest* dalam mendeteksi hoaks, yang selanjutnya dibahas berdasarkan hasil evaluasi pada masing-masing skenario pengujian.

#### 4.1.1 *Split Data*

Pada tahap ini, Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk menilai kemampuan model. Penelitian ini menggunakan rasio 80% data latih dan 20% data uji. Dari total 1.200 data, sebanyak 960 data digunakan sebagai data latih dan 240 data digunakan sebagai data uji pada *singlestage*. Pada *multistage stage 1* menghasilkan data train sebanyak 733 untuk aspek pendidikan, 174 untuk aspek politik, 53 untuk aspek bencana alam,

sedangkan data test menghasilkan sebanyak 183 untuk aspek pendidikan, 44 untuk aspek politik, dan 13 untuk aspek bencana alam. Pada *stage 2*, aspek pendidikan menghasilkan data train sebanyak 661 untuk data hoaks dan 71 untuk data non hoaks, sedangkan data test sebanyak 166 untuk hoaks dan 18 untuk data non hoaks. Pada aspek politik menghasilkan data train sebanyak 156 untuk data hoaks dan 18 untuk data non hoaks, sedangkan data test sebanyak 39 untuk data hoaks dan 5 untuk data non hoaks. Pada aspek bencana alam menghasilkan data train sebanyak 47 untuk data hoaks dan 5 untuk data non hoaks, sedangkan data test sebanyak 13 untuk data hoaks dan 1 untuk data non hoaks. Potongan *source code* pembagian data ditampilkan sebagai berikut.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)
```

Gambar 4. 1 Source Code Split Data

Gambar 4.1 menunjukkan proses pemisahan data dengan fungsi *train\_test\_split* menggunakan rasio 80% data latih dan 20% data uji. Pada skenario *single stage*, pembagian ini menghasilkan 960 data latih dan 240 data uji.

#### 4.1.2 Menghitung Kinerja Sistem

Pada tahap menghitung kinerja sistem dilakukan untuk menilai kemampuan model *Random Forest* dalam mendekripsi hoaks, baik pada skenario *single stage* maupun *multistage*. Evaluasi ini dilakukan untuk mengetahui sejauh mana model mampu membedakan informasi hoaks dan non-hoaks berdasarkan data uji yang tidak pernah dilihat oleh model sebelumnya.

Pengukuran performa dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* yang dihasilkan dari *confusion matrix*. Keempat metrik ini memberikan gambaran menyeluruh mengenai kemampuan model dalam membedakan kelas hoaks dan non-hoaks baik pada skenario *single stage* maupun *multistage*. Melalui proses evaluasi ini, dapat dilihat apakah model telah bekerja secara efektif dalam mengidentifikasi hoaks sesuai tujuan penelitian.

Berdasarkan nilai dari dalam klasifikasi *confusion matrix*, maka memperoleh nilai akurasi, *precision*, *recall*, dan *f1-Score* (Prasetyo & Nabiilah, 2023).

### 1. *Accuracy*

$$Akurasi = \frac{(TP + TN)}{(TP + TN + FN + FP)} \quad (4.1)$$

Keterangan:

- TP : *True Positive*
- TN : *True Negative*
- FP : *False Positive*
- FN : *False Negative*

Persamaan 4.1 digunakan untuk menghitung nilai akurasi, yaitu proporsi prediksi yang diklasifikasikan dengan benar oleh model terhadap seluruh data pengujian. Akurasi diperoleh dari perbandingan jumlah prediksi benar (True Positive dan True Negative) dengan total seluruh prediksi. Metrik ini digunakan untuk memberikan gambaran umum mengenai kinerja model dalam mengklasifikasikan data.

## 2. *Precision*

$$Precision = \frac{(TP)}{(TP + FP)} \quad (4.2)$$

Persamaan 4.2 digunakan untuk menghitung nilai *precision*, yaitu tingkat ketepatan model dalam memprediksi kelas positif. Nilai *precision* menunjukkan proporsi prediksi positif yang benar (*True Positive*) dibandingkan dengan seluruh prediksi positif yang dihasilkan model.

Kemudian, menghitung rata-rata nilai *precision* secara keseluruhan, dengan menggunakan rumus:

$$Precision_{avarage} = \frac{\sum precision_i}{n} \quad (4.3)$$

Selanjutnya, Persamaan 4.3 digunakan untuk menghitung nilai *precision* rata-rata dengan menjumlahkan nilai *precision* pada setiap kelas, kemudian dibagi dengan jumlah kelas. Perhitungan ini bertujuan untuk memberikan gambaran performa model secara keseluruhan tanpa dipengaruhi oleh perbedaan jumlah data pada tiap kelas.

## 3. *Recall*

$$Recall = \frac{(TP)}{(TP + FN)} \quad (4.4)$$

Persamaan 4.4 digunakan untuk menghitung nilai *recall*, yaitu kemampuan model dalam mengenali seluruh data yang benar-benar termasuk ke dalam kelas positif. Nilai *recall* menunjukkan proporsi *True Positive* terhadap seluruh data positif yang sebenarnya.

Kemudian, menghitung rata-rata nilai *recall* secara keseluruhan, dengan menggunakan rumus:

$$Recall_{avarage} = \frac{\sum recall_i}{n} \quad (4.5)$$

Selanjutnya, Persamaan 4.5 digunakan untuk menghitung nilai *recall* rata-rata dengan merata-ratakan nilai *recall* pada setiap kelas. Perhitungan ini bertujuan untuk menilai kemampuan model secara keseluruhan dalam menangkap data positif pada seluruh kelas tanpa dipengaruhi ketidakseimbangan jumlah data.

#### 4. *F1-score*

$$F1 - Score = \frac{(2 \times precision \times recall)}{(precision + recall)} \quad (4.6)$$

Persamaan 4.6 digunakan untuk menghitung nilai *F1-score*, yaitu nilai rata-rata harmonik antara *precision* dan *recall*. Metrik ini digunakan untuk menyeimbangkan ketepatan dan kelengkapan prediksi model, khususnya ketika terdapat ketidakseimbangan data antar kelas.

Kemudian, menghitung rata-rata nilai *f1-score* secara keseluruhan, dengan menggunakan rumus:

$$f1 - score_{avarage} = \frac{\sum_i^n f1_i}{n} \quad (4.7)$$

Selanjutnya, Persamaan 4.7 digunakan untuk menghitung nilai *F1-score* rata-rata dengan merata-ratakan nilai *F1-score* pada setiap kelas. Perhitungan ini bertujuan untuk mengevaluasi performa model secara keseluruhan tanpa dipengaruhi dominasi kelas tertentu.

#### 4.1.3 K-Fold Cross Validation

Pada bagian ini, *K-Fold Cross Validation* dengan nilai  $k = 5$  untuk mengevaluasi performa model secara lebih konsisten. Dataset terlebih dahulu dibagi menjadi data latih dan data uji menggunakan rasio 80:20, sehingga diperoleh 960 data latih dan 240 data uji. Data uji digunakan sebagai pengujian akhir dan tidak dilibatkan dalam proses *K-Fold*. Selanjutnya, proses *K-Fold Cross Validation* diterapkan pada data latih dengan membaginya ke dalam lima *fold* dengan ukuran yang relatif sama. Pada setiap iterasi, satu *fold* digunakan sebagai data uji *fold*, sedangkan empat *fold* lainnya digunakan sebagai data latih. Proses pelatihan dan evaluasi dilakukan sebanyak lima iterasi dengan kombinasi data yang berbeda. Seluruh hasil evaluasi dari masing-masing *fold* kemudian dirata-ratakan untuk memperoleh gambaran performa model yang lebih stabil sebelum dilakukan pengujian akhir pada data uji. Ilustrasi proses *K-Fold Cross Validation* ditunjukkan pada gambar berikut.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

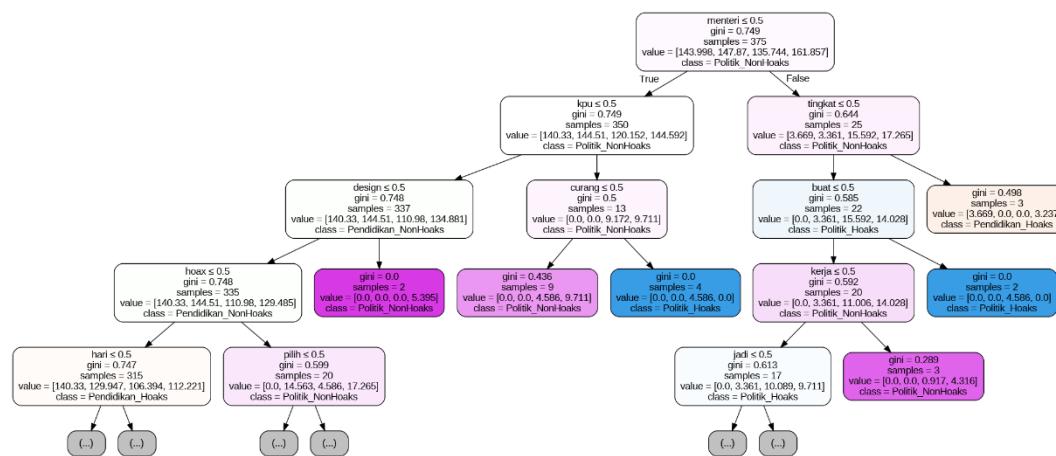
Gambar 4. 2 Ilustrasi 5-Fold Cross Validation

Gambar 4.2 memperlihatkan alur penerapan *5-Fold Cross Validation* pada proses evaluasi model. Pada setiap iterasi, satu *fold* digunakan sebagai data uji *fold*, sementara empat *fold* lainnya digunakan sebagai data latih. *Fold* yang berperan

sebagai data uji akan bergantian pada setiap iterasi hingga seluruh *fold* memperoleh peran yang sama. Melalui mekanisme ini, seluruh data latih dapat dimanfaatkan secara bergantian baik untuk pelatihan maupun pengujian sementara, sehingga evaluasi performa model menjadi lebih konsisten dan tidak bergantung pada satu pembagian data tertentu sebelum dilakukan pengujian akhir pada data uji.

## 4.2 Hasil Training

Pada gambar 4.3 adalah visualisasi dari satu pembangunan pohon keputusan (*Decision Tree*) dengan algoritma *Random Forest* pada skenario *singlestage hyperparameter* 100-30.



Gambar 4. 3 Visualiasi Salah Satu Pohon Keputusan *Random Forest*

Gambar 4.3 tersebut menunjukkan salah satu pohon keputusan yang terbentuk dalam algoritma *Random Forest*. Pohon ini menggambarkan proses pengambilan keputusan model berdasarkan pemisahan data pada setiap node menggunakan fitur teks dan nilai ambang tertentu. Pemilihan pemisahan dilakukan dengan mempertimbangkan nilai *Gini Index*, di mana nilai yang semakin kecil menunjukkan pemisahan kelas yang semakin baik.

Setiap node menampilkan informasi jumlah data (*samples*), distribusi kelas (*value*), serta kelas dominan yang dihasilkan. Proses klasifikasi berlangsung dari node akar hingga mencapai node daun, dan keputusan akhir pada setiap daun ditentukan berdasarkan kelas mayoritas. Perbedaan warna pada node digunakan untuk menunjukkan kelas yang dominan sehingga memudahkan interpretasi hasil klasifikasi.

Visualisasi ini menunjukkan bahwa satu pohon hanya merepresentasikan sebagian pola keputusan dalam model. Keputusan akhir *Random Forest* diperoleh dari gabungan prediksi seluruh pohon melalui mekanisme *majority voting*, sehingga model mampu menghasilkan prediksi yang lebih stabil dan robust dibandingkan satu pohon keputusan tunggal.

### 4.3 Hasil Pengujian

Pengujian model *Random Forest* dilakukan menggunakan data yang telah melalui tahap *preprocessing* dan ekstraksi fitur *Bag-of-Words*, dengan pembagian 80% data latih dan 20% data uji. Model kemudian dilatih menggunakan variasi jumlah pohon (*n\_estimator*) dan kedalaman maksimum pohon (*max\_depth*) untuk memperoleh konfigurasi dengan performa terbaik. Evaluasi dilakukan menggunakan dua pendekatan, yaitu *singlestage classification* dan *multistage classification*. Dataset yang digunakan memiliki distribusi kelas yang relatif seimbang, dengan jumlah data pendidikan hoaks dan politik hoaks masing-masing sebanyak 200 data, serta politik non-hoaks dan pendidikan non-hoaks sebanyak 169 dan 164 data. Kondisi ini memungkinkan model mempelajari pola tiap kelas dengan

baik. Penelitian ini menerapkan *K-Fold Cross Validation* pada setiap skenario untuk memperoleh evaluasi performa yang lebih stabil dan representatif.

#### 4.3.1 Hasil Skenario Uji 1

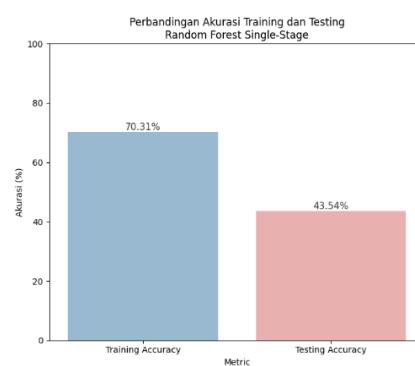
Hasil skenario uji 1 menggunakan pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi  $n\_estimator = 50$  dan  $max\_depth = 10$ . Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 2 *Confusion Matrix Singlestage*  $n\_estimator = 50$  dan  $max\_depth = 10$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	50,58%	75,00%	60,61%
Pendidikan non-hoaks	33,33%	27,27%	30,00%
Politik hoaks	51,52%	42,50%	46,58%
Politik non-hoaks	28,57%	23,53%	25,81%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	41,07%	42,08%	40,75%



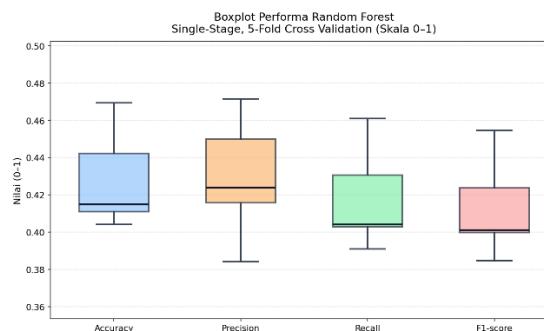
Gambar 4. 4 Perbandingan Akurasi *Singlestage Hyperparameter 50-10*

Tabel 4.2 dan Gambar 4.4 grafik perbandingan akurasi, model *Random Forest* yang menghasilkan akurasi data uji sebesar 43,54%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* yang masing-masing berada di antara 41–42%

menunjukkan bahwa performa model masih rendah dan belum merata pada seluruh kelas. Grafik tersebut memperlihatkan selisih yang cukup besar antara akurasi data latih dan data uji, yang mengindikasikan bahwa model belum mampu menggeneralisasi pola data dengan baik. Kondisi ini terjadi karena jumlah pohon yang relatif sedikit dan kedalaman pohon yang membatasi kemampuan model dalam menangkap karakteristik fitur teks yang kompleks, sehingga model masih kesulitan membedakan kelas secara konsisten, terutama pada kelas non-hoaks. Penelitian ini juga menerapkan *K-Fold Cross Validation* untuk mengevaluasi kestabilan performa model. Metode ini menghasilkan nilai performa rata-rata yang lebih konsisten.

Tabel 4. 3 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 50-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	44,21%	44,99%	43,05%	42,37%
2	46,93%	47,16%	46,10%	45,45%
3	41,49%	41,57%	40,42%	39,97%
4	40,41%	38,41%	39,09%	38,48%
5	41,09%	42,39%	40,28%	40,12%
Rata-rata	42,83% (27,1%)	42,90% (33,4%)	41,79 % (28,1%)	41,28% (27,1%)



Gambar 4. 5 Boxplot Performa Singlestage Hyperparameter 50-10

Tabel 4.3 hasil evaluasi menggunakan 5-Fold Cross Validation pada pendekatan *singlestage* model menghasilkan akurasi rata-rata sebesar 42,83%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai

42,90%, 41,79%, dan 41,28%. Perbedaan nilai antar *fold* relatif kecil, dengan rentang akurasi berada pada kisaran 40,41% hingga 46,93%, yang menunjukkan bahwa performa model cukup konsisten pada setiap *fold*. Pada Gambar 4.5, boxplot memperlihatkan sebaran nilai yang tidak terlalu lebar pada seluruh metrik evaluasi, sehingga kinerja model cenderung stabil meskipun tingkat akurasinya masih tergolong rendah. Kondisi ini mengindikasikan bahwa rendahnya performa model bukan disebabkan oleh ketergantungan pada pembagian data tertentu, melainkan keterbatasan pendekatan *singlestage* dalam menangkap kompleksitas dan keragaman pola pada data teks yang diklasifikasikan.

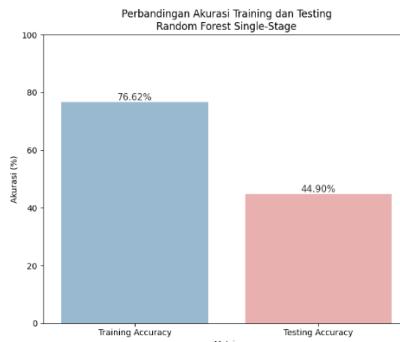
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi *n\_estimator* = 50 dan *max\_depth* = 20. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 4 *Confusion Matrix Singlestage n\_estimator = 50 dan max\_depth = 20*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	52,94%	67,50%	59,34%
Pendidikan non-hoaks	43,75%	42,42%	43,08%
Politik hoaks	43,24%	40,00%	41,56%
Politik non-hoaks	33,33%	26,47%	29,51%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	43,32%	44,10%	43,37%

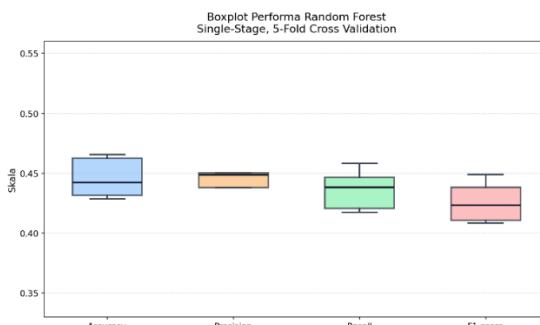


Gambar 4. 6 Perbandingan Akurasi *Singlestage Hyperparameter 50-20*

Tabel 4.4 hasil klasifikasi dan Gambar 4.6 grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 44,90%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 43,32%, 44,10%, dan 43,37%, yang menunjukkan bahwa performa model masih berada pada tingkat sedang dan belum sepenuhnya merata pada seluruh kelas. Secara per kelas, model mampu mengenali kelas pendidikan hoaks dengan *recall* sebesar 67,50%, namun masih mengalami kesulitan pada kelas politik non-hoaks yang hanya mencapai *recall* sebesar 26,47%. Gambar 4.6 memperlihatkan adanya selisih yang cukup besar antara akurasi data latih sebesar 76,62% dan akurasi data uji sebesar 44,90%, yang mengindikasikan bahwa model masih cenderung mengalami overfitting. Meskipun peningkatan kedalaman pohon memungkinkan model mempelajari pola data yang lebih kompleks, kondisi ini belum sepenuhnya diimbangi dengan kemampuan generalisasi yang baik, sehingga kesalahan klasifikasi, pada kelas non-hoaks masih relatif tinggi. Untuk memastikan kestabilan performa model, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* sehingga evaluasi yang diperoleh lebih konsisten.

Tabel 4. 5 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 50-20

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	42,85%	45,03%	41,73%	40,84%
2	44,21%	44,87%	43,82%	43,82%
3	46,25%	43,79%	44,66%	42,34%
4	43,15%	40,94%	42,06%	41,06%
5	46,57%	50,47%	45,81%	44,90%
Rata-rata	44,61% (17,2%)	45,02% (34,6%)	43,62 % (17,2%)	42,59% (17,5%)



Gambar 4. 7 Boxplot Performa Singlestage Hyperparameter 50-20

Tabel 4.5 hasil evaluasi menggunakan *5-Fold Cross Validation* pada pendekatan *singlestage* model menghasilkan akurasi rata-rata sebesar 44,61%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 45,02%, 43,62%, dan 42,59%. Perbedaan nilai antar *fold* relatif kecil, dengan rentang akurasi berada pada kisaran 42,85% hingga 46,57%, yang menunjukkan bahwa performa model cukup konsisten pada setiap *fold*. Kondisi ini diperkuat oleh boxplot pada Gambar 4.7 yang memperlihatkan sebaran nilai yang relatif sempit pada seluruh metrik evaluasi, sehingga kinerja model dapat dikatakan stabil pada berbagai pembagian data. Meskipun demikian, tingkat akurasi yang masih berada pada kategori sedang menunjukkan bahwa keterbatasan performa model bukan disebabkan oleh ketidakstabilan hasil pengujian, melainkan oleh keterbatasan pendekatan single-stage dalam menangkap kompleksitas dan keragaman pola pada data teks yang diklasifikasikan.

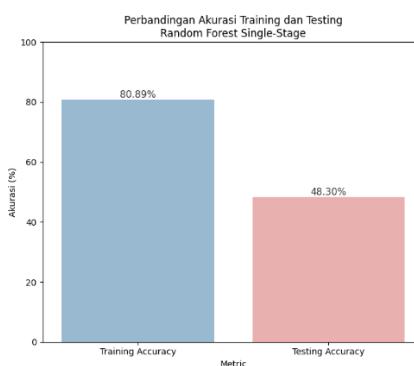
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi  $n\_estimator = 50$  dan  $max\_depth = 30$ . Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 6 Confusion Matrix Singlestage  $n\_estimator = 50$  dan  $max\_depth = 30$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	59,62%	77,50%	67,39%
Pendidikan non-hoaks	46,88%	45,45%	46,15%
Politik hoaks	45,71%	40,00%	42,67%
Politik non-hoaks	32,14%	26,47%	29,03%

Metric	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	46,09%	47,36%	46,31%



Gambar 4. 8 Perbandingan Akurasi *Singlestage Hyperparameter 50-30*

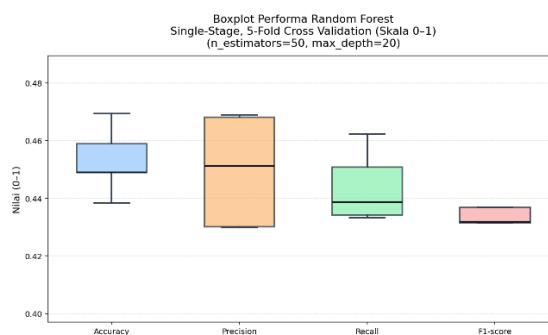
Tabel 4.6 hasil klasifikasi dan Gambar 4.8 grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 48,30%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 46,09%, 47,36%, dan 46,31%, yang menunjukkan adanya peningkatan performa dibandingkan konfigurasi dengan kedalaman pohon yang lebih rendah, meskipun kinerjanya belum sepenuhnya merata pada seluruh kelas. Secara per kelas, model mampu mengenali kelas pendidikan hoaks dengan recall yang cukup tinggi sebesar 77,50%, namun masih mengalami kesulitan dalam mengklasifikasikan kelas non-

hoaks, khususnya Politik non-hoaks yang hanya mencapai *recall* sebesar 26,47%.

Gambar 4.8 memperlihatkan adanya selisih yang cukup besar antara akurasi data latih sebesar 80,89% dan akurasi data uji sebesar 48,30%, yang mengindikasikan bahwa model masih cenderung mengalami *overfitting*. Peningkatan kedalaman pohon memungkinkan model menangkap karakteristik fitur teks yang lebih kompleks, namun kondisi tersebut belum sepenuhnya diimbangi dengan kemampuan generalisasi yang baik, sehingga kesalahan klasifikasi pada data uji masih relatif tinggi. Untuk memastikan kestabilan performa model, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* sehingga evaluasi yang diperoleh lebih konsisten.

Tabel 4. 7 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 50-30

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	44,89%	46,89%	43,85%	43,14%
2	46,93%	46,80%	46,21%	45,89%
3	44,89%	43,00%	43,41%	41,69%
4	45,89%	42,98%	45,08%	43,67%
5	43,83%	45,11%	43,31%	43,17%
Rata-rata	45,29% (11,7%)	44,96% (19,3%)	44,37 % (12,4%)	43,51% (15,2%)



Gambar 4. 9 Boxplot Performa Singlestage Hyperparameter 50-30

Tabel 4.7 hasil evaluasi menggunakan 5-Fold Cross Validation pada pendekatan *singlestage* menghasilkan akurasi rata-rata sebesar 45,29%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai

44,96%, 44,37%, dan 43,51%. Perbedaan nilai antar *fold* relatif kecil, dengan rentang akurasi berada pada kisaran 43,83% hingga 46,93%, yang menunjukkan bahwa performa model cukup konsisten pada setiap *fold*. Hal ini diperkuat oleh boxplot pada Gambar 4.9 yang memperlihatkan sebaran nilai yang relatif sempit pada seluruh metrik evaluasi, sehingga kinerja model dapat dikatakan stabil pada berbagai pembagian data. Meskipun demikian, peningkatan performa yang diperoleh masih tergolong terbatas, yang mengindikasikan bahwa keterbatasan model tidak disebabkan oleh ketidakstabilan pelatihan, melainkan oleh keterbatasan pendekatan *singlestage* dalam merepresentasikan kompleksitas dan keragaman pola pada data teks yang diklasifikasikan.

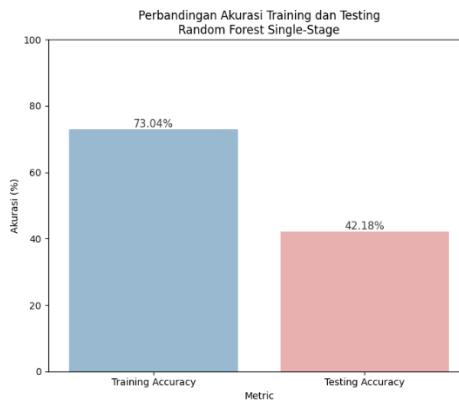
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi *n\_estimator* = 75 dan *max\_depth* = 10. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 8 Confusion Matrix Singlestage *n estimator* = 75 dan *max depth* = 10

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	54,55%	75,00%	63,16%
Pendidikan non-hoaks	29,03%	27,27%	28,12%
Politik hoaks	43,75%	35,00%	38,89%
Politik non-hoaks	31,03%	26,47%	28,57%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	39,59%	40,94%	39,69%



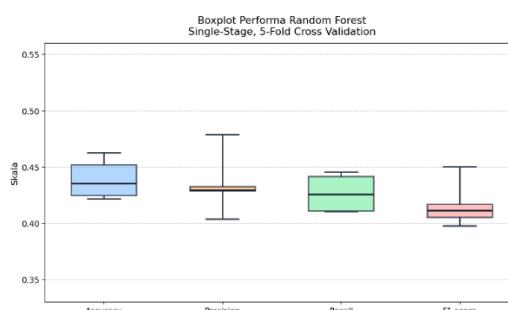
Gambar 4. 10 Perbandingan Akurasi *Singlestage Hyperparameter 75-10*

Tabel 4.8 hasil klasifikasi dan Gambar 4.10 grafik perbandingan akurasi, model *Random Forest singlестage* menghasilkan akurasi data uji sebesar 42,18%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 39,59%, 40,94%, dan 39,69%, yang menunjukkan bahwa performa model masih berada pada tingkat relatif rendah dan belum merata pada seluruh kelas. Secara per kelas, model mampu mengenali kelas pendidikan hoaks dengan *recall* yang cukup tinggi sebesar 75,00%, namun masih mengalami kesulitan pada kelas non-hoaks, khususnya pendidikan non-hoaks dan politik non-hoaks yang memiliki nilai *recall* di bawah 30%. Gambar 4.10 memperlihatkan adanya selisih yang cukup besar antara akurasi data latih sebesar 73,04% dan akurasi data uji sebesar 42,18%, yang mengindikasikan bahwa model belum sepenuhnya mampu melakukan generalisasi dengan baik. Peningkatan jumlah pohon memang memberikan perbaikan dalam proses pembelajaran model, namun kedalaman pohon yang relatif terbatas menyebabkan model belum optimal dalam menangkap pola teks yang lebih kompleks, terutama untuk membedakan kelas non-hoaks. Oleh karena itu, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* untuk

mengevaluasi kestabilan performa model sehingga hasil evaluasi yang diperoleh lebih konsisten.

Tabel 4. 9 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 75-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	42,17%	43,26%	41,11%	40,53%
2	43,53%	42,93%	42,56%	41,68%
3	46,25%	42,89%	44,15%	41,12%
4	42,46%	40,35%	41,02%	39,77%
5	45,20%	47,89%	44,56%	45,03%
Rata-rata	43,92% (17,6%)	43,46% (27,3%)	42,68% (16,5%)	41,62% (20,3%)



Gambar 4. 11 Boxplot Performa Singlestage Hyperparameter 75-10

Tabel 4.9 hasil evaluasi menggunakan 5-Fold Cross Validation pada pendekatan singlestage menghasilkan akurasi rata-rata sebesar 43,92%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 43,46%, 42,68%, dan 41,62%. Perbedaan nilai antar *fold* relatif kecil, dengan akurasi berada pada kisaran 42,17% hingga 45,20%, yang menunjukkan bahwa performa model relatif konsisten pada setiap *fold*. Kondisi ini diperkuat oleh boxplot pada Gambar 4.11 yang memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga kinerja model dapat dikatakan stabil pada berbagai pembagian data. Meskipun demikian, tingkat akurasi yang masih berada pada kategori sedang mengindikasikan bahwa keterbatasan performa model tidak disebabkan oleh ketidakstabilan proses pelatihan, melainkan oleh keterbatasan

pendekatan *singlestage* dalam menangkap kompleksitas dan keragaman pola pada data teks yang diklasifikasikan.

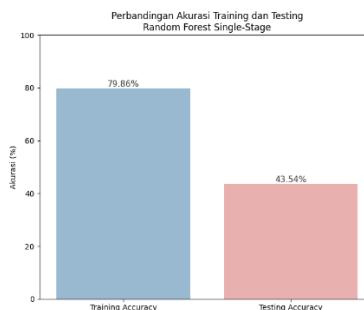
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi  $n\_estimator = 75$  dan  $max\_depth = 20$ . Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 10 *Confusion Matrix Singlestage n estimator = 75 dan max depth = 20*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	53,70%	72,50%	61,70%
Pendidikan non-hoaks	34,48%	30,30%	32,26%
Politik hoaks	45,47%	40,00%	42,67%
Politik non-hoaks	31,03%	26,47%	28,57%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	41,23%	42,32%	41,30%



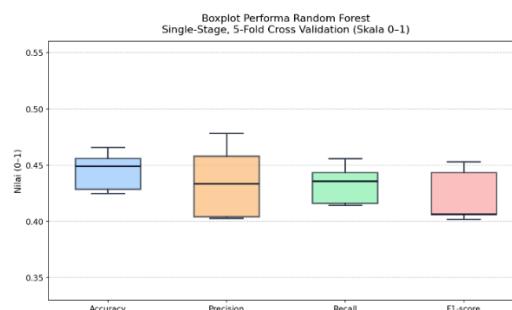
Gambar 4. 12 Perbandingan Akurasi *Singlestage Hyperparameter 75-20*

Tabel 4.10 hasil klasifikasi dan Gambar 4.12 grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 43,54%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 41,23%, 42,32%, dan 41,30%, yang menunjukkan bahwa performa model masih berada pada tingkat sedang dan belum merata pada seluruh kelas. Secara per kelas, model mampu mengenali kelas pendidikan hoaks dengan *recall* sebesar 72,50%,

namun masih mengalami kesulitan dalam mengklasifikasikan kelas non-hoaks, khususnya politik non-hoaks yang hanya mencapai *recall* sebesar 26,47%. Gambar 4.12 memperlihatkan selisih yang cukup besar antara akurasi data latih sebesar 79,86% dan akurasi data uji sebesar 43,54%, yang mengindikasikan bahwa model masih cenderung mengalami *overfitting*. Meskipun peningkatan jumlah pohon dan kedalaman pohon memungkinkan model mempelajari pola data yang lebih kompleks, kemampuan generalisasi model terhadap data uji masih terbatas, sehingga kesalahan klasifikasi pada beberapa kelas, terutama non-hoaks, masih relatif tinggi. Oleh karena itu, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* untuk memastikan bahwa evaluasi performa model yang diperoleh bersifat lebih stabil dan konsisten.

Tabel 4. 11 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 75-20

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	42,85%	43,31%	41,60%	40,60%
2	44,89%	45,79%	44,31%	44,34%
3	45,57%	40,39%	43,55%	40,15%
4	42,46%	40,27%	41,44%	40,64%
5	46,57%	47,83%	45,54%	45,30%
Rata-rata	44,47% (0,17%)	43,52% (0,33%)	43,29% (0,17%)	42,21% (0,24%)



Gambar 4. 13 Boxplot Performa Singlestage Hyperparameter 75-20

Tabel 4.11 hasil 5-Fold *Cross Validation*, model menghasilkan akurasi rata-rata sebesar 44,47% dengan standar deviasi 0,17%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 43,52% (0,33), 43,29%

(0,17), dan 42,21% (0,24), yang menunjukkan bahwa performa model relatif konsisten pada setiap fold. Hal ini diperkuat oleh *boxplot* pada Gambar 4.13 yang memperlihatkan sebaran nilai yang tidak terlalu lebar pada seluruh metrik evaluasi. Konsistensi tersebut mengindikasikan bahwa perubahan kedalaman pohon hingga nilai *max depth* = 20 belum memberikan performa yang signifikan. Namun, tingkat akurasi yang masih berada pada kategori menengah menunjukkan bahwa peningkatan kompleksitas model belum mampu secara optimal menangkap pola data, sehingga keterbatasan pendekatan *singlestage* masih memengaruhi performa secara keseluruhan.

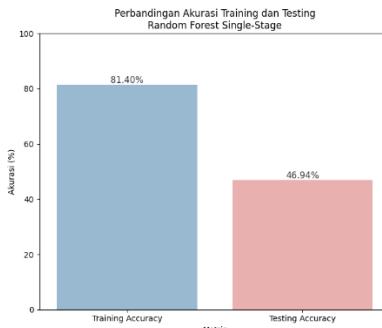
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi *n\_estimator* = 75 dan *max\_depth* = 30. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 12 *Confusion Matrix Singlestage n estimator = 75 dan max depth = 30*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	58,49%	77,50%	66,67%
Pendidikan non-hoaks	37,93%	33,33%	35,48%
Politik hoaks	46,15%	45,00%	45,57%
Politik non-hoaks	34,62%	26,47%	30,00%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	44,30%	45,58%	44,43%

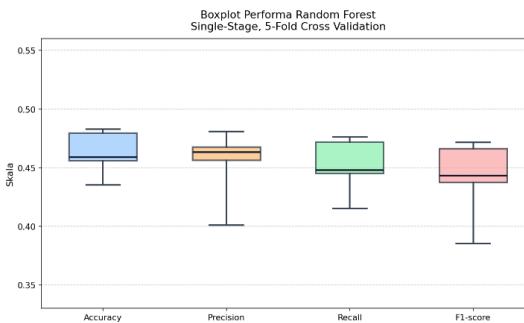


Gambar 4. 14 Perbandingan Akurasi Singlestage Kombinasi 75-30

Tabel 4.12 hasil klasifikasi dan Gambar 4.14 grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 46,94%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 44,30%, 45,58%, dan 44,43%, yang menunjukkan bahwa performa model berada pada tingkat sedang dan belum sepenuhnya merata pada seluruh kelas. Secara per kelas, model mampu mengenali kelas pendidikan hoaks dengan *recall* yang cukup tinggi sebesar 77,50%, namun masih mengalami kesulitan pada kelas non-hoaks, khususnya politik non-hoaks yang hanya mencapai *recall* sebesar 26,47%. Gambar 4.14 memperlihatkan adanya selisih yang cukup besar antara akurasi data latih sebesar 81,40% dan akurasi data uji sebesar 46,94%, yang mengindikasikan bahwa model masih cenderung mengalami *overfitting*. Meskipun peningkatan kedalaman pohon memungkinkan model menangkap pola fitur teks secara lebih mendalam, kemampuan generalisasi model terhadap data uji masih terbatas, sehingga kesalahan klasifikasi pada beberapa kelas, terutama kelas non-hoaks, masih relatif tinggi. Untuk memastikan kestabilan performa model, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* sehingga evaluasi yang diperoleh lebih konsisten.

Tabel 4. 13 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 75-30

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	45,57%	46,77%	44,48%	43,72%
2	48,29%	48,07%	47,62%	47,15%
3	43,53%	40,09%	41,52%	38,54%
4	47,94%	46,32%	47,17%	46,61%
5	45,89%	45,62%	44,79%	44,33%
Rata-rata	46,25% (19,3%)	45,37% (30,8%)	45,11% (24,4%)	44,07% (34,1%)



Gambar 4. 15 Boxplot Performa Singlestage Hyperparameter 75-30

Tabel 4.13 hasil 5-Fold *Cross Validation*, model menghasilkan akurasi rata-rata sebesar 46,25% dengan standar deviasi 19,3%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 45,37% (30,8), 45,11% (24,4), dan 44,07% (34,1), yang menunjukkan bahwa performa model konsisten antar *fold*. Hal ini diperkuat oleh *boxplot* pada Gambar 4.15 yang memperlihatkan sebaran nilai yang relatif sempit pada seluruh metrik evaluasi, sehingga variasi performa antar *fold* tergolong kecil. Konsistensi tersebut mengindikasikan bahwa model memiliki kemampuan generalisasi yang cukup stabil terhadap perbedaan pembagian data. Namun demikian, peningkatan akurasi yang diperoleh masih berada pada tingkat menengah, yang menunjukkan bahwa pendekatan *singlestage* masih memiliki keterbatasan dalam menangkap variasi dan struktur kompleks pada data hoaks secara menyeluruh.

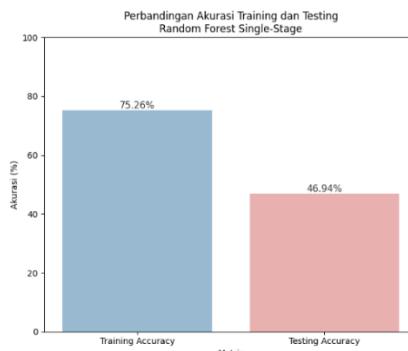
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi  $n\_estimator = 100$  dan  $max\_depth = 10$ . Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 14 Confusion Matrix Singlestage  $n\_estimator = 100$  dan  $max\_depth = 10$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	57,69%	75,00%	65,22%
Pendidikan non-hoaks	44,12%	45,45%	44,78%
Politik hoaks	43,75%	35,00%	38,89%
Politik non-hoaks	34,48%	29,41%	31,75%

Metric	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	45,01%	46,22%	45,16%



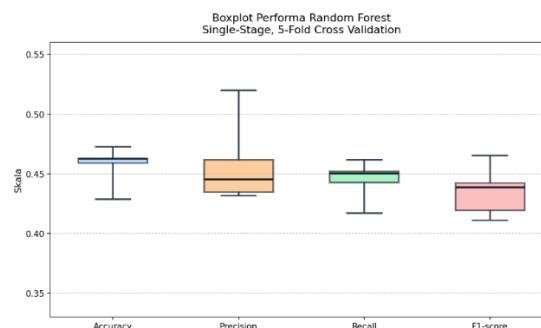
Gambar 4. 16 Perbandingan Akurasi *Singlestage Hyperparameter 100-10*

Tabel 4.14 hasil klasifikasi dan grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 46,94%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 45,01%, 46,22%, dan 45,16%, yang menunjukkan bahwa performa model berada pada tingkat sedang dan belum sepenuhnya merata pada seluruh kelas. Secara per kelas, model mampu mengklasifikasikan kelas pendidikan hoaks dengan cukup baik, ditunjukkan oleh nilai *recall* sebesar 75,00%, namun masih mengalami kesulitan pada kelas politik non-hoaks yang memiliki *recall* sebesar 29,41%. Grafik

perbandingan akurasi memperlihatkan adanya selisih yang cukup jelas antara akurasi data latih sebesar 75,26% dan akurasi data uji sebesar 46,94%, yang mengindikasikan bahwa kemampuan generalisasi model belum optimal. Kondisi ini menunjukkan bahwa peningkatan jumlah pohon tanpa diimbangi dengan kedalaman pohon yang lebih besar belum sepenuhnya mampu menangkap kompleksitas fitur teks, sehingga kesalahan klasifikasi masih terjadi, khususnya pada kelas non-hoaks. Oleh karena itu, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* untuk mengevaluasi kestabilan performa model sehingga hasil evaluasi yang diperoleh lebih konsisten.

Tabel 4. 15 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 100-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	42,85%	44,54%	41,71%	41,08%
2	46,25%	46,15%	45,19%	44,24%
3	46,25%	43,16%	44,26%	41,92%
4	45,89%	43,44%	45,03%	43,86%
5	47,26%	51,97%	46,17%	46,51%
Rata-rata	45,70% (16,7%)	45,85% (36,1%)	44,47% (16,8%)	43,52% (21,2%)



Gambar 4. 17 Boxplot Performa Singlestage Hyperparameter 100-10

Tabel 4.15 hasil 5-Fold *Cross Validation*, model menghasilkan akurasi rata-rata sebesar 45,70% dengan standar deviasi 16,7%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 45,85% (36,1), 44,47% (16,8), dan 43,52% (21,2), yang menunjukkan bahwa performa model relatif stabil

antar fold. Hal ini diperkuat oleh boxplot pada Gambar 4.17 yang memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga variasi performa antar fold tergolong kecil dan konsistensi model dapat dikatakan baik. Meskipun demikian, tingkat akurasi yang dihasilkan masih berada pada kategori menengah, yang mengindikasikan bahwa pendekatan *singlestage* belum sepenuhnya mampu menangkap kompleksitas dan keragaman pola informasi hoaks pada data secara menyeluruh.

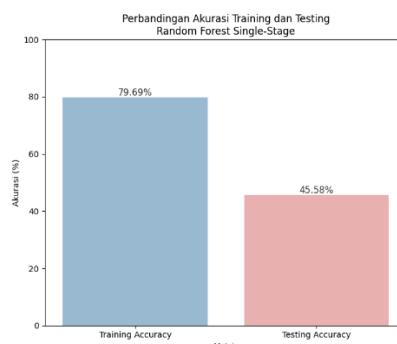
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi *n\_estimator* = 100 dan *max\_depth* = 20. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 16 *Confusion Matrix Singlestage n\_estimator = 100 dan max\_depth = 20*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	55,77%	72,50%	63,04%
Pendidikan non-hoaks	38,71%	36,36%	37,50%
Politik hoaks	47,22%	42,50%	44,47%
Politik non-hoaks	32,14%	26,47%	29,03%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	43,46%	44,46%	43,58%

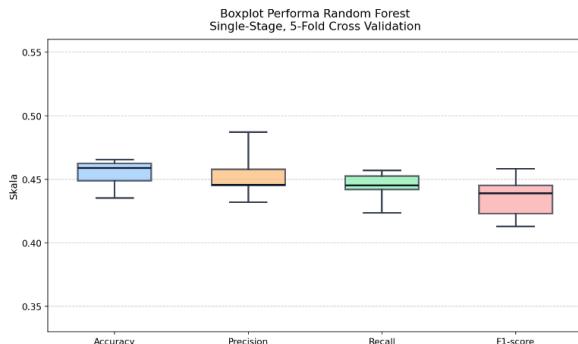


Gambar 4. 18 Perbandingan Akurasi *Singlestage Hyperparameter 100-20*

Tabel 4.16 hasil klasifikasi dan Gambar 4.18 grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 45,58%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 43,46%, 44,46%, dan 43,58%, yang menunjukkan bahwa performa model berada pada tingkat sedang dan belum merata pada seluruh kelas. Secara per kelas, model mampu mengenali kelas pendidikan hoaks dengan cukup baik, ditunjukkan oleh nilai recall sebesar 72,50%, namun masih mengalami kesulitan dalam mengklasifikasikan kelas non-hoaks, khususnya politik non-hoaks yang hanya mencapai recall sebesar 26,47%. Gambar 4.18 memperlihatkan adanya selisih yang cukup besar antara akurasi data latih sebesar 79,69% dan akurasi data uji sebesar 45,58%, yang mengindikasikan bahwa model masih cenderung mengalami *overfitting*. Meskipun peningkatan jumlah pohon dengan kedalaman sedang memungkinkan model mempelajari pola data yang lebih kompleks, kemampuan generalisasi terhadap data uji masih terbatas sehingga kesalahan klasifikasi, terutama pada kelas non-hoaks, masih relatif tinggi. Oleh karena itu, penelitian ini selanjutnya menerapkan *K-Fold Cross Validation* untuk mengevaluasi kestabilan performa model sehingga hasil evaluasi yang diperoleh lebih representatif dan konsisten.

Tabel 4. 17 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 100-20

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	43,53%	44,57%	42,36%	41,28%
2	44,89%	45,79%	44,20%	43,89%
3	46,25%	43,19%	44,53%	42,28%
4	45,89%	44,53%	45,25%	44,52%
5	46,57%	48,71%	45,68%	45,81%
Rata-rata	45,43% (12,3%)	45,36% (20,8%)	44,40% (12,8%)	43,56% (17,9%)



Gambar 4.19 Boxplot Performa *Singlestage Hyperparameter 100-20*

Tabel 4.17 hasil 5-Fold *Cross Validation*, model menghasilkan akurasi rata-rata sebesar 45,43% dengan standar deviasi 12,3%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 45,36% (20,8), 44,40% (12,8), dan 43,56% (17,9), yang menunjukkan bahwa performa model relatif konsisten pada setiap *fold*. Hal ini diperkuat oleh *boxplot* pada Gambar 4.19 yang memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga variasi performa antar *fold* tergolong kecil dan kinerja model dapat dikatakan stabil. Meskipun demikian, tingkat akurasi yang dihasilkan masih berada pada kategori sedang, yang mengindikasikan bahwa pendekatan *singlestage* belum sepenuhnya mampu merepresentasikan kompleksitas serta perbedaan karakteristik konten hoaks pada berbagai aspek secara optimal.

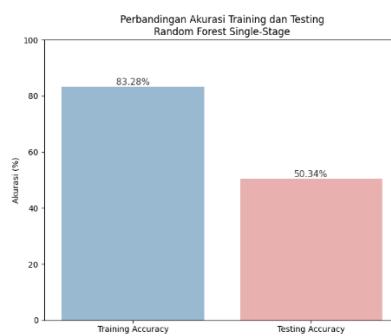
Pendekatan *singlestage classification* dengan parameter *Random Forest* berupa kombinasi  $n\_estimator = 100$  dan  $max\_depth = 30$ . Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Model dilatih menggunakan data dengan distribusi kelas yang seimbang.

Tabel 4. 18 *Confusion Matrix Singlestage n\_estimator = 100 dan max\_depth = 30*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan hoaks	61,70%	72,50%	66,67%
Pendidikan non-hoaks	45,45%	45,45%	45,45%
Politik hoaks	48,78%	50,00%	49,38%
Politik non-hoaks	38,46%	29,41%	33,33%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	48,60%	49,34%	48,71%

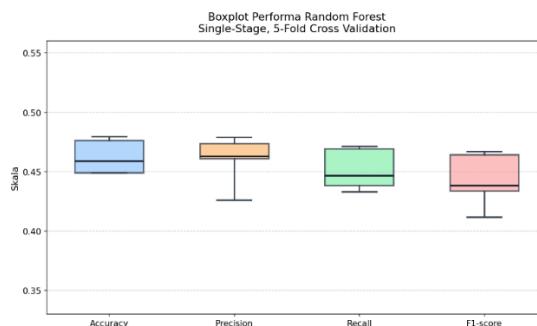
Gambar 4. 20 Perbandingan Akurasi *Singlestage Hyperparameter 100-30*

Tabel 4.18 hasil evaluasi dan Gambar 4.20 grafik perbandingan akurasi, model *Random Forest singlestage* menghasilkan akurasi data uji sebesar 50,34%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 48,60%, 49,34%, dan 48,71%, yang menunjukkan bahwa performa model berada pada tingkat sedang dan relatif lebih baik dibandingkan konfigurasi *singlestage* sebelumnya. Secara per kelas, model mampu mengklasifikasikan kelas pendidikan hoaks dengan baik, ditunjukkan oleh nilai *recall* sebesar 72,50%, serta menunjukkan peningkatan performa pada kelas politik hoaks dengan *recall* sebesar 50,00%. Namun demikian, performa pada kelas politik non-hoaks masih tergolong rendah dengan nilai *recall* sebesar 29,41%. Gambar 4.20 memperlihatkan adanya selisih yang cukup besar antara akurasi data latih sebesar 83,28% dan akurasi data uji sebesar 50,34%, yang mengindikasikan bahwa model masih mengalami

kecenderungan *overfitting*. Meskipun peningkatan jumlah pohon dan kedalaman pohon membantu model dalam menangkap karakteristik fitur teks yang lebih kompleks, kemampuan generalisasi terhadap data uji masih belum optimal. Oleh karena itu, penerapan *K-Fold Cross Validation* tetap diperlukan untuk memastikan bahwa evaluasi performa model yang diperoleh bersifat stabil.

Tabel 4. 19 Hasil Evaluasi 5-Fold Cross-Validation Singlestage Hyperparameter 100-30

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	44,89%	47,36%	43,83%	43,37%
2	47,61%	47,89%	46,92%	46,69%
3	44,89%	42,60%	43,28%	41,16%
4	47,94%	46,28%	47,13%	46,41%
5	45,89%	46,09%	44,65%	43,84%
Rata-rata	46,25% (0,14%)	46,04% (0,20%)	45,16% (0,17%)	44,29% (0,22%)



Gambar 4. 21 Boxplot Performa Singlestage Hyperparameter 100-30

Tabel 4.19 hasil 5-Fold *Cross Validation*, model menghasilkan akurasi rata-rata sebesar 46,25% dengan standar deviasi 0,14%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 46,04% (0,20), 45,16% (0,17), dan 44,29% (0,22), yang menunjukkan bahwa performa model relatif konsisten pada setiap *fold*. Hal ini diperkuat oleh *boxplot* pada Gambar 4.21 yang memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga variasi performa antar *fold* tergolong kecil dan kinerja model dapat dikatakan stabil. Meskipun demikian, tingkat akurasi yang dihasilkan masih berada pada kategori

sedang, yang mengindikasikan bahwa pendekatan *singlestage* dengan representasi fitur yang digunakan belum sepenuhnya mampu menangkap kompleksitas dan perbedaan karakteristik informasi hoaks secara optimal.

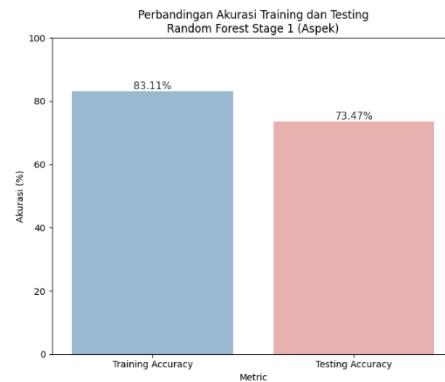
#### 4.3.2 Hasil Skenario Uji 2

Setelah melakukan pengujian menggunakan pendekatan *singlestage classification*, penelitian ini melanjutkan pengujian menggunakan pendekatan *multistage classification*. Pendekatan ini terdiri atas dua tahap klasifikasi. Tahap pertama mengklasifikasikan data berdasarkan aspek topik, yaitu pendidikan dan politik. Tahap kedua mengklasifikasikan data pada masing-masing aspek tersebut ke dalam kelas hoaks dan non-hoaks, sehingga proses klasifikasi dilakukan secara bertahap.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 10$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 20 *Confusion Matrix Multistage Stage 1 n estimator = 50 dan max depth = 10*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	70,73%	79,45%	74,84%
Politik	76,92%	67,57%	71,94%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	73,83%	73,51%	73,39%

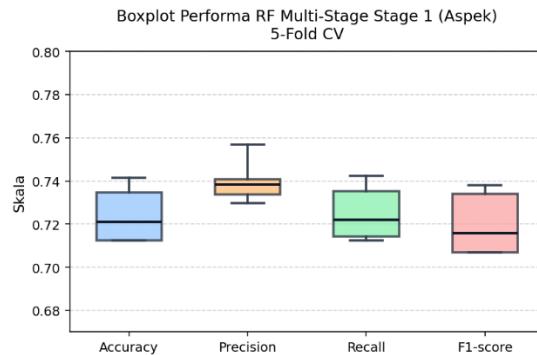


Gambar 4. 22 Perbandingan Akurasi *Multistage Stage 1 Hyperparameter 50-10*

Tabel 4.20 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 73,47%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 73,83%, 73,51%, dan 73,39%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.22 grafik perbandingan akurasi data latih dan data uji juga memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 21 Hasil Evaluasi *5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 50-10*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	74,14%	75,68%	74,23%	73,79%
2	73,46%	73,82%	73,50%	73,39%
3	72,10%	74,07%	72,20%	71,58%
4	71,23%	73,36%	71,43%	70,68%
5	71,23%	72,95%	71,23%	70,68%
Rata-rata	72,43% (0,13%)	73,98% (0,10%)	72,52% (0,13%)	72,02% (0,14%)



Gambar 4. 23 Boxplot Performa *Multistage Stage 1* Hyperparameter 50-10

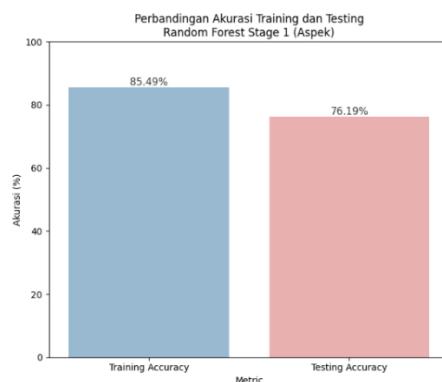
Tabel 4.21 hasil *5-Fold Cross Validation*, model *Random Forest* pada multistage Stage 1 menghasilkan akurasi rata-rata sebesar 72,43% dengan standar deviasi 0,13%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 73,98% (0,10), 72,52% (0,13), dan 72,02% (0,14), yang menunjukkan bahwa performa model relatif konsisten pada setiap *fold*. Pada Gambar 4.23 *boxplot* juga memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga kinerja model pada tahap penentuan aspek cenderung stabil dan menunjukkan perbedaan nilai yang kecil antar *fold*. Stabilitas ini menunjukkan bahwa model mampu mempelajari pola perbedaan aspek pendidikan dan politik secara konsisten pada berbagai subset data, yang didukung oleh karakteristik data aspek yang relatif seimbang dan perbedaan topik yang cukup jelas.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 20$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data

uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 22 Confusion Matrix Multistage Stage 1  $n$  estimator = 50 dan  $max\_depth$  = 20

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	74,36%	79,45%	76,82%
Politik	78,26%	72,97%	75,52%
<b>Metric</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	76,31%	76,21%	76,17%

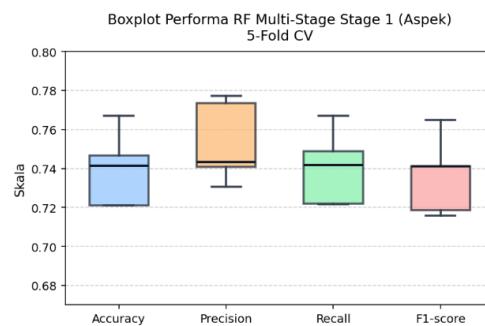


Gambar 4. 24 Perbandingan Akurasi Multistage Stage 1 Hyperparameter 50-20

Tabel 4.22 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* yang mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 76,19%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 76,31%, 76,21%, dan 76,17%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.24 grafik perbandingan akurasi data latih dan data uji juga memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 23 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 50-20

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	72,10%	73,07%	72,17%	71,85%
2	74,14%	74,31%	74,17%	74,11%
3	72,10%	74,07%	72,20%	71,58%
4	74,65%	77,34%	74,86%	74,12%
5	76,71%	77,73%	76,71%	76,49%
Rata-rata	73,94% (0,19%)	75,30% (0,20%)	74,02% (0,19%)	73,63% (0,20%)



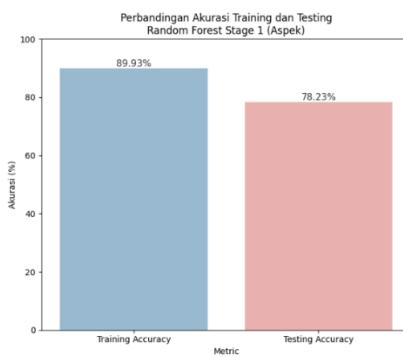
Gambar 4. 25 Boxplot Performa Multistage Stage 1 Hyperparameter 50-20

Tabel 4.23 hasil 5-Fold Cross Validation pada multistage Stage 1, model *Random Forest* menghasilkan akurasi rata-rata sebesar 73,94% dengan standar deviasi 0,19%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 75,30% (0,20), 74,02% (0,19), dan 73,63% (0,20). Nilai standar deviasi yang relatif kecil pada seluruh metrik menunjukkan bahwa variasi performa antar *fold* cenderung rendah, sehingga model mampu memberikan hasil yang konsisten pada berbagai subset data latih. Hal tersebut juga diperkuat oleh visualisasi boxplot pada Gambar 4.25 yang memperlihatkan sebaran nilai yang sempit serta tidak adanya perbedaan ekstrem antar *fold* pada setiap metrik evaluasi. Dengan demikian, model pada tahap penentuan aspek dapat dikatakan memiliki stabilitas dan konsistensi kinerja yang baik, serta mampu melakukan generalisasi dengan cukup efektif pada proses klasifikasi aspek.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 30$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 24 *Confusion Matrix Multistage Stage 1*  $n\_estimator = 50$  dan  $max\_depth = 30$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	76,62%	80,82%	78,67%
Politik	80,00%	75,68%	77,78%
<b>Macro Avg</b>	<b>78,31%</b>	<b>78,25%</b>	<b>78,22%</b>



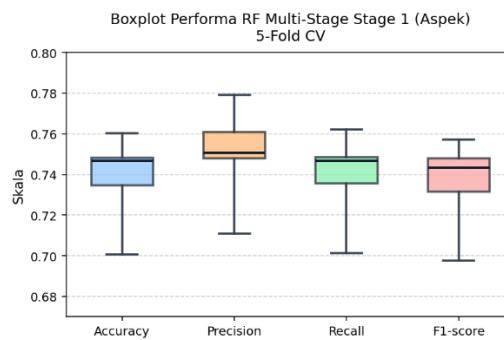
Gambar 4. 26 Perbandingan Akurasi *Multistage Stage 1* Hyperparameter 50-30

Tabel 4.24 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* yang mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 78,23%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 78,31%, 78,25%, dan 78,22%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.26 grafik perbandingan akurasi data latih dan data uji

memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 25 Hasil Evaluasi *5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 50-30*

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	73,46%	74,79%	73,54%	73,15%
2	74,82%	75,05%	74,86%	74,78%
3	70,06%	71,07%	70,14%	69,75%
4	76,02%	77,89%	76,20%	75,69%
5	74,65%	76,07%	74,65%	74,30%
Rata-rata	73,81% (0,22%)	74,97% (0,24%)	73,88% (0,22%)	73,53% (0,23%)



Gambar 4. 27 Boxplot Performa *Multistage Stage 1 Hyperparameter 50-30*

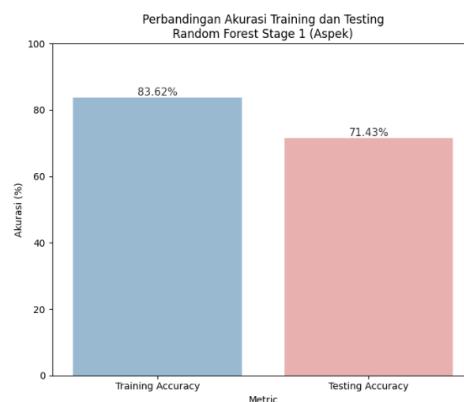
Tabel 4.25 hasil *5-Fold Cross Validation* pada *multistage Stage 1*, model *Random Forest* menghasilkan akurasi rata-rata sebesar 73,81% dengan standar deviasi 0,22%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 74,97% (0,24), 73,88% (0,22), dan 73,53% (0,23). Nilai standar deviasi yang relatif kecil pada seluruh metrik menunjukkan bahwa variasi performa antar *fold* cenderung rendah, sehingga model mampu memberikan hasil yang konsisten pada berbagai subset data latih. Hal ini juga diperkuat oleh visualisasi boxplot pada Gambar 4.27 yang memperlihatkan sebaran nilai yang sempit serta tidak adanya perbedaan ekstrem antar *fold* pada setiap metrik evaluasi. Dengan

demikian, model pada tahap penentuan aspek dapat dikatakan memiliki stabilitas dan konsistensi kinerja yang baik dalam mengklasifikasikan aspek.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 10$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 26 *Confusion Matrix Multistage Stage 1 n estimator = 75* dan  $max\_depth = 10$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	70,13%	73,97%	72,00%
Politik	72,86%	68,92%	70,83%
<i>Macro Avg</i>	71,49%	71,45%	71,42%



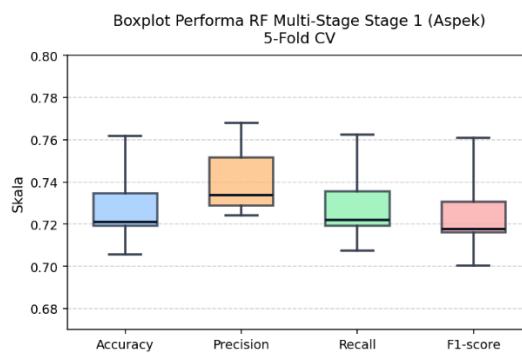
Gambar 4. 28 Perbandingan Akurasi *Multistage Stage 1 Hyperparameter 75-10*

Tabel 4.26 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 71,43%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 71,49%, 71,45%, dan 71,42%, yang menunjukkan

bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.28 grafik perbandingan akurasi data latih dan data uji memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 27 Hasil Evaluasi *5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 75-10*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	73,46%	75,14%	73,55%	73,06%
2	76,19%	76,78%	76,24%	76,07%
3	72,10%	73,36%	72,18%	71,77%
4	70,54%	72,41%	70,73%	70,04%
5	71,91%	72,88%	71,91%	71,61%
Rata-rata	72,84% (0,21%)	74,11% (0,18%)	72,92% (0,21%)	72,51% (0,22%)



Gambar 4. 29 Boxplot Performa *Multistage Stage 1 Hyperparameter 75-10*

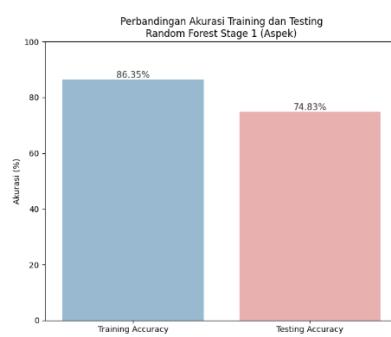
Tabel 4.27 hasil *5-Fold Cross Validation* pada *multistage Stage 1*, model *Random Forest* mampu menghasilkan akurasi rata-rata sebesar 72,84% dengan standar deviasi 0,21%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 74,11% (0,18), 72,92% (0,21), dan 72,51% (0,21). Nilai standar deviasi yang relatif kecil pada seluruh metrik menunjukkan bahwa variasi performa antar *fold* cenderung rendah, sehingga model mampu memberikan hasil

yang konsisten pada berbagai subset data latih. Hal ini juga diperkuat oleh visualisasi boxplot pada Gambar 4.29 yang memperlihatkan sebaran nilai yang sempit serta tidak adanya perbedaan ekstrem antar *fold* pada setiap metrik evaluasi. Dengan demikian, kinerja model pada tahap penentuan aspek dapat dikatakan stabil dan konsisten.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 20$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 28 *Confusion Matrix Multistage Stage 1*  $n\_estimator = 75$  dan  $max\_depth = 20$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	73,68%	76,71%	75,17%
Politik	76,06%	72,97%	74,48%
<i>Macro Avg</i>	74,87%	74,84%	74,83%



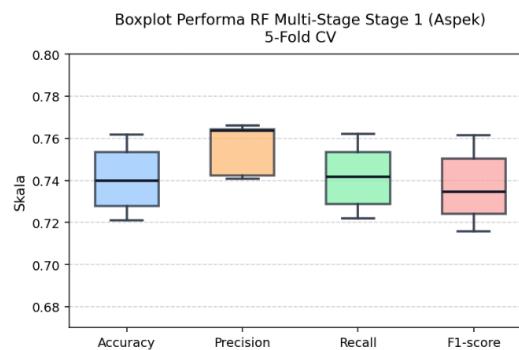
Gambar 4. 30 Perbandingan Akurasi *Multistage Stage 1* *Hyperparameter 75-20*

Tabel 4.28 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan

akurasi data uji sebesar 74,83%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 74,87%, 74,84%, dan 74,83%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.30 grafik perbandingan akurasi data latih dan data uji juga memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 29 Hasil Evaluasi *5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 75-20*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	72,78%	74,24%	72,87%	72,42%
2	76,19%	76,42%	76,22%	76,15%
3	72,10%	74,07%	72,20%	71,58%
4	73,97%	76,35%	74,17%	73,47%
5	75,34%	76,62%	75,34%	75,04%
Rata-rata	74,08% (0,17%)	75,54% (0,12%)	74,16% (0,16%)	73,73% (0,18%)



Gambar 4. 31 Boxplot Performa *Mutistage Stage 1 Hyperparameter 75-20*

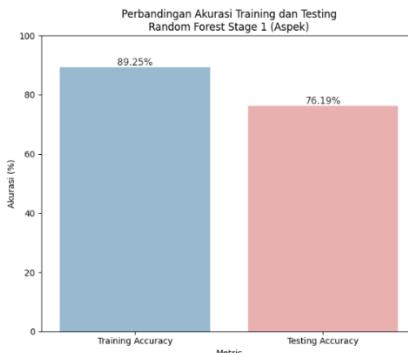
Tabel 4.29 hasil *5-Fold Cross Validation*, model *Random Forest* pada *multistage Stage 1* menghasilkan akurasi rata-rata sebesar 74,08% dengan standar deviasi 0,17%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 75,54% (0,12), 74,16% (0,16), dan 73,73% (0,18). Nilai standar

deviasi yang relatif kecil pada seluruh metrik menunjukkan bahwa variasi performa antar *fold* cenderung rendah, sehingga model mampu memberikan hasil yang konsisten pada berbagai subset data latih. Hal ini mengindikasikan bahwa proses pembelajaran model pada tahap penentuan aspek tidak bergantung pada satu pembagian data tertentu. Visualisasi boxplot pada Gambar 4.31 juga memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, yang semakin menegaskan bahwa kinerja model pada tahap pertama bersifat stabil dan menunjukkan perbedaan nilai yang kecil antar *fold*.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 30$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 30 Confusion Matrix Multistage Stage 1  $n\_estimator = 75$  dan  $max\_depth = 30$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	75,68%	76,71%	76,19%
Politik	76,71%	75,68%	76,19%
<b>Macro Avg</b>	<b>76,19%</b>	<b>76,19%</b>	<b>76,19%</b>

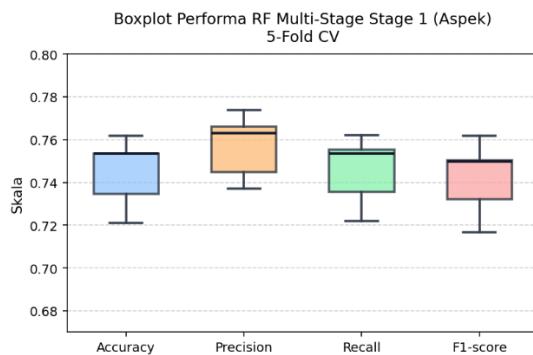


Gambar 4. 32 Perbandingan Akurasi *Multistage Stage 1 Hyperparameter 75-30*

Tabel 4.30 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 76,19%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 76,19%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.32 grafik perbandingan akurasi data latih dan data uji memperlihatkan adanya selisih, namun masih berada dalam batas yang wajar, sehingga model pada tahap pertama tetap mampu melakukan generalisasi aspek dengan cukup baik. Selain itu, evaluasi menggunakan *5-Fold Cross Validation* menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 31 Hasil Evaluasi *5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 75-30*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	73,46%	74,48%	73,53%	73,22%
2	76,19%	76,30%	76,21%	76,17%
3	72,10%	73,69%	72,19%	71,68%
4	75,34%	77,37%	75,52%	74,96%
5	75,34%	76,62%	75,34%	75,04%
Rata-rata	74,49% (0,16%)	75,69% (0,15%)	74,56% (0,16%)	74,21% (0,17%)



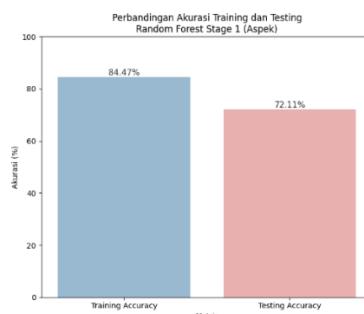
Gambar 4. 33 Boxplot Performa *Multistage Stage 1* Hyperparameter 75-30

Tabel 4.31 hasil *5-Fold Cross Validation* pada *multistage Stage 1*, model *Random Forest* menghasilkan akurasi rata-rata sebesar 74,49% dengan standar deviasi 0,16%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 75,69% (0,15), 74,56% (0,16), dan 74,21% (0,17). Nilai standar deviasi yang relatif kecil pada seluruh metrik menunjukkan bahwa variasi performa antar *fold* cenderung rendah, sehingga model mampu memberikan hasil yang konsisten pada berbagai subset data latih. Hal ini juga didukung oleh visualisasi boxplot pada Gambar 4.33 yang memperlihatkan sebaran nilai yang sempit tanpa perbedaan ekstrem antar *fold*, sehingga kinerja model pada tahap penentuan aspek dapat dikatakan stabil dan memiliki kemampuan generalisasi yang cukup baik.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 10$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 32 *Confusion Matrix Multistage Stage 1 n\_estimator = 100 dan max\_depth = 10*

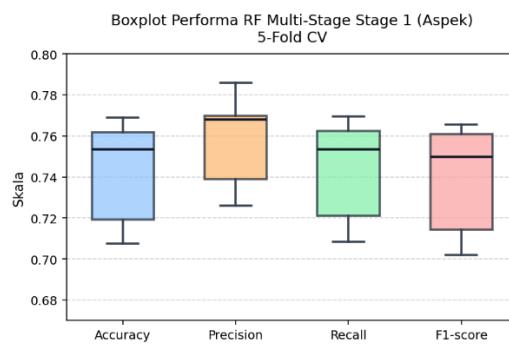
Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	70,51%	75,34%	72,85%
Politik	73,91%	68,92%	71,33%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	72,21%	72,13%	72,09%

Gambar 4. 34 Perbandingan Akurasi *Multistage Stage 1 Hyperparameter 100-10*

Tabel 4.32 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 72,11%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 72,21%, 72,13%, dan 72,09%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.34 grafik perbandingan akurasi data latih dan data uji juga memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 33 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 100-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	76,87%	78,57%	76,95%	76,55%
2	76,19%	76,78%	76,24%	76,07%
3	70,74%	72,60%	70,84%	70,19%
4	71,91%	73,89%	72,10%	71,43%
5	75,34%	76,98%	75,34%	74,96%
Rata-rata	74,21% (0,27%)	75,76% (0,24%)	74,29% (0,26%)	73,84% (0,28%)



Gambar 4. 35 Boxplot Performa Multistage Stage 1 Hyperparameter 100-10

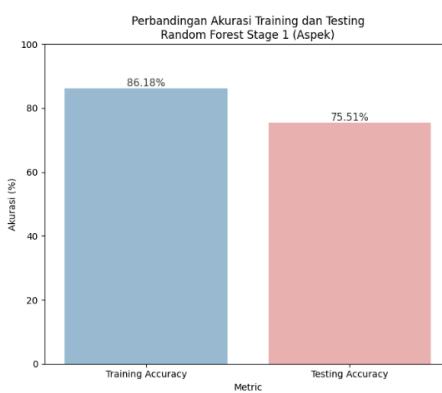
Tabel 4.33 hasil 5-Fold Cross Validation, model Random Forest pada multistage Stage 1 menghasilkan akurasi rata-rata sebesar 74,21% dengan standar deviasi 0,27%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 75,76% (0,24), 74,29% (0,26), dan 73,84% (0,28). Hasil ini menunjukkan bahwa performa model relatif konsisten pada setiap *fold*, meskipun terdapat variasi performa yang sedikit lebih besar dibandingkan beberapa konfigurasi sebelumnya. Pada Gambar 4.35 boxplot juga memperlihatkan sebaran nilai yang masih tergolong sempit pada seluruh metrik evaluasi, sehingga kinerja model pada tahap penentuan aspek tetap bersifat stabil dan tidak menunjukkan perbedaan nilai antar *fold*.

Pendekatan multistage classification tahap pertama menggunakan algoritma Random Forest dengan kombinasi *n\_estimator* = 100 dan *max\_depth* = 20 untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data

dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 34 *Confusion Matrix Multistage Stage 1*  $n\_estimator = 100$  dan  $max\_depth = 20$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	74,03%	78,08%	76,00%
Politik	77,14%	72,97%	75,00%
<b>Macro Avg</b>	<b>75,58%</b>	<b>75,53%</b>	<b>75,50%</b>



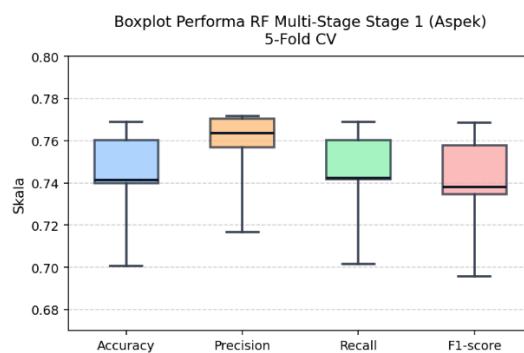
Gambar 4. 36 Perbandingan Akurasi *Multistage Stage 1* Hyperparameter 100-20

Tabel 4.34 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 75,51%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 75,58%, 75,53%, dan 75,50%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.36 grafik perbandingan akurasi data latih dan data uji juga memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan

evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 35 Hasil Evaluasi *5-Fold Cross-Validation Mutistage Stage 1 Hyperparameter 100-20*

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	74,14%	75,68%	74,23%	73,79%
2	76,87%	77,04%	76,89%	76,84%
3	70,06%	71,68%	70,15%	69,55%
4	73,97%	76,35%	74,17%	73,47%
5	76,02%	77,17%	76,02%	75,77%
Rata-rata	74,21% (0,26%)	75,58% (0,22%)	74,29% (0,25%)	73,88% (0,27%)



Gambar 4. 37 Boxplot Performa *Multistage Stage 1 Hyperparameter 100-20*

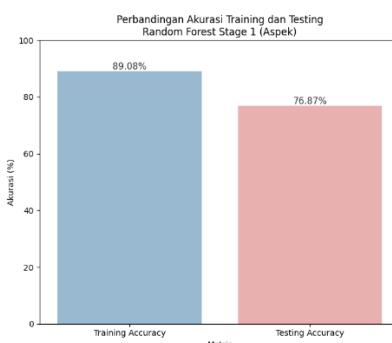
Tabel 4.35 hasil *5-Fold Cross Validation*, model *Random Forest* pada *multistage Stage 1* menghasilkan akurasi rata-rata sebesar 74,21% dengan standar deviasi 0,26%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 75,58% (0,22), 74,29% (0,25), dan 73,88% (0,27). Hasil ini menunjukkan bahwa performa model relatif konsisten pada setiap *fold*. Pada Gambar 4.37 boxplot juga memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga kinerja model pada tahap penentuan aspek cenderung stabil dan menunjukkan perbedaan nilai yang kecil antar fold.

Pendekatan *multistage classification* tahap pertama menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 30$  untuk mengklasifikasikan aspek pendidikan dan politik. Proses pembagian data

dilakukan menggunakan rasio 80:20, sehingga diperoleh 586 data latih dan 147 data uji. Dataset pada tahap ini terdiri atas 369 data politik dan 364 data pendidikan, sehingga model dilatih menggunakan data seimbang dengan distribusi kelas aspek.

Tabel 4. 36 *Confusion Matrix Multistage Stage 1 n\_estimator = 100 dan max\_depth = 30*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Pendidikan	76,00%	78,08%	77,03%
Politik	77,78%	75,68%	76,71%
<b>Macro Avg</b>	<b>76,89%</b>	<b>76,88%</b>	<b>76,87%</b>

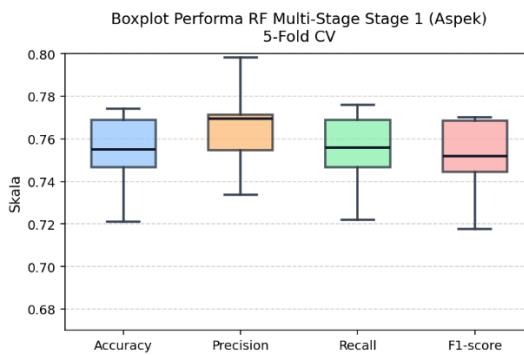


Gambar 4. 38 Perbandingan Akurasi *Multistage Stage 1 Hyperparameter 100-30*

Tabel 4.36 hasil pengujian *multistage classification* tahap pertama, model *Random Forest* mampu mengklasifikasikan aspek pendidikan dan politik dengan akurasi data uji sebesar 76,87%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 76,89%, 76,88%, dan 76,87%, yang menunjukkan bahwa performa model relatif seimbang pada kedua kelas aspek. Pada Gambar 4.38 grafik perbandingan akurasi data latih dan data uji juga memperlihatkan selisih yang tidak terlalu besar, sehingga model pada tahap pertama telah mampu melakukan generalisasi aspek dengan cukup baik. Penelitian ini menggunakan evaluasi *5-Fold Cross Validation* yang menunjukkan bahwa performa model pada tahap pertama relatif stabil dan konsisten antar *fold*.

Tabel 4. 37 Hasil Evaluasi 5-Fold Cross-Validation Multistage Stage 1 Hyperparameter 100-30

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	75,51%	77,12%	75,59%	75,17%
2	76,87%	76,94%	76,88%	76,86%
3	72,10%	73,36%	72,18%	71,77%
4	77,39%	79,82%	77,59%	77,00%
5	74,65%	75,46%	74,65%	74,45%
Rata-rata	75,30% (0,20%)	76,54% (0,23%)	75,38% (0,21%)	75,05% (0,21%)



Gambar 4. 39 Boxplot Performa Multistage Stage 1 Hyperparameter 100-30

Tabel 4.37 hasil 5-Fold Cross Validation, model Random Forest pada multistage Stage 1 menghasilkan akurasi rata-rata sebesar 75,30% dengan standar deviasi 0,20%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 76,54% (0,23), 75,38% (0,21), dan 75,05% (0,21). Hasil ini menunjukkan bahwa performa model relatif konsisten pada setiap *fold*, karena variasi nilai antar *fold* tergolong kecil. Pada Gambar 4.39 boxplot juga memperlihatkan sebaran nilai yang sempit pada seluruh metrik evaluasi, sehingga kinerja model pada tahap penentuan aspek cenderung stabil dan menunjukkan perbedaan nilai yang kecil antar *fold*, meskipun masih terdapat minor yang wajar akibat perbedaan distribusi data pada masing-masing *fold*.

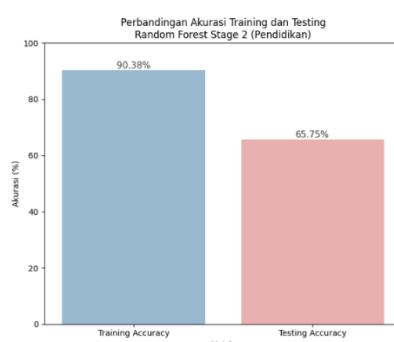
Setelah tahapan pertama (*Stage 1*) pada pendekatan multistage classification berhasil mengklasifikasikan data ke dalam kategori aspek secara cukup stabil dan konsisten, proses pengujian selanjutnya dilanjutkan ke tahap kedua

(*Stage 2*). Pada tahap ini, model difokuskan untuk melakukan klasifikasi lanjutan terhadap konten hoaks dan non-hoaks di dalam masing-masing aspek yang telah ditentukan pada *Stage 1*, sehingga evaluasi performa dapat dilakukan secara lebih spesifik dan mendalam sesuai karakteristik setiap aspek.

Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 10$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 38 *Confusion Matrix Multistage Stage 2 Aspek Pendidikan*  $n\_estimator = 50$  dan  $max\_depth = 10$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	68,18%	45,45%	54,55%
Hoaks	64,71%	82,50%	72,53%
<b>Metrix</b>			
<b>Macro Avg</b>	<b>66,44%</b>	<b>63,98%</b>	<b>63,54%</b>



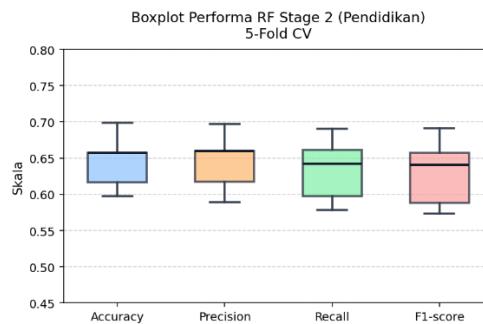
Gambar 4. 40 Perbandingan Akurasi *Multistage Stage 2 Aspek Pendidikan* *Hyperparameter 50-10*

Tabel 4.38 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* menghasilkan akurasi data uji sebesar 65,75%

dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 66,44%, 63,98%, dan 63,54%. Pada Gambar 4.40 grafik yang memaparkan perbedaan yang cukup besar antara akurasi data latih (90,38%) dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 39 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan Hyperparameter 50-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	61,64%	61,72%	59,70%	58,86%
2	65,75%	65,98%	66,10%	65,73%
3	65,75%	65,99%	64,24%	64,03%
4	69,86%	69,71%	69,05%	69,16%
5	59,72%	58,87%	57,81%	57,34%
Rata-rata	65,54% (3,96%)	64,45% (4,21%)	63,38% (4,61)	63,02% (4,89%)



Gambar 4. 41 Boxplot Performa Multistage Stage 2 Aspek Pendidikan Hyperparameter 50-10

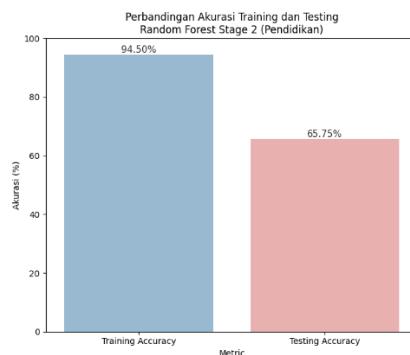
Tabel 4.39 hasil 5-Fold Cross Validation pada multistage Stage 2 aspek pendidikan, model menghasilkan akurasi rata-rata sebesar 65,54% dengan standar deviasi 3,96%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-

masing mencapai 64,45%, 63,38%, dan 63,02%. Variasi nilai yang cukup terlihat antar *fold* menunjukkan bahwa performa model pada tahap klasifikasi hoaks dan non-hoaks masih dipengaruhi oleh perbedaan distribusi data pada setiap *fold*. Hal ini diperkuat oleh boxplot pada Gambar 4.41 yang memperlihatkan sebaran nilai yang lebih lebar, sehingga dapat disimpulkan bahwa kestabilan model pada tahap kedua masih relatif terbatas dibandingkan tahap penentuan aspek.

Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 20$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 40 *Confusion Matrix Multistage Stage 2 Aspek Pendidikan*  $n\_estimator = 50$  dan  $max\_depth = 20$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	64,29%	54,55%	59,02%
Hoaks	66,67%	75,00%	70,59%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	65,48%	64,77%	64,80%

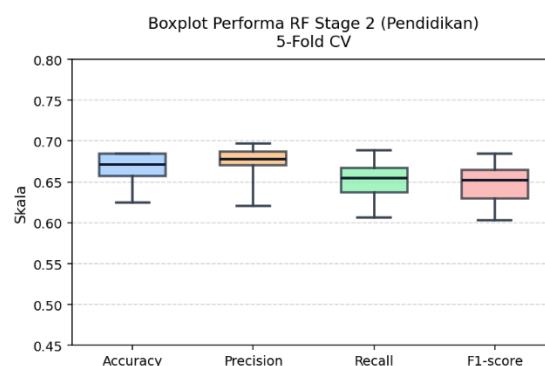


Gambar 4. 42 Perbandingan Akurasi *Multistage Stage 2 Aspek Pendidikan* *Hyperparameter 50-20*

Tabel 4.40 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* menghasilkan akurasi data uji sebesar 65,75% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 65,48%, 64,77%, dan 64,80%. Pada Gambar 4.42 memaparkan perbedaan yang cukup besar antara akurasi data latih (94,50%) dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini terjadi karena jumlah data pada tahap kedua relatif terbatas akibat proses penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model cenderung mempelajari pola spesifik pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4.41 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan Hyperparameter 50-20

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	65,75%	67,07%	63,71%	62,97%
2	68,49%	68,72%	68,86%	68,46%
3	67,12%	67,78%	65,49%	65,23%
4	68,49%	69,69%	66,74%	66,45%
5	62,50%	62,06%	60,62%	60,28%
Rata-rata	66,47% (24,9%)	67,06% (29,6%)	65,08% (3,12)	64,68% (31,6%)



Gambar 4.43 Boxplot Performa Multistage Stage 2 Aspek Pendidikan Hyperparameter 50-20

Tabel 4.41 hasil *5-Fold Cross Validation* pada *multistage Stage 2* aspek pendidikan, model *Random Forest* memperoleh akurasi rata-rata sebesar 66,47% dengan nilai *macro precision* 67,06%, *macro recall* 65,08%, dan *macro F1-score* 64,68%. Variasi nilai antar *fold* yang masih terlihat menunjukkan bahwa performa model belum sepenuhnya stabil, yang dipengaruhi oleh keterbatasan jumlah data pada tahap kedua serta penggunaan fitur *Bag of Words* berdimensi tinggi. Meskipun demikian, sebaran nilai pada boxplot pada Gambar 4.43 masih berada dalam rentang yang wajar, sehingga model tetap menunjukkan konsistensi performa pada sebagian besar fold.

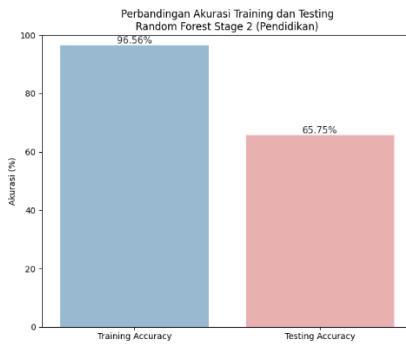
Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 30$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 42 *Confusion Matrix Multistage Stage 2* Aspek Pendidikan  $n\_estimator = 50$  dan  $max\_depth = 30$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	64,29%	54,55%	59,02%
Hoaks	66,67%	75,00%	70,59%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	65,48%	64,77%	64,80%

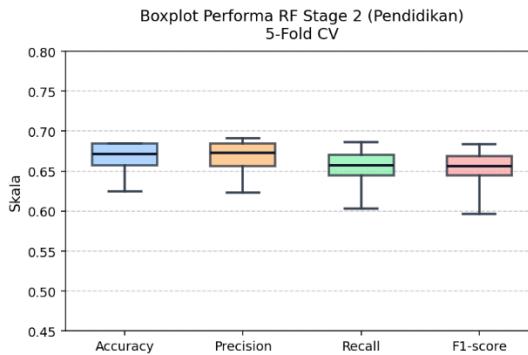


Gambar 4. 44 Perbandingan Akurasi *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 50-30*

Tabel 4.42 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* menghasilkan akurasi data uji sebesar 65,75% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 65,48%, 64,77%, dan 64,80%. Pada Gambar 4.44 memaparkan perbedaan yang cukup besar antara akurasi data latih 96,56% dan data uji menunjukkan adanya kecenderungan *overfitting*. Hal ini disebabkan oleh jumlah data pada tahap kedua yang relatif terbatas akibat proses penyaringan berdasarkan aspek, serta penggunaan representasi *Bag of Words* berdimensi tinggi yang membuat model cenderung mempelajari pola spesifik dan *noise* pada data latih. Selain itu, ketidakseimbangan kinerja antar kelas hoaks dan non-hoaks turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 43 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Pendidikan *Hyperparameter 50-30*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	68,49%	69,09%	67,00%	66,90%
2	68,49%	68,43%	68,59%	68,39%
3	65,75%	65,67%	64,50%	64,44%
4	67,12%	67,33%	65,75%	65,67%
5	62,50%	62,32%	60,31%	59,69%
Rata-rata	66,47% (24,9%)	66,57% (27%)	65,23% (31,4)	65,02% (33,2%)



Gambar 4. 45 Boxplot Performa *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 50-30*

Tabel 4.43 hasil *5-Fold Cross Validation* pada *multistage Stage 2* aspek pendidikan, model *Random Forest* memperoleh akurasi rata-rata sebesar 66,47% dengan nilai *macro precision* 66,57%, *macro recall* 65,23%, dan *macro F1-score* 65,02%. Sebaran nilai antar *fold* dengan boxplot pada Gambar 4.45 menunjukkan bahwa data relatif tidak terlalu lebar menunjukkan bahwa performa model cukup konsisten, meskipun kemampuan generalisasi masih terbatas akibat variasi data pada setiap fold.

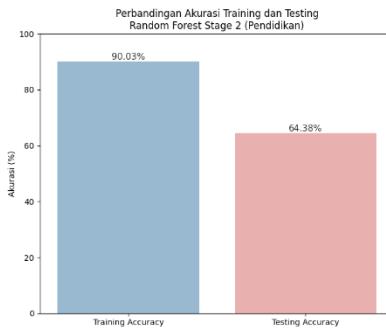
Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 10$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 44 *Confusion Matrix* *Multistage Stage 2* Aspek Pendidikan  $n\_estimator = 75$  dan  $max\_depth = 10$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	66,67%	42,42%	51,85%
Hoaks	63,46%	82,50%	71,74%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	65,06%	62,46%	61,80%

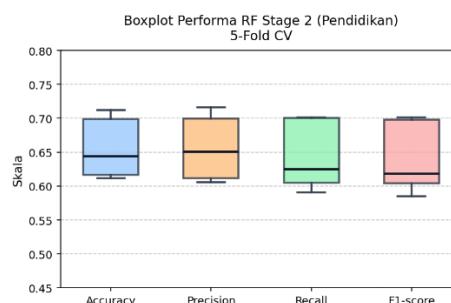


Gambar 4. 46 Perbandingan Akurasi *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 75-10*

Tabel 4.44 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* menghasilkan akurasi data uji sebesar 64,38% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 65,06%, 62,46%, dan 61,80%. Pada Gambar 4.46 menunjukkan selisih yang cukup besar antara akurasi data latih 90,03% dan data uji menunjukkan kecenderungan *overfitting*, yang dipengaruhi oleh keterbatasan jumlah data pada tahap kedua serta penggunaan fitur *Bag of Words* berdimensi tinggi sehingga kemampuan generalisasi model menjadi terbatas.

Tabel 4. 45 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Pendidikan *Hyperparameter 75-10*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	64,38%	65,06%	62,46%	61,79%
2	69,86%	69,93%	70,11%	69,81%
3	61,64%	61,15%	60,49%	60,38%
4	71,23%	71,64%	70,03%	70,13%
5	61,11%	60,54%	59,06%	58,51%
Rata-rata	65,64% (46,6%)	65,66% (50,2%)	64,43% (52,9)	64,12% (54,6%)



Gambar 4. 47 Boxplot Performa *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 75-10*

Tabel 4.45 hasil *5-Fold Cross Validation* pada *multistage Stage 2* aspek pendidikan, model *Random Forest* mampu menghasilkan akurasi rata-rata sebesar 65,64%. Nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing mencapai 65,66%, 64,43%, dan 64,12%. Hasil ini menunjukkan bahwa performa model relatif cukup stabil antar *fold*, meskipun masih terdapat variasi nilai yang terlihat pada Gambar 4.47 boxplot. Variasi tersebut mencerminkan keterbatasan jumlah data pada tahap kedua serta kompleksitas fitur teks, yang menyebabkan kemampuan generalisasi model belum sepenuhnya optimal.

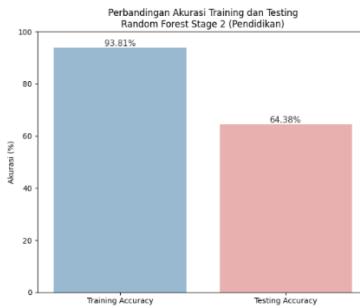
Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 20$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 46 *Confusion Matrix Multistage Stage 2 Aspek Pendidikan*  $n\_estimator = 75$  dan  $max\_depth = 20$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	64,00%	48,48%	55,17%
Hoaks	64,58%	77,50%	70,74%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	64,29%	62,99%	62,81%

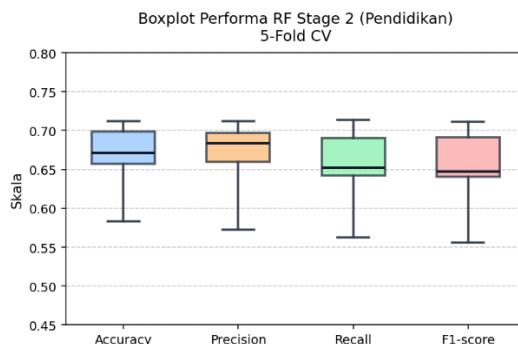


Gambar 4. 48 Perbandingan Akurasi *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 75-20*

Tabel 4.46 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* menghasilkan akurasi data uji sebesar 64,38% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 64,29%, 62,99%, dan 62,81%. Pada Gambar 4.48 memaparkan perbedaan yang cukup besar antara akurasi data latih 93,81% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini dipengaruhi oleh jumlah data pada tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model cenderung mempelajari pola spesifik pada data latih. Selain itu, ketidakseimbangan distribusi kelas hoaks dan non-hoaks turut membatasi kemampuan generalisasi model pada data uji.

Tabel 4. 47 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Pendidikan *Hyperparameter 75-20*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	67,12%	68,40%	65,22%	64,73%
2	71,23%	71,17%	71,36%	71,14%
3	69,86%	69,71%	69,05%	69,16%
4	65,75%	65,98%	64,24%	64,02%
5	58,33%	57,27%	56,25%	55,55%
Rata-rata	66,46% (50,3%)	66,50% (55%)	65,22% (57,8)	64,92% (60,2%)



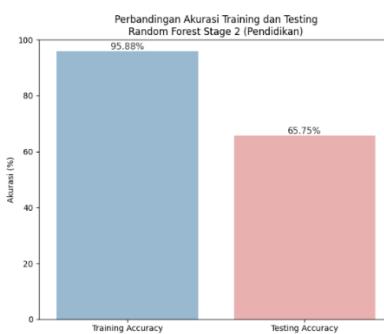
Gambar 4. 49 Boxplot Performa *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 75-20*

Tabel 4.47 hasil *5-Fold Cross Validation*, model *Random Forest* pada multistage *Stage 2* aspek pendidikan menghasilkan akurasi rata-rata sebesar 66,46% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 66,50%, 65,22%, dan 64,92%. Hasil ini menunjukkan bahwa performa model relatif cukup konsisten antar fold, meskipun masih terdapat variasi nilai yang disebabkan oleh perbedaan distribusi data pada setiap fold. Pada Gambar 4.40 boxplot memperlihatkan sebaran nilai yang moderat pada seluruh metrik evaluasi, sehingga kinerja model pada tahap klasifikasi hoaks dan non-hoaks aspek pendidikan dapat dikatakan stabil namun belum optimal dalam melakukan generalisasi.

Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 30$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 48 *Confusion Matrix Multistage Stage 2* Aspek Pendidikan  $n\_estimator = 75$  dan  $max\_depth = 30$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	66,67%	48,48%	56,14%
Hoaks	65,31%	80,00%	71,91%
<b>Metric</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	65,99%	64,24%	64,03%

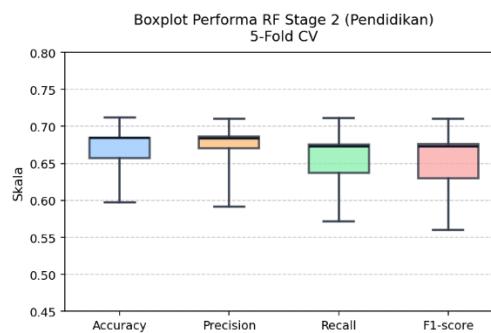


Gambar 4. 50 Perbandingan Akurasi *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 75-30*

Tabel 4.48 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* mampu menghasilkan akurasi data uji sebesar 65,75% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 65,99%, 64,24%, dan 64,03%. Pada Gambar 4.50 memaparkan perbedaan yang cukup besar antara akurasi data latih (95,88%) dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini dipengaruhi oleh keterbatasan jumlah data pada tahap kedua akibat penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang mendorong model mempelajari pola spesifik pada data latih. Selain itu, ketidakseimbangan distribusi kelas hoaks dan non-hoaks turut membatasi kemampuan generalisasi model pada data uji.

Tabel 4. 49 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan Hyperparameter 75-30

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	65,75%	67,07%	63,71%	62,97%
2	71,23%	71,00%	71,09%	71,03%
3	68,49%	68,37%	67,53%	67,61%
4	68,49%	68,65%	67,27%	67,29%
5	59,72%	59,13%	57,18%	55,97%
Rata-rata	66,46% (50,3%)	66,50% (55%)	65,36% (52,6)	64,97% (57,8%)



Gambar 4. 51 Boxplot Performa Multistage Stage 2 Aspek Pendidikan Hyperparameter 75-30

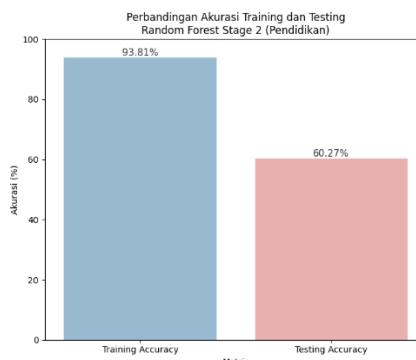
Tabel 4.49 menunjukkan hasil 5-Fold Cross Validation, model Random Forest pada multistage Stage 2 aspek pendidikan menghasilkan akurasi rata-rata sebesar 66,46% dengan nilai macro precision, macro recall, dan macro F1-score masing-masing 66,50%, 65,36%, dan 64,97%. Variasi nilai antar fold masih terlihat, terutama pada metrik F1-score, namun secara umum performa model berada pada kisaran yang relatif konsisten. Pada Gambar 4.51 boxplot menunjukkan sebaran nilai yang cukup moderat pada seluruh metrik evaluasi, sehingga kinerja model pada tahap klasifikasi hoaks dan non-hoaks aspek pendidikan dapat dikatakan stabil meskipun masih dipengaruhi oleh perbedaan distribusi data pada setiap fold.

Pendekatan multistage classification tahap kedua pada aspek pendidikan menggunakan algoritma Random Forest dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 10$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang

digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 50 *Confusion Matrix Multistage Stage 2* Aspek Pendidikan  $n\_estimator = 100$  dan  $max\_depth = 10$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	57,14%	48,48%	52,46%
Hoaks	62,22%	70,00%	65,88%
<i>Metrix</i>	<i>Precision Avg</i>	<i>Recall Avg</i>	<i>F1-Score Avg</i>
<i>Macro Avg</i>	59,68%	59,24%	59,17%



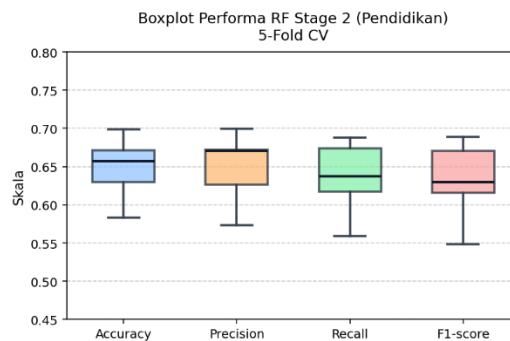
Gambar 4. 52 Perbandingan Akurasi *Multistage Stage 2* Aspek Pendidikan *Hyperparameter 100-10*

Tabel 4.50 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* mampu menghasilkan akurasi data uji sebesar 60,27% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 59,68%, 59,24%, dan 59,17%. Pada Gambar 4.52 memaparkan grafik perbedaan yang cukup besar antara akurasi data latih 93,81% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini dipengaruhi oleh jumlah data pada tahap kedua yang relatif terbatas akibat proses penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik dan *noise* pada data latih.

Selain itu, ketidakseimbangan performa antara kelas hoaks dan non-hoaks juga berkontribusi terhadap menurunnya kemampuan generalisasi model pada data uji.

Tabel 4. 51 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan Hyperparameter 100-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	65,75%	67,07%	63,71%	62,97%
2	67,12%	67,19%	67,34%	67,06%
3	63,01%	62,68%	61,74%	61,60%
4	69,86%	69,96%	68,78%	68,87%
5	58,33%	57,30%	55,93%	54,84%
Rata-rata	64,81% (43,8%)	64,84% (49,5%)	63,50% (5,07)	63,07% (54,6%)



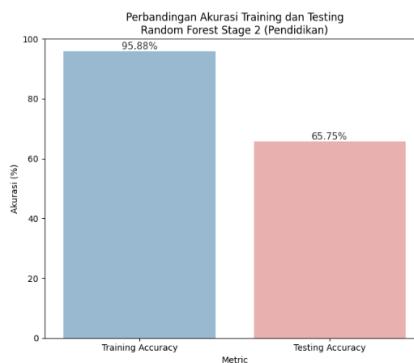
Gambar 4. 53 Boxplot Performa Multistage Stage 2 Aspek Pendidikan Hyperparameter 100-10

Tabel 4.51 hasil 5-Fold Cross Validation, model Random Forest pada multistage Stage 2 aspek pendidikan memperoleh akurasi rata-rata sebesar 64,81% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 64,84%, 63,50%, dan 63,07%. Hasil ini menunjukkan bahwa performa model relatif konsisten antar *fold*, yang juga tercermin dari sebaran nilai yang cukup merata pada boxplot pada Gambar 4.53 untuk seluruh metrik evaluasi. Meskipun demikian, kinerja model belum optimal karena jumlah data tahap kedua yang terbatas akibat penyaringan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang meningkatkan kecenderungan model mempelajari pola spesifik dan *noise*, sehingga kemampuan generalisasi pada data uji menjadi terbatas.

Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 20$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 52 *Confusion Matrix Multistage Stage 2 Aspek Pendidikan*  $n\_estimator = 100$  dan  $max\_depth = 20$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	64,29%	54,55%	59,02%
Hoaks	66,67%	75,00%	70,59%
<i>Macro Avg</i>	65,48%	64,77%	64,80%



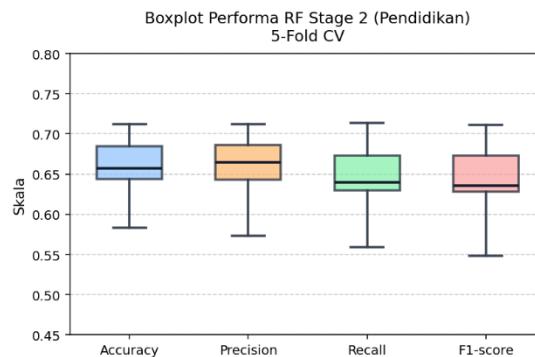
Gambar 4. 54 Perbandingan Akurasi *Multistage Stage 2 Aspek Pendidikan* *Hyperparameter 100-20*

Tabel 4.52 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* memperoleh akurasi data uji sebesar 65,75%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 65,48%, 64,77%, dan 64,80%. Pada Gambar 4.54 grafik memaparkan perbedaan yang cukup besar antara akurasi data latih 95,88% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh

jumlah data tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik pada data latih, sehingga kemampuan generalisasi model pada data uji belum optimal.

Tabel 4. 53 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Pendidikan *Hyperparameter 100-20*

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	65,75%	66,44%	63,97%	63,53%
2	71,23%	71,17%	71,36%	71,14%
3	68,49%	68,65%	67,27%	67,29%
4	64,38%	64,29%	62,99%	62,81%
5	58,33%	57,30%	55,93%	54,84%
Rata-rata	65,33% (48,5%)	65,57% (52,8%)	64,30% (57,1)	63,92% (60,6%)



Gambar 4. 55 Boxplot Performa *Mutistage Stage 2* Aspek Pendidikan *Hyperparameter 100-20*

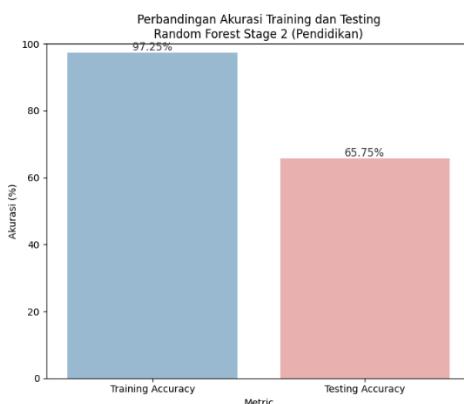
Tabel 4.53 hasil 5-Fold Cross Validation pada *multistage Stage 2* aspek pendidikan, model *Random Forest* menghasilkan akurasi rata-rata sebesar 65,33% dengan nilai *macro precision* 65,57%, *macro recall* 64,30%, dan *macro F1-score* 63,92%. Hasil ini menunjukkan bahwa performa model masih belum stabil pada setiap *fold*. Pada Gambar 4.55 boxplot memperlihatkan sebaran nilai yang cukup lebar pada seluruh metrik evaluasi, sehingga menunjukkan adanya variasi performa antar *fold*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas

serta penggunaan fitur *Bag of Words* berdimensi tinggi, yang membuat kemampuan generalisasi model belum optimal.

Pendekatan *multistage classification* tahap kedua pada aspek pendidikan menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 30$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi tahap pertama dengan total 364 data, yang dibagi menggunakan rasio 80:20 menjadi 291 data latih dan 73 data uji. Distribusi label hoaks dan non-hoaks yang relatif seimbang pada data latih maupun data uji.

Tabel 4. 54 *Confusion Matrix Multistage Stage 2* Aspek Pendidikan  $n\_estimator = 100$  dan  $max\_depth = 30$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	65,38%	51,52%	57,63%
Hoaks	65,96%	77,50%	71,26%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	65,67%	64,51%	64,45%



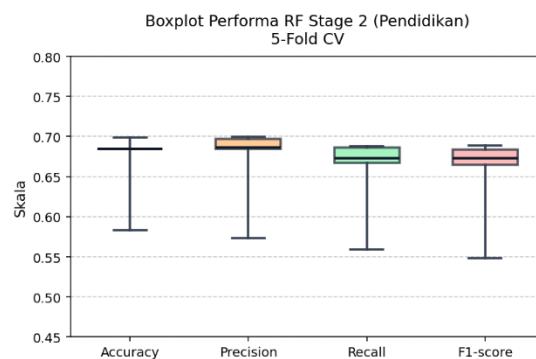
Gambar 4. 56 Perbandingan Akurasi *Multistage Stage 2* Aspek Pendidikan *Hyperparameter* 100-30

Tabel 4.54 hasil pengujian *multistage classification* tahap kedua pada aspek pendidikan, model *Random Forest* menghasilkan akurasi data uji sebesar 65,75% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing

sebesar 65,67%, 64,51%, dan 64,45%. Pada Gambar 4.56 memaparkan grafik selisih yang cukup besar antara akurasi data latih 97,25% dan data uji, yang mengindikasikan kecenderungan *overfitting*. Kondisi ini terjadi karena jumlah data pada tahap kedua relatif terbatas setelah proses pemisahan berdasarkan aspek, sementara representasi fitur *Bag of Words* memiliki dimensi yang tinggi sehingga model cenderung mempelajari pola spesifik pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model. Hasil 5-Fold Cross Validation menunjukkan variasi performa antar fold yang masih cukup terlihat, namun boxplot memperlihatkan sebaran nilai yang relatif terkendali pada seluruh metrik, sehingga kinerja model pada tahap ini dapat dikatakan cukup stabil meskipun belum optimal.

Tabel 4. 55 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Pendidikan Hyperparameter 100-30

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	68,49%	69,69%	66,74%	66,45%
2	68,49%	68,43%	68,59%	68,39%
3	68,49%	68,65%	67,27%	67,29%
4	69,86%	69,96%	68,78%	68,87%
5	58,33%	57,30%	55,93%	54,84%
Rata-rata	66,73% (47,3%)	66,81% (53,5%)	65,46% (53,9)	65,17% (58,4%)



Gambar 4. 57 Boxplot Performa Multistage Stage 2 Aspek Pendidikan Hyperparameter 100-30

Tabel 4.55 hasil *5-Fold Cross Validation* pada *multistage classification* tahap kedua aspek pendidikan, model *Random Forest* menunjukkan akurasi rata-rata sebesar 66,73%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 66,81%, 65,46%, dan 65,17%. Hasil ini menunjukkan performa yang relatif stabil antar *fold*, meskipun masih terdapat penurunan nilai pada salah satu *fold* yang memengaruhi rata-rata keseluruhan. Pada Gambar 4.57 boxplot memperlihatkan sebaran nilai metrik evaluasi yang cukup rapat, sehingga kinerja model dapat dikatakan konsisten. Namun, performa tersebut belum optimal karena jumlah data pada tahap kedua yang terbatas serta penggunaan fitur *Bag of Words* berdimensi tinggi membuat model masih sensitif terhadap variasi data dan pola spesifik pada subset tertentu.

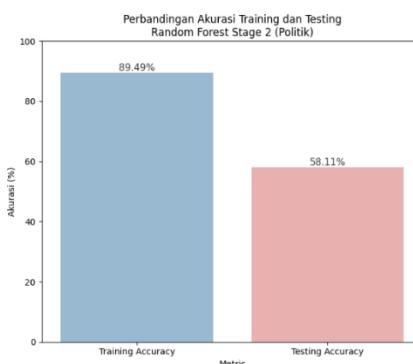
Setelah dilakukan analisis dan evaluasi kinerja model pada *multistage classification* tahap kedua untuk aspek pendidikan, pembahasan selanjutnya difokuskan pada tahap kedua aspek politik. Pada tahap ini, model *Random Forest* kembali diterapkan untuk mengklasifikasikan berita hoaks dan non-hoaks yang telah terfilter berdasarkan aspek politik, dengan tujuan mengevaluasi konsistensi performa model serta pengaruh perbedaan karakteristik data pada aspek yang berbeda.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 10$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji.

Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 56 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 50 dan max depth = 10*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	54,84%	50,00%	52,31%
Hoaks	60,47%	65,00%	62,65%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	67,65%	57,50%	57,48%



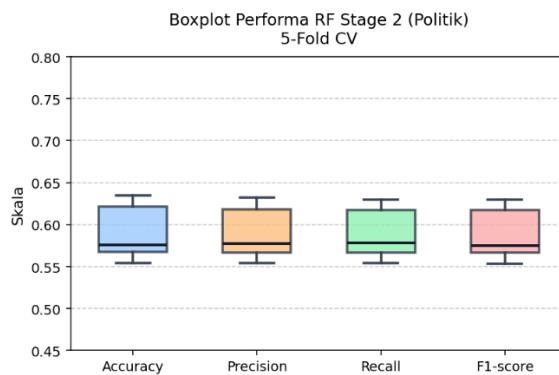
Gambar 4. 58 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik Hyperparameter 50-10*

Tabel 4.56 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* menghasilkan akurasi data uji sebesar 58,11% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 67,65%, 57,50%, dan 57,48%. Pada Gambar 4.58 memaparkan perbedaan yang cukup besar antara akurasi data latih 89,49% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini dipengaruhi oleh jumlah data tahap kedua yang relatif terbatas setelah pemisahan berdasarkan aspek politik, serta karakteristik fitur *Bag of Words* berdimensi tinggi yang membuat model cenderung mempelajari

pola spesifik pada data latih. Selain itu, variasi distribusi kelas hoaks dan non-hoaks pada data politik turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 57 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 50-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	55,40%	55,40%	55,44%	55,33%
2	63,51%	63,20%	62,94%	62,96%
3	62,16%	61,83%	61,69%	61,71%
4	56,75%	56,65%	56,69%	56,63%
5	57,53%	57,74%	57,80%	57,50%
Rata-rata	59,07% (35,5%)	58,96% (33,7%)	58,91% (32,4)	58,82% (33,2%)



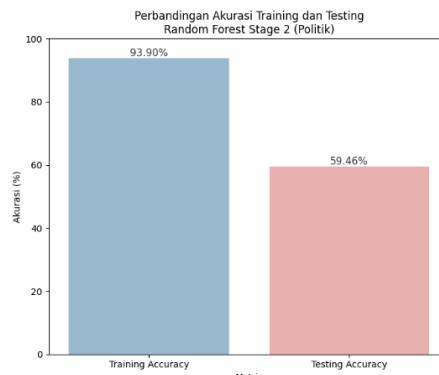
Gambar 4. 59 Boxplot Performa Multistage Stage 2 Aspek Politik Hyperparameter 50-10

Tabel 4.57 hasil evaluasi 5-Fold Cross Validation pada multistage classification tahap kedua aspek politik dengan kombinasi  $n\_estimators = 50$  dan  $max\_depth = 10$ , model Random Forest menghasilkan akurasi rata-rata sebesar 59,07%, dengan *macro precision* 58,96%, *macro recall* 58,91%, dan *macro F1-score* 58,82%. Pada Gambar 4.59 boxplot menunjukkan sebaran nilai performa yang relatif sempit pada seluruh metrik, menandakan kestabilan model antar *fold* meskipun capaian performanya masih terbatas. Hal ini mengindikasikan bahwa model belum mampu menangkap pola yang kuat pada data politik, yang dipengaruhi oleh jumlah data tahap kedua yang terbatas serta kompleksitas fitur *Bag of Words* berdimensi tinggi, sehingga kemampuan generalisasi model masih belum optimal.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 20$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 58 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 50 dan max depth = 20*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	56,25%	52,94%	54,55%
Hoaks	61,90%	65,00%	63,41%
<b>Macro Avg</b>	<b>59,08%</b>	<b>58,97%</b>	<b>58,98%</b>



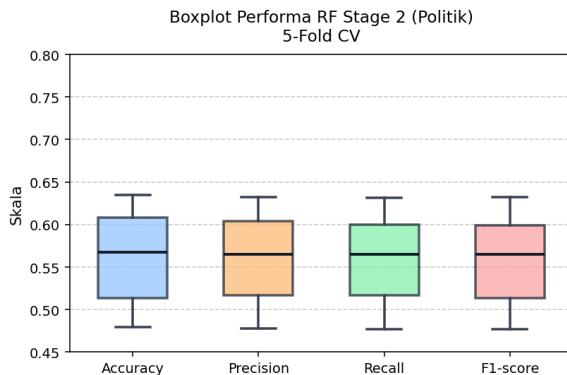
Gambar 4. 60 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik Hyperparameter 50-20*

Tabel 4.58 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* menghasilkan akurasi data uji sebesar 59,46% dengan nilai *macro precision* 59,08%, *macro recall* 58,97%, dan *macro F1-score* 58,98%. Pada Gambar 4.60 memaparkan perbedaan yang cukup besar antara

akurasi data latih 93,90% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini dipengaruhi oleh jumlah data politik tahap kedua yang relatif terbatas serta representasi fitur *Bag of Words* berdimensi tinggi, sehingga model cenderung mempelajari pola spesifik pada data latih. Boxplot hasil *5-Fold Cross Validation* memperlihatkan sebaran nilai metrik yang relatif konsisten antar fold, namun pada rentang performa yang masih rendah, menandakan bahwa kemampuan generalisasi model pada aspek politik belum optimal.

Tabel 4. 59 Hasil Evaluasi *5-Fold Cross-Validation* Mutistage Stage 2 Aspek Politik *Hyperparameter* 50-20

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	51,35%	51,69%	51,69%	51,35%
2	63,51%	63,22%	63,16%	63,18%
3	60,81%	60,42%	60,00%	59,92%
4	56,75%	56,47%	56,47%	56,47%
5	47,94%	47,74%	47,72%	47,69%
Rata-rata	59,07% (64,5%)	55,91% (62,9%)	55,81% (62,1)	55,72% (62,7%)



Gambar 4. 61 Boxplot Performa *Mutistage Stage 2* Aspek Politik *Hyperparameter* 50-20

Tabel 4.59 hasil evaluasi *5-Fold Cross Validation* pada multistage classification tahap kedua aspek politik, model *Random Forest* menghasilkan akurasi rata-rata sebesar 59,07%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 55,91%, 55,81%, dan 55,72%. Pada Gambar 4.61 memaparkan hasil boxplot yang menunjukkan sebaran nilai relatif lebar pada seluruh metrik, menandakan performa model yang belum stabil antar *fold*.

Perbedaan performa ini dipengaruhi oleh jumlah data politik yang terbatas serta kompleksitas fitur *Bag of Words* berdimensi tinggi, sehingga model cenderung kesulitan menangkap pola yang konsisten dan belum mampu melakukan generalisasi secara optimal.

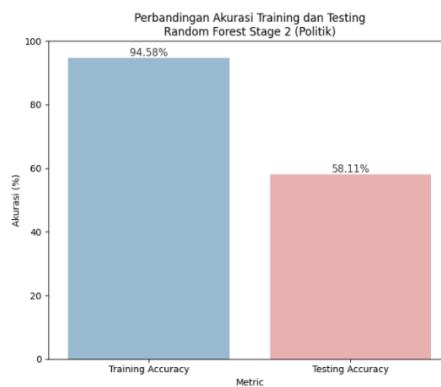
Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 50$  dan  $max\_depth = 30$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 60 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 50 dan max depth = 30*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	54,84%	50,00%	52,31%
Hoaks	60,47%	65,00%	62,65%

Metrix	Precision Avg	Recall Avg	F1-Score Avg
Macro Avg	57,65%	57,50%	57,48%

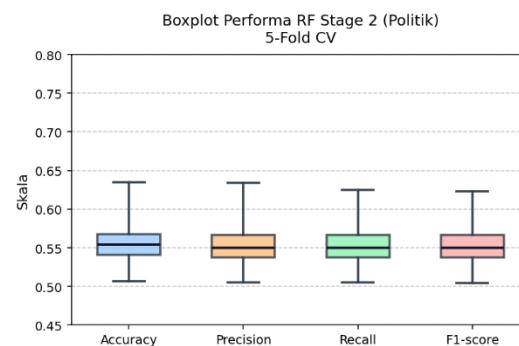


Gambar 4. 62 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik Hyperparameter 50-30*

Tabel 4.60 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* menghasilkan akurasi data uji sebesar 58,11% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 57,65%, 57,50%, dan 57,48%. Pada Gambar 4.62 memaparkan perbedaan yang cukup besar antara akurasi data latih 94,58% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini dipengaruhi oleh keterbatasan jumlah data pada aspek politik serta penggunaan fitur *Bag of Words* berdimensi tinggi, sehingga model cenderung mempelajari pola spesifik pada data latih dan belum mampu melakukan generalisasi secara optimal pada data uji.

Tabel 4. 61 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 50-30

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	54,05%	53,75%	53,75%	53,75%
2	56,75%	56,65%	56,69%	56,63%
3	63,51%	63,39%	62,50%	62,35%
4	55,40%	55,02%	55,00%	55,00%
5	50,68%	50,48%	50,49%	50,45%
Rata-rata	56,08% (47,2%)	55,86% (47,8%)	55,68% (44,3)	55,63% (43,8%)



Gambar 4. 63 Boxplot Performa *Multistage Stage 2 Aspek Politik Hyperparameter 50-30*

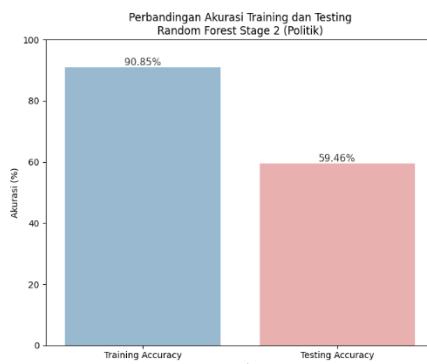
Tabel 4.61 hasil *5-Fold Cross Validation* pada *multistage classification* tahap kedua aspek politik, model *Random Forest* menghasilkan akurasi rata-rata sebesar 56,08%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 55,86%, 55,68%, dan 55,63%. Pada Gambar 4.63 boxplot

menunjukkan sebaran nilai yang relatif lebar pada seluruh metrik, menandakan performa model yang belum stabil antar *fold*. Kondisi ini mengindikasikan kemampuan generalisasi yang masih terbatas, yang dipengaruhi oleh jumlah data aspek politik yang relatif sedikit serta kompleksitas fitur *Bag of Words* berdimensi tinggi, sehingga model belum mampu menangkap pola umum secara optimal.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 10$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 62 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 75 dan max depth = 10*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	55,56%	58,82%	57,14%
Hoaks	63,16%	60,00%	61,54%
<i>Macro Avg</i>	59,36%	59,41%	59,34%

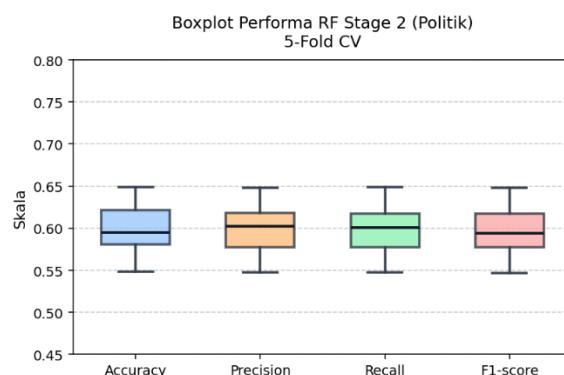


Gambar 4. 64 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik Hyperparameter 75-10*

Tabel 4.62 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* menghasilkan akurasi data uji sebesar 59,46% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 59,36%, 59,41%, dan 59,34%. Pada Gambar 4.64 memaparkan grafik perbedaan yang cukup besar antara akurasi data latih 90,85% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 63 Hasil Evaluasi 5-Fold Cross-Validation Multistage Stage 2 Aspek Politik Hyperparameter 75-10

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	59,45%	60,19%	60,07%	59,42%
2	61,16%	61,83%	61,69%	61,71%
3	64,86%	64,76%	64,85%	64,76%
4	58,10%	57,76%	57,72%	57,72%
5	54,79%	54,72%	54,77%	54,65%
Rata-rata	59,87% (38,4%)	59,85% (38,3%)	59,82% (38,3)	59,65% (38,4%)



Gambar 4. 65 Boxplot Performa *Multistage Stage 2* Aspek Politik Hyperparameter 75-10

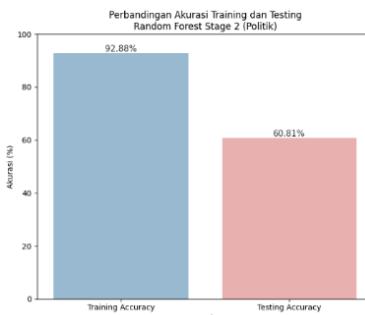
Tabel 4.63 aspek politik diperoleh nilai akurasi rata-rata sebesar 59,87%, dengan *macro precision*, *macro recall*, dan *macro F1-score* masing-masing

59,85%, 59,82%, dan 59,65%. Pada Gambar 4.65 memaparkan hasil boxplot yang menunjukkan sebaran nilai relatif sempit pada seluruh metrik evaluasi, menandakan performa model yang cukup konsisten antar *fold* meskipun masih berada pada tingkat akurasi menengah. Variasi nilai yang muncul pada beberapa *fold* mengindikasikan bahwa keterbatasan jumlah data pada tahap kedua serta karakteristik fitur *Bag of Words* berdimensi tinggi masih memengaruhi kestabilan performa model.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 20$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 64 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 75 dan max depth = 20*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	57,14%	58,82%	57,97%
Hoaks	64,10%	62,50%	63,29%
<b>Metrix</b>			
Macro Avg	60,62%	60,66%	60,63%

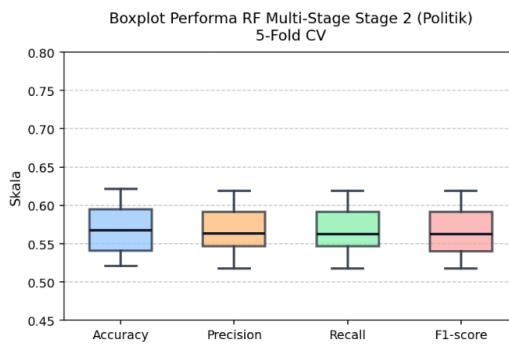


Gambar 4. 66 Perbandingan Akurasi *Multistage Stage 2* Aspek Politik *Hyperparameter 75-20*

Tabel 4.64 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* menghasilkan akurasi data uji sebesar 60,81% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 60,62%, 60,66%, dan 60,63%. Pada Gambar 4.66 memaparkan hasil perbedaan yang cukup besar antara akurasi data latih 92,88% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek politik, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 65 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Politik *Hyperparameter 75-20*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	54,05%	54,68%	54,63%	54,02%
2	59,45%	59,19%	59,19%	59,19%
3	62,16%	61,91%	61,91%	61,91%
4	56,75%	56,32%	56,25%	56,24%
5	52,05%	51,73%	51,74%	51,72%
Rata-rata	56,89% (40,5%)	56,76% (39,3%)	56,74% (39,5)	59,61% (40,4%)



Gambar 4. 67 Boxplot Performa *Multistage Stage 2* Aspek Politik *Hyperparameter 75-20*

Tabel 4.65 hasil evaluasi *5-Fold Cross Validation* pada *multistage classification* tahap kedua aspek politik model *Random Forest* menghasilkan akurasi rata-rata sebesar 56,89%, dengan *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 56,76%, 56,74%, dan 59,61%. Pada Gambar 4.67 memaparkan hasil boxplot yang menunjukkan sebaran nilai performa relatif sempit pada seluruh metrik, menandakan kestabilan kinerja model antar fold meskipun berada pada tingkat performa yang masih moderat. Hasil ini mengindikasikan bahwa model belum optimal dalam melakukan generalisasi, yang dipengaruhi oleh keterbatasan jumlah data pada tahap kedua serta kompleksitas fitur *Bag of Words* berdimensi tinggi pada aspek politik.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 75$  dan  $max\_depth = 30$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif

seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 66 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 75 dan max depth = 30*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	58,82%	58,82%	58,82%
Hoaks	65,00%	65,00%	65,00%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
<i>Macro Avg</i>	61,91%	61,91%	61,91%

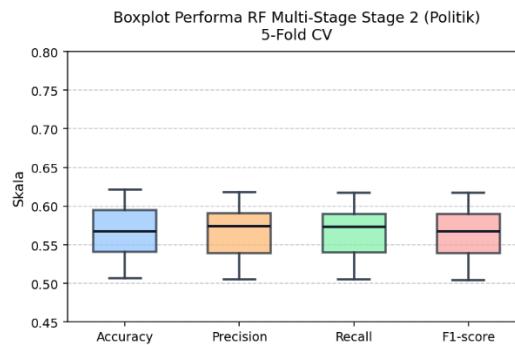


Gambar 4. 68 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik Hyperparameter 75-30*

Tabel 4.66 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* yang menghasilkan akurasi data uji sebesar 62,16%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 61,91%, 61,91%, dan 61,91%. Pada Gambar 4.68 memaparkan hasil perbedaan yang cukup besar antara akurasi data latih 95,93% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data pada tahap kedua yang relatif terbatas akibat proses penyaringan berdasarkan aspek politik, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang mendorong model mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 67 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 75-30

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	56,75%	57,44%	57,35%	56,75%
2	54,05%	53,94%	53,97%	53,91%
3	62,16%	61,83%	61,69%	61,71%
4	59,45%	59,07%	58,97%	58,98%
5	50,68%	50,48%	50,49%	50,45%
Rata-rata	56,62% (44,8%)	56,55% (44,3%)	56,49% (43,6)	56,35% (43,7%)



Gambar 4. 69 Boxplot Performa Multistage Stage 2 Aspek Politik Hyperparameter 75-30

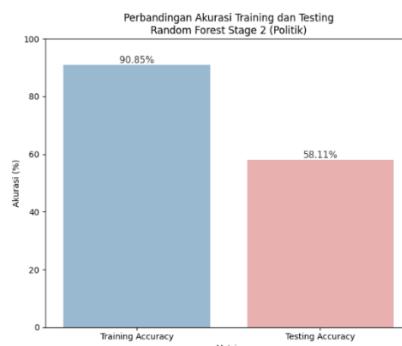
Tabel 4.69 hasil evaluasi *multistage classification* tahap kedua pada aspek politik menggunakan 5-Fold Cross Validation model Random Forest menghasilkan rata-rata akurasi sebesar 56,62%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 56,55%, 56,49%, dan 56,35%. Pada Gambar 4.69 memaparkan hasil boxplot performa yang menunjukkan sebaran nilai relatif sempit pada seluruh metrik, menandakan konsistensi kinerja model antar *fold*. Namun, nilai performa yang masih berada pada kisaran menengah mengindikasikan bahwa model belum optimal, yang dipengaruhi oleh keterbatasan jumlah data pada tahap kedua aspek politik serta kompleksitas fitur *Bag of Words* berdimensi tinggi yang menyulitkan model dalam menangkap pola general yang stabil.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma Random Forest dengan kombinasi *n\_estimator* = 100 dan *max\_depth* = 10 untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang

digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 68 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 100 dan max depth = 10*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	54,05%	58,82%	56,34%
Hoaks	62,16%	57,50%	59,74%
<i>Metrix</i>	<i>Precision Avg</i>	<i>Recall Avg</i>	<i>F1-Score Avg</i>
<i>Macro Avg</i>	58,11%	58,16%	58,04%



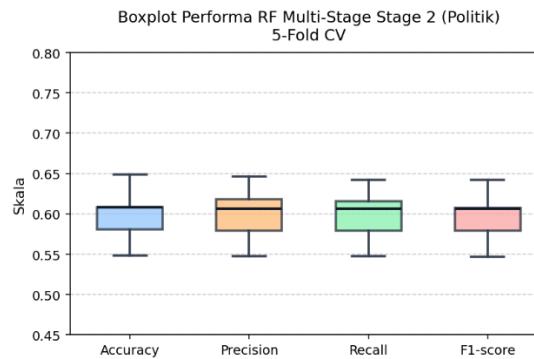
Gambar 4. 70 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik Hyperparameter 100-10*

Tabel 4.68 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* menghasilkan akurasi data uji sebesar 58,11% dengan nilai macro precision, macro recall, dan macro F1-score masing-masing 58,11%, 58,16%, dan 58,04%. Pada Gambar 4.70 memaparkan hasil perbedaan yang cukup besar antara akurasi data latih 90,85% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek politik, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model cenderung

mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 69 Hasil Evaluasi 5-Fold Cross-Validation Mutistage Stage 2 Aspek Politik Hyperparameter 100-10

<b><i>Fold</i></b>	<b><i>Accuracy (%)</i></b>	<b><i>Precision Macro (%)</i></b>	<b><i>Recall Macro (%)</i></b>	<b><i>F1-Score Macro (%)</i></b>
1	60,81%	61,77%	61,54%	60,74%
2	64,86%	64,62%	64,19%	64,21%
3	60,81%	60,62%	60,66%	60,63%
4	58,10%	57,91%	57,94%	57,91%
5	54,79%	54,72%	54,77%	54,65%
Rata-rata	59,87% (37,2%)	59,93% (37,7%)	59,82% (35,9)	59,63% (35,6%)



Gambar 4. 71 Boxplot Performa Multistage Stage 2 Aspek Politik Hyperparameter 100-10

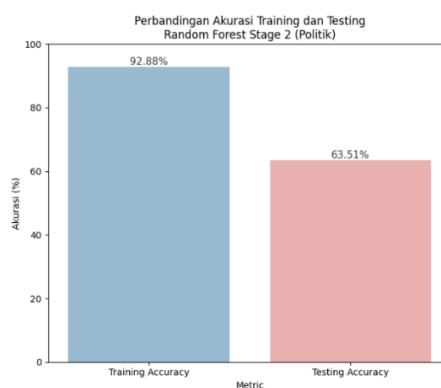
Tabel 4.69 hasil evaluasi *multistage classification* tahap kedua pada aspek politik menggunakan 5-Fold Cross Validation model Random Forest menghasilkan rata-rata akurasi sebesar 59,87%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 59,93%, 59,82%, dan 59,63%. Pada Gambar 4.71 boxplot performa menunjukkan sebaran nilai yang relatif sempit pada seluruh metrik, yang mengindikasikan bahwa kinerja model cukup konsisten antar *fold*. Meskipun demikian, capaian performa yang masih berada pada kisaran menengah menunjukkan bahwa model belum optimal, yang dipengaruhi oleh jumlah data tahap kedua aspek politik yang terbatas serta penggunaan fitur *Bag of Words*

berdimensi tinggi yang menyulitkan model dalam menangkap pola umum secara stabil.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 20$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 70 *Confusion Matrix Multistage Stage 2 Aspek Politik*  $n\_estimator = 100$  dan  $max\_depth = 20$

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	60,61%	58,82%	65,97%
Hoaks	65,85%	67,50%	66,67%
<b>Metrix</b>			
<i>Macro Avg</i>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
	63,23%	63,16%	63,18%



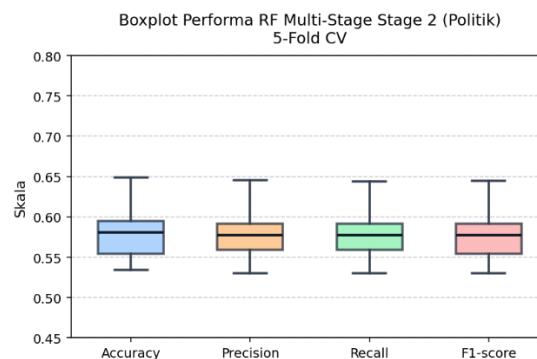
Gambar 4. 72 Perbandingan Akurasi *Multistage Stage 2 Aspek Politik* *Hyperparameter* 100-20

Tabel 4.70 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* yang mampu menghasilkan akurasi data uji sebesar

63,51% dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing 63,23%, 63,16%, dan 63,18%. Pada Gambar 4.72 memaparkan grafik perbedaan yang cukup besar antara akurasi data latih 92,88% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas akibat penyaringan berdasarkan aspek politik, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 71 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Politik *Hyperparameter 100-20*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	55,40%	55,91%	55,88%	55,39%
2	59,45%	59,19%	59,19%	59,19%
3	64,86%	64,58%	64,41%	64,44%
4	58,10%	57,76%	57,72%	57,72%
5	53,24%	52,99%	52,99%	52,99%
Rata-rata	58,25% (43,7%)	58,08% (43%)	58,03% (42,4)	57,95% (43,2%)



Gambar 4. 73 Boxplot Performa *Mutistage Stage 2* Aspek Politik *Hyperparameter 100-20*

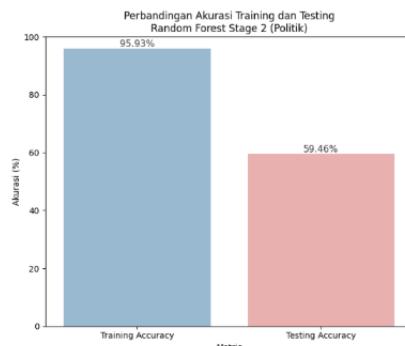
Tabel 4.71 hasil evaluasi *multistage classification* tahap kedua pada aspek politik menggunakan 5-Fold Cross Validation model Random Forest menghasilkan rata-rata akurasi sebesar 58,25%, dengan nilai *macro precision*, *macro recall*, dan

*macro F1-score* masing-masing 58,08%, 58,03%, dan 57,95%. Pada Gambar 4.71 memaparkan hasil boxplot performa menunjukkan sebaran nilai yang relatif sempit pada seluruh metrik evaluasi, yang mengindikasikan bahwa kinerja model cukup konsisten pada setiap fold. Namun, capaian performa yang masih berada pada kisaran menengah menunjukkan bahwa model belum optimal sepenuhnya. Hal ini dipengaruhi oleh keterbatasan jumlah data pada tahap kedua aspek politik akibat proses penyaringan multistage, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang menyulitkan model dalam menangkap pola umum yang stabil dan berdampak pada kemampuan generalisasi model.

Pendekatan *multistage classification* tahap kedua pada aspek politik menggunakan algoritma *Random Forest* dengan kombinasi  $n\_estimator = 100$  dan  $max\_depth = 30$  untuk mengklasifikasikan berita hoaks dan non-hoaks. Data yang digunakan merupakan hasil seleksi dari tahap pertama dengan total 369 data, yang kemudian dibagi menggunakan rasio 80:20 menjadi 295 data latih dan 74 data uji. Distribusi label hoaks dan non-hoaks pada data latih maupun data uji relatif seimbang, sehingga memungkinkan model mempelajari pola klasifikasi secara lebih representatif pada aspek politik.

Tabel 4. 72 *Confusion Matrix Multistage Stage 2 Aspek Politik n estimator = 100 dan max depth = 30*

Kelas	Confusion Matrix		
	Precision	Recall	F1-Score
Non-Hoaks	56,25%	52,94%	54,55%
Hoaks	61,90%	65,00%	63,41%
<b>Metrix</b>	<b>Precision Avg</b>	<b>Recall Avg</b>	<b>F1-Score Avg</b>
Macro Avg	59,08%	58,97%	58,98%

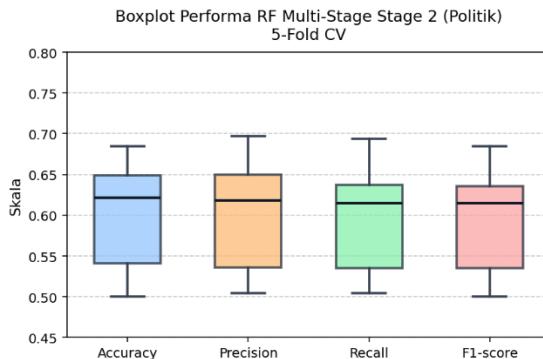


Gambar 4. 74 Perbandingan Akurasi *Multistage Stage 2* Aspek Politik *Hyperparameter 100-30*

Tabel 4.72 hasil pengujian *multistage classification* tahap kedua pada aspek politik, model *Random Forest* yang mampu menghasilkan akurasi data uji sebesar 59,46%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 59,08%, 58,97%, dan 58,98%. Pada Gambar 4.74 memaparkan hasil hasil perbedaan yang cukup besar antara akurasi data latih 95,93% dan data uji menunjukkan adanya kecenderungan *overfitting*. Kondisi ini disebabkan oleh jumlah data tahap kedua yang relatif terbatas akibat proses penyaringan berdasarkan aspek, serta penggunaan fitur *Bag of Words* berdimensi tinggi yang membuat model lebih mudah mempelajari pola spesifik dan *noise* pada data latih. Selain itu, distribusi kelas hoaks dan non-hoaks yang belum sepenuhnya seimbang turut memengaruhi kemampuan generalisasi model pada data uji.

Tabel 4. 73 Hasil Evaluasi 5-Fold Cross-Validation *Mutistage Stage 2* Aspek Politik *Hyperparameter 100-30*

Fold	Accuracy (%)	Precision Macro (%)	Recall Macro (%)	F1-Score Macro (%)
1	50,00%	50,44%	50,44%	49,99%
2	62,16%	61,81%	61,47%	61,45%
3	54,05%	53,57%	53,52%	53,51%
4	64,86%	64,98%	63,75%	63,56%
5	68,49%	69,66%	69,39%	68,46%
Rata-rata	59,91% (76,7%)	60,09% (79,7%)	59,71% (77%)	59,39% (75,3%)



Gambar 4. 75 Boxplot Performa *Multistage Stage 2* Aspek Politik *Hyperparameter* 100-30

Tabel 4.73 hasil evaluasi *multistage classification* tahap kedua pada aspek politik menggunakan *5-Fold Cross Validation* model *Random Forest* menghasilkan rata-rata akurasi sebesar 59,91%, dengan nilai *macro precision*, *macro recall*, dan *macro F1-score* masing-masing sebesar 60,09%, 59,71%, dan 59,39%. Pada Gambar 4.75 memaparkan hasil boxplot performa yang menunjukkan sebaran nilai relatif moderat pada seluruh metrik, yang mengindikasikan adanya variasi performa antar *fold* meskipun masih berada dalam rentang yang wajar. Nilai performa yang berada pada kisaran menengah ini menunjukkan bahwa model belum sepenuhnya optimal, yang dipengaruhi oleh keterbatasan jumlah data pada tahap kedua aspek politik serta kompleksitas fitur *Bag of Words* berdimensi tinggi yang menyulitkan model dalam menangkap pola general yang stabil.

#### 4.4 Pembahasan

Berdasarkan hasil uji coba pada Skenario dengan pendekatan *single stage classification* dan *multistage classification*, analisis difokuskan pada kinerja model *Random Forest* berdasarkan variasi parameter *n\_estimator* dan *max\_depth*.

Pembahasan ini memberikan gambaran ringkas mengenai stabilitas dan kualitas prediksi model pada setiap skenario.

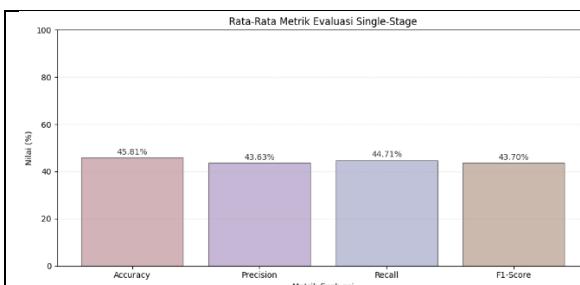
#### 4.4.1 Rangkuman Hasil Evaluasi Model

Bagian ini bertujuan untuk merangkum dan membandingkan hasil performa model *Random Forest* dengan pendekatan klasifikasi *singlestage* dan *multistage* dengan data balance. Evaluasi ini dilakukan untuk mengidentifikasi performa pada setiap konfigurasi *hyperparameter* untuk menentukan performa terbaik berdasarkan akurasi, *precision*, *recall*, *f1-score*.

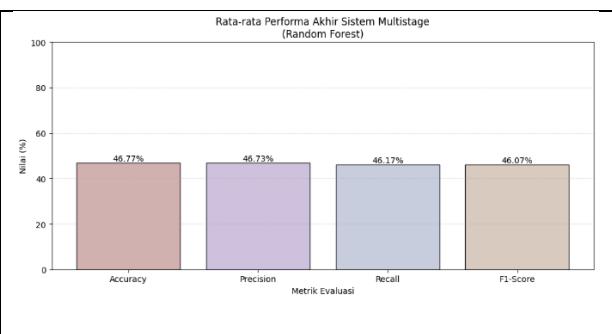
Tabel 4. 74 Hasil Evaluasi Model *Random Forest*

Uji Coba	Type of Stage Classification	Dataset	Evaluasi			
			Akurasi	Precision	Recall	F1-score
1	SingleStage	Balance	43,54%	41,07%	42,08%	40,75%
2			44,90%	43,32%	44,10%	43,37%
3			48,30%	46,09%	47,36%	46,31%
4			42,18%	39,59%	40,94%	39,69%
5			43,54%	41,23%	42,32%	41,30%
6			46,94%	44,30%	45,58%	44,43%
7			46,94%	45,01%	46,22%	45,16%
8			45,58%	43,46%	44,46%	43,58%
9			50,34%	48,60%	49,34%	48,71%
Rata-Rata			45,36%	43,97%	45,02%	44,06%
10	Multistage Stage 1	Balance	73,47%	73,83%	73,51%	73,39%
11			76,19%	76,31%	76,21%	76,17%
12			78,23%	78,31%	78,25%	78,22%
13			71,43%	71,49%	71,45%	71,42%
14			74,83%	74,87%	74,84%	74,83%
15			76,19%	76,19%	76,19%	76,19%
16			72,11%	72,21%	72,13%	72,09%
17			75,51%	75,58%	75,53%	75,50%
18			76,87%	76,89%	76,88%	76,87%
19	Multistage Stage 2 Aspek Pendidikan	Balance	65,75%	66,44%	63,98%	63,54%
20			65,75%	65,48%	64,77%	64,80%
21			65,75%	65,48%	64,77%	64,80%
22			64,38%	65,06%	62,46%	61,80%
23			64,38%	64,29%	62,99%	62,81%
24			65,75%	65,99%	64,24%	64,03%
25			60,27%	59,68%	59,24%	59,17%
26			65,75%	65,48%	64,77%	64,80%
27			65,75%	65,67%	64,51%	64,45%
28	Multistage Stage 2 Aspek Politik	Balance	58,11%	57,65%	57,50%	57,48%
29			59,46%	59,08%	58,97%	58,98%
30			58,11%	57,65%	57,50%	57,48%

Uji Coba	Type of Stage Classification	Dataset	Evaluasi			
			Akurasi	Precision	Recall	F1-score
31			59,46%	59,36%	59,41%	59,34%
32			60,81%	60,62%	60,66%	60,63%
33			62,16%	61,91%	61,91%	61,91%
34			58,11%	58,11%	58,16%	58,04%
35			63,51%	63,23%	63,16%	63,18%
36			59,46%	59,08%	58,97%	58,98%
Performa Total Multistage			46,76%	46,72%	46,16%	46,06%



Gambar 4. 76 Hasil Rata-Rata Evaluasi Singlestage



Gambar 4. 77 Hasil Rata-Rata Evaluasi Multistage

Tabel 4.74 rangkuman hasil evaluasi serta visualisasi grafik pada Gambar 4.76 dan Gambar 4.77, rata-rata performa pendekatan *singlestage classification* menunjukkan kinerja yang berada pada tingkat sedang dan relatif seimbang pada seluruh metrik evaluasi. Rata-rata akurasi yang diperoleh pada *singlestage* mencapai 45,36%, dengan nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 43,97%, 45,02%, dan 44,06%. Grafik rata-rata metrik evaluasi *singlestage* memperlihatkan bahwa keempat metrik berada pada hasil performa yang relatif berdekatan, yang mengindikasikan bahwa model *Random Forest* mampu memberikan performa yang konsisten ketika proses klasifikasi hoaks dilakukan secara langsung dalam satu ruang klasifikasi tanpa pemisahan aspek.

Sebaliknya, grafik rata-rata performa akhir sistem *multistage* menunjukkan nilai yang sedikit lebih tinggi, dengan akurasi sebesar 46,76%, serta *precision*,

*recall*, dan *F1-score* masing-masing sebesar 46,72%, 46,16%, dan 46,06%. Nilai ini tidak merepresentasikan kinerja satu model tunggal, melainkan merupakan performa *end-to-end* sistem *multistage*, yang diperoleh dari hasil perkalian kinerja *Stage 1* dan *Stage 2*. Hal ini terlihat pada grafik, di mana seluruh metrik *multistage* berada pada kisaran yang hampir seragam, mencerminkan akumulasi kesalahan dari dua tahap klasifikasi yang saling bergantung.

Perbedaan nilai performa antara pendekatan *single-stage* dan *multistage* menunjukkan adanya perbedaan alur proses klasifikasi yang digunakan. Pada pendekatan *multistage*, proses klasifikasi diawali dengan pengelompokan data berdasarkan aspek, sehingga pada tahap selanjutnya model hanya memproses data yang telah berada pada aspek yang sama. Kondisi ini membuat model pada tahap lanjutan dapat mempelajari pola hoaks yang lebih spesifik sesuai dengan aspek yang dibahas. Sebaliknya, pada pendekatan *singlestage*, model harus mengklasifikasikan seluruh data sekaligus dalam satu proses, sehingga harus menangani variasi kelas dan konteks yang lebih beragam dalam satu ruang klasifikasi.

Hasil dari grafik dan tabel menunjukkan bahwa keunggulan pendekatan *multistage* tidak hanya dilihat dari nilai performa numerik akhir. Pada pendekatan ini, proses klasifikasi dibagi ke dalam beberapa tahap, sehingga model dapat mempelajari pola hoaks berdasarkan aspek secara lebih terarah. Pembagian tahap tersebut membantu model memahami karakteristik data pada setiap aspek dengan lebih jelas. Oleh karena itu, meskipun perbedaan nilai performa antara *singlestage* dan *multistage* relatif kecil, pendekatan *multistage* tetap relevan karena mampu

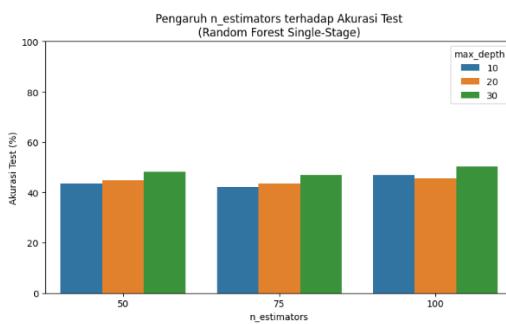
menggambarkan proses klasifikasi hoaks berbasis aspek secara lebih baik. Hasil ini menunjukkan adanya perbedaan fokus antara kedua pendekatan, pada *singlestage* menekankan pencapaian performa numerik akhir, sedangkan *multistage* menekankan alur dan proses klasifikasi yang lebih terstruktur.

#### 4.4.2 Pengaruh Jumlah *n\_estimator* pada Metode *Random Forest*

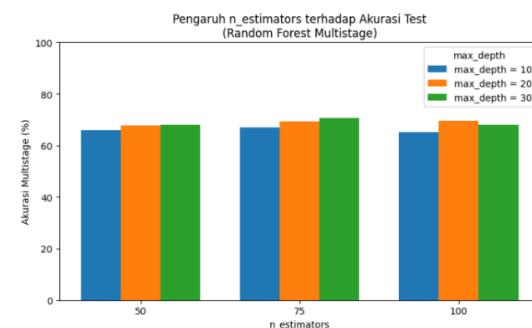
Jumlah *n\_estimators* menentukan banyaknya pohon keputusan pada algoritma *Random Forest*. Variasi parameter ini diuji untuk mengetahui pengaruhnya terhadap akurasi pengujian pada pendekatan *single-stage* dan *multistage*.

Tabel 4. 75 Pengaruh *n\_estimator* *Singlestage* dan *Multistage*

<i>N estimator</i>	<i>singlestage</i>	<i>multistage</i>
50	45,58%	67,23%
75	44,22%	68,93%
100	47,62%	67,63%



Gambar 4. 78 Grafik Pengaruh *n\_estimator* *Singlestage*



Gambar 4. 79 Grafik Pengaruh *n\_estimator* *Multistage*

Tabel 4.75 hasil parameter *n\_estimator* menunjukkan pengaruh yang cukup jelas terhadap performa model *Random Forest*, baik pada pendekatan *singlestage* maupun *multistage*. Nilai *n\_estimator* merepresentasikan jumlah pohon keputusan yang dibangun dalam suatu *ensemble*, semakin banyak pohon yang digunakan,

semakin stabil prediksi model karena keputusan akhir diperoleh dari proses agregasi banyak pohon, sehingga variansi dapat ditekan dan model tidak terlalu bergantung pada pola yang terbentuk dari satu pohon tertentu. Hal ini terlihat pada Gambar 4.78 dan Gambar 4.79 yang menampilkan perubahan akurasi pengujian pada berbagai jumlah pohon dan kedalaman pohon.

Pada pendekatan *singlestage*, peningkatan jumlah *n\_estimator* menunjukkan kecenderungan peningkatan performa meskipun tidak bersifat linear. Ketika *n\_estimator* = 50, rata-rata akurasi pengujian yang diperoleh sebesar 45,58%. Pada *n\_estimator* = 75, akurasi sedikit menurun menjadi 44,22%, namun kembali meningkat secara cukup signifikan pada *n\_estimator* = 100 dengan nilai akurasi tertinggi sebesar 47,62%. Pola ini menunjukkan bahwa pada jumlah pohon yang lebih besar, model *singlestage* menjadi lebih stabil dalam menghasilkan keputusan, sehingga mampu menangkap pola data secara lebih komprehensif. Pada *n\_estimator* = 75 mengindikasikan bahwa peningkatan jumlah pohon tidak selalu langsung menghasilkan kenaikan performa, terutama ketika masih dipengaruhi oleh kedalaman pohon dan distribusi fitur yang dipelajari.

Berbeda dengan pendekatan *singlestage*, pada *multistage* secara konsisten menghasilkan performa numerik yang lebih tinggi. Akurasi tertinggi pada pendekatan *multistage* dicapai ketika *n\_estimator* = 75 dengan nilai sebesar 68,93%, lebih tinggi dibandingkan seluruh konfigurasi *singlestage*. Pada *n\_estimator* = 50, akurasi *multistage* berada pada 67,23%, sedangkan ketika jumlah pohon ditingkatkan menjadi *n\_estimator* = 100, akurasi sedikit menurun menjadi 67,63%. Pola ini menunjukkan bahwa pada pendekatan *multistage* terdapat titik

optimal jumlah pohon, di mana penambahan pohon hingga batas tertentu mampu meningkatkan stabilitas prediksi, namun penambahan lebih lanjut tidak selalu diikuti oleh peningkatan performa.

Performa *multistage* yang lebih tinggi dibandingkan *singlestage* terjadi karena mekanisme klasifikasi bertingkat yang digunakan mampu mengurangi kompleksitas masalah pada tahap akhir. Pada *multistage* terdiri dari dua tahapan klasifikasi yang saling bergantung, di mana *Stage 1* berfungsi untuk mengelompokkan data berdasarkan aspek, sehingga *Stage 2* hanya mempelajari pola hoaks pada ruang data yang lebih spesifik dan tematik. Kesalahan pada *Stage 1* dapat terpropagasi ke *Stage 2*, pembatasan ruang klasifikasi ini tetap memberikan keuntungan dalam membedakan pola hoaks dibandingkan pendekatan *singlestage* yang harus mempelajari seluruh variasi data secara langsung.

Dengan demikian, dapat disimpulkan bahwa peningkatan *n\_estimator* pada pendekatan *singlestage* memberikan dampak positif terhadap performa model hingga jumlah pohon yang lebih besar, namun secara keseluruhan performanya masih berada di bawah pendekatan *multistage*. Sementara itu, pada pendekatan *multistage* terdapat titik optimal pada jumlah pohon menengah, yaitu *n\_estimator* = 75. Hal ini menegaskan bahwa pemilihan jumlah pohon keputusan perlu disesuaikan dengan karakteristik hasil klasifikasi yang digunakan, dan pendekatan *multistage* lebih efektif secara numerik dalam konteks deteksi hoaks berbasis aspek.

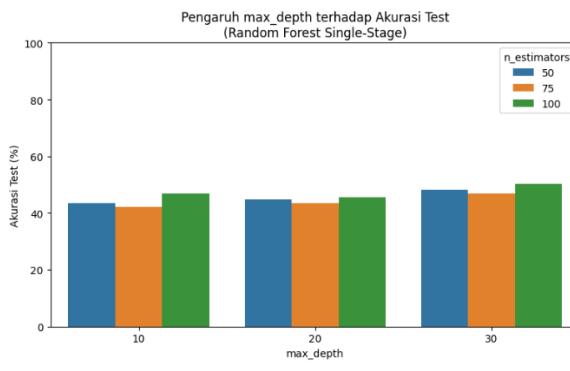
#### 4.4.3 Pengaruh *Max\_depth* pada *Random Forest*

Parameter *max\_depth* menentukan kedalaman maksimum pohon keputusan pada algoritma *Random Forest*. Variasi nilai *max\_depth* diuji untuk menganalisis

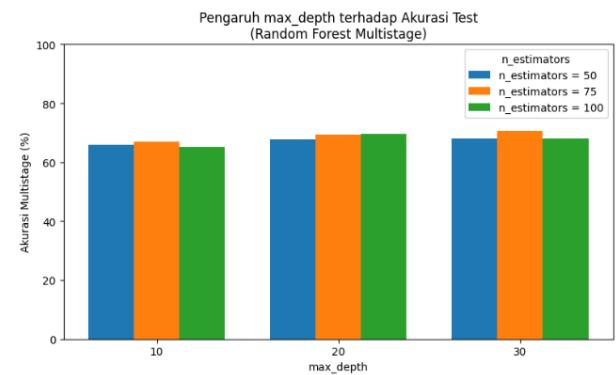
pengaruh tingkat kompleksitas pohon terhadap akurasi pengujian pada pendekatan *single-stage* dan *multistage*.

Tabel 4. 76 Pengaruh *max\_depth* *Singlestage* dan *Multistage*

<b>Max Depth</b>	<b>singlestage</b>	<b>multistage</b>
10	44,22%	65,96%
20	44,67%	68,93%
30	48,53%	68,90%



Gambar 4. 80 Grafik Pengaruh *Max\_Depth Singlestage*



Gambar 4. 81 Grafik Pengaruh *Max\_Depth Multistage*

Tabel 4.76, parameter *max\_depth* menunjukkan pengaruh yang cukup jelas terhadap performa model *Random Forest*, baik pada pendekatan *singlestage* maupun *multistage*. Parameter *max\_depth* mengontrol kedalaman maksimum pohon keputusan, sehingga menentukan sejauh mana model dapat mempelajari hubungan kompleks antar fitur. Semakin besar nilai *max\_depth*, semakin kompleks struktur pohon yang terbentuk, yang memungkinkan model menangkap pola data yang lebih rinci, namun juga berpotensi meningkatkan risiko *overfitting* apabila tidak diimbangi dengan karakteristik data yang memadai.

Pada pendekatan *singlestage*, peningkatan nilai *max\_depth* diikuti oleh peningkatan performa pengujian secara bertahap. Ketika *max\_depth* = 10, rata-rata akurasi pengujian yang diperoleh sebesar 44,22%. Nilai ini meningkat menjadi

44,67% pada  $max\_depth = 20$ , dan kembali meningkat secara lebih nyata hingga mencapai 48,53% pada  $max\_depth = 30$ . Pola ini menunjukkan bahwa pada *singlestage*, peningkatan kedalaman pohon membantu model dalam mempelajari struktur data teks yang lebih kompleks dan heterogen. Hal tersebut juga terlihat pada Gambar 4.80, di mana seluruh variasi  $n\_estimator$  menunjukkan kecenderungan peningkatan akurasi seiring bertambahnya nilai  $max\_depth$ , yang mengindikasikan bahwa model *singlestage* masih memperoleh manfaat dari struktur pohon yang lebih dalam untuk menangkap keragaman pola lintas aspek dalam satu ruang klasifikasi.

Sebaliknya, pada pendekatan *multistage*, peningkatan nilai  $max\_depth$  memberikan peningkatan performa yang lebih terbatas. Akurasi pengujian meningkat dari 65,96% pada  $max\_depth = 10$  menjadi 68,93% pada  $max\_depth = 20$ , namun tidak menunjukkan peningkatan yang berarti ketika  $max\_depth = 30$  dengan akurasi sebesar 68,90%. Pola ini menunjukkan bahwa penambahan kedalaman pohon pada *multistage* tidak selalu diikuti oleh peningkatan performa yang signifikan. Hal ini juga terlihat pada Gambar 4.81, di mana kurva akurasi *multistage* cenderung mendatar pada nilai  $max\_depth$  yang lebih tinggi.

Fenomena tersebut dapat dijelaskan oleh karakteristik *multistage classification*, khususnya pada *Stage 2*, di mana data yang diproses telah melalui penyaringan berdasarkan aspek sehingga jumlah dan keragamannya relatif lebih terbatas. Dalam kondisi ini, pohon keputusan dengan kedalaman yang terlalu besar berpotensi mempelajari pola yang terlalu spesifik terhadap data latih, sehingga tidak lagi memberikan keuntungan berarti dalam meningkatkan akurasi data uji.

Oleh karena itu, nilai *max\_depth* = 20 hingga 30 dapat dianggap sebagai rentang yang relatif optimal pada pendekatan *multistage*, karena mampu menjaga keseimbangan antara kemampuan representasi model dan stabilitas generalisasi.

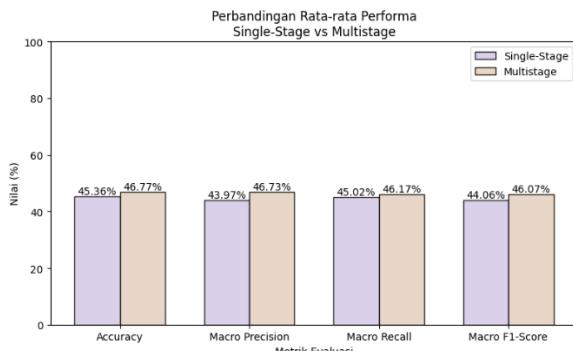
Hasil ini menunjukkan bahwa peningkatan *max\_depth* memberikan dampak positif yang lebih kuat pada pendekatan *singlestage* dibandingkan *multistage*. Pada *singlestage*, model masih diuntungkan oleh struktur pohon yang lebih dalam untuk menangkap kompleksitas data teks yang beragam, sedangkan pada *multistage*, kompleksitas model perlu dikendalikan agar tidak memperkuat kecenderungan *overfitting* pada data yang lebih sempit di tahap lanjutan. Hal ini pemilihan nilai *max\_depth* harus disesuaikan dengan hasil klasifikasi yang digunakan serta karakteristik distribusi data pada setiap tahap pemodelan.

#### 4.5 Perbandingan *SingleStage* dan *Multistage Classification*

Bagian ini membahas perbandingan performa rata-rata antara pendekatan *singlestage* dan *multistage*. Perbandingan dilakukan menggunakan metrik akurasi, *precision*, *recall*, dan *f1-score* untuk melihat pengaruh strategi klasifikasi dan teknik penyeimbangan data terhadap kinerja model *Random Forest*.

Tabel 4. 77 Perbandingan *Singlestage* dan *Multistage*

<b>Matrix</b>	<b>Singlestage</b>	<b>Multistage</b>
Akurasi	45,36%	46,77%
Presisi	43,97%	46,73%
Recall	45,02%	46,17%
F1-Score	44,06%	46,07%



Gambar 4. 82 Grafik Perbandingan *Singlestage* dan *Multistage*

Tabel 4.77 dan visualisasi grafik pada Gambar 4.82, rata-rata performa pendekatan *multistage classification* menunjukkan nilai yang sedikit lebih tinggi dibandingkan pendekatan *singlestage classification* pada seluruh metrik evaluasi. Pada *single-stage*, rata-rata akurasi yang diperoleh sebesar 45,36%, dengan nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 43,97%, 45,02%, dan 44,06%. Sementara itu, pendekatan *multistage* menghasilkan rata-rata akurasi sebesar 46,77%, serta nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 46,73%, 46,17%, dan 46,07%. Grafik perbandingan tersebut memperlihatkan bahwa seluruh metrik evaluasi pada pendekatan *multistage* secara konsisten berada di atas *singlestage*.

Performa numerik yang lebih tinggi pada pendekatan *multistage* menunjukkan bahwa pemisahan proses klasifikasi ke dalam beberapa tahap mampu membantu model dalam mempelajari pola data secara lebih terstruktur. Pada tahap pertama, model berperan menyederhanakan ruang klasifikasi dengan menentukan aspek, sehingga tahap selanjutnya hanya memproses data yang telah homogen secara tematik. Kondisi tersebut memungkinkan model *Random Forest* pada tahap lanjutan untuk memanfaatkan fitur teks secara lebih fokus dan relevan, yang

berdampak pada peningkatan nilai evaluasi akhir dibandingkan pendekatan *singlestage* yang harus menangani seluruh variasi kelas dalam satu ruang klasifikasi yang sama.

Selain itu, penggunaan fitur *Bag of Words* berdimensi tinggi pada jumlah data *Stage 2* yang relatif lebih terbatas tetap menjadi tantangan tersendiri bagi pendekatan *multistage*. Meskipun pembagian tahap membantu mengurangi kompleksitas kelas, keterbatasan jumlah data pada tahap lanjutan dapat membatasi potensi peningkatan performa secara signifikan. Sebaliknya, pendekatan *singlestage* memanfaatkan seluruh data dalam satu proses pembelajaran, sehingga memiliki konteks data yang lebih luas dan stabil, meskipun harus menghadapi kompleksitas klasifikasi yang lebih tinggi.

Hasil perbandingan ini menunjukkan bahwa pendekatan *multistage* mampu memberikan performa numerik yang lebih unggul dibandingkan *singlestage* pada penelitian ini. Namun, keunggulan tersebut diperoleh dengan konsekuensi kompleksitas arsitektur sistem yang lebih tinggi. Dengan demikian, penelitian ini menegaskan adanya *trade-off* antara performa numerik akhir dan kompleksitas sistem, di mana *multistage* unggul secara moderat dari sisi akurasi dan konsistensi metrik, sedangkan *singlestage* memiliki alur klasifikasi yang lebih sederhana dan efisien.

Adapun perbedaan hasil performa antara penelitian Setyadin et al. (2025) dan penelitian ini dipengaruhi oleh perbedaan karakteristik dataset serta kompleksitas pendekatan klasifikasi yang diterapkan. Penelitian Setyadin et al. melaporkan kinerja yang sangat tinggi dengan nilai akurasi sebesar 98,9%,

precision 99,0%, recall 100%, dan F1-score 99,5%, yang dicapai menggunakan dataset berukuran besar dan seimbang sebanyak 12.000 artikel berita yang bersumber dari situs resmi Kominfo, sehingga pola bahasa yang dipelajari model relatif konsisten dan mudah dipisahkan.

Sementara itu, penelitian ini menggunakan dataset dengan distribusi kelas yang tidak merata pada beberapa aspek serta menerapkan pendekatan *multistage classification*, yang secara alami memiliki tingkat kesulitan lebih tinggi karena melibatkan proses klasifikasi bertahap dan potensi propagasi kesalahan antar tahap. Selain itu, variasi gaya bahasa dan konteks pada teks artikel menyebabkan ruang fitur menjadi lebih heterogen, sehingga berdampak pada penurunan nilai performa absolut dibandingkan penelitian Setyadin et al. Meskipun demikian, pendekatan yang digunakan dalam penelitian ini lebih merepresentasikan kondisi nyata penyebaran hoaks, sehingga hasil yang diperoleh tetap memiliki relevansi dan kontribusi ilmiah yang kuat.

Dalam hal ini sebagaimana isu hoaks menjadi salah satu tantangan besar dalam interaksi digital, Islam telah memberikan pedoman yang jelas mengenai pentingnya kehati-hatian dalam menerima dan menyebarkan informasi. Prinsip ini ditegaskan dalam firman Allah SWT dalam Q.S. An-Nur ayat 15.

إِذْ تَلْقَوْنَاهُ بِالْسِتَّةِ كُمْ وَقَوْلُونَ بِأَفْوَاهِكُمْ مَا لَيْسَ لَكُمْ بِهِ عِلْمٌ وَتَحْسِبُونَهُ هَيْنَا وَهُوَ عِنْدَ اللَّهِ عَظِيمٌ

“(Ingatlah) di waktu kamu menerima berita bohong itu dari mulut ke mulut dan kamu katakan dengan mulutmu apa yang tidak kamu ketahui sedikit juga, dan kamu menganggapnya suatu yang ringan saja. Padahal dia pada sisi Allah adalah besar” (Q.S. An-Nur ayat 15).

Ayat ini mengingatkan bahwa menyebarkan informasi tanpa bukti merupakan tindakan serius dalam pandangan Allah Swt, meskipun pelakunya menganggapnya sebagai hal sepele. Larangan ini sejalan dengan fokus penelitian yang bertujuan mengenali dan mengklasifikasikan berita bohong agar tidak tersebar di masyarakat. Melalui algoritma Random Forest yang diuji pada *single stage* dan *multistage*, penelitian ini membantu mencegah penyebaran konten palsu dengan cara mengidentifikasi pola linguistik yang sering muncul pada berita hoaks. Dengan demikian, teknologi ini mendukung nilai-nilai Islam dalam menjaga kehati-hatian sebelum menyampaikan informasi.

Maraknya penyebaran hoaks di media sosial menunjukkan bahwa banyak pengguna terburu-buru menyebarkan informasi tanpa memastikan kebenarannya. Kondisi ini menimbulkan dampak serius, seperti kesalahpahaman publik, kerusakan reputasi, hingga terciptanya keresahan sosial. Penelitian ini sebagai respon ilmiah terhadap masalah tersebut, dengan membangun model *Random Forest* yang berfungsi membantu proses verifikasi otomatis terhadap berita atau klaim tertentu sehingga informasi yang beredar dapat dipilah berdasarkan tingkat kebenarannya. Setiap muslim diperintahkan untuk tidak langsung membagikan kabar yang ia dengar sebelum memastikan valid dan sumbernya. Prinsip ini ditegaskan secara jelas oleh Rasulullah SAW dalam sebuah hadis yang melarang penyebaran informasi yang belum terverifikasi.

كُنْتِ بِالْمُرْءَ كَذِبًا أَنْ يُحَدِّثَ بِكُلِّ مَا سَمِعَ

“Cukuplah seseorang itu dianggap pendusta bila ia menyampaikan semua yang ia dengar.” (HR. Muslim no. 5 (*Muqaddimah Shahih Muslim*)).

Hadis tersebut menegaskan bahwa seseorang dapat terjatuh ke dalam perilaku dusta hanya karena menceritakan setiap informasi yang ia dengar tanpa terlebih dahulu memastikan kebenarannya. Rasulullah SAW memberi peringatan bahwa sikap tergesa-gesa dalam menyampaikan berita dapat menyebabkan tersebarnya kabar palsu, menimbulkan fitnah, dan merusak keharmonisan sosial. Prinsip kehati-hatian ini menunjukkan bahwa Islam menempatkan verifikasi atau *tabayyun* sebagai bagian penting dalam menjaga kebenaran informasi yang beredar di masyarakat. Dalam konteks penelitian ini, pesan moral hadis tersebut memiliki relevansi yang kuat. Model *Random Forest* yang dikembangkan di penelitian ini pada dasarnya berfungsi sebagai alat bantu untuk melakukan proses verifikasi secara sistematis terhadap berita maupun klaim yang tersebar di media sosial. Dengan memberikan prediksi apakah suatu informasi termasuk hoaks atau bukan, model ini membantu pengguna bersikap lebih selektif sebelum menyebarkan suatu konten. Hal ini sejalan dengan nilai kehati-hatian yang diajarkan dalam Islam agar umat tidak mudah menyebarkan setiap informasi yang diterima tanpa dasar yang jelas. Dengan demikian, penggunaan teknologi dalam penelitian ini dapat dipandang sebagai upaya mendukung praktik *tabayyun* modern yang membantu masyarakat menghindari perilaku penyebaran informasi yang tidak valid sebagaimana diperingatkan dalam hadis tersebut.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan serangkaian uji coba dan analisis yang telah dilakukan terhadap sistem deteksi hoaks berbasis algoritma *Random Forest* menggunakan pendekatan *single stage classification* dan *multistage classification*, diperoleh beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil merancang dan mengimplementasikan sistem deteksi hoaks pada postingan artikel berbahasa Indonesia menggunakan algoritma *Random Forest* dengan dua pendekatan klasifikasi, yaitu *singlestage* dan *multistage*. Proses penelitian meliputi tahapan prapemrosesan teks, ekstraksi fitur menggunakan metode *Bag of Words*, pelatihan model, serta evaluasi performa menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*. Hasil pengujian menunjukkan bahwa pendekatan *singlestage* menghasilkan performa rata-rata dengan akurasi sebesar 45,36%, nilai *precision* 43,97%, *recall* 45,02%, dan *F1-score* 44,06%. Pendekatan ini mampu memberikan performa yang relatif konsisten karena seluruh data diproses dalam satu ruang klasifikasi tanpa pemisahan aspek. Sementara itu, pendekatan *multistage* menghasilkan performa akhir sistem yang sedikit lebih tinggi, dengan akurasi sebesar 46,76%, serta nilai *precision* 46,72%, *recall* 46,16%, dan *F1-score* 46,06%. Hasil tersebut menunjukkan bahwa pemisahan proses klasifikasi

berdasarkan aspek mampu membantu model dalam mempelajari pola hoaks secara lebih terarah.

2. Analisis terhadap pengaruh parameter *Random Forest* menunjukkan bahwa nilai *n\_estimator* dan *max\_depth* memberikan dampak yang berbeda pada masing-masing pendekatan klasifikasi. Pada pendekatan *singlestage*, peningkatan nilai *max\_depth* membantu model dalam menangkap kompleksitas data teks yang beragam karena seluruh kelas dipelajari secara bersamaan, sedangkan peningkatan jumlah pohon (*n\_estimator*) cenderung meningkatkan stabilitas prediksi model. Pada pendekatan *multistage*, performa terbaik diperoleh pada kombinasi jumlah pohon dan kedalaman pohon menengah, karena data pada tahap lanjutan telah dikelompokkan berdasarkan aspek sehingga memiliki ruang klasifikasi yang lebih sempit. Secara keseluruhan, pendekatan *singlestage* memiliki alur klasifikasi yang lebih sederhana dan mudah diterapkan, sedangkan pendekatan *multistage* memberikan struktur klasifikasi yang lebih terarah berbasis aspek. Oleh karena itu, pemilihan pendekatan klasifikasi perlu disesuaikan dengan tujuan sistem, karakteristik data, serta kebutuhan analisis yang diinginkan dalam penerapan deteksi hoaks.

## 5.2 Saran

Berdasarkan hasil penelitian ini, terdapat beberapa saran untuk penelitian selanjutnya:

1. Menggunakan dataset yang lebih besar dan lebih beragam untuk meningkatkan kualitas model. Pada penelitian ini, jumlah data hoaks dan non-

hoaks pada beberapa aspek masih terbatas, sehingga memengaruhi kestabilan performa model, terutama pada kelas non-hoaks. Penelitian selanjutnya disarankan untuk memperluas sumber dataset dengan menambahkan data dari berbagai platform resmi seperti *TurnBackHoax*, Kominfo, maupun arsip berita daring terpercaya. Selain itu, penambahan variasi topik hoaks juga diperlukan agar model mampu mempelajari pola bahasa yang lebih kompleks dan menghasilkan performa klasifikasi yang lebih kuat pada seluruh aspek

2. Selain itu, penelitian selanjutnya juga dapat mencoba menganalisis berbagai aspek lain selain pendidikan, politik, dan bencana alam, seperti hoaks kesehatan, isi keagamaan, sosial. Penambahan aspek tersebut memungkinkan model untuk mempelajari variasi pola penyebaran hoaks pada kategori yang lebih beragam, sehingga hasil klasifikasi dapat mencerminkan kondisi nyata penyebaran hoaks.
3. Menguji dengan mempertimbangkan penggunaan algoritma lain untuk membandingkan hasil performa klasifikasi. Meskipun Random Forest memperoleh hasil yang stabil pada pendekatan klasifikasi singlestage dan multistage. Model lain seperti BiLSTM, RoBERTa, atau IndoBERT berpotensi menawarkan kemampuan pemahaman konteks kalimat yang lebih kuat, terutama untuk mendeteksi hoaks dengan pola bahasa, atau metode lain seperti XGBoost, LightGBM, ataupun *Support Vector Machine* dapat diuji untuk melihat apakah model-model tersebut mampu memberikan akurasi, presisi, *recall*, atau F1-score yang lebih tinggi

## DAFTAR PUSTAKA

- Abedinia, A., & Seydi, V. (2024). Building semi-supervised decision trees with semi-cart algorithm. *International Journal of Machine Learning and Cybernetics*, 15(10), 4493–4510. <https://doi.org/10.1007/s13042-024-02161-z>
- Adiguna, V. B., Aqqad, M., Pt, J. S., & Dp, H. R. (2024). *Improving Performance Sentiment Analysis Movie Review Film using Random Forest with Feature Selection Information Gain*. 8(1). <https://doi.org/10.29099/ijair.v8i1.1.1227>
- Afuan, L. (2024). Enhanced Fall Detection using Optimized Random Forest Classifier on Wearable Sensor Data. *Journal of Applied Data Sciences*, 6(1), 213–224. <https://doi.org/10.47738/jads.v6i1.498>
- Agustinah, A., Nasution, S. M., Rahmat S., A. D., Andriani, A. D., & Teguh, D. F. (2024). Pengembangan Dan Implementasi Literasi Digital Untuk Menangkal Hoaks Kebencanaan Di Cianjur. *Jurnal AbdiMas Nusa Mandiri*, 6(2), 124–134. <https://doi.org/10.33480/abdimas.v6i2.5660>
- Algiriyyage, N., Sampath, R., Prasanna, R., & Doyle, E. E. H. (2021). *Identifying Disaster-related Tweets: A Large-Scale Detection Model Comparison*.
- Almuayqil, S. N., Humayun, M., Jhanjhi, N. Z., Almufareh, M. F., & Javed, D. (2022). Framework for Improved Sentiment Analysis via Random Minority Oversampling for User Tweet Review Classification. *Electronics*, 11(19), 3058. <https://doi.org/10.3390/electronics11193058>
- Amaly, N., & Armiah, A. (2021). Peran Kompetensi Literasi Digital Terhadap Konten Hoaks dalam Media Sosial. *Alhadharah: Jurnal Ilmu Dakwah*, 20(2), 43. <https://doi.org/10.18592/alhadharah.v20i2.6019>
- Arentania, M. N., Afriliani, C., Razzaq, A., & Nugraha, M. Y. (2025). Peran Literasi Digital Dalam Menangkal Hoaks Keagamaan Di Media Sosial Pada Remaja. *NAAFI: JURNAL ILMIAH MAHASISWA*, 1(3), 396–404. <https://doi.org/10.62387/naafijurnalilmiahmahasiswa.v2i2.172>
- Arintasari, S. A., Wibowo, W., & Dewi, M. R. (2023). *Klasifikasi Berita pada Laman Website TurnBackHoax.id Menggunakan Algoritma*. 1, 1–12.
- Aryanto, R. R. (2021). *Studi Komparasi Model Klasifikasi Berbasis Pembelajaran Mesin Untuk Sistem Rekomendasi Pemilihan Program Studi Sarjana [Universitas Islam Indonesia]*. <https://dspace.uii.ac.id/handle/123456789/37194>

- Barreñada, L., Dhiman, P., Timmerman, D., Boulesteix, A.-L., & Van Calster, B. (2024). Understanding overfitting in random forest for probability estimation: A visualization and simulation study. *Diagnostic and Prognostic Research*, 8(1), 14. <https://doi.org/10.1186/s41512-024-00177-1>
- Batoebara, M. U., & Hasugian, B. S. (2023). Isu Hoaks Meningkat Menjadi Potensi Kekacauan Informasi. *Device: Journal of Information System, Computer Science and Information Technology*, 4(2), 64–79. <https://doi.org/10.46576/device.v4i2.4044>
- Bulut, M., Bulut, A., Kaban, A., & Kirbas, A. (2023). *Keyword Survey and Thematic Focuses in Educational Research: A Review of 2023*.
- Carrasco, L., Urrutia, F., & Abeliuk, A. (2025). *Zero-Shot Decision Tree Construction via Large Language Models* (No. arXiv:2501.16247). arXiv. <https://doi.org/10.48550/arXiv.2501.16247>
- Chandra, M. I. A., & Yusuf, R. (2024). *Visualisasi Kata Kunci Pemberitaan Pemilu 2024 Menggunakan Spacy Dan Wordcloud*. 5.
- Darmawan, B., Laksito, A. D., Yudianto, M. R. A., & Sidauruk, A. (2023). *Analisis Perbandingan Ekstraksi Fitur Teks pada Sentimen Analisis Kenaikan Harga BBM*. 11.
- Desriansyah, M. D., Sari, I. U., & Zulfahmi, Z. (2025). Analisis Efektivitas Algoritma Machine Learning dalam Deteksi Hoaks: Pada Berita Digital Berbahasa Indonesia. *Jurnal Sistem Informasi Dan Informatika*, 3(2), 63–69. <https://doi.org/10.47233/jiska.v3i1.2024>
- DickiPrabowo, R., Widaningrum, I., & Karaman, J. (2025a). Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma K-Nearest Neighbor (knn) Dan Support Vector Machine (svm). *JIKO (Jurnal Informatika dan Komputer)*, 9(1), 93. <https://doi.org/10.26798/jiko.v9i1.1424>
- DickiPrabowo, R., Widaningrum, I., & Karaman, J. (2025b). Sistem Deteksi Berita Hoax Pemilu 2024 Indonesia Menggunakan Algoritma K-Nearest Neighbor (knn) Dan Support Vector Machine (svm). *JIKO (Jurnal Informatika dan Komputer)*, 9(1), 93–111. <https://doi.org/10.26798/jiko.v9i1.1424>
- Doddy Suryadi, Fajar Nugraha, & Misbach, T. M. (2024). Penyebaran Hoax Politik Pada Media Sosial. *Jurnal komunikan*, 1(1). <https://doi.org/10.30993/jurnalkomunikan.v1i1.290>
- Dulkiah, Moh., & Setia, P. (2020). *Pola Penyebaran Hoaks di Kalangan Mahasiswa Perguruan Tinggi Islam di Bandung Jawa Barat* | Dulkiah |

*Jurnal SMART (Studi Masyarakat, Religi, dan Tradisi).*  
<https://doi.org/10.18784/smart.v6i2.978>

Fadli, M., & Saputra, R. A. (2023). Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke. *Jurnal Teknik*, 12(2).  
<https://doi.org/10.31000/jt.v12i2.9099>

Fardhina, A., Siregar, R. M., Br Sibarani, M. R. W., Br Ginting, I. C., & Pratama, A. (2025). Sistem Deteksi Berita Hoaks berbasis Algoritma Natural Language Processing (NLP) menggunakan BERT. *Jurnal Manajemen Informatika, Sistem Informasi dan Teknologi Komputer (JUMISTIK)*, 4(1), 450–461. <https://doi.org/10.70247/jumistik.v4i1.156>

Ferdita Nugraha, A., Aziza, R. F. A., & Pristyanto, Y. (2022). Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing. *Jurnal Infomedia*, 7(1), 39. <https://doi.org/10.30811/jim.v7i1.2959>

GeeksforGeeks. (2025, July 3). *Random Forest Hyperparameter Tuning in Python*. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/random-forest-hyperparameter-tuning-in-python/>

Hadi, H. P., & Sukamto, T. S. (2020). Klasifikasi Jenis Laporan Masyarakat Dengan K-Nearest Neighbor Algorithm. *JOINS (Journal of Information System)*, 5(1), 77–85. <https://doi.org/10.33633/joins.v5i1.3355>

Hassan, F. M. (2021). *Multistage feature-assisted deep learning and its application in fine-grained fake news detection* [D\_ph, University of Birmingham]. <https://etheses.bham.ac.uk/id/eprint/11673/>

Hati, C. R. S., & Sulistiani, H. (2025). Implementation of CNN Algorithm for Indonesian Hoax News Detection on Online News Portals. *Journal of Applied Informatics and Computing*, 9(3), 765–774. <https://doi.org/10.30871/jaic.v9i3.9403>

Huda, D. N. I., Prianto, C., & Awangga, R. M. (2023). Analisis Sentimen Perbandingan Layanan Jasa Pengiriman Kurir Pada Ulasan Play Store Menggunakan Metode Decision Tree Dan Random Forest. *Jurnal Ilmiah Informatika*, 11(02), 150–158. <https://doi.org/10.33884/jif.v11i02.7952>

Indra, Hamdani, A. U., Setiawati, S., Mentari, Z. D., & Purnomo, M. H. (2024). Comparison of K-NN, SVM, and Random Forest Algorithm for Detecting Hoax on Indonesian Election 2024. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, 13(1), 166–179. <https://doi.org/10.23887/janapati.v13i1.76079>

- Irhamdhika, G. (2022). Mitigasi Hoax Di Era Disrupsi Melalui Literasi Digital. *J-IKA*, 9(1), 39–46. <https://doi.org/10.31294/kom.v9i1.12610>
- Israt Jahan, Md Nazmul Hasan, Syed Nurul Islam, Lima Akter, Md Khaledur Rahman Onik, Ashraful Islam, & Sm Mahamudul Hasan. (2024). Advanced machine learning techniques for fake news detection: A comprehensive analysis. *Magna Scientia Advanced Research and Reviews*, 12(2), 203–212. <https://doi.org/10.30574/msarr.2024.12.2.0198>
- Jamilah, S., & Missouri, R. (2024). Peran Etika Berbahasa dalam Menangkal Hoaks dan Ujaran Kebencian di Era Digital. *Pendiri: Jurnal Riset Pendidikan*, 1(2), 67–75. <https://doi.org/10.63866/pendiri.v1i2.62>
- Juliswara, V., & Muryanto, F. (2022). Model Penanggulangan Hoax Mengenai Berita Covid 19 untuk Pengembangan Literasi Digital Masyarakat di Indonesia. *JIIP - Jurnal Ilmiah Ilmu Pendidikan*, 5(7), 2587–2596. <https://doi.org/10.54371/jiip.v5i7.725>
- Khasanah, T. L., Annafajrin, V. R., Rizky, M. R. N., Arifudin, F. I., & Athallah, M. N. (2025). *Analisis Literasi Digital Masyarakat Indonesia Terkait Hoaks Libur Sekolah Saat Ramadhan Tahun 2025*. 2(6).
- Maulidya, E. N., Natalia, A., Yulianti, I., & Havez, M. (2023). Strategi Penanggulangan Informasi Hoax dan Terorisme di Media Sosial Oleh Unit Polisi Virtual Provinsi Lampung. *Jurnal Dakwah dan Komunikasi*, 8(1), 139. <https://doi.org/10.29240/jdk.v8i1.7822>
- Momole, G. M. (2022). *Perbandingan Naïve Bayes dan Random Forest Dalam Klasifikasi Bahasa Daerah* | *JATISI*. <https://jurnal.mdp.ac.id/index.php/jatisi/article/view/1857>
- Muharam, M. G. (2020). *Machine Learning for Hoax text classification*. <https://kaggle.com/code/muhammadghazimuharam/machine-learning-for-hoax-text-classification>
- Nayoga, B. P., Adipradana, R., Suryadi, R., & Suhartono, D. (2021). Hoax Analyzer for Indonesian News Using Deep Learning Models. *Procedia Computer Science*, 179, 704–712. <https://doi.org/10.1016/j.procs.2021.01.059>
- Nugroho, M. W. (2025). *Analisis Performa Algoritma Random Forest dalam Mengatasi Overfitting pada Model Prediksi*. 4. <https://doi.org/10.35870/jtik>
- Nuwairah, N., Munsyi, M., Amrullah, A., Ariani, A., & Aslamiah, R. (2024a). Klasifikasi Konten Pada Situs Web Dakwah Indonesia Menggunakan Text Mining Untuk Deteksi Paham Islam Radikal. *Technologia : Jurnal Ilmiah*, 15(4), 799. <https://doi.org/10.31602/tji.v15i4.16333>

- Nuwairah, N., Munsyi, M., Amrullah, A., Ariani, A., & Aslamiah, R. (2024b). Klasifikasi Konten Pada Situs Web Dakwah Indonesia Menggunakan Text Mining Untuk Deteksi Paham Islam Radikal. *Technologia : Jurnal Ilmiah*, 15(4), 799. <https://doi.org/10.31602/tji.v15i4.16333>
- Oktaviana, T., & Hasfi, N. (2020). Analisis Elemen Berita Hoaks Di Website. *Interaksi Online*, 8(3), 21–31.
- Prasetyo, S. Y., & Nabiilah, G. Z. (2023). Perbandingan Model Machine Learning pada Klasifikasi Tumor Otak Menggunakan Fitur Discrete Cosine Transform. *Jurnal Teknologi Terpadu*, 9(1), 29–34. <https://doi.org/10.54914/jtt.v9i1.605>
- Pratama, R., & Basuki, S. (2025a). Klasifikasi Hoax Vs Non-Hoax Pada Berita Bencana Alam Berbahasa Indonesia Menggunakan Word Embedding. *Jurnal Komputer, Informasi dan Teknologi*, 5(1), 13. <https://doi.org/10.53697/jkomitek.v5i1.2338>
- Pratama, R., & Basuki, S. (2025b). Klasifikasi Hoax Vs Non-Hoax Pada Berita Bencana Alam Berbahasa Indonesia Menggunakan Word Embedding. *Jurnal Komputer, Informasi dan Teknologi*, 5(1), 13. <https://doi.org/10.53697/jkomitek.v5i1.2338>
- Rahmadhany, A., Aldila Safitri, A., & Irwansyah, I. (2021). Fenomena Penyebaran Hoax dan Hate Speech pada Media Sosial. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 3(1), 30–43. <https://doi.org/10.47233/jtekstis.v3i1.182>
- Ridwan, R., Hermaliani, E. H., & Ernawati, M. (2024). Penerapan: Penerapan Metode SMOTE Untuk Mengatasi Imbalanced Data Pada Klasifikasi Ujaran Kebencian. *Computer Science (CO-SCIENCE)*, 4(1), 80–88. <https://doi.org/10.31294/coscience.v4i1.2990>
- Ropikoh, I. A., Abdulhakim, R., Enri, U., & Sulistiyowati, N. (2021a). Penerapan Algoritma Support Vector Machine (SVM) untuk Klasifikasi Berita Hoax Covid-19. *Journal of Applied Informatics and Computing*, 5(1), 64–73. <https://doi.org/10.30871/jaic.v5i1.3167>
- Ropikoh, I. A., Abdulhakim, R., Enri, U., & Sulistiyowati, N. (2021b). Penerapan Algoritma Support Vector Machine (SVM) untuk Klasifikasi Berita Hoax Covid-19. *Journal of Applied Informatics and Computing*, 5(1), 64–73. <https://doi.org/10.30871/jaic.v5i1.3167>
- Rosanti, C., Artanto, F. A., & Saputra, R. E. (2025). *Regresi Dengan Ekstrasi Fitur Neural Bag of Words Pada Analisis Sentimen Pengguna Aplikasi Bank Digital Syariah*. 10(3).

- Roshinta, T. A., Kumala, E., & Dinata, I. F. (2023). Sistem Deteksi Berita Hoax Berbahasa Indonesia Bidang Kesehatan. *remik*, 7(2), 1167–1173. <https://doi.org/10.33395/remik.v7i2.12369>
- Salsabila, A. A. (2023). Pentingnya Literasi di Era Digital dalam Menghadapi Hoaks di Media Sosial. *Inspirasi Dunia: Jurnal Riset Pendidikan dan Bahasa*, 3(1), 45–54. <https://doi.org/10.58192/insdun.v3i1.1775>
- Saptoyo, R. D. A., & Galih, B. (2024, January 3). *Kominfo Temukan 12.547 Konten Hoaks 5 Tahun Terakhir, Terbanyak soal Kesehatan dan Penipuan*. KOMPAS.com. <https://www.kompas.com/cekfakta/read/2024/01/03/171700782/kominfo-temukan-12.547-konten-hoaks-5-tahun-terakhir-terbanyak-soal>
- Sari, R. A. P., Kacung, S., & Santoso, B. (2025). *Analisis Sentimen Layanan Kesehatan Bpjs Menggunakan Metode Svm*. 7(2).
- Setyadin, R. D., Winasis, R. H., & Triyono, G. (2025). Fake News Detection using the Random Forest Algorithm. *Sistemasi: Jurnal Sistem Informasi*, 14(3), 1142–1153. <https://doi.org/10.32520/stmsi.v14i3.4995>
- Suci Amaliah, Nusrang, M., & Aswi, A. (2022). Penerapan Metode Random Forest Untuk Klasifikasi Varian Minuman Kopi di Kedai Kopi Konijiwa Bantaeng. *VARIANSI: Journal of Statistics and Its application on Teaching and Research*, 4(3), 121–127. <https://doi.org/10.35580/variansiunm31>
- Sumantri, D., Syarifuddin Syarifuddin, & Alif Bahtiar Pamulaan. (2024). Pengaruh Hoax Terhadap Pendidikan Karakter dalam Lingkungan Remaja. *Al Yazidiy Jurnal Sosial Humaniora dan Pendidikan*, 6(2), 129–138. <https://doi.org/10.55606/ay.v6i2.1130>
- Tambunan, T., Yohanna, M., & Silalahi, A. P. (2023). Penerapan Metode Random Forest Dalam Mendeteksi Berita Hoax. *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, 7(2), 301–306. <https://doi.org/10.46880/jmika.Vol7No2.pp301-306>
- Thelagathoti, R. K., Chandel, D. S., Tom, W. A., Jiang, C., Krzyzanowski, G., Olou, A., & Fernando, M. R. (2025). Machine Learning-Based Ensemble Feature Selection and Nested Cross-Validation for miRNA Biomarker Discovery in Usher Syndrome. *Bioengineering*, 12(5), 497. <https://doi.org/10.3390/bioengineering12050497>
- Triyono, L., Gernowo, R., Prayitno, P., Rahaman, M., & Yudantoro, T. R. (2023). Fake News Detection in Indonesian Popular News Portal Using Machine Learning For Visual Impairment. *JOIV: International Journal on Informatics Visualization*, 7(3), 726–732. <https://doi.org/10.30630/joiv.7.3.1243>

Wijiyanto, W., Pradana, A. I., Sopangi, S., & Atina, V. (2024). Teknik K-Fold Cross Validation untuk Mengevaluasi Kinerja Mahasiswa. *Jurnal Algoritma*, 21(1). <https://doi.org/10.33364/algoritma/v.21-1.1618>

Yoga Religia, Agung Nugroho, & Wahyu Hadikristanto. (2021). Analisis Perbandingan Algoritma Optimasi pada Random Forest untuk Klasifikasi Data Bank Marketing. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(1), 187–192. <https://doi.org/10.29207/resti.v5i1.2813>

Yopita, D. B. (2024). Analisis Penyebaran Hoax Di Media Sosial Dan Dampaknya Terhadap Masyarakat. *Jurnal Pendidikan, Bahasa dan Budaya*, 3(2), 252–258. <https://doi.org/10.55606/jpbb.v3i2.3201>

Zhang, Y., Luo, L., Ji, X., & Dai, Y. (2021). Improved Random Forest Algorithm Based on Decision Paths for Fault Diagnosis of Chemical Process with Incomplete Data. *Sensors*, 21(20), 6715. <https://doi.org/10.3390/s21206715>

# **LAMPIRAN**

## Lampiran I. Source Code Ekstraksi Fitur BoW

```
from sklearn.feature_extraction.text import CountVectorizer

# =====
#           FITUR & LABEL
# =====

X_text = df_aspek_bersih[["narasi_final"]].values
y = df_aspek_bersih[["Label_Aspek_Hoaks"]].values
vectorizer = CountVectorizer(
    analyzer="word",
    ngram_range=(1, 1),
    min_df=2
)
X_bow = vectorizer.fit_transform(X_text)

print("Jumlah data:", X_bow.shape[0])
print("Jumlah fitur (vocab):", X_bow.shape[1])
```

## Source Code Bangun Model Singlestage

### Split Data 80:20 Singlestage

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X_bow,
    y,
    test_size=0.20,
    random_state=42,
    stratify=y
)

print("Ukuran data latih :", X_train.shape)
print("Ukuran data uji   :", X_test.shape)
```

### Distribusi Data Singlestage

```
import pandas as pd

print("Distribusi label TRAIN:")
print(pd.Series(y_train).value_counts())

print("\nDistribusi label TEST:")
print(pd.Series(y_test).value_counts())

Distribusi label TRAIN:
Pendidikan_Hoaks      160
Politik_Hoaks         160
Politik_NonHoaks     135
Pendidikan_NonHoaks   131
Name: count, dtype: int64

Distribusi label TEST:
Politik_Hoaks         40
Pendidikan_Hoaks      40
Politik_NonHoaks      34
Pendidikan_NonHoaks    33
Name: count, dtype: int64
```

## Source Code Random Forest Singlestage salah satu *hyperparameter*

```

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

valid_classes = ["Pendidikan_Hoaks", "Pendidikan_NonHoaks", "Politik_Hoaks",
                 "Politik_NonHoaks"]

# =====
#           Inisialisasi model Random Forest (Single-Stage)
# =====
rf_single_4class = RandomForestClassifier(
    n_estimators=50,
    max_depth=10,
    max_features='sqrt',
    min_samples_split=10,
    min_samples_leaf=2,
    random_state=42,
    n_jobs=-1,
    class_weight="balanced"
)

# =====
#           Latih model
# =====
rf_single_4class.fit(X_train, y_train)

# Prediksi
y_pred_train = rf_single_4class.predict(X_train)
y_pred_test = rf_single_4class.predict(X_test)

# =====
#           Evaluasi Akurasi
# =====
acc_train = accuracy_score(y_train, y_pred_train)
acc_test = accuracy_score(y_test, y_pred_test)

print(f"\nAkurasi TRAIN Single-Stage: {acc_train*100:.2f}%")
print(f"Akurasi TEST  Single-Stage: {acc_test*100:.2f}%")

# =====
#           Confusion Matrix
# =====
cm = confusion_matrix(y_test, y_pred_test, labels=valid_classes)

cm_df = pd.DataFrame(cm, index=valid_classes, columns=valid_classes)

plt.figure(figsize=(8, 6))
sns.heatmap(cm_df, annot=True, fmt="d", cmap="Blues",
            linewidths=0.5, linecolor="black")
plt.title("Confusion Matrix RF (Single-Stage, 4 Kelas)\n(n_estimators=50, max_depth=10)")
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.tight_layout()
plt.show()

```

```

# =====
#           Classification Report
# =====
print("\n== LAPORAN KLASIFIKASI (Single-Stage) ==")
print(classification_report(
    y_test,
    y_pred_test,
    labels=valid_classes,
    target_names=valid_classes,
    digits=4
))

```

### Source Code Bangun Model Random Forest Multistage Stage 1

```

# =====
#           MULTISTAGE - STAGE 1 (MENENTUKAN ASPEK)
# =====
X_stage1 = X_bow

# Label aspek (Pendidikan / Politik)
y_stage1 = df_aspek_bersih["Aspek"].astype(str).values

#
valid_aspek = ["Pendidikan", "Politik"]
mask = pd.Series(y_stage1).isin(valid_aspek).values

X_stage1 = X_stage1[mask]
y_stage1 = y_stage1[mask]

print("Shape X_stage1:", X_stage1.shape)
print("Shape y_stage1:", y_stage1.shape)

print("\n== Distribusi Aspek (Sebelum Split) ==")
print(pd.Series(y_stage1).value_counts())

```

### Split Data 80:20 MultiStage Stage 1

```

X_train_s1, X_test_s1, y_train_s1, y_test_s1 = train_test_split(
    X_stage1,
    y_stage1,
    test_size=0.20,
    random_state=42,
    stratify=y_stage1
)

print("\nUkuran data latih Stage 1:", X_train_s1.shape)
print("Ukuran data uji Stage 1:", X_test_s1.shape)

print("\nDistribusi Aspek TRAIN Stage 1:")
print(pd.Series(y_train_s1).value_counts())
print("\nDistribusi Aspek TEST Stage 1:")
print(pd.Series(y_test_s1).value_counts())

Ukuran data latih Stage 1: (586, 1864)
Ukuran data uji Stage 1: (147, 1864)

Distribusi Aspek TRAIN Stage 1:
Politik      295
Pendidikan   291
Name: count, dtype: int64

Distribusi Aspek TEST Stage 1:
Politik      74
Pendidikan   73
Name: count, dtype: int64

```

*Source Code Random Forest Multistage Stage 1 salah satu hyperparameter*

```
rf_stage1 = RandomForestClassifier(  
    n_estimators=50,  
    max_depth=10,  
    max_features="sqrt",  
    min_samples_split=10,  
    min_samples_leaf=2,  
    random_state=42,  
    n_jobs=-1,  
    class_weight="balanced"  
)  
  
# 6) Training  
rf_stage1.fit(X_train_s1, y_train_s1)  
  
# Prediksi  
pred_train_s1 = rf_stage1.predict(X_train_s1)  
pred_test_s1 = rf_stage1.predict(X_test_s1)  
  
# Akurasi  
acc_train_s1 = accuracy_score(y_train_s1, pred_train_s1)  
acc_test_s1 = accuracy_score(y_test_s1, pred_test_s1)  
  
print(f"\nAkurasi TRAIN Stage 1 (Aspek): {acc_train_s1*100:.2f}%")  
print(f"Akurasi TEST Stage 1 (Aspek): {acc_test_s1*100:.2f}%")  
  
# Confusion Matrix  
cm_s1 = confusion_matrix(y_test_s1, pred_test_s1, labels=valid_aspek)  
cm_s1_df = pd.DataFrame(cm_s1, index=valid_aspek, columns=valid_aspek)  
  
plt.figure(figsize=(6, 5))  
sns.heatmap(cm_s1_df, annot=True, fmt="d", cmap="Blues",  
            linewidths=0.5, linecolor="black")  
plt.title("Confusion Matrix Stage 1 (Klasifikasi Aspek)\nRandom Forest")  
plt.xlabel("Prediksi")  
plt.ylabel("Aktual")  
plt.tight_layout()  
plt.show()  
  
# Classification Report  
print("\n==== LAPORAN KLASIFIKASI STAGE 1 (ASPEK) ===")  
print(classification_report(  
    y_test_s1,  
    pred_test_s1,  
    labels=valid_aspek,  
    target_names=valid_aspek,  
    digits=4  
)
```

*Source Code Bangun Model Random Forest Multistage Stage 2 Aspek Pendidikan*

```
# =====  
# STAGE 2 - DATA PENDIDIKAN  
# =====  
  
df_aspek_bersih["Aspek"] = df_aspek_bersih["Aspek"].astype(str)  
  
valid_aspek = ["Pendidikan", "Politik"]  
mask_valid = df_aspek_bersih["Aspek"].isin(valid_aspek)  
  
df_aspek_bersih = df_aspek_bersih[mask_valid].reset_index(drop=True)
```

```

# 3) Filter khusus Pendidikan
mask_pendidikan = (df_aspek_bersih["Aspek"] == "Pendidikan").values

X_pendidikan = X_bow[mask_pendidikan]

y_pendidikan = df_aspek_bersih.loc[mask_pendidikan, "label"].astype(int).values

print("== STAGE 2 (PENDIDIKAN) ==")
print("Jumlah data Pendidikan :", X_pendidikan.shape[0])
print("Jumlah fitur BoW      :", X_pendidikan.shape[1])

print("\nDistribusi label (0=NonHoaks, 1=Hoaks):")
print(pd.Series(y_pendidikan).value_counts().sort_index())

# =====
# Filter Data khusus Pendidikan
# =====
mask_pendidikan = (df_aspek_bersih["Aspek"] == "Pendidikan").values

X_pendidikan = X_bow[mask_pendidikan]

y_pendidikan = df_aspek_bersih.loc[mask_pendidikan, "label"].astype(int).values

print("== STAGE 2 (PENDIDIKAN) ==")
print("Jumlah data Pendidikan :", X_pendidikan.shape[0])
print("Jumlah fitur BoW      :", X_pendidikan.shape[1])

print("\nDistribusi label (0=NonHoaks, 1=Hoaks):")
print(pd.Series(y_pendidikan).value_counts().sort_index())

== STAGE 2 (PENDIDIKAN) ==
Jumlah data Pendidikan : 364
Jumlah fitur BoW      : 1864

Distribusi label (0=NonHoaks, 1=Hoaks):
0    164
1    200
Name: count, dtype: int64

```

## Split Data 80:20

```

from sklearn.model_selection import train_test_split

X_train_p, X_test_p, y_train_p, y_test_p = train_test_split(
    X_pendidikan,
    y_pendidikan,
    test_size=0.20,
    random_state=42,
    stratify=y_pendidikan
)

print("\nUkuran data latih Pendidikan:", X_train_p.shape)
print("Ukuran data uji  Pendidikan:", X_test_p.shape)

print("\nDistribusi TRAIN:", pd.Series(y_train_p).value_counts().sort_index())
print("Distribusi TEST : ", pd.Series(y_test_p).value_counts().sort_index())

```

*Source Code Random Forest Multistage Stage 2 aspek pendidikan salah satu hyperparameter*

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

valid_binary = [0, 1] # 0=NonHoaks, 1=Hoaks

rf_stage2_pendidikan = RandomForestClassifier(
    n_estimators=50,
    max_depth=10,
    max_features="sqrt",
    min_samples_split=10,
    min_samples_leaf=2,
    random_state=42,
    n_jobs=-1,
    class_weight="balanced"
)
# Train
rf_stage2_pendidikan.fit(X_train_p, y_train_p)

# Prediksi
pred_train_p = rf_stage2_pendidikan.predict(X_train_p)
pred_test_p = rf_stage2_pendidikan.predict(X_test_p)

# Akurasi
acc_train_p = accuracy_score(y_train_p, pred_train_p)
acc_test_p = accuracy_score(y_test_p, pred_test_p)

print(f"\nAkurasi TRAIN Stage 2 Pendidikan: {acc_train_p*100:.2f}%")
print(f"Akurasi TEST Stage 2 Pendidikan: {acc_test_p*100:.2f}%")

# Confusion Matrix
cm_p = confusion_matrix(y_test_p, pred_test_p, labels=valid_binary)
cm_p_df = pd.DataFrame(cm_p, index=["NonHoaks(0)", "Hoaks(1)"], columns=["NonHoaks(0)", "Hoaks(1)"])

plt.figure(figsize=(6, 5))
sns.heatmap(cm_p_df, annot=True, fmt="d", cmap="Blues", linewidths=0.5, linecolor="black")
plt.title("Confusion Matrix Stage 2 (Pendidikan)\nRandom Forest")
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.tight_layout()
plt.show()

# Classification Report
print("\n== LAPORAN KLASIFIKASI STAGE 2 (PENDIDIKAN) ==")
print(classification_report(
    y_test_p,
    pred_test_p,
    labels=valid_binary,
    target_names=["NonHoaks", "Hoaks"],
    digits=4
)))
```

## Source Code Bangun Model Random Forest Multistage Stage 2 Aspek Politik

```
# =====
# STAGE 2 - DATA POLITIK
# =====

mask_politik = (df_aspek_bersih["Aspek"].astype(str) == "Politik").values

X_politik = X_bow[mask_politik]
y_politik = df_aspek_bersih.loc[mask_politik, "label"].astype(int).values

print("Jumlah data Politik:", X_politik.shape[0])
print("Distribusi label Politik (0=NonHoaks, 1=Hoaks):")
print(pd.Series(y_politik).value_counts().sort_index())

Jumlah data Politik: 369
Distribusi label Politik (0=NonHoaks, 1=Hoaks):
0    169
1    200
Name: count, dtype: int64
```

Split Data 80:20

```
from sklearn.model_selection import train_test_split

X_train_pol, X_test_pol, y_train_pol, y_test_pol = train_test_split(
    X_politik,
    y_politik,
    test_size=0.20,
    random_state=42,
    stratify=y_politik
)

print("\nUkuran data latih Politik:", X_train_pol.shape)
print("Ukuran data uji Politik : ", X_test_pol.shape)

print("\nDistribusi TRAIN Politik:")
print(pd.Series(y_train_pol).value_counts().sort_index())

print("\nDistribusi TEST Politik:")
print(pd.Series(y_test_pol).value_counts().sort_index())
```

Source Code Random Forest Multistage Stage 2 aspek politik salah satu hyperparameter

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns

valid_binary = [0, 1] # 0=NonHoaks, 1=Hoaks

rf_stage2_politik = RandomForestClassifier(
    n_estimators=50,
    max_depth=10,
    max_features="sqrt",
    min_samples_split=10,
    min_samples_leaf=2,
    random_state=42,
    n_jobs=-1,
    class_weight="balanced"
)
```

```

# Training
rf_stage2_politik.fit(X_train_pol, y_train_pol)

# Prediksi
pred_train_pol = rf_stage2_politik.predict(X_train_pol)
pred_test_pol = rf_stage2_politik.predict(X_test_pol)

# Akurasi
acc_train_pol = accuracy_score(y_train_pol, pred_train_pol)
acc_test_pol = accuracy_score(y_test_pol, pred_test_pol)

print(f"\nAkurasi TRAIN Stage 2 Politik: {acc_train_pol*100:.2f}%")
print(f"Akurasi TEST Stage 2 Politik: {acc_test_pol*100:.2f}%")

cm_pol = confusion_matrix(y_test_pol, pred_test_pol, labels=valid_binary)
cm_pol_df = pd.DataFrame(
    cm_pol,
    index=["NonHoaks(0)", "Hoaks(1)"],
    columns=["NonHoaks(0)", "Hoaks(1)"]
)

plt.figure(figsize=(6, 5))
sns.heatmap(cm_pol_df, annot=True, fmt="d", cmap="Blues",
            linewidths=0.5, linecolor="black")
plt.title("Confusion Matrix Stage 2 (Politik)\nRandom Forest")
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.tight_layout()
plt.show()

print("\n==== LAPORAN KLASIFIKASI STAGE 2 (POLITIK) ===")
print(classification_report(
    y_test_pol,
    pred_test_pol,
    labels=valid_binary,
    target_names=["NonHoaks", "Hoaks"],
    digits=4
))

```

## Lampiran II. Graphical User Interface

### Deteksi Hoaks Artikel Bahasa Indonesia

Pipeline: aspek (keyword) → preprocessing → Bag of Words → Random Forest (6 kelas).

Prediksi tunggal Prediksi batch (CSV) Confusion Matrix Catatan

**Judul**

Instruksi Gubernur Jateng tentang penilangan bagi yg tidak bermasker di muka umum Rp.150.000 menggunakan e-tilang Via apps PIKOBAR

**Narasi**

9th.Seluruh Anggota Grup Sesuai Instruksi Gubernur Jawa Tengah Hasil Rapat Tim Gugus Tugas Covid 19 Jateng sbb:

**Prediksi**

**Hasil**

Aspek terdeteksi: Pendidikan  
Prediksi: Pendidikan\_Hoaks  
Status: Hoaks  
Confidence: 32.02%

**Pendidikan\_Hoaks**

Pendidikan_Hoaks	32%
Politik_Hoaks	18%
Bencana_Hoaks	15%
Pendidikan_NonHoaks	14%

**Pendidikan\_Hoaks**

Pendidikan_Hoaks	32%
Politik_Hoaks	18%
Bencana_Hoaks	15%
Pendidikan_NonHoaks	14%
Politik_NonHoaks	12%
Bencana_NonHoaks	9%

Tabel probabilitas

Kelas_ID	Nama_Kelas	Probabilitas
1	Pendidikan_Hoaks	0.3202432113541138
3	Politik_Hoaks	0.17656312299683172
5	Bencana_Hoaks	0.1516271787911094
2	Pendidikan_NonHoaks	0.144554093264526
4	Politik_NonHoaks	0.12126369920232959
6	Bencana_NonHoaks	0.08574869439108948

☰ Contoh input

Judul	Narasi
Beredar kabar beasiswa kuliah gratis untuk semua warga	Beredar informasi bahwa semua warga akan mendapatkan beasiswa...
Isu pemilu diundur	Pesan berantai menyebutkan pemilu akan diundur karena alasan...

## Deteksi Hoaks Artikel Bahasa Indonesia

Pipeline: aspek (keyword) → preprocessing → Bag of Words → Random Forest (6 kelas).

Prediksi tunggal   Prediksi batch (CSV)   Confusion Matrix   Catatan

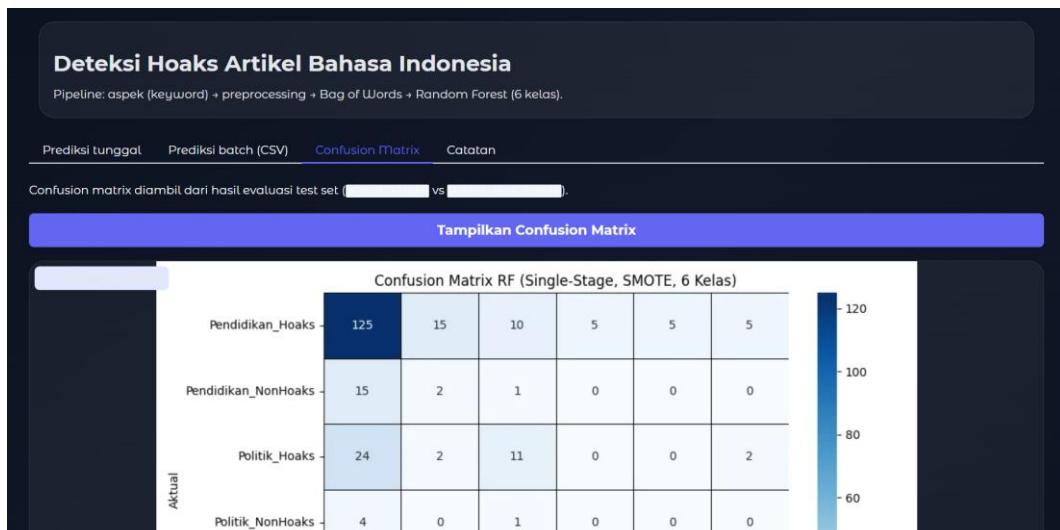
Format CSV: minimal ada kolom **Kolom** opsiional.

Drop File Here  
- OR -  
Click to Upload

Proses batch

Hasil batch

1	2	3
---	---	---



## Deteksi Hoaks Artikel Bahasa Indonesia

Pipeline: aspek (keyword) → preprocessing → Bag of Words → Random Forest (6 kelas).

Prediksi tunggal   Prediksi batch (CSV)   Confusion Matrix   Catatan

Catatan penting:

- Model mengikuti langkah preprocessing yang sama seperti kode training.
- Jika narasi terlalu pendek, setelah stopword removal + stemming bisa menjadi kosong, sehingga prediksi gagal.
- Penentuan aspek menggunakan keyword, sehingga bisa bias jika teks mengandung kata kunci lintas topik.

### Lampiran III. Perhitungan Manual Random Forest

Contoh hasil data kata unik dari ekstraksi fitur BOW

Doc	Instruksi	Gubernur	Masker	Resmi	Vaksin	Covid	Label
Doc 1	1	1	1	1	0	0	1 (hoaks)
Doc 2	0	1	0	1	1	1	0 (non-hoaks)
Doc 3	0	0	0	1	1	1	0 (non-hoaks)

Tabel 3. 14 contoh *bootstrapping*

Pohon	Hasil <i>Bootstrapping</i>
Pohon 1	[Doc 1, Doc 1, Doc 2]
Pohon 2	[Doc 2, Doc 3, Doc 3]
Pohon 3	[Doc 1, Doc 2, Doc 3]

#### Pohon 1

Bootstrap : [Doc 1, Doc 1, Doc 2]

Label : [1, 1, 0]

#### Hitung Proporsi Kelas :

$$P_1 = \frac{2}{3} = 0,667, P_0 = \frac{1}{3} = 0,333$$

Hitung Gini Index dengan menggunakan rumus pada persamaan 3.3

$$\begin{aligned} Gini &= 1 - (0,667^2 + 0,333^2) \\ &= 1 - (0,444 + 0,110) \\ &= 1 - 0,554 \\ &= 0,446 \text{ (tidak murni)} \end{aligned}$$

Setelah dilakukan bootstrap sampling pada pohon 1, diperoleh Gini Index 0,446 yang menunjukkan ketidakmurnian data di node tersebut. Untuk mengurangi *impurity*, dilakukan uji split menggunakan dua fitur yaitu instruksi dan resmi.

#### Uji split tiap fitur pada pohon 1 (menggunakan fitur instruksi)

Doc 1 = instruksi = 1 (muncul 2 kali)

Doc 2 = instruksi = 0 (muncul 1 kali)

- **Right**

(instruksi = 1)  $\rightarrow$  (Doc 1, Doc 2)  $\rightarrow$  label [1,1]

Estimasi kelas di dalam node :

$$P_1 = \frac{2}{2} = 1, P_0 = \frac{0}{2} = 0$$

#### Hitung *Gini Index Node Kanan*

$$\begin{aligned} Gini_{Right} &= 1 - (1^2 + 0^2) \\ &= 1 - (1 + 0) \\ &= 1 - 0 \\ &= 0 \text{ (murni)} \end{aligned}$$

- **Left**

(instruksi = 0)  $\rightarrow$  (Doc 2)  $\rightarrow$  Label [0]

Estimasi probabilitas kelas di dalam node :

$$P_0 = \frac{1}{1} = 1, P_1 = \frac{0}{1} = 0$$

#### Hitung *Gini Index Node Kiri*

$$\begin{aligned} Gini_{Left} &= 1 - (1^2 + 0^2) \\ &= 1 - (1 + 0) \\ &= 1 - 0 \\ &= 0 \text{ (murni)} \end{aligned}$$

### Gini Split Fitur Instruksi Pohon 1

Hitung Gini Split dengan menggunakan rumus 3.6. Pada penelitian ini pemisahan dilakukan secara biner (binary split), sehingga digunakan rumus pada persamaan 3.7.

Maka,

$$Gini_{Split} = \frac{2}{3} \cdot 0 + \frac{1}{3} \cdot 0 = 0 \text{ (murni)}$$

#### Uji split tiap fitur pada pohon 1 (menggunakan fitur resmi)

Contoh menggunakan fitur “resmi”

Doc 1 = resmi = 0  $\rightarrow$  muncul dua kali

Doc 2 = resmi = 1  $\rightarrow$  muncul dua kali

- **Right**

(resmi = 1)  $\rightarrow$  (Doc 1, Doc 2)  $\rightarrow$  label [1,1]

Estimasi kelas di dalam node :

$$P_1 = \frac{2}{2} = 1, P_0 = \frac{0}{2} = 0$$

### Hitung *Gini Index Node Kanan*

$$\begin{aligned}Gini_{Right} &= 1 - (1^2 + 0^2) \\&= 1 - (1 + 0) \\&= 1 - 0 \\&= 0 \text{ (murni)}\end{aligned}$$

#### - *Left*

(resmi = 0)  $\rightarrow$  (Doc 2)  $\rightarrow$  label [0]

$$P_0 = \frac{1}{1} = 1, P_1 = \frac{0}{1} = 0$$

### Hitung *Gini Index Node Kiri*

$$\begin{aligned}Gini_{Left} &= 1 - (1^2 + 0^2) \\&= 1 - (1 + 0) \\&= 1 - 0 \\&= 0 \text{ (murni)}\end{aligned}$$

Gini split fitur resmi :

$$Gini_{Split} = \frac{2}{3} \cdot 0 + \frac{1}{3} \cdot 0 = 0 \text{ (murni)}$$

## Pohon 2

Bootstrap : [Doc 2, Doc 2, Doc 3]

Label : [0, 0, 0]

### Hitung Proporsi Kelas :

$$P_0 = \frac{3}{3} = 1, P_1 = \frac{0}{3} = 0$$

### *Gini Index Pohon 2*

$$\begin{aligned}Gini &= 1 - (1^2 + 0^2) \\&= 1 - (1 + 0) \\&= 1 - 1 \\&= 0 \text{ (murni)}\end{aligned}$$

## Pohon 3

Bootstrap : [Doc 1, Doc 2, Doc 3]

Label : [1, 0, 0]

### Hitung Proporsi Kelas :

$$P_1 = \frac{1}{3} = 0,333, P_0 = \frac{2}{3} = 0,667 \text{ (belum murni)}$$

### ***Gini Index***

$$\begin{aligned} Gini &= 1 - (0,333^2 + 0,667^2) \\ &= 1 - (0,110 + 0,444) \\ &= 1 - 0,554 \\ &= 0,446 \text{ (belum murni)} \end{aligned}$$

### Uji split tiap fitur pada pohon 3 (menggunakan fitur instruksi)

Doc 1 = instruksi = 1

Doc 2 = instruksi = 0

Doc 3 = instruksi = 0

#### - ***Right***

(instruksi = 1)  $\rightarrow$  (Doc 1)  $\rightarrow$  label [1]

Estimasi probabilitas kelas di dalam node :

$$P_1 = \frac{1}{1} = 1, P_0 = \frac{0}{1} = 0$$

#### Hitung ***Gini Index Node Kanan***

$$\begin{aligned} Gini_{Right} &= 1 - (1^2 + 0^2) \\ &= 1 - (1 + 0) \\ &= 1 - 0 \\ &= 0 \text{ (murni)} \end{aligned}$$

#### - ***Left***

(instruksi = 0)  $\rightarrow$  (Doc 2, Doc 3)  $\rightarrow$  label [0,0]

Estimasi probabilitas kelas di dalam node :

$$P_0 = \frac{2}{2} = 1, P_1 = \frac{0}{2} = 0$$

$$\begin{aligned} Gini_{Left} &= 1 - (1^2 + 0^2) \\ &= 1 - (1 + 0) \\ &= 1 - 0 \\ &= 0 \text{ (murni)} \end{aligned}$$

### ***Gini Split Instruksi Pohon 3***

$$Gini_{Split} = \frac{2}{3} \cdot 0 + \frac{1}{3} \cdot 0 = 0 \text{ (murni)}$$

#### **Uji split fitur resmi pada pohon 3**

Doc 1 = resmi = 1

Doc 2 = resmi = 0

Doc 3 = resmi = 0

##### **- *Right***

(resmi = 1) → (Doc 2, Doc 3) → label [0,0]

Estimasi probabilitas kelas di dalam node :

$$P_1 = \frac{1}{2} = 1, P_0 = \frac{0}{2} = 0$$

$$Gini_{Right} = 1 - (1^2 + 0^2)$$

$$= 1 - (1 + 0)$$

$$= 1 - 0$$

$$= 0 \text{ (murni)}$$

##### **- *Left***

(resmi = 0) → (Doc 1) → label [1]

Estimasi probabilitas kelas di dalam node :

$$P_0 = \frac{2}{2} = 1, P_1 = \frac{0}{2} = 0$$

$$Gini_{Left} = 1 - (1^2 + 0^2)$$

$$= 1 - (1 + 0)$$

$$= 1 - 0$$

$$= 0 \text{ (murni)}$$

### ***Gini Split resmi Pohon 3***

$$Gini_{Split} = \frac{2}{3} \cdot 0 + \frac{1}{3} \cdot 0 = 0 \text{ (murni)}$$

Setelah seluruh pohon keputusan pada algoritma *Random Forest* dibentuk melalui proses *bootstrap sampling* dan pemilihan fitur terbaik berdasarkan nilai *Gini Index*, setiap pohon menghasilkan satu prediksi kelas terhadap data uji. Prediksi akhir *Random Forest* kemudian ditentukan menggunakan mekanisme *majority voting*, yaitu

dengan memilih kelas yang paling sering muncul (modus) dari seluruh prediksi pohon. Pada penelitian ini digunakan contoh tiga pohon keputusan, sehingga untuk setiap dokumen diperoleh tiga hasil prediksi. Kelas yang memiliki jumlah kemunculan terbanyak di antara ketiga prediksi tersebut ditetapkan sebagai prediksi akhir model. Pendekatan ini bertujuan untuk meningkatkan stabilitas dan akurasi klasifikasi dengan mengurangi pengaruh kesalahan prediksi dari satu pohon keputusan tertentu. Berikut adalah tabel hasil majority voting.

Data	Prediksi Pohon 1	Prediksi Pohon 2	Prediksi Pohon 3	Modus (Voting)	Prediksi Akhir
Doc 1	1	0	1	1 (muncul 2x)	1
Doc 2	0	0	0	0 (muncul 3x)	0
Doc 3	0	0	0	0 (muncul 3x)	0

Berdasarkan hasil *majority voting* pada Tabel tersebut, dapat dilihat bahwa Dokumen 1 diprediksi sebagai hoaks karena dua dari tiga pohon keputusan memberikan prediksi kelas 1. Sementara itu, Dokumen 2 dan Dokumen 3 diprediksi sebagai non-hoaks karena seluruh pohon keputusan secara konsisten menghasilkan prediksi kelas 0.