

**OPTIMASI *HYPERPARAMETER RANDOM FOREST* DENGAN
GRID SEARCH UNTUK KLASIFIKASI SENTIMEN PUBLIK
PADA ULASAN PRODUK IPHONE**

SKRIPSI

Oleh :
YUDISTIRA RAMADANI
NIM. 210605110123



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

**OPTIMASI *HYPERPARAMETER RANDOM FOREST* DENGAN
GRID SEARCH UNTUK KLASIFIKASI SENTIMEN PUBLIK
PADA ULASAN PRODUK IPHONE**

SKRIPSI

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

Oleh :
YUDISTIRA RAMADANI
NIM. 210605110123

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

HALAMAN PERSETUJUAN

**OPTIMASI *HYPERPARAMETER RANDOM FOREST* DENGAN
GRID SEARCH UNTUK KLASIFIKASI SENTIMEN PUBLIK
PADA ULASAN PRODUK IPHONE**

SKRIPSI

Oleh :
YUDISTIRA RAMADANI
NIM. 210605110123

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 05 Juni 2025

Pembimbing I,



Tri Mukti Lestari, M.Kom
NIP. 19911108 202012 2 005

Pembimbing II,



Dr. M. Amin Hariyadi, M.T
NIP. 19670018 200501 1 001

Mengetahui,

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




Dr. Irfan Achrul Kurniawan M.MT., IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

**OPTIMASI *HYPERPARAMETER RANDOM FOREST* DENGAN
GRID SEARCH UNTUK KLASIFIKASI SENTIMEN PUBLIK
PADA ULASAN PRODUK IPHONE**

SKRIPSI

Oleh :

YUDISTIRA RAMADANI
NIM. 210605110123

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 20 Juni 2025

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. Zainal Abidin, M.Kom</u> NIP. 19760613 200501 1 004	()
Anggota Penguji I	: <u>Fatchurrochman, M.Kom</u> NIP. 19700731 200501 1 002	()
Anggota Penguji II	: <u>Tri Mukti Lestari, M.Kom</u> NIP. 19911108 202012 2 005	()
Anggota Penguji III	: <u>Dr. M. Amin Hariyadi, M.T</u> NIP. 19670018 200501 1 001	()

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




Dr. Do Fachrul Kurniawan M.MT., IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Yudistira Ramadani
NIM : 210605110123
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Optimasi *Hyperparameter Random forest* Dengan
Grid Search Untuk Klasifikasi Sentimen Publik
Pada Ulasan Produk Iphone

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, 05 Juni 2025

Yang membuat pernyataan,



Yudistira Ramadani
NIM.210605110123

MOTTO

It wont be easy, just try it ya

HALAMAN PERSEMBAHAN

Segala puji syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala, karena atas rahmat dan petunjuk-Nya, skripsi ini dapat diselesaikan dengan baik dan lancar. Shalawat serta salam senantiasa tercurah kepada Rasulullah Shallallahu 'alaihi wasallam, yang telah membimbing umat dari zaman jahiliyah menuju cahaya Islam. Skripsi ini tidak akan terselesaikan tanpa dukungan dan kontribusi dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih, penulis mempersembahkan karya ini kepada semua pihak yang telah berperan dalam proses penelitian dan penyusunan skripsi ini :

1. Keluarga tercinta, Bapak Yoga Pramito dan Ibu Mas Uda sebagai orang tua yang selalu memberikan kasih sayang, dukungan, dan doa tanpa henti kepada penulis, serta Yasmin Talita Sakhi, adik tercinta yang menjadi sumber semangat dalam setiap langkah. Mereka senantiasa memberikan motivasi dan dukungan penuh hingga penulis dapat menyelesaikan skripsi ini dengan baik dan lancar.
2. Ibu Tri Mukti Lestari, M.Kom selaku dosen pembimbing I serta Dr. M. Amin Hariyadi, M.T selaku dosen pembimbing II, yang telah memberikan ilmu, arahan, dan bimbingan kepada penulis selama proses penelitian ini. Dengan penuh kesabaran, mereka senantiasa memberikan motivasi, saran, serta dukungan, sehingga penulis dapat menyelesaikan skripsi ini dengan baik dan lancar.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirabbilalamin, segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu wa Ta'ala atas rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "*Optimasi Hyperparameter Random forest dengan Grid Search untuk Klasifikasi Sentimen Publik pada Ulasan Produk iPhone*" dengan baik dan lancar. Skripsi ini diajukan sebagai salah satu syarat kelulusan dalam memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulisan skripsi ini tentu tidak terlepas dari dukungan, bantuan, serta motivasi dari berbagai pihak, baik secara langsung maupun tidak langsung. Oleh karena itu, dengan penuh rasa hormat dan apresiasi, penulis menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah berkontribusi dalam proses penelitian dan penyusunan skripsi ini :

1. Prof. Dr. H. M. Zainuddin, MA., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Dr. Sri Hariani, M.Si., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Ir. Fachrul Kurniawan, M.MT, IPU., selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

4. Tri Mukti Lestari, M.Kom., selaku pembimbing I, yang telah dengan penuh dedikasi meluangkan waktu untuk memberikan bimbingan, arahan, dan motivasi yang berharga selama proses penyusunan skripsi ini hingga dapat diselesaikan dengan baik.
5. Dr. M. Amin Hariyadi, M.T., selaku pembimbing II, yang dengan tulus telah memberikan arahan, dan dengan cermat meluruskan kesalahan skripsi ini.
6. Dr. Zainal Abidin, M.Kom., selaku Ketua Penguji yang telah memberikan banyak saran dan arahan dari seminar proposal hingga sidang skripsi penulis.
7. Fatchurrohman, M.Kom., selaku Anggota Penguji 1 yang telah memberikan saran dan masukan berharga sepanjang proses, mulai dari seminar proposal hingga sidang skripsi.
8. Nia Farichah, S.Si, selaku staf administrasi Program Studi, yang dengan penuh kesabaran selalu membantu penulis dalam mengurus berbagai keperluan akademik.
9. Segenap dosen, Laboran, dan jajaran staff Program Studi Teknik Informatika yang telah berbagi ilmu dan turut berkontribusi, baik secara langsung maupun tidak langsung, dalam mendukung proses penyusunan skripsi ini.
10. Keluarga tercinta, Bapak Yoga Pramito dan Ibu Mas Uda sebagai orang tua yang selalu memberikan kasih sayang, dukungan, dan doa tanpa henti kepada penulis, serta Yasmin Talita Sakhi, adik tercinta yang menjadi sumber semangat dalam setiap langkah. Mereka senantiasa memberikan

motivasi dan dukungan penuh hingga penulis dapat menyelesaikan skripsi ini dengan baik dan lancar.

11. Kepada wanita teristimewa, Alfina Difa Salsabilah, yang telah menjadi pendukung terbaik dalam perjalanan panjang ini. Terima kasih atas setiap doa, kesabaran, serta kehadiran yang selalu mendengarkan keluh kesah, memberikan semangat, dan menguatkan penulis, bahkan di saat-saat paling sulit sekalipun. Dukungan dan kebersamaan yang diberikan telah menjadi salah satu sumber kekuatan utama bagi penulis dalam menyelesaikan perjalanan ini.
12. Teman-teman “Kelas E”, yang telah menjadi bagian tak terpisahkan sejak awal semester dan ikut mewarnai perjalanan penulis hingga saat ini. Kebersamaan dalam rutinitas bermain badminton tidak hanya menjadi sarana melepas penat, tetapi juga menghadirkan momen berharga penuh semangat dan keseruan. Terima kasih atas dukungan, motivasi, dan kebersamaan yang selalu diberikan, serta atas kesempatan untuk mengembangkan hobi badminton bersama. Kenangan bersama Kelas E akan senantiasa menjadi bagian berharga dalam perjalanan hidup penulis.
13. Kepada teman-teman Teknik Informatika angkatan 2021 "ASTER" serta seluruh keluarga besar Teknik Informatika UIN Malang, yang telah memberikan dukungan, bantuan, dan semangat selama proses penyusunan skripsi ini.

14. Seluruh pihak yang sudah terlibat, baik secara langsung maupun tidak langsung dari awal perkuliahan hingga akhir penulisan skripsi ini yang tidak dapat penulis sebutkan satu persatu.

15. Diri sendiri yang sudah mampu berjuang hingga detik ini. Terimakasih karena sudah berusaha semaksimal mungkin untuk menyelesaikan salah satu tanggung jawab. Semoga ilmu yang didapatkan penulis dapat bermanfaat untuk orang lain.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan, baik dalam hal keilmuan maupun penyajian. Oleh karena itu, penulis dengan terbuka menerima kritik dan saran yang konstruktif guna perbaikan di masa mendatang. Semoga karya ini dapat memberikan manfaat bagi berbagai pihak yang berkepentingan.

Malang, 05 Juni 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xv
ABSTRAK	xvi
مستخلص البحث.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II STUDI PUSTAKA	6
2.1 Penelitian Terkait.....	6
2.2 <i>Grid Search</i>	12
2.3 <i>Random forest</i>	14
2.4 <i>Preprocessing Text</i>	17
2.5 Term Frequency-Inverse Document Frequency (TF-IDF).....	18
BAB III METODE PENELITIAN	20
3.1 Desain Penelitian	20
3.2 Pengumpulan Data.....	21
3.3 Desain Sistem	24
3.3.1 <i>Preprocessing Text</i>	25
3.3.1.1 <i>Case Folding</i>	26
3.3.1.2 <i>Cleaning</i>	26
3.3.1.3 <i>Tokenization</i>	27
3.3.1.4 <i>Normalization</i>	28
3.3.1.5 <i>Stopword Removal</i>	28
3.3.1.6 <i>Stemming</i>	29
3.3.2 Pembobotan TF-IDF	30
3.3.3 <i>Split Data</i>	32
3.3.4 <i>Grid Search</i>	32
3.3.4.1 <i>Parameter Grid Search</i>	33
3.3.5 <i>Pemodelan Random forest</i>	36
3.3.6 <i>Evaluasi Confusion Matrix</i>	41
3.4 Skenario Pengujian	43
BAB IV UJI COBA DAN PEMBAHASAN.....	46

4.1	Data Penelitian.....	46
4.2	<i>Preprocessing Text</i>	47
4.2.1	<i>Case Folding</i>	48
4.2.2	<i>Cleaning</i>	49
4.2.3	<i>Tokenization</i>	50
4.2.4	<i>Normalization</i>	52
4.2.5	<i>Stopword Removal</i>	53
4.2.6	<i>Stemming</i>	54
4.3	Pembobotan TF-IDF (<i>Term Frequency-Inverse Document Frequency</i>).....	56
4.4	Uji Coba.....	58
4.4.1	Implementasi <i>Grid Search</i> Pada Model Split 80:20	59
4.4.2	Evaluasi Model Split 80:20 dengan Parameter Terbaik.....	59
4.4.3	Implementasi <i>Grid Search</i> Pada Model Split 70:30	61
4.4.4	Evaluasi Model Split 70:30 dengan Parameter Terbaik.....	62
4.4.5	Implementasi <i>Grid Search</i> Pada Model Split 60:40	63
4.4.6	Evaluasi Model Split 60:40 dengan Parameter Terbaik.....	64
4.5	Performa <i>Random forest</i> Setelah Optimasi <i>Hyperparameter</i>	66
4.5.1	Split 80:20	66
4.5.2	Split 70:30	69
4.5.3	Split 60:40	71
4.7	Integrasi Penelitian	75
	BAB V KESIMPULAN DAN SARAN	80
5.1	Kesimpulan.....	80
5.2	Saran	81
	DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi <i>Grid Search</i>	12
Gambar 2.2 <i>K-fold Cross Validation</i> dan <i>Grid Search</i>	14
Gambar 2.3 Alur Metode <i>Random forest</i>	15
Gambar 3.1 Desain Penelitian	20
Gambar 3.2 Desain Sistem	25
Gambar 3.3 Skenario Pengujian	44
Gambar 4.1 Distribusi Sentimen	46
Gambar 4.2 <i>Source Code Case Folding</i>	48
Gambar 4.3 Hasil <i>Case Folding</i>	48
Gambar 4.4 <i>Source Code Cleaning</i>	49
Gambar 4.5 Hasil <i>Cleaning</i>	50
Gambar 4.6 <i>Source Code Tokenization</i>	51
Gambar 4.7 Hasil <i>Tokenization</i>	51
Gambar 4.8 <i>Source Code Normalization</i>	52
Gambar 4.9 Hasil <i>Normalization</i>	52
Gambar 4.10 <i>Source Code Stopword Removal</i>	53
Gambar 4.11 Hasil <i>Stopword Removal</i>	54
Gambar 4.12 <i>Source Code Stemming</i>	54
Gambar 4.13 Hasil <i>Stemming</i>	55
Gambar 4.14 Jumlah Data Setelah <i>Preprocessing</i>	55
Gambar 4.15 <i>Confusion Matrix</i> Model Split 80:20	60
Gambar 4.16 <i>Confusion Matrix</i> Model Split 70:30	62
Gambar 4.17 <i>Confusion Matrix</i> Model Split 60:40	65
Gambar 4.18 <i>Confusion Matrix</i> Model Split 80:20 Sebelum Optimasi	67
Gambar 4.19 <i>Confusion Matrix</i> Model Split 70:30 Sebelum Optimasi	69
Gambar 4.20 <i>Confusion Matrix</i> Model Split 60:40 Sebelum Optimasi	71
Gambar 4.21 Performa Model Berdasarkan Split Data	73

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	9
Tabel 3.1 Penjelasan Fitur.....	22
Tabel 3.2 Penyajian Contoh Data Utama.....	22
Tabel 3.3 Pelabelan Ulasan Secara Manual.....	24
Tabel 3.4 Penerapan <i>Case folding</i>	26
Tabel 3.5 Penerapan <i>Cleaning</i>	26
Tabel 3.6 Penerapan <i>Tokenization</i>	27
Tabel 3.7 Penerapan <i>Normalization</i>	28
Tabel 3.8 Penerapan <i>Stopword removal</i>	29
Tabel 3.9 Penerapan <i>Stemming</i>	29
Tabel 3.10 <i>Term Frequency</i>	30
Tabel 3.11 Menghitung DF (<i>Document Frequency</i>).....	31
Tabel 3.12 Menghitung IDF (<i>Inverse Document Frequency</i>).....	31
Tabel 3.13 <i>Term Frequency - Inverse Document Frequency</i>	32
Tabel 3.14 Ruang Pencarian <i>Grid Search</i>	33
Tabel 3.15 Contoh Hasil TF-IDF	34
Tabel 3.16 Contoh Kombinasi Parameter	35
Tabel 3.17 Contoh Perhitungan <i>K-Fold</i> Kombinasi <i>Hyperparameter</i>	35
Tabel 3.18 Contoh Hasil TF-IDF Data Baru.....	40
Tabel 3.19 Kondisi <i>Confusion matrix</i>	41
Tabel 3.20 Skenario Pengujian	45
Tabel 4.1 Penyajian Data Penelitian	47
Tabel 4.2 Data Sampel Perhitungan TF-IDF Hasil <i>Preprocessing</i>	56
Tabel 4.3 Hasil Pembobotan TF Hingga IDF	57
Tabel 4.4 Hasil Pembobotan TF-IDF	57
Tabel 4.5 Distribusi Label Pada Tiap Split	59
Tabel 4.6 Kombinasi <i>Hyperparameter</i> Terbaik (Model Split 80:20)	59
Tabel 4.7 Evaluasi Model Split 80:20.....	60
Tabel 4.8 Kombinasi <i>Hyperparameter</i> Terbaik (Model Split 70:30)	61
Tabel 4.9 Evaluasi Model Split 70:30.....	63
Tabel 4.10 Kombinasi <i>Hyperparameter</i> Terbaik (Model Split 60:40)	64
Tabel 4.11 Evaluasi Model Split 60:40.....	65
Tabel 4.12 <i>Confusion Matrix</i> Model RF dan (RF + <i>Grid Search</i>) Split 80:20	67
Tabel 4.13 Metrik Evaluasi Model RF dan (RF + <i>Grid Search</i>) Split 80:20.....	68
Tabel 4.14 <i>Confusion Matrix</i> Model RF dan (RF + <i>Grid Search</i>) Split 70:30	69
Tabel 4.15 Metrik Evaluasi Model RF dan (RF + <i>Grid Search</i>) Split 70:30.....	70
Tabel 4.16 <i>Confusion Matrix</i> Model RF dan (RF + <i>Grid Search</i>) Split 60:40	72
Tabel 4.17 Metrik Evaluasi Model RF dan (RF + <i>Grid Search</i>) Split 60:40.....	72

ABSTRAK

Ramadani, Yudistira. 2025. **Optimasi *Hyperparameter Random forest* Dengan *Grid Search* Untuk Klasifikasi Sentimen Publik Pada Ulasan Produk iPhone.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Tri Mukti Lestari, M.Kom (II) Dr. M. Amin Hariyadi, M.T.

Kata kunci: *Random forest*, *Grid Search*, Klasifikasi Sentimen, *Hyperparameter*, Ulasan iPhone.

Ulasan pengguna iPhone mengandung sentimen yang penting untuk dianalisis guna mendukung pengambilan keputusan berbasis opini publik. Tantangan muncul karena jumlah data yang besar dan tidak terstruktur, sehingga dibutuhkan pendekatan klasifikasi otomatis. Penelitian ini menggunakan algoritma *Random forest* dengan optimasi *hyperparameter* menggunakan teknik *Grid Search*. Dataset berisi 3.062 ulasan diproses melalui tahapan *Preprocessing* teks, pembobotan TF-IDF, dan pelabelan manual menjadi sentimen positif dan negatif. Total 162 kombinasi *hyperparameter* diuji. Parameter terbaik ditemukan melalui evaluasi rata-rata akurasi pada validasi silang 5-fold. Hasil menunjukkan bahwa model sebelum optimasi menghasilkan akurasi 82.59%, sedangkan setelah optimasi meningkat menjadi 83.76%. Pada konfigurasi terbaik dengan split data 70:30, diperoleh *precision* 84.97%, *recall* 92.80%, dan *F1-score* 88.71%. Optimasi *hyperparameter* terbukti meningkatkan performa model dalam klasifikasi sentimen publik terhadap ulasan produk iPhone secara signifikan.

ABSTRACT

Ramadani, Yudistira. 2025. **Optimization of *Random forest Hyperparameters with Grid Search for Public Sentiment Classification on iPhone Product Reviews***. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Promotor: (I) Tri Mukti Lestari, M.Kom. (II) Dr. M. Amin Hariyadi, M.T.

Key words: *Random forest, Grid Search, Sentiment Classification, Hyperparameter, iPhone Reviews*

iPhone user reviews contain sentiment that is important to analyze to support public opinion-based decision-making. Challenges arise due to the large amount of unstructured data, so an automated classification approach is needed. This research uses *Random forest* algorithm with *hyperparameter* optimization using *Grid Search* technique. The dataset of 3,062 reviews was processed through text *Preprocessing*, TF-IDF weighting, and manual labeling into positive and negative sentiments. A total of 162 *hyperparameter* combinations were tested. The best parameters were found through evaluating the average accuracy on 5-fold cross validation. The results show that the model before optimization yields an accuracy of 82.59%, while after optimization it increases to 83.76%. In the best configuration with a 70:30 data split, *precision* 84.97%, *recall* 92.80%, and *F1-score* 88.71% were obtained. *Hyperparameter* optimization is proven to significantly improve the model's performance in public sentiment classification of iPhone product reviews.

مستخلص البحث

رمضان، يوديستيرا 2025. تحسين معيار الغابة العشوائية المفرط مع البحث الشبكي لتصنيف المشاعر العامة على مراجعات منتجات الآيفون. أطروحة. برنامج دراسة هندسة المعلوماتية، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية، مولانا مالك إبراهيم مالانج. المشرف: (أولاً) تري موكتي ليستاري، م. كوم (ثانياً) د. م. أمين هريادي، م. ت.

الكلمات المفتاحية : الغابة العشوائية، البحث الشبكي، تصنيف المشاعر، المقياس الفائق، مراجعات الآيفون

تحتوي مراجعات مستخدمي الآيفون على مشاعر من المهم تحليلها لدعم عملية اتخاذ القرارات القائمة على الرأي العام. تنشأ التحديات بسبب الكمية الكبيرة من البيانات غير المهيكلة، لذلك هناك حاجة إلى نهج تصنيف تلقائي. ويستخدم هذا البحث خوارزمية الغابة العشوائية مع تحسين المعلمة الفائقة باستخدام تقنية البحث الشبكي. تمت معالجة مجموعة البيانات المكونة من 3,062 مراجعة من خلال المعالجة المسبقة للنص، وترجيح TF-IDF، والتصنيف اليدوي إلى مشاعر إيجابية وسلبية. تم اختبار ما مجموعه 162 مجموعة من مجموعات العلامات الفائقة. تم العثور على أفضل العلامات من خلال تقييم متوسط الدقة على 5 أضعاف التحقق من صحة 5 أضعاف. تُظهر النتائج أن النموذج قبل التحسين يحقق دقة تبلغ 82.59%، بينما ترتفع هذه الدقة بعد التحسين إلى 83.76%. في أفضل تكوين مع تقسيم البيانات بنسبة 70:30، تم الحصول على دقة 84.97%، واسترجاع 92.80%، ودرجة F1 88.71%. وقد ثبت أن تحسين المعامل الفائق يحسن بشكل كبير من أداء النموذج في تصنيف المشاعر العامة لمراجعات منتجات

iPhone.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat telah membawa dampak signifikan terhadap berbagai aspek kehidupan manusia, termasuk bagaimana masyarakat mengakses informasi dan membuat keputusan, khususnya dalam hal pembelian produk (Darioshi & Lahav, 2021). Kemajuan teknologi ini telah mengubah cara konsumen mencari dan memverifikasi informasi, dari metode konvensional seperti bertanya kepada teman atau membaca ulasan di media cetak, menjadi lebih modern dengan memanfaatkan platform digital (Belharar & Chakor, 2021).

Dalam beberapa tahun terakhir, iPhone tetap menjadi salah satu smartphone paling diminati di pasar global. Pada tahun 2024, iPhone 15 Pro Max tercatat sebagai smartphone terlaris di dunia, dengan pangsa pasar mencapai 4,4%, diikuti oleh iPhone 15 (4,3%) dan iPhone 15 Pro (3,7%) (Yonatan, 2024). Meskipun mengalami sedikit penurunan penjualan di beberapa wilayah seperti China, iPhone tetap menjadi produk unggulan Apple, dengan total pendapatan mencapai \$69,7 miliar pada kuartal terakhir 2023, meningkat 6% secara *year-on-year* (YoY) (Harjadi, 2024).

Di Indonesia, pangsa pasar iPhone juga mengalami pertumbuhan. Hingga Oktober 2024, Apple menguasai sekitar 12% pasar smartphone di Indonesia, meningkat dibandingkan tahun sebelumnya (Siahaan, 2024). Tren ini sejalan

dengan meningkatnya jumlah perangkat smartphone aktif di Indonesia, yang mencapai 354 juta unit, menjadikan Indonesia sebagai salah satu pasar terbesar untuk perangkat mobile (Yonatan, 2024).

Produk iPhone tetap menjadi salah satu merek smartphone paling diminati di pasar global. Popularitasnya didukung oleh berbagai faktor, seperti inovasi teknologi, ekosistem yang terintegrasi, serta loyalitas pengguna yang tinggi (Desika, 2024). Popularitas ini tercermin dalam banyaknya ulasan yang dipublikasikan di berbagai platform *e-commerce* seperti Amazon, yang menggambarkan pengalaman langsung konsumen, baik dalam bentuk dukungan maupun kritik. Dengan volume ulasan yang terus bertambah, analisis manual menjadi sulit dilakukan. Oleh karena itu, diperlukan analisis sentimen untuk memahami pola penerimaan publik serta mendukung strategi pemasaran dan pengembangan produk (Hossain & Rahman, 2023).

Analisis sentimen adalah teknik NLP untuk mengklasifikasikan opini menjadi positif, netral, atau negatif (Aftab dkk., 2023). Penelitian ini menggunakan *Random forest*, metode ensemble learning yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi klasifikasi (Siemers & Bajorath, 2023). Studi sebelumnya (Salungweni dkk., 2024) melaporkan akurasi 83%, namun tanpa mempertimbangkan pengaruh *hyperparameter*. Sementara itu, Penelitian oleh (N. Zhu dkk., 2022) menunjukkan bahwa optimasi *hyperparameter* pada *Random forest* dapat secara signifikan meningkatkan kinerja deteksi klasifikasi. Oleh karena itu, penelitian ini menggunakan *Grid Search* untuk menguji pengaruh tuning beberapa *hyperparameter* penting yaitu *n_estimators*, *max_depth*, *max_features*,

min_samples_split, dan *min_samples_leaf* terhadap performa klasifikasi sentimen publik pada rekomendasi produk iPhone.

Dalam era digital, tidak semua informasi dapat langsung dipercaya, sehingga perlu verifikasi sebelum menyimpulkan. Islam menekankan kehati-hatian dalam menerima informasi, sebagaimana firman Allah dalam Surat Al-Hujurat ayat 6 :

يَا أَيُّهَا الَّذِينَ آمَنُوا إِن جَاءَكُمْ فَاسِقٌ بِنَبَأٍ فَتَبَيَّنُوا أَن تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصِحِّحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ

"Hai orang-orang yang beriman, jika seseorang yang fasik datang kepadamu membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu." (Q.S. Al-Hujarat:6)

Ayat ini menekankan pentingnya meneliti kebenaran informasi sebelum mempercayai atau menyebarkannya. Tafsir Tahlili dari Kementerian Agama RI menjelaskan bahwa menerima berita tanpa verifikasi, terutama dari sumber yang tidak terpercaya, dapat menimbulkan kesalahpahaman dan dampak merugikan (Tafsir Al Hujurat Ayat 6 Kemenag RI, 2024). Dalam penelitian ini, analisis sentimen digunakan untuk memahami opini publik terhadap suatu topik. Namun, agar hasilnya objektif dan akurat, diperlukan metode yang mampu memverifikasi data dan menghindari kesimpulan bias. Prinsip kehati-hatian dalam Islam menjadi pedoman dalam memastikan bahwa proses analisis dilakukan secara sistematis, transparan, dan bertanggung jawab.

Penelitian ini berkontribusi dalam mengembangkan metode yang lebih akurat untuk memahami sentimen publik terhadap ulasan pengguna iPhone di platform *e-commerce*. Dengan menerapkan *Random forest*, penelitian ini memberikan wawasan tentang kinerja algoritma dalam analisis sentimen berbahasa Indonesia

terhadap produk teknologi. Hasilnya diharapkan bermanfaat bagi produsen smartphone, penjual, dan calon pembeli dalam membuat keputusan yang lebih informatif berdasarkan analisis sentimen yang akurat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah :

1. Bagaimana performa metode *Random forest* dalam mengklasifikasikan sentimen publik terhadap ulasan pengguna iPhone berdasarkan metrik evaluasi *acuration*, *precision*, *recall*, dan *F1-score*?
2. Bagaimana pengaruh optimasi *hyperparameter* *n_estimators*, *max_depth*, *max_features*, *min_samples_split*, dan *min_samples_leaf* terhadap hasil akurasi model *Random forest* dalam klasifikasi sentimen publik terhadap ulasan pengguna iPhone?

1.3 Batasan Masalah

Untuk memfokuskan penelitian dan mendapatkan hasil yang optimal, beberapa batasan masalah dalam penelitian ini adalah:

1. Data yang digunakan adalah dataset ulasan pengguna iPhone, yang merupakan data sekunder diperoleh dari website Kaggle.com dan terdiri dari 3.062 ulasan.
2. Klasifikasi sentimen dibagi menjadi dua kategori: positif dan negatif.
3. Metode klasifikasi yang digunakan adalah *Random forest*.

4. *Hyperparameter* yang dioptimalkan adalah *n_estimators*, *max_depth*, *max_features*, *min_samples_split*, dan *min_samples_leaf*.
5. Evaluasi performa klasifikasi menggunakan *confusion matrix*.

1.4 Tujuan Penelitian

Berdasarkan latar belakang dan pernyataan masalah di atas adalah :

1. Menganalisis performa metode *Random forest* dalam mengklasifikasikan sentimen publik terhadap ulasan pengguna iPhone berdasarkan metrik evaluasi seperti *acuration*, *precision*, *recall*, dan *F1-score*.
2. Mengevaluasi pengaruh optimasi *hyperparameter* (*n_estimators*, *max_depth*, *max_features*, *min_samples_split*, dan *min_samples_leaf*) terhadap perubahan nilai akurasi model *Random forest* menggunakan *confusion matrix*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat berkontribusi pada pengembangan ilmu pengetahuan di bidang *text mining*, khususnya analisis sentimen berbahasa Indonesia menggunakan metode *Random forest* dengan optimasi *hyperparameter* untuk meningkatkan performa klasifikasi. Selain itu, hasil penelitian ini juga dapat menjadi acuan bagi berbagai pemangku kepentingan, termasuk produsen smartphone, penjual, dan calon pembeli, dalam memahami opini publik terhadap produk iPhone di platform *e-commerce* serta dalam membuat keputusan berbasis data.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian sebelumnya dapat dijadikan referensi untuk penelitian ini, memberikan panduan yang membantu penulis dalam menyelesaikan penelitian.

Penelitian (G & B.Sumathi, 2020) mengeksplorasi analisis sentimen umpan balik pelanggan menggunakan algoritma *Random forest* yang dioptimalkan dengan metode *Grid Search* untuk penyetelan *hyperparameter*. Dengan menerapkan *Grid Search*, akurasi model meningkat dari 84,53% menjadi 90,02% melalui penyetelan parameter seperti jumlah pohon dan variabel maksimum dalam setiap pohon. Data yang digunakan terdiri dari 1500 ulasan produk, terbagi menjadi ulasan positif dan negatif. Hasil menunjukkan bahwa pendekatan ini lebih efektif dibandingkan model tanpa penyetelan parameter, menegaskan pentingnya penyetelan *hyperparameter* dalam meningkatkan kinerja klasifikasi.

(Saepudin dkk., 2024) dalam penelitiannya menganalisis sentimen ulasan pengguna Shopee dengan membandingkan kinerja SVM, *Random forest*, dan *Logistic Regression*. Dataset terdiri dari 3.000 ulasan dari Google Play Store, dengan 90% data untuk pelatihan dan 10% untuk pengujian. Pelabelan menggunakan TextBlob menghasilkan 307 ulasan positif, 2.537 netral, dan 156 negatif. Setelah *Preprocessing*, hasil pengujian menunjukkan *Random forest* memiliki akurasi tertinggi (94%), diikuti oleh SVM (91%) dan *Logistic Regression*

(86%). Penelitian ini memberikan wawasan bagi Shopee dalam memahami persepsi pelanggan terhadap layanannya.

Penelitian (Rohman dkk., 2025) mengoptimalkan *hyperparameter* model *Random forest* untuk klasifikasi diabetes menggunakan dataset Pima Indian Diabetes. Dengan menerapkan teknik *Grid Search* dan *Random Search*, penelitian ini bertujuan meningkatkan akurasi model dalam diagnosis dini diabetes. Metodologi mencakup pengumpulan data, *Preprocessing*, pembagian dataset, dan evaluasi menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*. Hasil menunjukkan bahwa penyetelan *hyperparameter* secara signifikan meningkatkan kinerja model, dengan *Random Search* mencapai akurasi 0,75, presisi 0,64, dan *recall* 0,69. Temuan ini menegaskan pentingnya penyetelan *hyperparameter* dalam meningkatkan kinerja model untuk aplikasi machine learning dalam diagnosis medis diabetes.

(Miftahusalam dkk., 2022) meneliti efektivitas *Random forest*, SVM, dan Naïve Bayes dalam analisis sentimen terkait kebijakan penghapusan tenaga honorer di Indonesia. Sebanyak 1.147 tweet dikumpulkan dan diproses melalui *Preprocessing* serta *feature extraction* menggunakan BoW dan TF-IDF. Hasil penelitian menunjukkan bahwa *Random forest* dengan random oversampling memiliki akurasi (66,67%), diikuti SVM (65,33%) dan Naïve Bayes (64%). Evaluasi juga menggunakan *F1-score*, di mana *Random forest* unggul dengan nilai 0,67. Studi ini mengungkap adanya keseimbangan opini publik terhadap kebijakan tersebut, memberikan wawasan bagi pemerintah dalam pengambilan keputusan.

(Jennifer dkk., 2024) menganalisis sentimen pengguna TikTok terkait kasus pembunuhan Wayan Mirna Salihin yang melibatkan Jessica Kumala Wongso. Dengan menggunakan metode *Random forest Classifier*, analisis dilakukan terhadap 15.000 komentar TikTok. Hasilnya menunjukkan bahwa mayoritas komentar bersifat positif (52%) mendukung Jessica, diikuti oleh opini netral (36%) dan negatif (12%). Model *Random forest* yang dibangun menunjukkan akurasi, presisi, *recall*, dan *F1-score* di atas 80% untuk semua kelas sentimen, yang menunjukkan efektivitasnya dalam klasifikasi. Pemetaan confusion matrix juga menunjukkan kemampuan model dalam membedakan komentar positif, negatif, dan netral dengan baik.

Penelitian (Hidayat & Sanjaya, 2025) menganalisis sentimen publik terhadap kebijakan TKDN produk Apple di Indonesia menggunakan *Random forest*. Data dikumpulkan dari platform X, diproses dengan TF-IDF, dan diklasifikasikan menjadi negatif, netral, dan positif. Model mencapai 91% akurasi, dengan *precision* dan *recall* masing-masing 91%, tetapi cenderung mengklasifikasikan sentimen sebagai netral akibat ketidakseimbangan kelas. Studi ini memberikan wawasan bagi industri dan pembuat kebijakan serta membuka peluang eksplorasi teknik penyeimbangan kelas atau model yang lebih kompleks.

Penelitian (Miftahusalam dkk., 2023) membandingkan Naive Bayes dan *Random forest* dalam analisis sentimen ulasan BCA Mobile dari 1–31 Januari 2023. Data diproses melalui *Preprocessing*, penghapusan data tidak relevan, pelabelan sentimen, dan pembobotan TF-IDF. Hasil menunjukkan *Random forest* unggul dengan akurasi 93,48% dan *F1-score* 90,48%, sedangkan Naive Bayes memperoleh

akurasi 92,31% dan *F1-score* 88,34%. Temuan ini menunjukkan *Random forest* lebih efektif untuk klasifikasi sentiment.

Penelitian (Dana dkk., 2024) membandingkan algoritma *Decision Tree* dan *Random forest* dengan penerapan *hyperparameter* tuning dalam mendeteksi risiko stroke berdasarkan berbagai variabel pasien. Menggunakan dataset dari Kaggle, penelitian ini menganalisis faktor-faktor seperti usia, jenis kelamin, riwayat hipertensi, dan status merokok. Hasil menunjukkan bahwa model *Random forest* memberikan akurasi lebih tinggi (0,97) dibandingkan *Decision Tree* (0,95), dan setelah penerapan *hyperparameter* tuning, akurasi *Random forest* meningkat menjadi 0,98. Penelitian ini menekankan pentingnya deteksi dini stroke dan menunjukkan bahwa *Random forest* dengan *hyperparameter* tuning adalah model yang lebih efektif untuk prediksi risiko stroke, memberikan kontribusi signifikan dalam upaya pencegahan dan pengelolaan penyakit ini.

Berdasarkan beberapa penelitian yang telah dijelaskan sebelumnya, penulis menyusun sebuah Tabel 2.1 yang membandingkan setiap penelitian tersebut.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Perbedaan
1	G & B.Sumathi., (2020)	<i>Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction</i>	<i>Random forest</i>	Model mengalami peningkatan performa hingga 6% dengan pemilihan <i>hyperparameter</i> optimal	<ul style="list-style-type: none"> - Objek penelitian adalah ulasan pengguna iphone - Jumlah Dataset yang digunakan berupa 3091 komentar. - tiga variasi split data
2	Saepudin dkk., (2024)	Perbandingan Algoritma SVM, <i>Random forest</i> , dan	SVM, <i>Random forest</i> , <i>Logistic</i>	<i>Random forest</i> memiliki akurasi tertinggi dibandingkan	<ul style="list-style-type: none"> - Objek Penelitian. - Menggunakan parameter n estimator, max

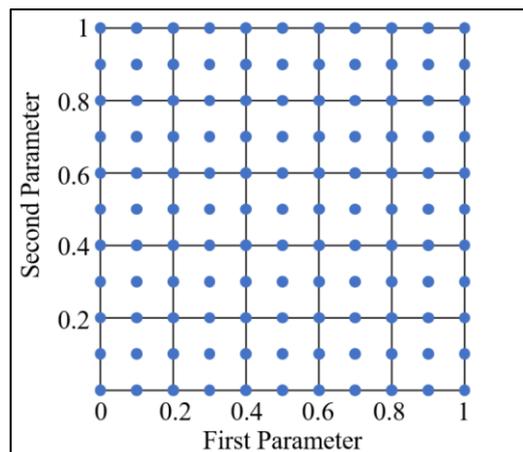
No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Perbedaan
		<i>Logistic Regression</i> untuk Klasifikasi Sentimen Shopee	<i>Regression</i>	SVM dan <i>Logistic Regression</i> .	<i>depth</i> , dan <i>min sample</i> - Variasi nilai <i>n estimator</i> - tiga variasi split data
3	Rohman dkk., (2025)	<i>Hyperparameter Tuning of Random forest Algorithm for Diabetes Classification</i>	<i>Random forest</i>	Hasil penelitian menunjukkan bahwa penyetelan <i>hyperparameter</i> secara signifikan meningkatkan kinerja model <i>Random forest</i> , dengan <i>Random Search</i> memberikan hasil terbaik	- Objek penelitian adalah ulasan pengguna produk iphone - Jumlah dataset yang digunakan - <i>Hyperparameter</i> yang digunakan adalah <i>n_estimators</i> dan <i>max_features</i> - tiga variasi split data
4	Miftahusalam dkk., (2022)	Analisis Sentimen Publik terhadap Kebijakan Penghapusan Tenaga Honorer di Indonesia	<i>Random forest</i> , SVM, <i>Naïve Bayes</i>	<i>Random forest</i> memiliki akurasi tertinggi, diikuti oleh SVM dan <i>Naïve Bayes</i> .	- Objek penelitian adalah ulasan pengguna iphone - Dataset yang digunakan 3062 - Splitting data tiga variasi - Menggunakan parameter khusus pada <i>Random forest</i>
5	Jennifer dkk., (2024)	Analisis Sentimen Pada Pengguna Tiktok (Studi Kasus: Jessica-Mirna)	<i>Random forest</i>	Performa model cukup baik dengan akurasi 92%	- Objek penelitian - Terdapat parameter khusus - Splitting data tiga variasi
6	Hidayat & Sanjaya (2025)	Analisis Sentimen Publik Terhadap Penjualan iphone 16 dan Kebijakan TKDN di Indonesia	<i>Random forest</i>	Akurasi <i>Random forest</i> sangat tinggi	- Objek Penelitian berupa ulasan pengguna iphone - Dataset 3062 - Menggunakan parameter khusus - Menggunakan tiga variasi split data
7	Miftahusalam dkk., (2023)	Analisis Sentimen <i>Review</i>	<i>Random forest dan</i>	Metode <i>Random forest</i> menunjukkan akurasi yang lebih	- Objek penelitian berupa ulasan pengguna iphone

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Perbedaan
		Aplikasi <i>BCA Mobile</i>	<i>Naive Bayes</i>	tinggi dibandingkan <i>Naive Bayes</i>	<ul style="list-style-type: none"> - Dataset penelitian 3062 - Menggunakan Parameter Khusus - Menggunakan satu metode klasifikasi - Menggunakan tiga variasi split data
8	Dana dkk., (2024)	Perbandingan Algoritma <i>Decision Tree</i> dan <i>Random forest</i> dengan <i>Hyperparameter Tuning</i> dalam Mendeteksi Penyakit Stroke	<i>Random forest</i> dan <i>Decision Tree</i>	Model <i>Random forest</i> memiliki performa yang tinggi dibandingkan dengan <i>Decision Tree</i> dan mengalami kenaikan akurasi setelah tuning <i>hyperparameter</i>	<ul style="list-style-type: none"> - Objek penelitian berupa ulasan pengguna iphone - Menggunakan variasi split data - Penggunaan <i>hyperparameter</i> khusus

Tabel 2.1 menyajikan ringkasan penelitian terdahulu yang menggunakan algoritma *Random forest* dan beberapa metode lainnya dalam analisis sentimen di berbagai platform. Penelitian yang dirangkum mencakup analisis sentimen pada umpan balik pelanggan, ulasan pengguna Shopee, opini publik di media sosial terkait kebijakan, serta komentar di TikTok. Dari penelitian yang telah dilakukan, metode *Random forest* sering digunakan karena kemampuannya dalam menangani data dalam jumlah besar dan memberikan hasil klasifikasi yang akurat. Hasil penelitian menunjukkan bahwa *Random forest* tidak hanya unggul dalam akurasi, tetapi juga efektif dalam berbagai konteks, seperti diagnosis medis dan analisis opini publik, menegaskan pentingnya penyetelan *hyperparameter* untuk meningkatkan kinerja model.

2.2 Grid Search

Peningkatan akurasi model dapat dioptimalkan melalui penerapan *cross-validation* dengan metode *Grid Search* (Yan dkk., 2022). *Grid Search* adalah metode eksplorasi sistematis yang digunakan untuk menemukan *hyperparameter* terbaik dalam rangka meningkatkan kinerja model klasifikasi. Dengan menguji semua kombinasi *hyperparameter* yang telah ditentukan sebelumnya, metode ini mampu mengidentifikasi kombinasi optimal yang memberikan performa terbaik bagi model (Fajri & Primajaya, 2023). Berikut adalah ilustrasi dari *Grid Search* (Hien dkk., 2020).



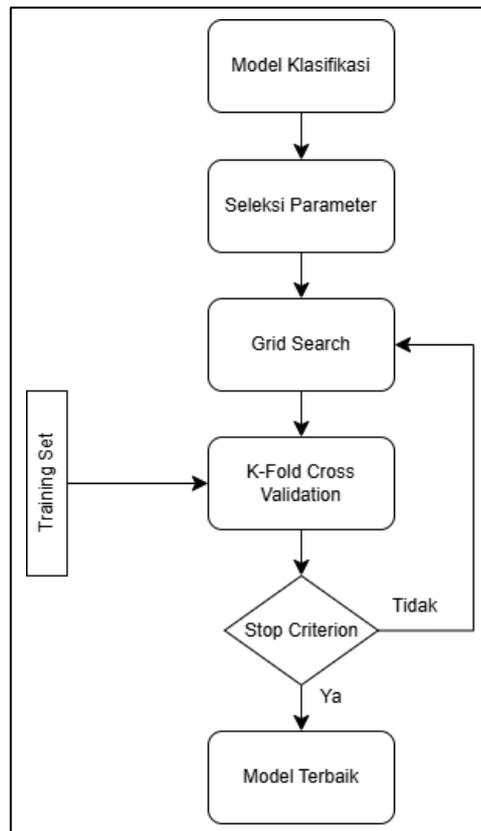
Gambar 2.1 Ilustrasi *Grid Search*

Proses ini dilakukan dengan membangun grid berisi berbagai kombinasi *hyperparameter* yang mungkin digunakan, lalu melatih dan menguji model pada set validasi atau *cross-validation* untuk setiap kombinasi tersebut. Kombinasi *hyperparameter* yang menghasilkan kinerja terbaik pada data validasi atau *cross-validation* akan dipilih sebagai konfigurasi optimal. Langkah-langkah dalam algoritma *Grid Search* adalah sebagai berikut (Shams dkk., 2023):

1. Menentukan *hyperparameter* yang akan dioptimasi serta rentang nilai yang akan diuji.

2. Membuat *grid* yang berisi semua kemungkinan kombinasi *hyperparameter*.
3. Untuk setiap kombinasi *hyperparameter* dalam *grid*:
 - a. Latih model dengan data pelatihan menggunakan kombinasi *hyperparameter* saat ini.
 - b. Evaluasi model menggunakan metrik kinerja pada data validasi atau *cross-validation* ($CV = 5$).
 - c. Catat hasil kinerja dari setiap kombinasi *hyperparameter*.
4. Pilih kombinasi *hyperparameter* yang menghasilkan metrik kinerja terbaik.

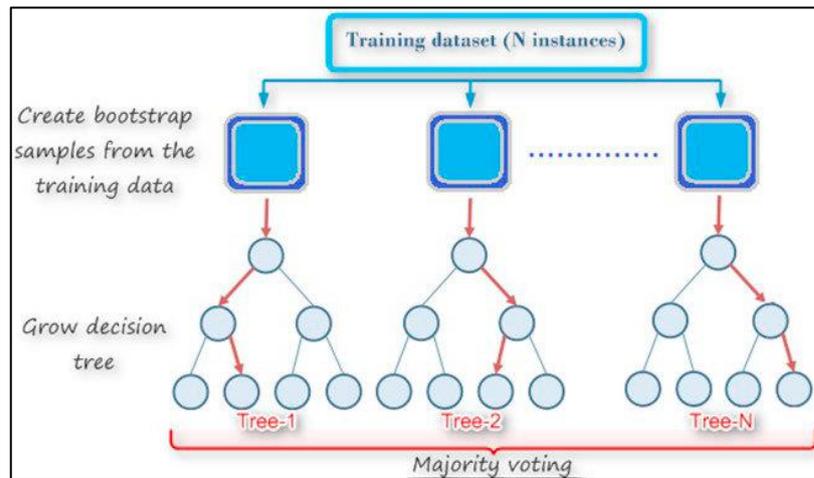
Metode *Grid Search* sering kali digabungkan dengan *k-fold cross-validation*, yang menciptakan indeks evaluasi untuk model klasifikasi (Yan dkk., 2022). *K-fold cross-validation* memungkinkan pengulangan data pelatihan dan pengujian sebanyak k kali, di mana $1/k$ dari dataset digunakan sebagai data uji pada setiap iterasi (Singh dkk., 2021). Dari proses ini, diperoleh akurasi model untuk setiap iterasi, dan kinerja model klasifikasi dievaluasi berdasarkan rata-rata akurasi yang diperoleh. Selanjutnya, parameter *classifier* disesuaikan berdasarkan hasil pencarian *grid*, dan akurasi *classifier* dihitung kembali. Proses optimasi ini berlanjut hingga semua kombinasi parameter dievaluasi, dengan tujuan untuk menghasilkan nilai akurasi maksimum pada Gambar 2.1 (Toha dkk., 2022).



Gambar 2.2 *K-fold Cross Validation* dan *Grid Search*

2.3 *Random forest*

Random forest adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi, yang mengandalkan teknik *ensemble learning* dengan menggabungkan banyak pohon keputusan (*Decision Trees*) untuk meningkatkan akurasi dan mengurangi risiko *overfitting* (Che dkk., 2023). Algoritma ini bekerja dengan cara membangun sejumlah pohon keputusan dari subset acak data pelatihan yang diambil dengan metode *bootstrap* (sampling dengan pengembalian). Setiap pohon dalam hutan ini memberikan prediksi, dan hasil akhir ditentukan berdasarkan suara terbanyak untuk klasifikasi atau rata-rata untuk regresi (Sage dkk., 2020). Alur metode *Random forest* direpresentasikan oleh Gambar 2.3 (Syafia dkk., 2023).



Gambar 2.3 Alur Metode *Random forest*

Proses kerja algoritma *Random forest* dimulai dengan memanfaatkan data training yang tersedia. Data ini dipecah menjadi beberapa subset secara acak menggunakan teknik *bootstrap*. Metode *bootstrap* memungkinkan pemilihan sampel dengan pengulangan dari data asli, sehingga setiap subset mungkin mengandung elemen data yang sama secara berulang. Subset ini kemudian digunakan untuk membangun pohon keputusan secara individual (Novikova dkk., 2022). Dalam proses pembangunan pohon, terdapat pembatasan pada kedalaman pohon dan jumlah fitur yang dapat digunakan pada setiap pemisahan, yang bertujuan untuk mencegah model dari *overfitting*.

Menurut (Contreras dkk., 2021), algoritma *Random forest* memiliki berbagai parameter yang berperan dalam membangun pohon keputusan secara acak. Namun, beberapa parameter utama yang paling berpengaruh terhadap hasil prediksi dan membantu mencegah *overfitting* adalah sebagai berikut :

1. *N Estimators* : Menentukan jumlah pohon dalam hutan. Nilai parameter ini dapat disesuaikan antara 10 hingga 100. Secara default, nilainya 100.

2. *Max Depth* : Mengontrol kedalaman maksimum dari setiap pohon dalam model. Secara default, nilainya *None* (sampai semua daun murni).
3. *Criterion* : Mengukur kualitas pemisahan (*split*) pada setiap node. *Random forest* mendukung dua kriteria utama, yaitu *Gini impurity* dan *entropy*, yang masing-masing digunakan untuk mengukur ketidakmurnian dan informasi yang diperoleh dari suatu pemisahan.
4. *Min Samples Split* : Menentukan jumlah minimum sampel yang diperlukan untuk membagi suatu node dalam pohon. Secara default, nilainya adalah 2.
5. *Min Samples Leaf* : Jumlah minimum sampel yang diizinkan untuk menjadi node daun. Dari 1 hingga jumlah total sampel. Secara default, nilainya 1.
6. *Max Features* : Menentukan jumlah fitur yang dipertimbangkan saat mencari pemisahan terbaik. Nilai defaultnya bisa berupa *auto*, *sqrt*, atau *log2*.

Saat membangun pohon, pada setiap node, algoritma memilih subset acak dari fitur yang tersedia untuk menentukan pemisahan terbaik. Pendekatan ini tidak hanya menambah keragaman antar pohon, tetapi juga mengurangi kemungkinan korelasi di antara mereka. Proses ini diulang sebanyak $n_estimators$, yang merupakan jumlah pohon yang ingin dibangun dalam model (Sipper & Moore, 2020).

Setelah semua pohon keputusan selesai dibangun, model siap untuk melakukan prediksi terhadap data baru. Proses prediksi dilakukan dengan mengumpulkan hasil dari setiap pohon keputusan. Setiap pohon akan mengklasifikasikan data baru dan mencatat hasilnya. Setelah semua hasil

terkumpul, dilakukan agregasi untuk menghitung jumlah prediksi untuk setiap kelas dari semua pohon. Kelas yang mendapatkan jumlah prediksi terbanyak akan menjadi hasil akhir untuk data baru tersebut, mengikuti prinsip voting mayoritas yang menjadi dasar dari metode ensemble learning (Abdulkareem & Abdulazeez, 2021). Dengan cara ini, *Random forest* mampu memberikan prediksi yang lebih akurat dan robust dibandingkan dengan menggunakan satu pohon keputusan saja.

2.4 *Preprocessing Text*

Preprocessing Text merupakan langkah krusial dalam menormalkan istilah yang terdapat dalam kalimat. Tujuannya adalah untuk memastikan bahwa data yang digunakan dalam pelatihan adalah akurat dan fitur yang diekstraksi bersifat konsisten, sehingga memudahkan proses pengolahan data (Palomino & Aider, 2022). Proses *Preprocessing* teks diperlukan untuk menstandarisasi teks menjadi bentuk yang lebih alami dengan menghapus kata-kata yang tidak relevan, sehingga dapat mengoptimalkan kalimat yang akan digunakan dalam proses klasifikasi (Gifari dkk., 2022). Dalam penelitian ini, terdapat enam tahapan *Preprocessing* teks yang akan dijelaskan sebagai berikut :

1. *Cleaning* merupakan proses penghapusan data teks dari kalimat yang tidak penting, tidak baku, dan tidak relevan.
2. *Case folding* merupakan proses mengubah huruf kapital yang ada di seluruh kalimat menjadi huruf kecil.
3. *Normalization* adalah proses menstandarkan teks dengan mengubah kata tidak baku, singkatan, atau istilah slang menjadi bentuk baku.

4. Tokenizing merupakan proses memecah kalimat pada data menjadi pecahan kata tunggal.
5. *Stopword removal* merupakan proses penghapusan data teks yang berisi kata tidak memiliki makna dan tidak relevan pada topik.
6. *Stemming* merupakan proses perubahan data teks pada kata imbuhan menjadi kata dasar.

2.5 Term Frequency-Inverse Document Frequency (TF-IDF)

Proses *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah salah satu metode pembobotan yang umum digunakan dalam perolehan informasi dan text mining (Artama dkk., 2020). Proses TF-IDF bertujuan untuk menghasilkan representasi vektor yang menggambarkan tingkat kepentingan setiap kata dalam dokumen terhadap keseluruhan korpus. Proses perhitungan TF-IDF membantu memfilter dan menonjolkan kata-kata kunci yang memiliki kontribusi tinggi terhadap makna dan sentimen dalam setiap dokumen. Semakin sering sebuah kata atau frase muncul dalam sebuah teks, maka semakin tinggi pula skor yang diperoleh algoritma TF-IDF. Setiap kata yang ada akan memiliki nilai yang ditetapkan padanya. Nilai-nilai ini kemudian akan diurutkan dari yang terbesar ke terkecil atau sebaliknya. Hasilnya adalah representasi numerik yang memungkinkan analisis sentimen yang lebih akurat dengan mempertimbangkan bobot relatif kata-kata dalam konteks yang telah ditentukan. Pembobotan ditentukan dengan dengan mengalikan term frequency (TF) frekuensi kemunculan suatu kata dalam sebuah dokumen dengan inverse document frequency (IDF), yaitu jumlah kemunculan kata dalam kumpulan dokumen.

Berikut ini adalah rumus yang digunakan dalam menghitung nilai *Inverse Document Frequency* (IDF) :

$$IDF = \log\left(\frac{N}{ni}\right) \quad (2.1)$$

Dalam perhitungan TF-IDF menggunakan persamaan 2.2 berikut :

$$TFIDF = TF \times IDF \quad (2.2)$$

Keterangan:

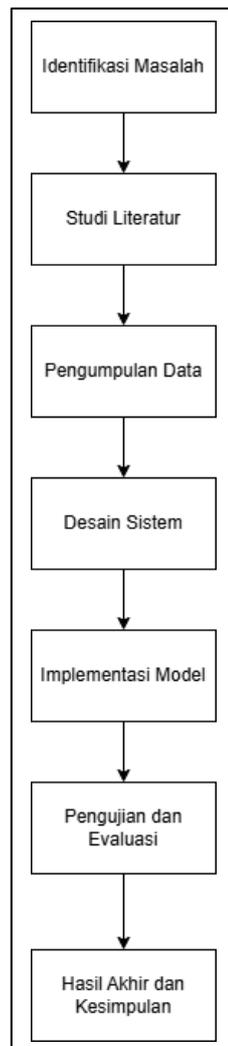
- TF = Jumlah kemunculan term (i) pada sebuah dokumen
- IDF = Jumlah (i) term pada seluruh dokumen
- N = Jumlah dokumen
- ni = Jumlah dokumen yang mengandung (i) term

BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian disusun untuk memberikan panduan yang jelas dalam pelaksanaan sebuah penelitian. Dengan desain penelitian yang terstruktur, setiap tahapan penelitian dapat dilakukan secara sistematis, sehingga hasil yang diperoleh lebih akurat dan sesuai dengan tujuan yang telah ditetapkan.



Gambar 3.1 Desain Penelitian

Gambar 3.1 menunjukkan tahapan-tahapan dalam penelitian ini. Langkah pertama adalah identifikasi masalah, di mana penulis menentukan permasalahan yang akan diteliti. Setelah itu, dilakukan studi literatur untuk mempelajari penelitian sebelumnya yang relevan. Selanjutnya, penulis melakukan pengumpulan data, yaitu mengambil data dari website Kaggle.com. Setelah data terkumpul, penelitian berlanjut ke desain sistem, di mana sistem dirancang untuk mengolah dan menganalisis data tersebut. Tahap berikutnya adalah implementasi model, di mana model diterapkan sesuai dengan kebutuhan penelitian. Setelah itu, dilakukan pengujian dan evaluasi untuk melihat kinerja model yang telah dibuat. Tahap terakhir adalah hasil akhir dan kesimpulan, yang berisi ringkasan dari seluruh proses penelitian dan temuan yang diperoleh.

3.2 Pengumpulan Data

Pada tahap pengumpulan data, penelitian ini menggunakan dataset ulasan pengguna iPhone yang diperoleh dari Kaggle. Data diambil dari sumber iPhone Customer Reviews NLP Dataset yang dapat diakses melalui tautan berikut: <https://www.kaggle.com/datasets/mrmars1010/iphone-customer-reviews-nlp>.

Dataset ini berisi 3.062 ulasan dari pengguna iPhone, yang mencerminkan berbagai opini terhadap produk tersebut. Data dikumpulkan dalam format .csv, dan terdiri dari beberapa 11 fitur.

Atribut-atribut dalam dataset ini menyajikan berbagai informasi penting yang diperlukan untuk menganalisis sentimen publik terhadap ulasan pengguna iPhone. Informasi ini menjadi dasar dalam proses analisis untuk memahami pola sentimen publik terhadap produk iPhone secara lebih komprehensif Tabel 3.1.

Tabel 3.1 Penjelasan Fitur

No	Atribut	Keterangan
1	productAsin	Kode unik yang diberikan untuk setiap produk iPhone di Amazon.
2	country	Negara asal pengguna yang memberikan ulasan.
3	date	Tanggal ketika ulasan diposting oleh pengguna.
4	isVerified	Status apakah ulasan berasal dari pembelian yang diverifikasi (True jika terverifikasi, False jika tidak).
5	ratingScore	Skor rating yang diberikan oleh pengguna terhadap produk iPhone (dalam skala 1–5).
6	reviewTitle	Judul yang ditulis oleh pengguna untuk ulasan mereka.
7	reviewDescription	Isi atau deskripsi lengkap dari ulasan pengguna mengenai produk iPhone.
8	reviewUrl	Tautan langsung menuju ulasan yang diposting di Amazon.
9	reviewedIn	Bahasa yang digunakan dalam ulasan tersebut.
10	variant	Varian produk iPhone yang diulas, seperti warna atau kapasitas penyimpanan.
11	variantAsin	Kode unik untuk varian spesifik dari produk iPhone tersebut.

Atribut-atribut ini memberikan informasi penting yang membantu dalam analisis sentimen terhadap ulasan pengguna iPhone. Ulasan tersebut sangat relevan untuk dieksplorasi guna memahami pola sentimen publik terhadap produk iPhone secara keseluruhan Tabel 3.2.

Tabel 3.2 Penyajian Contoh Data Utama

reviewTitle	reviewDescription	reviewUrl
Product	100% genuine	https://www.amazon.in/gp/customer-reviews/R1P9A1G2E3ZB7L/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98
Good product	Happy to get the iPhone 13 in Amazon offer	https://www.amazon.in/gp/customer-reviews/R1XI5GCQ8WH1LV/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98
Too smooth and effective battery life	5 star 	https://www.amazon.in/gp/customer-reviews/R3J3YN2SLDE72Y/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98
Apple 13 256 GB	Great Product... Nothing to say, as the Name APPLE is own identity of this	https://www.amazon.in/gp/customer-reviews/R2ZMQV9EEYDF4K/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98

reviewTitle	reviewDescription	reviewUrl
	product.. All is well, recommendation is 100% . Just grab this product else you missed something....	
Good	Good product 👍👍	https://www.amazon.in/gp/customer-reviews/R1XF1WDLPLST8I/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98
Good	Nice product	https://www.amazon.in/gp/customer-reviews/R2QA6GV3Q80RB2/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98
Excellent	Excellent product	https://www.amazon.in/gp/customer-reviews/R322E11GD7VP55/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9BQS98
Great	Loved it	https://www.amazon.in/gp/customer-reviews/RZQ10EXG69IQX/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9HRYFZ
I love it	Nice	https://www.amazon.in/gp/customer-reviews/R66RSI0SENEH2/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G9HRYFZ
Improved version of iPhone 11	Nothing new features only model changed	https://www.amazon.in/gp/customer-reviews/R2QO7SAC741QEC/ref=cm_cr_getr_d_rvw_ttl?ie=UTF8&ASIN=B09G93H3BR

Setelah proses pengumpulan data, penelitian ini dilanjutkan dengan tahap pelabelan secara manual untuk mengkategorikan ulasan berdasarkan sentimen yang terkandung di dalamnya. Karena dataset awalnya berisi ulasan dalam bahasa Inggris, maka dilakukan proses penerjemahan ke dalam bahasa Indonesia agar analisis sentimen lebih sesuai dengan konteks pengguna lokal.

Dalam penelitian ini, terdapat dua label sentimen yang digunakan, yaitu positif dan negatif. Proses pelabelan dilakukan dengan bantuan Mumtazza Awalia Rahmani S.S., yang berperan sebagai validator untuk memastikan bahwa setiap ulasan diklasifikasikan secara akurat sesuai dengan sentimen yang diekspresikan

oleh pengguna. Validasi ini bertujuan untuk meningkatkan akurasi dalam pengelompokan data sehingga hasil analisis sentimen dapat lebih representatif. Menurut (Arham dkk., 2022), sentimen positif mencerminkan opini yang mendukung atau bernada optimis, sentimen negatif menunjukkan ketidakpuasan atau kritik. Contoh hasil pelabelan komentar Tabel 3.3.

Tabel 3.3 Pelabelan Ulasan Secara Manual

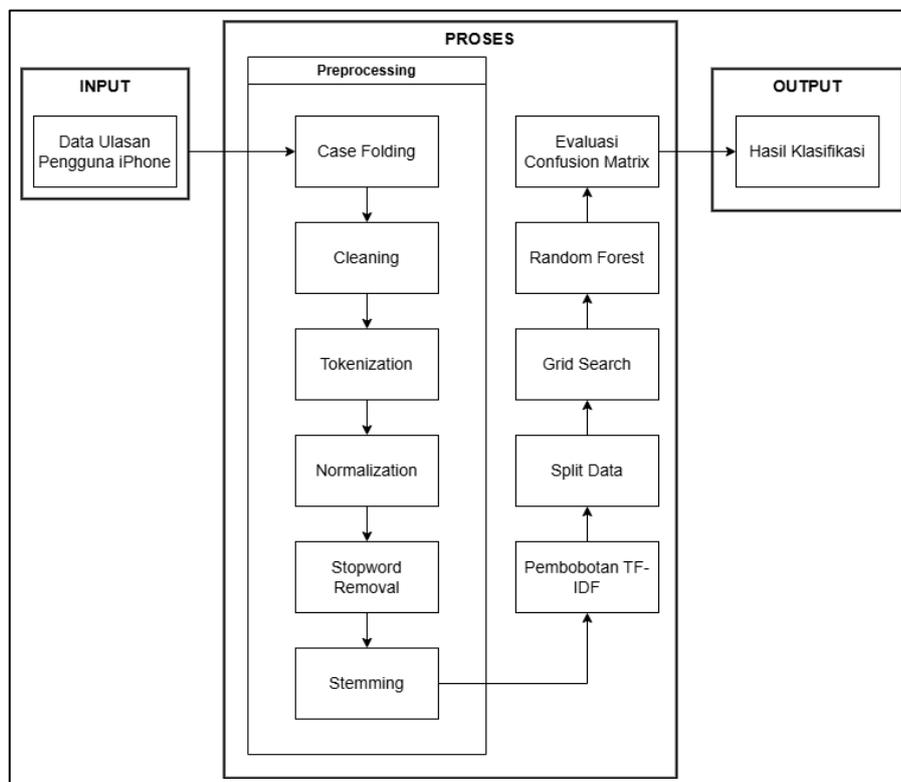
No	Ulasan	Label
1	100% asli	Positif
2	Senang mendapatkan iPhone 13 di penawaran Amazon	Positif
3	Panggilan video telepon sangat buruk dan berisik dan panggilan turun di antara percakapan. Cacat muncul nanti setelah jendela kembali ditutup! Pengalaman yang sangat buruk 😞	Negatif
4	Produk hebat ... tidak ada yang perlu dikatakan, karena nama Apple adalah identitas produk ini sendiri .. semuanya baik -baik saja, rekomendasi adalah 100%. Raih saja produk ini lagi, Anda melewatkan sesuatu	Positif
5	Perangkat saya tidak berfungsi (dimatikan) Saya tidak dapat mengunduh faktur dan garansi dokumenter. Dapatkah saya mendapatkan pusat layanan yang menanyakan dokumen	Negatif

Data yang telah dilabeli ini akan menjadi dasar untuk menerapkan metode *Random forest* dalam analisis sentimen pada tahap berikutnya. Melalui proses ini, penulis berupaya untuk mengidentifikasi tren, pola, dan kecenderungan sentimen publik terhadap produk iPhone berdasarkan ulasan pengguna. Dengan demikian, hasil penelitian ini diharapkan dapat memberikan gambaran yang lebih jelas mengenai opini masyarakat, terutama dalam mempertimbangkan keputusan pembelian berdasarkan ulasan yang terdapat di media sosial.

3.3 Desain Sistem

Desain sistem dalam penelitian ini dirancang untuk membangun sebuah platform analisis sentimen yang optimal dalam mengolah data ulasan produk

iPhone berdasarkan pengguna. Sistem ini dikembangkan menggunakan bahasa pemrograman *Python*. Gambaran keseluruhan dari desain sistem Gambar 3.2.



Gambar 3.2 Desain Sistem

Setiap tahapan dalam desain sistem akan diuraikan secara mendetail pada bagian berikut. Penjelasan ini bertujuan untuk memberikan pemahaman yang jelas mengenai proses yang dilakukan dalam pengembangan sistem ini.

3.3.1 *Preprocessing Text*

Penelitian ini menerapkan enam tahapan *Preprocessing* untuk menyiapkan data sebelum proses klasifikasi sentimen, yaitu *cleaning*, *case folding*, *tokenization*, *normalization*, *stopword removal*, dan *stemming*. Proses ini bertujuan untuk memastikan data lebih terstruktur dan siap digunakan dalam pemodelan. Pembahasan lebih lanjut mengenai proses ini dapat ditemukan pada bagian berikut.

3.3.1.1 Case Folding

Pada tahap *case folding*, semua huruf dalam teks diubah menjadi huruf kecil (*lowercase*). Langkah ini dilakukan untuk memastikan bahwa teks diproses dalam format yang seragam, menghilangkan perbedaan antara huruf besar dan kecil yang tidak relevan untuk analisis. Sebagai contoh, kata "Apple" akan diubah menjadi "apple", dan kata "Raih" akan menjadi "raih" seperti Tabel 3.4.

Tabel 3.4 Penerapan *Case folding*

No	Sebelum <i>Case folding</i>	Setelah <i>Case folding</i>
1	100% asli	100% asli
2	Senang mendapatkan iPhone 13 di penawaran Amazon	senang mendapatkan iphone 13 di penawaran amazon
3	Panggilan video telepon sangat buruk dan berisik dan panggilan turun di antara percakapan. Cacat muncul nanti setelah jendela kembali ditutup! Pengalaman yang sangat buruk 😞	panggilan video telepon sangat buruk dan berisik dan panggilan turun di antara percakapan. cacat muncul nanti setelah jendela kembali ditutup! pengalaman yang sangat buruk 😞
4	Produk hebat ... tidak ada yang perlu dikatakan, karena nama Apple adalah identitas produk ini sendiri .. semuanya baik -baik saja, rekomendasi adalah 100%. Raih saja produk ini lagi, Anda melewatkan sesuatu	produk hebat ... tidak ada yang perlu dikatakan, karena nama apple adalah identitas produk ini sendiri .. semuanya baik -baik saja, rekomendasi adalah 100%. raih saja produk ini lagi, anda melewatkan sesuatu
5	Perangkat saya tidak berfungsi (dimatikan) Saya tidak dapat mengunduh faktur dan garansi dokumenter. Dapatkah saya mendapatkan pusat layanan yang menanyakan dokumen	perangkat saya tidak berfungsi (dimatikan) saya tidak dapat mengunduh faktur dan garansi dokumenter. dapatkah saya mendapatkan pusat layanan yang menanyakan dokumen

3.3.1.2 Cleaning

Proses *cleaning* bertujuan untuk menghapus elemen-elemen yang tidak relevan dalam teks, seperti simbol, emoji, dan tanda baca. Hal ini dilakukan agar data menjadi lebih bersih dan siap untuk diproses lebih lanjut seperti Tabel 3.5.

Tabel 3.5 Penerapan *Cleaning*

No	Sebelum <i>Cleaning</i>	Setelah <i>Cleaning</i>
1	100% asli	100 asli
2	senang mendapatkan iphone 13 di penawaran amazon	senang mendapatkan iphone 13 di penawaran amazon
3	panggilan video telepon sangat buruk dan berisik dan panggilan turun di antara	panggilan video telepon sangat buruk dan berisik dan panggilan turun di antara

No	Sebelum <i>Cleaning</i>	Setelah <i>Cleaning</i>
	percakapan. cacat muncul nanti setelah jendela kembali ditutup! pengalaman yang sangat buruk 😞	percakapan cacat muncul nanti setelah jendela kembali ditutup pengalaman yang sangat buruk
4	produk hebat ... tidak ada yang perlu dikatakan, karena nama apple adalah identitas produk ini sendiri .. semuanya baik -baik saja, rekomendasi adalah 100%. raih saja produk ini lagi, anda melewatkan sesuatu	produk hebat tidak ada yang perlu dikatakan karena nama apple adalah identitas produk ini sendiri semuanya baik baik saja rekomendasi adalah 100 raih saja produk ini lagi anda melewatkan sesuatu
5	perangkat saya tidak berfungsi (dimatikan) saya tidak dapat mengunduh faktur dan garansi dokumenter. dapatkah saya mendapatkan pusat layanan yang menanyakan dokumen	perangkat saya tidak berfungsi dimatikan saya tidak dapat mengunduh faktur dan garansi dokumenter dapatkah saya mendapatkan pusat layanan yang menanyakan dokumen

3.3.1.3 Tokenization

Proses *tokenization* bertujuan untuk memecah teks menjadi unit-unit kecil yang disebut *token*, biasanya berupa kata-kata atau frasa pendek. Dengan proses ini, teks yang sebelumnya berupa kalimat utuh akan diubah menjadi daftar kata yang berdiri sendiri, sehingga memudahkan analisis pada tahap berikutnya Tabel 3.6.

Tabel 3.6 Penerapan *Tokenization*

No	Sebelum <i>Tokenization</i>	Setelah <i>Tokenization</i>
1	100 asli	["100", "asli"]
2	senang mendapatkan iphone 13 di penawaran amazon	["senang", "mendapatkan", "iphone", "13", "di", "penawaran", "amazon"]
3	panggilan video telepon sangat buruk dan berisik dan panggilan turun di antara percakapan cacat muncul nanti setelah jendela kembali ditutup pengalaman yang sangat buruk	["panggilan", "video", "telepon", "sangat", "buruk", "dan", "berisik", "dan", "panggilan", "turun", "di", "antara", "percakapan", "cacat", "muncul", "nanti", "setelah", "jendela", "kembali", "ditutup", "pengalaman", "yang", "sangat", "buruk"]
4	produk hebat tidak ada yang perlu dikatakan karena nama apple adalah identitas produk ini sendiri semuanya baik baik saja rekomendasi adalah 100 raih saja produk ini lagi anda melewatkan sesuatu	["produk", "hebat", "tidak", "ada", "yang", "perlu", "dikatakan", "karena", "nama", "apple", "adalah", "identitas", "produk", "ini", "sendiri", "semuanya", "baik", "baik", "saja", "rekomendasi", "adalah", "100", "raih", "saja", "produk", "ini", "lagi", "anda", "melewatkan", "sesuatu"]
5	perangkat saya tidak berfungsi dimatikan saya tidak dapat mengunduh faktur dan garansi dokumenter dapatkah saya mendapatkan pusat layanan yang menanyakan dokumen	["perangkat", "saya", "tidak", "berfungsi", "dimatikan", "saya", "tidak", "dapat", "mengunduh", "faktur", "dan", "garansi", "dokumenter", "dapatkah", "saya", "mendapatkan", "pusat", "layanan", "yang", "menanyakan", "dokumen"]

3.3.1.4 Normalization

Tahap *normalization* bertujuan untuk mengubah teks yang tidak baku atau berisi slang menjadi bentuk yang lebih baku, sehingga dapat dipahami oleh model dan meningkatkan kualitas analisis sentimen. Proses ini dilakukan dengan cara mengganti kata-kata tidak baku yang terdapat dalam file *colloquial-indonesian-lexicon.csv* (Nasalsabila, t.t.). Kamus ini berisi pasangan kata slang dan kata baku yang sesuai. Kata tidak baku yang ditemukan dalam teks akan diganti dengan kata baku, sementara kata yang tidak terdapat dalam kamus tetap dipertahankan Tabel 3.7.

Tabel 3.7 Penerapan *Normalization*

No	Sebelum <i>Normalization</i>	Setelah <i>Normalization</i>
1	["100", "asli"]	["100", "asli"]
2	["senang", "mendapatkan", "iphone", "13", "di", "penawaran", "amazon"]	["senang", "mendapat", "iphone", "13", "di", "penawaran", "amazon"]
3	["panggilan", "video", "telepon", "sangat", "buruk", "dan", "berisik", "dan", "panggilan", "turun", "di", "antara", "percakapan", "cacat", "muncul", "nanti", "setelah", "jendela", "kembali", "ditutup", "pengalaman", "yang", "sangat", "buruk"]	["panggilan", "video", "telepon", "sangat", "buruk", "dan", "berisik", "dan", "panggilan", "turun", "di", "antara", "percakapan", "cacat", "muncul", "nanti", "setelah", "jendela", "kembali", "tutup", "pengalaman", "yang", "sangat", "buruk"]
4	["produk", "hebat", "tidak", "ada", "yang", "perlu", "dikatakan", "karena", "nama", "apple", "adalah", "identitas", "produk", "ini", "sendiri", "semuanya", "baik", "baik", "saja", "rekomendasi", "adalah", "100", "raih", "saja", "produk", "ini", "lagi", "anda", "melewatkan", "sesuatu"]	["produk", "hebat", "tidak", "ada", "yang", "perlu", "katakan", "karena", "nama", "apple", "adalah", "identitas", "produk", "ini", "sendiri", "semua", "baik", "baik", "saja", "rekomendasi", "adalah", "100", "raih", "saja", "produk", "ini", "lagi", "anda", "lewat", "sesuatu"]
5	["perangkat", "saya", "tidak", "berfungsi", "dimatikan", "saya", "tidak", "dapat", "mengunduh", "faktur", "dan", "garansi", "dokumenter", "dapatkah", "saya", "mendapatkan", "pusat", "layanan", "yang", "menanyakan", "dokumen"]	["perangkat", "saya", "tidak", "fungsi", "mati", "saya", "tidak", "dapat", "unduh", "faktur", "dan", "garansi", "dokumentasi", "dapatkah", "saya", "dapat", "pusat", "layanan", "yang", "tanya", "dokumen"]

3.3.1.5 Stopword Removal

Tahap *Stopword removal* adalah proses menghapus kata-kata yang dianggap tidak memberikan informasi penting atau tidak memengaruhi analisis dalam data

teks. Stopword biasanya berupa kata-kata umum yang sering muncul dalam teks, seperti kata sambung, kata depan, atau kata ganti (misalnya: "ini", "karena", "dan", "saya", "itu", "ya", dll.) Tabel 3.8.

Tabel 3.8 Penerapan *Stopword removal*

No	Sebelum <i>Stopword removal</i>	Setelah <i>Stopword removal</i>
1	["100", "asli"]	["100", "asli"]
2	["senang", "mendapat", "iphone", "13", "di", "penawaran", "amazon"]	["senang", "mendapat", "iphone", "13", "penawaran", "amazon"]
3	["panggilan", "video", "telepon", "sangat", "buruk", "dan", "berisik", "dan", "panggilan", "turun", "di", "antara", "percakapan", "cacat", "muncul", "nanti", "setelah", "jendela", "kembali", "tutup", "pengalaman", "yang", "sangat", "buruk"]	["panggilan", "video", "telepon", "buruk", "berisik", "panggilan", "turun", "percakapan", "cacat", "muncul", "jendela", "tutup", "pengalaman", "buruk"]
4	["produk", "hebat", "tidak", "ada", "yang", "perlu", "katakan", "karena", "nama", "apple", "adalah", "identitas", "produk", "ini", "sendiri", "semua", "baik", "baik", "saja", "rekomendasi", "adalah", "100", "raih", "saja", "produk", "ini", "lagi", "anda", "lewat", "sesuatu"]	["produk", "hebat", "katakan", "nama", "apple", "identitas", "produk", "sendiri", "baik", "rekomendasi", "100", "raih", "produk", "lagi", "lewat", "sesuatu"]
5	["perangkat", "saya", "tidak", "fungsi", "mati", "saya", "tidak", "dapat", "unduh", "faktur", "dan", "garansi", "dokumentasi", "dapatkah", "saya", "dapat", "pusat", "layanan", "yang", "tanya", "dokumen"]	["perangkat", "fungsi", "mati", "unduh", "faktur", "garansi", "dokumentasi", "dapat", "pusat", "layanan", "tanya", "dokumen"]

3.3.1.6 *Stemming*

Stemming adalah proses penting dalam pengolahan teks untuk mengubah kata-kata menjadi bentuk dasar atau kata dasar (*root word*). Proses ini bertujuan untuk menyederhanakan analisis dengan mengeliminasi variasi bentuk kata yang memiliki makna serupa. Dengan kata lain, kata-kata seperti "menyukai", "disukai", atau "penyuka" akan diubah menjadi kata dasar "suka" pada Tabel 3.9.

Tabel 3.9 Penerapan *Stemming*

No	Sebelum <i>Stemming</i>	Setelah <i>Stemming</i>
1	["100", "asli"]	["100", "asli"]
2	["senang", "mendapat", "iphone", "13", "penawaran", "amazon"]	["senang", "dapat", "iphone", "13", "tawar", "amazon"]
3	["panggilan", "video", "telepon", "buruk", "berisik", "panggilan", "turun", "cakup"]	["panggil", "video", "telepon", "buruk", "berisik", "panggil", "turun", "cakap"]

No	Sebelum Stemming	Setelah Stemming
	"percakapan", "cacat", "muncul", "jendela", "tutup", "pengalaman", "buruk"]	"cacat", "muncul", "jendela", "tutup", "pengalaman", "buruk"]
4	["produk", "hebat", "katakan", "nama", "apple", "identitas", "produk", "sendiri", "baik", "rekomendasi", "100", "raih", "produk", "lagi", "lewat", "sesuatu"]	["produk", "hebat", "kata", "nama", "apple", "identitas", "produk", "diri", "baik", "rekomendasi", "100", "raih", "produk", "lagi", "lewat", "suatu"]
5	["perangkat", "fungsi", "mati", "unduh", "faktur", "garansi", "dokumentasi", "dapat", "pusat", "layanan", "tanya", "dokumen"]	["perangkat", "fungsi", "mati", "unduh", "faktur", "garansi", "dokumentasi", "dapat", "pusat", "layan", "tanya", "dokumen"]

3.3.2 Pembobotan TF-IDF

Selanjutnya, dilakukan pembobotan kata menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk menentukan frekuensi kemunculan suatu term dalam setiap dokumen. Term Frequency (TF) menghitung probabilitas kemunculan sebuah term dalam dokumen tertentu, seperti dokumen D1 hingga D5. Langkah-langkah proses pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah sebagai berikut:

- a. Menghitung frekuensi kemunculan setiap kata *Term Frequency* (TF) pada setiap dokumen, seperti yang ditampilkan Tabel 3.10.

Tabel 3.10 *Term Frequency*

Kata	TF (Term Frequency)				
	D1	D2	D3	D4	D5
iphone	0	1	0	0	0
buruk	0	0	2	0	0
produk	0	0	0	2	0
layanan	0	0	0	0	1
dokumen	0	0	0	0	1
panggilan	0	0	2	0	0
rekomendasi	0	0	0	1	0
garansi	0	0	0	0	1
pengalaman	0	0	1	0	0
pusat	0	0	0	0	1

- b. Menghitung DF (Document Frequency) Tabel 3.11.

Tabel 3.11 Menghitung DF (*Document Frequency*)

Kata	TF (Term Frequency)					DF
	D1	D2	D3	D4	D5	
iphone	0	1	0	0	0	1
buruk	0	0	2	0	0	1
produk	0	0	0	2	0	1
layanan	0	0	0	0	1	1
dokumen	0	0	0	0	1	1
panggilan	0	0	2	0	0	1
rekomendasi	0	0	0	1	0	1
garansi	0	0	0	0	1	1
pengalaman	0	0	1	0	0	1
pusat	0	0	0	0	1	1

c. Menghitung IDF (Inverse Document Frequency) Tabel 3.12 menggunakan persamaan 2.1.

Tabel 3.12 Menghitung IDF (*Inverse Document Frequency*)

Kata	IDF
	$\log\left(\frac{N}{ni}\right)$
iphone	$\log\left(\frac{5}{1}\right) = 0.699$
buruk	$\log\left(\frac{5}{1}\right) = 0.699$
produk	$\log\left(\frac{5}{1}\right) = 0.699$
layanan	$\log\left(\frac{5}{1}\right) = 0.699$
dokumen	$\log\left(\frac{5}{1}\right) = 0.699$
panggilan	$\log\left(\frac{5}{1}\right) = 0.699$
rekomendasi	$\log\left(\frac{5}{1}\right) = 0.699$
garansi	$\log\left(\frac{5}{1}\right) = 0.699$
pengalaman	$\log\left(\frac{5}{1}\right) = 0.699$
pusat	$\log\left(\frac{5}{1}\right) = 0.699$

- d. Melakukan perhitungan bobot TF-IDF pada Tabel 3.13 dengan menggunakan rumus yang tercantum pada Persamaan 2.2.

Tabel 3.13 *Term Frequency - Inverse Document Frequency*

Kata	TF-IDF(TERM FREQUENCY)				
	D1	D2	D3	D4	D5
iphone	0.00	0.699	0.00	0.00	0.00
buruk	0.00	0.00	1.398	0.00	0.00
produk	0.00	0.00	0.00	1.398	0.00
layanan	0.00	0.00	0.00	0.00	0.699
dokumen	0.00	0.00	0.00	0.00	0.699
panggilan	0.00	0.00	1.398	0.00	0.00
rekomendasi	0.00	0.00	0.00	0.699	0.00
garansi	0.00	0.00	0.00	0.00	0.699
pengalaman	0.00	0.00	0.699	0.00	0.00
pusat	0.00	0.00	0.00	0.00	0.699

3.3.3 *Split Data*

Tahapan penting selanjutnya adalah data dibagi menjadi dua bagian, satu bagian digunakan untuk pelatihan (*training*) dan bagian lainnya untuk pengujian (*testing*). Untuk rasio pembagian data, pada penelitian ini digunakan tiga variasi yaitu 80:20, 70:30, dan 60:40. Rasio pembagian data ini digunakan untuk mengevaluasi bagaimana perubahan dalam pembagian data mempengaruhi kinerja model (Bichri dkk., 2024).

3.3.4 *Grid Search*

Pemilihan *hyperparameter* yang optimal berperan penting dalam meningkatkan performa model *Random forest*. Salah satu metode yang digunakan dalam penelitian ini adalah *Grid Search*, yang secara sistematis mengeksplorasi kombinasi *hyperparameter* terbaik berdasarkan metrik evaluasi tertentu. Menurut

(G & B.Sumathi, 2020), proses *Grid Search* melibatkan pengujian semua kombinasi yang telah ditentukan serta evaluasi model menggunakan metrik seperti akurasi untuk menentukan konfigurasi optimal :

1. Membagi dataset menjadi *training set* dan *testing set* untuk mengevaluasi performa model.
2. Mendefinisikan *Random forest Classifier* sebagai model utama.
3. Menentukan grid *hyperparameter* yang akan diuji.
4. Mendefinisikan fungsi *Grid Search*, yang menerima model, parameter grid, dan dataset sebagai input.
5. Mendefinisikan fungsi objektif, yang mengevaluasi kombinasi *hyperparameter* berdasarkan akurasi model.

$$accuracy = f(hyperparameters) \quad (3.1)$$

6. Menjalankan iterasi untuk mencoba setiap kombinasi.
7. Menggunakan setiap kombinasi *hyperparameter* dalam fungsi objektif dan mengukur performanya.
8. Menentukan kombinasi *hyperparameter* terbaik yang memberikan akurasi tertinggi pada dataset uji.

$$hyperparams *= \arg \max f(hyperparams) \quad (3.2)$$

3.3.4.1 Parameter *Grid Search*

Parameter yang digunakan dalam proses *Grid Search* pada penelitian ini ditampilkan pada Tabel 3.14.

Tabel 3.14 Ruang Pencarian *Grid Search*

Parameter	Value 1	Value 2	Value 3
<i>n_estimators</i>	200	400	600

Parameter	Value 1	Value 2	Value 3
<i>max_features</i>	sqrt	log2	–
<i>max_depth</i>	600	800	None
<i>min_samples_split</i>	2	5	10
<i>min_samples_leaf</i>	1	2	4

Berikut adalah contoh perhitungan manual untuk proses *Grid Search* pada

Random forest :

1. Terdapat data contoh hasil TF-IDF pada Tabel 3.15

Tabel 3.15 Contoh Hasil TF-IDF

Kata	D1 (Positif)	D2 (Positif)	D3 (Negatif)	D4 (Positif)	D5 (Negatif)
iphone	0.000	0.699	0.000	0.000	0.000
buruk	0.000	0.000	1.398	0.000	0.000
produk	0.000	0.000	0.000	1.398	0.000
layanan	0.000	0.000	0.000	0.000	0.699
dokumen	0.000	0.000	0.000	0.000	0.699
panggilan	0.000	0.000	1.398	0.000	0.000
rekomendasi	0.000	0.000	0.000	0.699	0.000
garansi	0.000	0.000	0.000	0.000	0.699
pengalaman	0.000	0.000	0.699	0.000	0.000
pusat	0.000	0.000	0.000	0.000	0.699

Dataset ini akan dibagi menjadi 3 fold

Fold 1 :

D1, D3 Testing

D2, D4, D5 Training

Fold 2 :

D2, D4 Testing

D1, D3, D5 Training

Fold 3 :

D5 Testing

D1, D2, D3, D4 Training

2. Kombinasi *Grid Hyperparameter*

Dari Tabel 3.14, terdapat total 162 kombinasi *hyperparameter*, Tabel 3.16 mencontohkan 8 kombinasi dari total 162 kombinasi *hyperparameter*.

Tabel 3.16 Contoh Kombinasi Parameter

<i>n_estimators</i>	<i>max_features</i>	<i>max_depth</i>	<i>min_samples_split</i>	<i>min_samples_leaf</i>
200	sqrt	600	2	1
200	sqrt	600	2	2
200	sqrt	600	2	4
200	sqrt	600	5	1
200	log2	800	5	2
400	sqrt	None	10	4
600	log2	800	5	1
600	sqrt	600	2	1

3. Perhitungan *Grid Search* dengan *K-Fold*

Untuk setiap kombinasi *hyperparameter*, model dilatih pada training set dan diuji pada testing set. Sebagai contoh, hasil akurasi untuk setiap kombinasi pada Tabel 3.17.

Tabel 3.17 Contoh Perhitungan *K-Fold* Kombinasi *Hyperparameter*

<i>n_estimators</i>	<i>max_features</i>	<i>max_depth</i>	<i>min_samples_split</i>	<i>min_samples_leaf</i>	Avg Acc (%)
200	sqrt	600	2	1	86%
200	sqrt	600	2	2	84%
200	log2	800	5	1	87%
400	sqrt	800	5	2	89%
400	log2	None	10	4	90%
600	sqrt	None	5	1	89%
600	log2	600	2	2	86%
600	sqrt	800	10	1	92%

4. Menentukan *Hyperparameter* Terbaik

Berdasarkan contoh hasil evaluasi, akan ditentukan *hyperparameter* terbaik dengan akurasi tertinggi 92% diperoleh dengan $n_estimators = 400$, $max_features = \text{sqrt}$, $max_depth = 800$, $min_sample_split = 10$, dan $min_sample_leaf = 1$ pada contoh evaluasi model, maka ini adalah contoh kombinasi *hyperparameter* terbaik.

3.3.5 Pemodelan *Random forest*

Setelah dilakukan proses pencarian parameter terbaik, maka selanjutnya membangun model *Random forest* berdasarkan parameter terbaik. Algoritma ini bekerja dengan membuat banyak *Decision Tree* selama proses pelatihan, dan hasil akhir ditentukan berdasarkan voting mayoritas dari prediksi masing-masing pohon. Berikut adalah tahapan dalam algoritma *Random forest* (Jackins dkk., 2021):

1. *Bootstrap Sampling*, dari dataset latih yang tersedia, algoritma mengambil sampel secara acak dengan pengembalian (*bagging*) untuk membentuk subset data yang berbeda bagi setiap *Decision Tree*. Dalam proses klasifikasi, setiap pohon keputusan dalam *Random forest* tidak menggunakan semua fitur yang tersedia, tetapi hanya memilih \sqrt{m} fitur secara acak, di mana m adalah jumlah total prediktor (fitur) yang tersedia dalam dataset (T. Zhu, 2020).
2. Pembangunan *Decision Tree*, setiap pohon keputusan dilatih menggunakan subset data hasil *bootstrap sampling*. Dalam proses ini, pemisahan (*splitting*) pada setiap node dilakukan dengan memilih fitur terbaik

berdasarkan *Gini Index* menggunakan persamaan berikut (Suci Amaliah dkk., 2022):

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

Keterangan :

p_i : probabilitas setiap kelas pada node
 c : jumlah kelas

3. Dari fitur-fitur yang dipilih secara acak tersebut, selanjutnya dilakukan pencarian titik pemisah terbaik dengan menggunakan kriteria Gini. Berikut adalah persamaan yang digunakan (Suci Amaliah dkk., 2022) :

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n}\right) Gini(S_i) \quad (3.4)$$

Keterangan :

c : jumlah kelas
 n_i : jumlah sampel dalam subset
 n : jumlah sampel di node yang diberikan

4. Proses pemilihan fitur dan pencarian titik pemisah terbaik akan terus berlanjut untuk setiap node dalam pohon hingga pohon tersebut sepenuhnya terbentuk.
5. Setelah semua pohon dalam *Random forest* selesai dilatih, hasil prediksi akhir ditentukan menggunakan voting mayoritas melalui persamaan berikut:

$$\hat{y} = mode\{h_1(x), h_2(x), \dots, h_b(x)\} \quad (3.5)$$

Keterangan :

\hat{y} : kelas prediksi akhir untuk data x
 $h_b(x)$: prediksi kelas dari pohon keputusan ke- b untuk data x
 $mode$: kelas dengan jumlah suara terbanyak

Contoh perhitungan untuk membangun pohon keputusan dalam *Random forest* adalah sebagai berikut:

1. Terdapat data hasil dari TF-IDF pada Tabel 3.15
2. Membagi Data TF-IDF dengan *Bootstrap* dengan nilai 5 pada $n_estimator$ sebagai contoh nilai terbaiknya

Subset 1:

$$X1 = [D1, D3, D5]$$

$$y1 = [\text{Positif}, \text{Negatif}, \text{Negatif}]$$

Subset 2:

$$X2 = [D2, D4, D5]$$

$$y2 = [\text{Positif}, \text{Positif}, \text{Negatif}]$$

Subset 3:

$$X3 = [D1, D2, D3]$$

$$y3 = [\text{Positif}, \text{Positif}, \text{Negatif}]$$

Subset 4:

$$X4 = [D2, D3, D5]$$

$$y4 = [\text{Positif}, \text{Negatif}, \text{Negatif}]$$

Subset 5:

$$X5 = [D1, D4, D5]$$

$$y5 = [\text{Positif}, \text{Positif}, \text{Negatif}]$$

Dengan $max_features = "sqrt"$ sebagai contoh nilai terbaik, berdasarkan Tabel 3.15 Contoh Hasil TF-IDF, terdapat 10 fitur kata, maka setiap pohon

hanya akan menggunakan $\sqrt{10} \approx 3$ fitur secara acak dalam proses pemisahan.

3. Menentukan Split Terbaik dengan *Gini Index*

Menggunakan $min_samples_split = 2$ sebagai contoh nilai terbaik, artinya setiap node harus memiliki minimal 2 sampel sebelum dapat dibagi.

$$Gini(D1) = 1 - \left(\left(\frac{1}{3} \right)^2 + \left(\frac{2}{3} \right)^2 \right) = 1 - \frac{5}{9} = 0.44$$

$$Gini(D2) = 1 - \left(\left(\frac{2}{3} \right)^2 + \left(\frac{1}{3} \right)^2 \right) = 1 - \frac{5}{9} = 0.44$$

$$Gini(D3) = 1 - \left(\left(\frac{2}{3} \right)^2 + \left(\frac{1}{3} \right)^2 \right) = 1 - \frac{5}{9} = 0.44$$

$$Gini(D4) = 1 - \left(\left(\frac{2}{3} \right)^2 + \left(\frac{1}{3} \right)^2 \right) = 1 - \frac{5}{9} = 0.44$$

$$Gini(D5) = 1 - \left(\left(\frac{2}{3} \right)^2 + \left(\frac{1}{3} \right)^2 \right) = 1 - \frac{5}{9} = 0.44$$

4. Membangun Pohon Keputusan

Menggunakan $max_depth = 5$ sebagai contoh nilai terbaik, yang berarti setiap pohon maksimal memiliki kedalaman 5 level. Karena $min_samples_leaf = 1$ sebagai contoh nilai terbaik, maka setiap *leaf node* minimal harus memiliki 1 sampel sebelum menjadi keputusan akhir.

Pohon Keputusan 1 (Subset 1)

Jika nilai TF-IDF buruk ≥ 1 , maka Negatif

Jika nilai TF-IDF iphone ≥ 0.5 , maka Positif

Jika tidak memenuhi kedua aturan di atas, maka Negatif

Pohon Keputusan 2 (Subset 2)

Jika nilai TF-IDF produk ≥ 1 , maka Positif

Jika nilai TF-IDF garansi ≥ 0.5 , maka Negatif

Jika tidak memenuhi kedua aturan di atas, maka Positif

Pohon Keputusan 3 (Subset 3)

Jika nilai TF-IDF panggilan ≥ 1 , maka Negatif

Jika nilai TF-IDF iphone ≥ 0.5 , maka Positif

Jika tidak memenuhi kedua aturan di atas, maka Positif

Pohon Keputusan 4 (Subset 4)

Jika nilai TF-IDF buruk ≥ 1 , maka Negatif

Jika nilai TF-IDF rekomendasi ≥ 0.5 , maka Positif

Jika tidak memenuhi kedua aturan di atas, maka Negatif

Pohon Keputusan 5 (Subset 5)

Jika nilai TF-IDF produk ≥ 1 , maka Positif

Jika nilai TF-IDF pusat ≥ 0.5 , maka Negatif

Jika tidak memenuhi kedua aturan di atas, maka Positif

5. Prediksi Data Baru

Misal terdapat data baru dengan hasil TF-IDF Tabel 3.18

Tabel 3.18 Contoh Hasil TF-IDF Data Baru

Kata	Nilai TF-IDF
iphone	0.6
buruk	0.0
produk	1.2
layanan	0.3
dokumen	0.1
panggilan	0.4
rekomendasi	0.7
garansi	0.2
pengalaman	0.0
pusat	0.5

Prediksi berdasarkan 5 Pohon Keputusan:

Pohon 1: $\text{iphone} \geq 0.5 \rightarrow \text{Positif}$

Pohon 2: $\text{produk} \geq 1 \rightarrow \text{Positif}$

Pohon 3: $\text{iphone} \geq 0.5 \rightarrow \text{Positif}$

Pohon 4: $\text{rekomendasi} \geq 0.5 \rightarrow \text{Positif}$

Pohon 5: $\text{produk} \geq 1 \rightarrow \text{Positif}$

6. Voting Akhir

Hasil prediksi kelas untuk data baru berdasarkan dua pohon keputusan :

$$\hat{y} = \text{mode}\{\text{Positif}, \text{Positif}, \text{Positif}, \text{Positif}, \text{Positif}\} = \text{Positif}$$

Jadi, data baru Tabel 3.18 diklasifikasikan sebagai positif karena itu adalah kelas yang paling sering muncul.

3.3.6 Evaluasi *Confusion Matrix*

Confusion matrix dipilih untuk mengevaluasi kinerja algoritma *Random forest* dalam penelitian ini. Teknik ini akan menghitung nilai *accuracy*, *precision*, *recall*, dan *f1 score* dari model *machine learning* yang telah dibangun. Terdapat empat kasus pada kondisi tertentu dalam perhitungan *confusion matrix* (Fahmy Amin, 2022) Tabel 3.19.

Tabel 3.19 Kondisi *Confusion matrix*

		Predict Values	
		1	0
Aktual Values	1	True Positif (TP)	False Negatif (FN)
	0	False Positif (FP)	True Negatif (TN)

True Positive (TP) terjadi saat prediksi dan data aktual sesuai. *True Negative* (TN) menunjukkan prediksi benar meskipun data aktual tidak sesuai. *False Positive* (FP) terjadi ketika prediksi salah meskipun data aktual benar, sedangkan *False Negative* (FN) berarti baik prediksi maupun data aktual salah (Chicco & Jurman, 2022). Dalam konteks ini, *True Positive* (TP) mengacu pada jumlah komentar yang memiliki sentimen positif secara aktual dan diprediksi oleh model sebagai komentar positif. *True Negative* (TN) merupakan jumlah komentar yang memiliki sentimen negatif secara aktual dan juga diprediksi oleh model sebagai komentar negatif. *False Positive* (FP) terjadi ketika komentar yang memiliki sentimen negatif secara aktual diprediksi oleh model sebagai komentar positif. Sebaliknya, *False Negative* (FN) terjadi ketika komentar yang memiliki sentimen positif secara aktual diprediksi oleh model sebagai komentar negatif.

Accuracy mengukur sejauh mana model memberikan prediksi yang benar secara keseluruhan. Nilai ini dihitung sebagai rasio antara jumlah prediksi benar (positif dan negatif) dengan total prediksi. Semakin tinggi *accuracy*, semakin baik kinerja model dalam menghasilkan prediksi yang tepat. Perhitungan *accuracy* ini dapat dilakukan dengan menggunakan rumus sebagai berikut (Kurnianto & Febriawan, 2023) :

$$Accuracy = \frac{(TP + TN)}{TP + TN + FN + FP} \times 100\% \quad (3.6)$$

Precision mengukur keakuratan prediksi positif model, dihitung sebagai rasio antara prediksi positif yang benar dengan total prediksi positif. Nilai *precision* yang tinggi menunjukkan bahwa model lebih andal dalam mengidentifikasi kategori

positif. Nilai *precision* dapat dihitung menggunakan rumus sebagai berikut (Zulqornain dkk., 2021) :

$$Precision = \frac{(TP)}{TP + FP} \times 100\% \quad (3.7)$$

Recall mengukur kemampuan model dalam mengidentifikasi seluruh data positif yang sebenarnya. Nilai ini dihitung dengan membandingkan jumlah prediksi positif yang benar terhadap total data positif yang tersedia. Semakin tinggi *recall*, semakin baik model dalam mendeteksi kelas positif. Nilai *recall* dapat dihitung dengan persamaan sebagai berikut (Zulqornain dkk., 2021) :

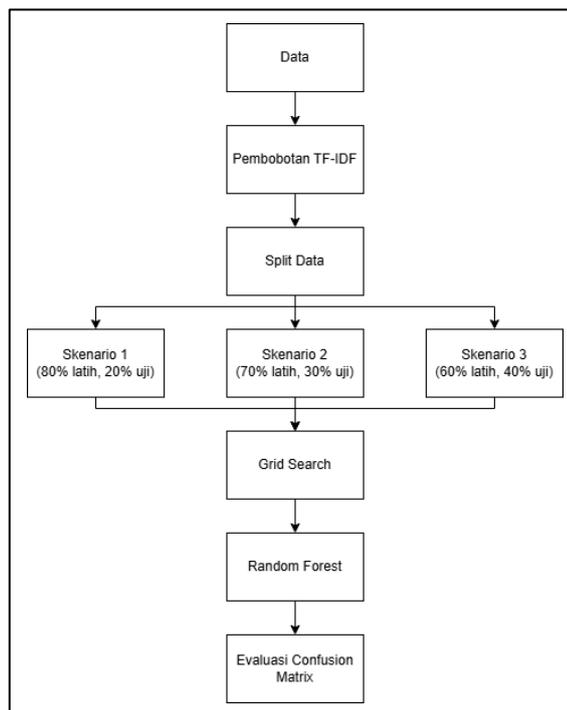
$$Recall = \frac{(TP)}{TP + FN} \times 100\% \quad (3.8)$$

F1 Score menggabungkan *precision* dan *recall* dalam satu metrik untuk menilai keseimbangan keduanya. Dihitung sebagai rata-rata harmonik dari *precision* dan *recall*, *f1 score* memberikan bobot yang sama pada keduanya. Metrik ini sangat penting, terutama saat terdapat ketidakseimbangan kelas dalam data, karena membantu mengevaluasi kinerja model secara lebih adil. Persamaan yang digunakan adalah sebagai berikut (Zulqornain dkk., 2021) :

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (3.9)$$

3.4 Skenario Pengujian

Pengujian dalam penelitian ini dilakukan untuk mengevaluasi kinerja algoritma *Random forest* dalam klasifikasi sentimen publik terhadap ulasan pengguna terhadap produk iPhone. Gambar menampilkan skenario pengujian.



Gambar 3.3 Skenario Pengujian

Gambar 3.3 menunjukkan alur skenario pengujian yang digunakan dalam penelitian ini. Proses dimulai dari data yang telah melalui tahap pembobotan menggunakan TF-IDF, kemudian dilakukan proses pembagian data menjadi tiga skenario berbeda, yaitu skenario 1 dengan pembagian 80% data latih dan 20% data uji, skenario 2 dengan 70% data latih dan 30% data uji, serta skenario 3 dengan 60% data latih dan 40% data uji. Masing-masing skenario data latih kemudian digunakan dalam proses pencarian parameter terbaik menggunakan *Grid Search*. Setelah parameter terbaik diperoleh, model *Random forest* dilatih menggunakan parameter tersebut, dan hasilnya dievaluasi menggunakan *confusion matrix* untuk menilai performa model pada masing-masing skenario. Tabel 3.20 menunjukkan skenario pengujian berdasarkan rasio pembagian data yang digunakan.

Tabel 3.20 Skenario Pengujian

Skenario	Split Ratio	Kombinasi <i>Hyperparameter Random forest</i>
1	80:20	$n_estimators = [200, 400, 600]$ $max_features = ['sqrt', 'log2']$ $max_depth = [600, 800, None]$ $min_samples_split = [2, 5, 10]$ $min_samples_leaf = [1, 2, 4]$
2	70:30	$n_estimators = [200, 400, 600]$ $max_features = ['sqrt', 'log2']$ $max_depth = [600, 800, None]$ $min_samples_split = [2, 5, 10]$ $min_samples_leaf = [1, 2, 4]$
3	60:40	$n_estimators = [200, 400, 600]$ $max_features = ['sqrt', 'log2']$ $max_depth = [600, 800, None]$ $min_samples_split = [2, 5, 10]$ $min_samples_leaf = [1, 2, 4]$

Dengan demikian, terdapat tiga skenario pengujian yang masing-masing menggunakan variasi rasio pembagian data latih dan data uji, yaitu 80:20, 70:30, dan 60:40. Setiap skenario diuji menggunakan kombinasi *hyperparameter* pada algoritma *Random forest*. Pengujian ini bertujuan untuk mengevaluasi pengaruh rasio pembagian data dan konfigurasi *hyperparameter* terhadap kinerja model dalam mengklasifikasikan sentimen publik terhadap ulasan produk iPhone. Hasil dari masing-masing skenario akan dibandingkan untuk menentukan konfigurasi terbaik yang memberikan performa klasifikasi optimal.

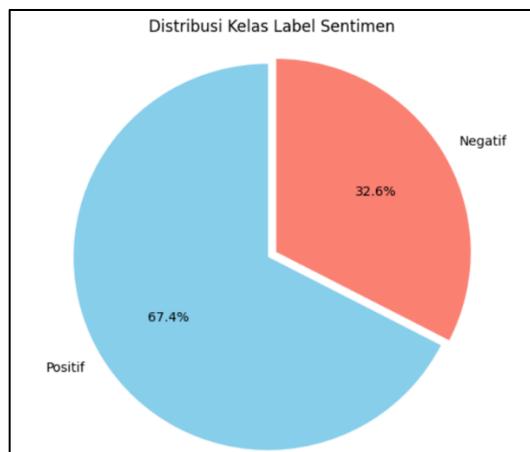
BAB IV

UJI COBA DAN PEMBAHASAN

Pada bab ini diuraikan hasil pengujian model klasifikasi sentimen terhadap ulasan pengguna iPhone menggunakan algoritma *Random forest*, serta evaluasi performa model berdasarkan metrik yang telah ditentukan

4.1 Data Penelitian

Merujuk pada penjelasan sebelumnya, data dalam penelitian ini berkaitan dengan klasifikasi sentimen pada ulasan pelanggan iPhone. Dataset diperoleh dari platform Kaggle yang berisi kumpulan komentar atau ulasan dari pengguna iPhone terkait pengalaman mereka terhadap produk tersebut.



Gambar 4.1 Distribusi Sentimen

Distribusi kelas label sentimen dari ulasan pelanggan menunjukkan bahwa 67,4% dari total ulasan memiliki sentimen positif, mencerminkan kepuasan pengguna terhadap produk iPhone. Di sisi lain, 32,6% dari total ulasan mencerminkan sentimen negatif, yang menunjukkan ketidakpuasan atau keluhan dari sebagian pengguna. Distribusi ini memberikan gambaran umum tentang

pandangan pengguna terhadap iPhone, dengan mayoritas ulasan cenderung positif, sebagaimana disajikan lebih lanjut dalam Tabel 4.1.

Tabel 4.1 Penyajian Data Penelitian

No	Ulasan	Label
1	Produk ini tiba-tiba digantung selama panggilan dan saya tidak bisa memotong panggilan. Pengalaman terburuk dengan produk ini. Tidak ada nilai uang	Negatif
2	Harapan itu bagus !! iPhone adalah yang terbaik !!	Positif
3	Kualitas kabel tidak bagus	Negatif
4	Fitur bagus, warna tidak menurut. Aku	Negatif
5	Tingkatkan kualitas foto Anda ... video sangat mengesankan 😊	Positif
6	Sangat buruk	Negatif
7	Brand bagus	Positif
8	Telepon yang bagus	Positif
9	Telah melacak dan membandingkan kinerja baterai dengan ponsel yang sama yang dibeli dari croma.mobile yang saya beli secara online memiliki lebih sedikit kinerja baterai dibandingkan dengan yang dibeli secara offline. Batrai iPhone 13 Dibeli secara online, Mengalir 5-6% semalam dibandingkan dengan ponsel yang dibeli 1-2% semalam .. saya telah memeriksa kinerja ini dengan penyedia layanan yang sama di kedua telepon.. Berinvestasilah dengan hati-hati.. Produk yang tersedia secara online dengan diskon besar mungkin memiliki masalah..	Negatif
10	Ponsel dengan daya tahan baterai terburuk, sangat terbatas dalam menggunakan ponsel dan Apple mengatakan demi keamanan.	Negatif

Data yang terdapat pada Tabel 4.1 menampilkan 10 sampel teks mentah yang mencakup berbagai variasi penulisan, termasuk penggunaan kata-kata non-baku, singkatan, dan emotikon. Data ini merupakan langkah awal yang akan diproses lebih lanjut dalam tahap uji coba.

4.2 *Preprocessing Text*

Pada tahap ini, dilakukan *Preprocessing* untuk membersihkan dan menormalkan data teks sebelum digunakan oleh model. Proses ini mencakup enam tahapan, yaitu *case folding*, *cleaning*, *tokenizing*, *normalization*, *stopword removal*, dan *stemming*, guna memastikan data berada dalam format yang konsisten dan siap untuk tahap selanjutnya.

4.2.1 Case Folding

Tahapan case folding diterapkan pada kolom *translated_review* untuk mengubah seluruh huruf menjadi huruf kecil. Proses ini dilakukan menggunakan metode `.lower()` pada setiap entri teks. Gambar 4.2 menampilkan kode program yang digunakan serta hasil setelah *case folding* diterapkan.

```
def case_folding(text):
    return text.lower()

df['translated_review'] = df['translated_review'].fillna('')

['case_folded'] = df['translated_review'].apply(case_folding)
print("Case Folding Results:")
df[['translated_review', 'case_folded']].head()
```

Gambar 4.2 Source Code Case Folding

Kode pada Gambar 4.2 menunjukkan fungsi `case_folding()` yang menggunakan metode `.lower()` untuk mengubah setiap teks menjadi huruf kecil. Sebelum dilakukan proses ini, nilai kosong pada kolom *translated_review* terlebih dahulu diganti dengan string kosong untuk menghindari error. Kemudian, fungsi tersebut diterapkan ke seluruh baris pada kolom *translated_review*, dan hasilnya disimpan dalam kolom baru bernama *case_folded*. Gambar 4.3 menampilkan hasil perubahan setelah dilakukan case folding, di mana seluruh huruf besar telah dikonversi menjadi huruf kecil.

translated_review	case_folded
Produk ini tiba -tiba digantung selama panggil...	produk ini tiba -tiba digantung selama panggil...
Harapan itu bagus !! iPhone adalah yang terbai...	harapan itu bagus !! iphone adalah yang terbai...
Kualitas kabel tidak bagus	kualitas kabel tidak bagus
Fitur bagus, warna tidak menurut. Aku	fitur bagus, warna tidak menurut. aku
Tingkatkan kualitas foto Anda ... video sangat...	tingkatkan kualitas foto anda ... video sangat...

Gambar 4.3 Hasil Case Folding

Gambar 4.3 menunjukkan hasil dari proses *case folding*, di mana seluruh huruf kapital pada teks telah berhasil dikonversi menjadi huruf kecil. Misalnya, kata “iPhone” diubah menjadi “iphone”, dan “Produk” menjadi “produk”. Perubahan ini bertujuan untuk menyamakan representasi kata agar tidak terjadi perbedaan pemrosesan antara kata yang seharusnya identik namun ditulis dengan kapitalisasi berbeda.

4.2.2 Cleaning

Tahapan *cleaning* diterapkan pada kolom *case_folded* untuk menghapus elemen-elemen yang tidak relevan seperti URL, mention, hashtag, serta karakter khusus. Proses ini dilakukan menggunakan fungsi *re.sub()* dari pustaka *re* untuk menyaring pola-pola tersebut dari teks. Gambar 4.4 menampilkan kode program yang digunakan serta hasil setelah tahap *cleaning* diterapkan.

```
def cleaning(text):
    if not text:
        return ""

    text = re.sub(r"http\S+|www\S+|https\S+", "", text)
    text = re.sub(r"@w+|#w+|[\^w\s]", "", text)
    return text.strip()

df['cleaned'] = df['case_folded'].apply(cleaning)
print("Cleaning Results:")
df[['case_folded', 'cleaned']].head()
```

Gambar 4.4 Source Code Cleaning

Pada Gambar 4.4 terlihat bahwa teks telah dibersihkan dari elemen-elemen yang tidak diperlukan seperti tautan (URL), simbol “@” dan “#”, serta karakter-karakter khusus lainnya. Jika sebelumnya sebuah teks mengandung tautan atau simbol tertentu, maka setelah tahap *cleaning*, elemen-elemen tersebut telah dihapus

sehingga hanya menyisakan kata-kata utama yang relevan untuk proses analisis.

Gambar 4.5 merupakan hasil dari proses *cleaning*.

case_folded	cleaned
produk ini tiba -tiba digantung selama panggil...	produk ini tiba tiba digantung selama panggilan...
harapan itu bagus !! iphone adalah yang terbai...	harapan itu bagus iphone adalah yang terbaik
kualitas kabel tidak bagus	kualitas kabel tidak bagus
fitur bagus, warna tidak menurut. aku	fitur bagus warna tidak menurut aku
tingkatkan kualitas foto anda ... video sangat...	tingkatkan kualitas foto anda video sangat me...

Gambar 4.5 Hasil *Cleaning*

Gambar 4.5 merupakan hasil dari proses *cleaning* data pada kolom “case_folded” yang menghasilkan perbaikan dalam bentuk penyederhanaan kalimat dan penghapusan kata-kata yang tidak relevan. Dalam tahap ini, beberapa elemen yang dianggap tidak memberikan kontribusi terhadap analisis, seperti simbol, angka berlebih, dan tanda baca, dihilangkan untuk meningkatkan kualitas data. Sebagai contoh, kata “tiba-tiba” disederhanakan menjadi “tiba tiba”, dan tanda baca seperti koma serta karakter khusus lainnya dihapus. Proses ini bertujuan untuk memastikan bahwa teks yang digunakan dalam tahap analisis berikutnya lebih bersih, seragam, dan representatif terhadap isi komentar sebenarnya.

4.2.3 Tokenization

Dalam tahap pengolahan teks, salah satu proses yang memiliki peran penting adalah tokenization. Proses ini menjadi fondasi dalam berbagai teknik analisis teks lanjutan, seperti ekstraksi fitur, analisis sentimen, maupun pemodelan machine learning. Gambar 4.6 menampilkan kode program untuk tahap *tokenization*.

```

def tokenization(text):
    return text.split()

df['tokenized'] = df['cleaned'].apply(tokenization)
print("Tokenization Results:")
df[['cleaned', 'tokenized']].head()

```

Gambar 4.6 *Source Code Tokenization*

Gambar 4.6 menunjukkan proses tokenisasi dengan mendefinisikan fungsi `tokenization` yang memisahkan teks berdasarkan spasi menggunakan `.split()`. Fungsi ini kemudian diterapkan ke kolom `cleaned` menggunakan `.apply()`, dan hasilnya disimpan pada kolom baru bernama `tokenized`. Dengan cara ini, setiap kalimat dibagi menjadi daftar kata, yang akan digunakan pada tahap pemrosesan teks berikutnya. Gambar 4.7 merupakan hasil dari proses tokenization.

cleaned	tokenized
produk ini tiba tiba digantung selama panggilan...	[produk, ini, tiba, tiba, digantung, selama, p...
harapan itu bagus iphone adalah yang terbaik	[harapan, itu, bagus, iphone, adalah, yang, te...
kualitas kabel tidak bagus	[kualitas, kabel, tidak, bagus]
fitur bagus warna tidak menurut aku	[fitur, bagus, warna, tidak, menurut, aku]
tingkatkan kualitas foto anda video sangat me...	[tingkatkan, kualitas, foto, anda, video, sang...

Gambar 4.7 Hasil *Tokenization*

Gambar 4.7 hasil tokenisasi menunjukkan bahwa setiap teks pada kolom `cleaned` telah berhasil diubah menjadi daftar kata pada kolom `tokenized`. Misalnya, kalimat “kualitas kabel tidak bagus” dipisah menjadi token-token seperti “kualitas”, “kabel”, “tidak”, “bagus”. Hasil ini menandakan bahwa proses tokenisasi berjalan dengan baik, menghasilkan potongan kata yang siap digunakan untuk analisis lanjutan.

4.2.4 Normalization

Setelah teks dipecah menjadi token, langkah berikutnya adalah memastikan konsistensi bentuk kata yang digunakan. Hal ini dilakukan untuk menghindari makna ganda atau kesalahan interpretasi yang bisa muncul akibat variasi penulisan kata dalam bahasa sehari-hari. Berikut adalah kode untuk tahap *normalization*.

```

slang_dict = pd.read_csv("/kaggle/input/colloquial-indonesian-lexicon/colloquial-indonesian-lexicon.csv")
slang_mapping = dict(zip(slang_dict['slang'], slang_dict['formal']))

def normalization(words):
    return [slang_mapping[word] if word in slang_mapping else word for word in words]

df['normalized'] = df['tokenized'].apply(normalization)
print("Normalization Results:")
df[['tokenized', 'normalized']].head()

```

Gambar 4.8 Source Code Normalization

Gambar 4.8 melakukan proses normalisasi dengan memanfaatkan kamus pasangan kata tidak baku dan kata baku dari file CSV. Data tersebut dibaca dan diubah menjadi dictionary `slang_mapping`. Fungsi `normalization` kemudian mengganti setiap kata dalam daftar token dengan bentuk bakunya jika tersedia dalam kamus. Hasil normalisasi disimpan dalam kolom baru bernama `normalized`. Gambar 4.9 menampilkan hasil *normalization*.

tokenized	normalized
[produk, ini, tiba, tiba, digantung, selama, p...	[produk, ini, tiba, tiba, digantung, selama, p...
[harapan, itu, bagus, iphone, adalah, yang, te...	[harapan, itu, bagus, iphone, adalah, yang, te...
[kualitas, kabel, tidak, bagus]	[kualitas, kabel, tidak, bagus]
[fitur, bagus, warna, tidak, menurut, aku]	[fitur, bagus, warna, tidak, menurut, aku]
[tingkatkan, kualitas, foto, anda, video, sang...	[tingkatkan, kualitas, foto, anda, video, sang...

Gambar 4.9 Hasil Normalization

Gambar hasil normalisasi menunjukkan bahwa sebagian kata dalam kolom `tokenized` telah diperiksa dan, jika ditemukan dalam kamus, diganti dengan bentuk bakunya di kolom `normalized`. Sebagian besar token tetap sama karena tidak

termasuk dalam daftar slang, menandakan bahwa teks sudah cukup formal atau tidak mengandung banyak kata tidak baku.

4.2.5 Stopword Removal

Setelah teks dinormalisasi, langkah berikutnya difokuskan pada penyaringan kata agar hanya informasi yang benar-benar relevan yang dipertahankan. Proses ini membantu menyederhanakan data dan mengurangi gangguan dari kata-kata yang terlalu umum, sehingga analisis dapat lebih terarah dan efisien. Berikut adalah gambar kode untuk langkah penghapusan kata-kata umum (*stopword*).

```
import nltk
from nltk.corpus import stopwords
stop_words = set(stopwords.words("indonesian"))

def remove_stopwords(words):
    return [word for word in words if word not in stop_words]

df['stopwords_removed'] = df['normalized'].apply(remove_stopwords)
print("Stopword Removal Results:")
df[['normalized', 'stopwords_removed']].head()
```

Gambar 4.10 Source Code Stopword Removal

Gambar 4.10 menggunakan pustaka NLTK (*Natural Language Toolkit*) untuk menghapus stopwords dari data teks dalam bahasa Indonesia. Pertama, stopwords dalam bahasa Indonesia diambil menggunakan `stopwords.words("indonesian")` dan disimpan dalam variabel `stop_words`. Fungsi `remove_stopwords` menerima daftar kata (`words`) dan mengembalikan daftar kata yang tidak termasuk dalam daftar stopwords. Fungsi ini diterapkan pada kolom `normalized` dari DataFrame `df`, menghasilkan kolom baru bernama `stopwords_removed` yang berisi teks tanpa stopwords. Hasilnya kemudian ditampilkan dengan mencetak lima baris pertama

dari kolom `normalized` dan `stopwords_removed`. Gambar 4.11 hasil proses *stopword removal*.

normalized	stopwords_removed
[produk, ini, tiba, tiba, digantung, selama, p...	[produk, digantung, panggilan, memotong, pangg...
[harapan, itu, bagus, iphone, adalah, yang, te...	[harapan, bagus, iphone, terbaik]
[kualitas, kabel, tidak, bagus]	[kualitas, kabel, bagus]
[fitur, bagus, warna, tidak, menurut, aku]	[fitur, bagus, warna]
[tingkatkan, kualitas, foto, anda, video, sang...	[tingkatkan, kualitas, foto, video, mengesankan]

Gambar 4.11 Hasil *Stopword Removal*

Gambar tersebut menunjukkan hasil dari proses *stopword removal* pada data teks yang telah dinormalisasi. Pada kolom "normalized", teks masih mengandung kata-kata umum (stopword). Setelah proses *stopword removal*, kolom "stopwords_removed" hanya menyisakan kata-kata yang dianggap penting atau bermakna, seperti "iphone", "produk", "fitur", "kualitas", dan sebagainya.

4.2.6 Stemming

Setelah proses penghapusan *stopwords*, tahap selanjutnya adalah menyamakan bentuk kata agar kata yang berbeda namun bermakna sama dikenali sebagai satu kesatuan. Hal ini membantu meningkatkan akurasi analisis teks.

Berikut adalah gambar kode untuk tahap *stemming*.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stemming(words):
    return [stemmer.stem(word) for word in words]

df['stemming'] = df['stopwords_removed'].apply(stemming)
print("Stemming Results:")
df[['stopwords_removed', 'stemming']].head()
```

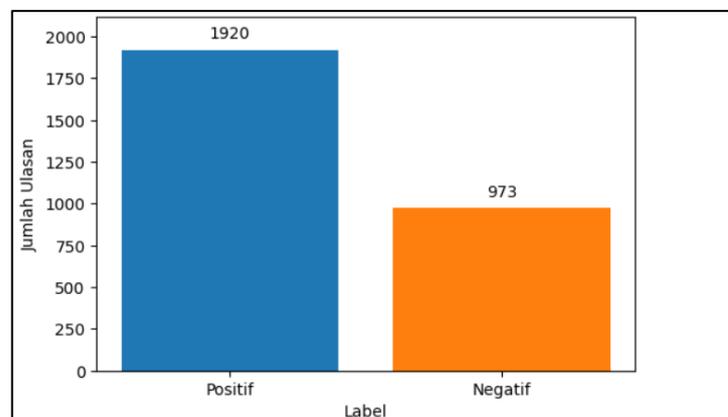
Gambar 4.12 Source Code Stemming

Gambar 4.12 menggunakan library Sastrawi untuk melakukan proses stemming pada teks. Fungsi stemming dibuat untuk mengubah setiap kata dalam daftar menjadi bentuk dasarnya menggunakan stemmer dari Sastrawi. Kemudian, fungsi ini diterapkan ke kolom `stopwords_removed` pada DataFrame, dan hasilnya disimpan dalam kolom baru bernama `stemming`. Proses ini membantu menyederhanakan variasi kata agar analisis teks lebih efektif. Gambar 4.13 menampilkan hasil tahap *stemming*.

<code>stopwords_removed</code>	<code>stemming</code>
[produk, digantung, panggilan, memotong, pangg...	[produk, gantung, panggil, potong, panggil, al...
[harapan, bagus, iphone, terbaik]	[harap, bagus, iphone, baik]
[kualitas, kabel, bagus]	[kualitas, kabel, bagus]
[fitur, bagus, warna]	[fitur, bagus, warna]
[tingkatkan, kualitas, foto, video, mengesankan]	[tingkat, kualitas, foto, video, kesan]

Gambar 4.13 Hasil *Stemming*

Hasil *stemming* menunjukkan bahwa kata-kata dalam kolom `stopwords_removed` telah diproses untuk dikembalikan ke bentuk dasarnya. Misalnya, kata "digantung" menjadi "gantung" dan "terbaik" menjadi "baik". Proses ini bertujuan untuk menyamakan bentuk kata yang berbeda namun memiliki makna dasar yang sama, sehingga analisis teks menjadi lebih efektif.



Gambar 4.14 Jumlah Data Setelah *Preprocessing*

Setelah seluruh tahapan *Preprocessing* selesai dilakukan, jumlah data dalam masing-masing kelas sentimen mengalami perubahan. Data berlabel positif kini berjumlah 1920, sementara data berlabel negatif berjumlah 973, yang mencerminkan distribusi akhir yang akan digunakan dalam proses klasifikasi.

4.3 Pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*)

Setelah melalui tahapan *Preprocessing*, langkah berikutnya adalah mengubah data teks menjadi representasi numerik agar dapat diproses oleh algoritma machine learning. Dalam penelitian ini, pendekatan yang digunakan adalah TF-IDF, sebuah metode yang efektif untuk mengekstraksi fitur dari data teks berdasarkan pentingnya suatu kata dalam kumpulan dokumen. Pendekatan ini memungkinkan setiap dokumen direpresentasikan sebagai vektor yang mencerminkan karakteristik unik dari kontennya.

Tabel 4.2 merupakan hasil *Preprocessing* dari Tabel 4.1 yang kemudian data dari tabel ini akan digunakan untuk menghitung pembobotan TF-IDF dengan rumus yang telah dijelaskan dalam persamaan 2.1 dan 2.2.

Tabel 4.2 Data Sampel Perhitungan TF-IDF Hasil *Preprocessing*

No	Ulasan	Label
1	produk gantung panggil potong panggil alam buruk produk nilai uang	Negatif
2	harap bagus iphone baik	Positif
3	kualitas kabel bagus	Negatif
4	fitur bagus warna	Negatif
5	tingkat kualitas foto video kesan	Positif
6	buruk	Negatif
7	brand bagus	Positif
8	telepon yang bagus	Positif
9	baterai iphone beli online boros malam banding ponsel beli.	Negatif
10	ponsel daya baterai buruk batas gunakan ponsel apple kata aman	Negatif

Term	TF x IDF									
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
buruk	0.52	0	0	0	0	0.52	0	0	0	0.52
baterai	0	0	0	0	0	0	0	0	0.70	0.70

Tabel 4.4 menunjukkan hasil pembobotan TF-IDF, di mana term "ponsel" memiliki bobot tertinggi pada D10 sebesar 1.4, menunjukkan bahwa kata ini lebih spesifik dalam dokumen tersebut dibandingkan dokumen lainnya. Sementara itu, term "iphone" memiliki bobot 0.70 pada D2 dan D9, yang menandakan keberadaannya di beberapa dokumen namun dengan distribusi yang terbatas. Term "buruk" memiliki bobot 0.52 pada D1, D6, dan D10, mengindikasikan bahwa kata ini cukup signifikan dalam konteks negatif di beberapa dokumen. Selain itu, term "baterai" memiliki bobot 0.70 di D9 dan D10, menunjukkan bahwa topik terkait daya tahan baterai cukup menonjol dalam dokumen tersebut. Term "bagus" muncul lebih sering dibandingkan beberapa kata lain, tetapi bobotnya relatif kecil (0.30) karena IDF-nya lebih rendah.

4.4 Uji Coba

Setelah melalui tahapan dan pembobotan TF-IDF, langkah selanjutnya *Preprocessing* adalah melakukan uji coba model klasifikasi. Pengujian dilakukan dengan tiga variasi pembagian data, yaitu 80:20, 70:30, dan 60:40. Pada setiap split, dilakukan *tuning hyperparameter* menggunakan *Grid Search* untuk mendapatkan kombinasi parameter terbaik. Evaluasi model dilakukan menggunakan *confusion matrix* guna mengukur akurasi, presisi, *recall*, dan *F1-score* dari hasil klasifikasi. Tabel 4.5 menyajikan distribusi label pada tiap split.

Tabel 4.5 Distribusi Label Pada Tiap Split

Split Data	Data Latih	Positif (Latih)	Negatif (Latih)	Data Uji	Positif (Uji)	Negatif (Uji)
80:20	2.314	1.529	785	579	391	188
70:30	2.025	1.323	702	868	597	271
60:40	1.735	1.142	593	1.158	778	380

4.4.1 Implementasi *Grid Search* Pada Model Split 80:20

Pada tahap ini, *Grid Search* digunakan untuk mencari kombinasi terbaik dari lima *hyperparameter Random forest*, yaitu *n_estimators*, *max_features*, *max_depth*, *min_samples_split*, dan *min_samples_leaf*. Proses evaluasi dilakukan dengan metode *5-Fold Cross Validation* pada data latih dengan rasio split 80:20.

Tabel 4.6 Kombinasi *Hyperparameter* Terbaik (Model Split 80:20)

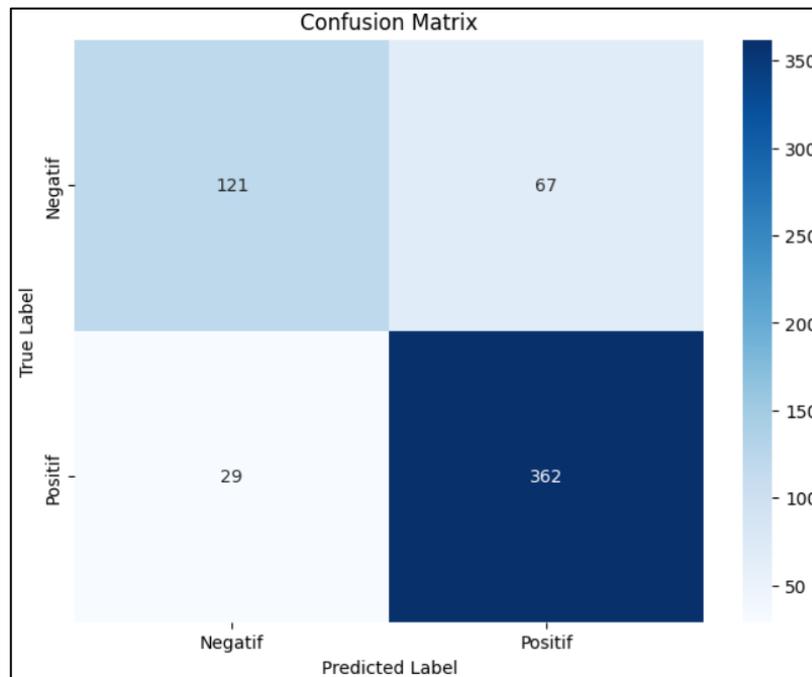
<i>Hyperparameter</i>	Nilai Terbaik
<i>n_estimators</i>	200
<i>max_features</i>	sqrt
<i>max_depth</i>	600
<i>min_samples_split</i>	5
<i>min_samples_leaf</i>	1
Avg CV Accuracy	82.15%

Dari total 162 kombinasi *hyperparameter* yang diuji, kombinasi terbaik pada Tabel 4.6 diperoleh pada *n_estimators* = 200, *max_features* = sqrt, *max_depth* = 600, *min_samples_split* = 5, *min_samples_leaf* = 1. Kombinasi tersebut menghasilkan akurasi rata-rata tertinggi sebesar 82.15% pada validasi silang.

4.4.2 Evaluasi Model Split 80:20 dengan Parameter Terbaik

Bagian ini mengulas evaluasi model dengan pembagian data latih dan data uji menggunakan rasio 80:20, dengan menerapkan parameter optimal yang telah

ditentukan sebelumnya. Hasil evaluasi yang diperoleh melalui *confusion matrix* ditunjukkan pada Gambar 4.15.



Gambar 4.15 *Confusion Matrix* Model Split 80:20

Berdasarkan *confusion matrix*, terdapat 121 ulasan negatif yang terklasifikasi dengan benar sebagai negatif, sementara 67 ulasan negatif mengalami kesalahan klasifikasi sebagai positif. Di sisi lain, sebanyak 29 ulasan positif tidak tepat teridentifikasi sebagai negatif, sedangkan 362 ulasan positif berhasil dikategorikan dengan benar sebagai positif. Nilai *precision*, *recall*, dan *F1-score* dapat ditemukan Tabel 4.7.

Tabel 4.7 Evaluasi Model Split 80:20

Metrik	Nilai (%)
<i>Precision</i>	84.38
<i>Recall</i>	92.58
F1 Score	88.29
Akurasi	83.42

Hasil Tabel 4.7 menunjukkan bahwa akurasi model mencapai 83.42%, dengan nilai untuk *precision*, *recall*, dan *F1-score* masing-masing sebesar 84.38%, 92.58%, dan 88.29%. Berikut adalah perhitungan rinci untuk setiap metrik :

$$Accuracy = \frac{(362 + 121)}{(362 + 121 + 29 + 67)} \times 100\% = 83.42\%$$

$$Precision = \frac{(362)}{362 + 67} \times 100\% = 84.38\%$$

$$Recall = \frac{(362)}{362 + 29} \times 100\% = 92.58\%$$

$$F1\ Score = \frac{2 \times 84.38 \times 92.58}{84.38 + 92.58} \times 100\% = 88.29\%$$

4.4.3 Implementasi *Grid Search* Pada Model Split 70:30

Pada tahap ini, *Grid Search* juga diterapkan untuk data latih dengan skenario pembagian data 70:30 guna menemukan kombinasi optimal dari lima *hyperparameter* *Random forest*, yaitu *n_estimators*, *max_features*, *max_depth*, *min_samples_split*, dan *min_samples_leaf*. Proses evaluasi dilakukan menggunakan 5-Fold Cross Validation terhadap seluruh kombinasi parameter.

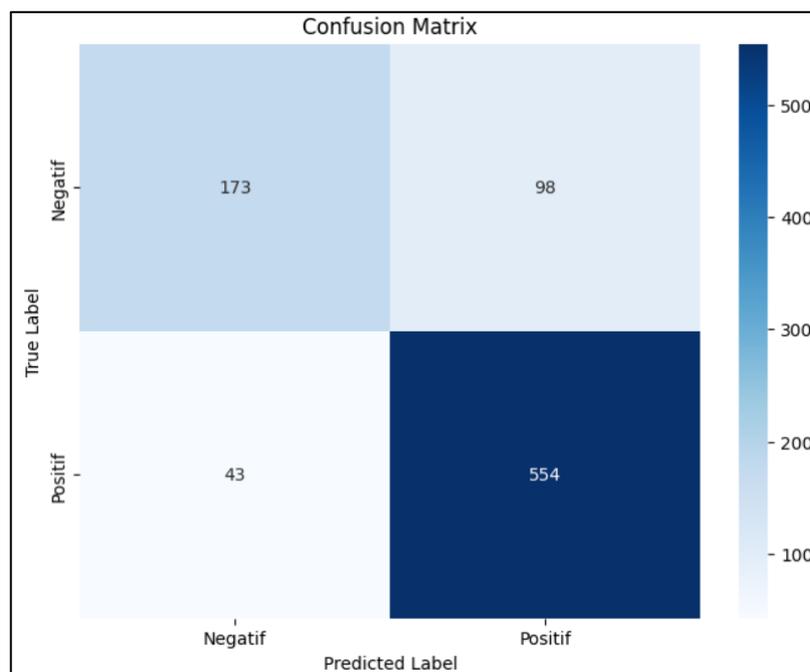
Tabel 4.8 Kombinasi *Hyperparameter* Terbaik (Model Split 70:30)

<i>Hyperparameter</i>	Nilai Terbaik
<i>n_estimators</i>	200
<i>max_features</i>	sqrt
<i>max_depth</i>	600
<i>min_samples_split</i>	5
<i>min_samples_leaf</i>	1
Avg CV Accuracy	81.53%

Dari total 162 kombinasi *hyperparameter* yang diuji, performa terbaik diperoleh pada $n_estimators = 200$, $max_features = \text{sqrt}$, $max_depth = 600$, $min_samples_split = 5$, $min_samples_leaf = 1$. Kombinasi ini menghasilkan akurasi rata-rata validasi silang sebesar 81.53%, yang merupakan hasil terbaik pada konfigurasi split 70:30.

4.4.4 Evaluasi Model Split 70:30 dengan Parameter Terbaik

Bagian ini membahas evaluasi model dengan pembagian data latih dan data uji menggunakan rasio 70:30, menggunakan parameter terbaik yang telah diperoleh sebelumnya. Gambar 4.16 menampilkan hasil evaluasi *confusion matrix*.



Gambar 4.16 *Confusion Matrix* Model Split 70:30

Dari Gambar 4.16, tercatat ada 173 ulasan negatif yang berhasil diklasifikasikan secara akurat sebagai negatif, sementara 98 ulasan negatif mengalami kesalahan prediksi sebagai positif. Di sisi lain, terdapat 43 ulasan positif

yang salah terprediksi sebagai negatif, sedangkan 554 ulasan positif berhasil teridentifikasi dengan benar sebagai positif. Nilai *precision*, *recall*, dan *F1-score* dapat ditemukan Tabel 4.9.

Tabel 4.9 Evaluasi Model Split 70:30

Metrik	Nilai (%)
<i>Precision</i>	84.97
<i>Recall</i>	92.80
F1 Score	88.71
Akurasi	83.76

Hasil Tabel 4.9 menunjukkan bahwa model memiliki akurasi model mencapai 83.76%, dengan nilai untuk *precision*, *recall*, dan *F1-score* masing-masing sebesar 84.97%, 92.80%, dan 88.71%. Berikut adalah detail perhitungan untuk masing-masing metrik:

$$Accuracy = \frac{(554 + 173)}{(554 + 173 + 43 + 98)} \times 100\% = 83.76\%$$

$$Precision = \frac{(554)}{554 + 98} \times 100\% = 84.97\%$$

$$Recall = \frac{(554)}{554 + 43} \times 100\% = 92.80\%$$

$$F1\ Score = \frac{2 \times 84.97 \times 92.80}{84.97 + 92.80} \times 100\% = 88.71\%$$

4.4.5 Implementasi *Grid Search* Pada Model Split 60:40

Pada skenario pembagian data latih dan uji sebesar 60:40, *Grid Search* kembali digunakan untuk mengevaluasi kombinasi terbaik dari lima *hyperparameter Random forest*, yaitu *n_estimators*, *max_features*, *max_depth*,

min_samples_split, dan *min_samples_leaf*. Evaluasi dilakukan menggunakan metode 5-Fold Cross Validation terhadap seluruh 162 kombinasi parameter.

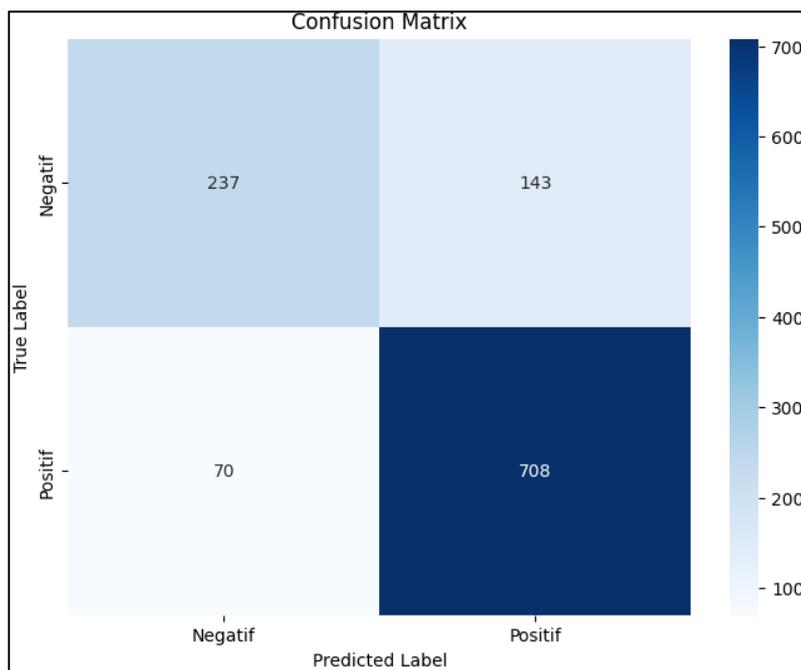
Tabel 4.10 Kombinasi *Hyperparameter* Terbaik (Model Split 60:40)

<i>Hyperparameter</i>	Nilai Terbaik
<i>n_estimators</i>	200
<i>max_features</i>	sqrt
<i>max_depth</i>	600
<i>min_samples_split</i>	2
<i>min_samples_leaf</i>	1
Avg CV Accuracy	80.52%

Dari seluruh pengujian, kombinasi dengan performa terbaik berdasarkan Tabel 4.10, ditemukan pada $n_estimators = 200$, $max_features = \text{sqrt}$, $max_depth = 600$, $min_samples_split = 2$, $min_samples_leaf = 1$. Kombinasi tersebut memberikan akurasi rata-rata validasi silang sebesar 80.52%, yang menjadi hasil terbaik untuk konfigurasi data split 60:40. Kombinasi *hyperparameter* ini selanjutnya akan di implementasikan ke dalam model untuk dilatih ulang pada data latih sebanyak 60% dengan total 1.735 data.

4.4.6 Evaluasi Model Split 60:40 dengan Parameter Terbaik

Evaluasi model *Random forest* dilakukan dengan menggunakan parameter terbaik yang diperoleh dari proses *Grid Search*. Data dibagi dengan rasio 60:40, di mana 60% digunakan untuk pelatihan dan 40% untuk pengujian. Gambar 4.17 menampilkan hasil evaluasi model.



Gambar 4.17 *Confusion Matrix* Model Split 60:40

Berdasarkan hasil evaluasi model, ulasan negatif yang terprediksi dengan benar sebagai negatif sebanyak 237, sedangkan 143 ulasan negatif mengalami kesalahan klasifikasi sebagai positif. Sementara itu, 70 ulasan positif salah teridentifikasi sebagai negatif, sementara 708 ulasan positif berhasil dikategorikan dengan benar sebagai positif. Adapun nilai *precision*, *recall*, dan *F1-score* disajikan Tabel 4.11.

Tabel 4.11 Evaluasi Model Split 60:40

Metrik	Nilai (%)
<i>Precision</i>	83.20
<i>Recall</i>	91.00
F1 Score	86.92
Akurasi	81.61

Hasil Tabel 4.11 menunjukkan bahwa model memiliki akurasi mencapai 81.61%, dengan nilai untuk *precision*, *recall*, dan *F1-score* masing-masing sebesar

83.20%, 91.00%, dan 86.92%. Berikut adalah penjabaran terperinci untuk setiap metrik :

$$Accuracy = \frac{(708 + 237)}{(708 + 237 + 70 + 143)} \times 100\% = 81.61\%$$

$$Precision = \frac{(708)}{708 + 70} \times 100\% = 83.20\%$$

$$Recall = \frac{(708)}{708 + 143} \times 100\% = 91.00\%$$

$$F1\ Score = \frac{2 \times 83.20 \times 91.00}{83.20 + 91.00} \times 100\% = 86.92\%$$

4.5 Performa *Random forest* Setelah Optimasi *Hyperparameter*

Analisa dilakukan dengan *confusion matrix* dan *classification report* untuk mengevaluasi secara menyeluruh performa model *Random forest* setelah optimasi *hyperparameter* pada masing-masing skenario pembagian data.

4.5.1 Split 80:20

Pada bagian ini ditampilkan perbandingan performa model sebelum dan sesudah optimasi *hyperparameter* pada skenario pembagian data 80:20. Analisis ini bertujuan untuk melihat bagaimana perubahan parameter yang dioptimalkan dapat meningkatkan kemampuan model dalam klasifikasi sentimen, khususnya dalam mengenali ulasan positif dan negatif dengan lebih akurat. Gambar 4.18 menunjukkan hasil evaluasi model sebelum dilakukan optimasi *hyperparameter*.



Gambar 4.18 *Confusion Matrix* Model Split 80:20 Sebelum Optimasi

Evaluasi dilakukan dengan membandingkan nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), serta metrik akurasi, presisi, *recall*, dan *F1-score* untuk melihat perbedaan performa kedua model secara lebih menyeluruh.

Tabel 4.12 *Confusion Matrix* Model RF dan (RF + *Grid Search*) Split 80:20

Kondisi	RF	RF + <i>Grid Search</i>
TP	358	362
TN	120	121
FP	68	67
FN	33	29

Hasil Tabel 4.12 menunjukkan bahwa model dengan optimasi *hyperparameter* mampu meningkatkan jumlah True Positive (TP) dari 358 menjadi 362, yang berarti model lebih baik dalam mengidentifikasi ulasan positif dengan benar. Selain itu, False Negative (FN) berhasil dikurangi dari 33 menjadi 29,

menunjukkan peningkatan dalam menangkap ulasan positif yang sebelumnya terklasifikasi salah. False Positive (FP) mengalami sedikit penurunan dari 68 menjadi 67, menunjukkan perbaikan kecil dalam menghindari kesalahan klasifikasi ulasan negatif sebagai positif. True Negative (TN) juga mengalami peningkatan dari 120 menjadi 121, yang berarti model sedikit lebih baik dalam mempertahankan akurasi klasifikasi ulasan negatif.

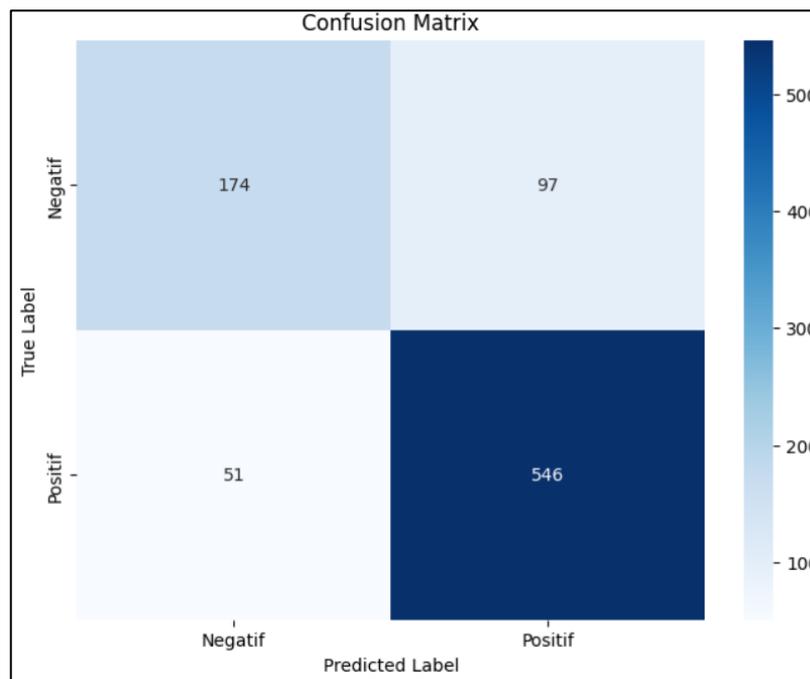
Tabel 4.13 Metrik Evaluasi Model RF dan (RF + *Grid Search*) Split 80:20

Metrik	RF	RF + <i>Grid Search</i>
<i>Precision</i> (%)	84.04	84.38
<i>Recall</i> (%)	91.56	92.58
<i>F1-score</i> (%)	87.64	88.29
Akurasi (%)	82.56	83.42

Hasil Tabel 4.13 menunjukkan bahwa model dengan optimasi *hyperparameter* mengalami peningkatan pada semua metrik evaluasi dibandingkan dengan model sebelum optimasi. *Precision* naik dari 84.04% menjadi 84.38%, menunjukkan sedikit peningkatan dalam akurasi klasifikasi ulasan positif. *Recall* meningkat dari 91.56% menjadi 92.58%, yang berarti model lebih efektif dalam menangkap ulasan positif yang benar. *F1-score* juga mengalami kenaikan dari 87.64% menjadi 88.29%, menandakan keseimbangan yang lebih baik antara *precision* dan *recall*. Selain itu, akurasi keseluruhan meningkat dari 82.56% menjadi 83.42%, yang menunjukkan bahwa optimasi *hyperparameter* membantu meningkatkan kinerja model secara keseluruhan.

4.5.2 Split 70:30

Perbandingan performa model *random forest* sebelum dan sesudah optimasi *hyperparameter* pada skenario split data 70:30 disajikan pada bagian ini. Berikut Gambar 4.19 evaluasi model sebelum optimasi *hyperparameter*.



Gambar 4.19 *Confusion Matrix* Model Split 70:30 Sebelum Optimasi

Analisis dilakukan dengan membandingkan True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), serta metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* guna memahami perbedaan performa antara kedua model secara lebih komprehensif.

Tabel 4.14 *Confusion Matrix* Model RF dan (RF + *Grid Search*) Split 70:30

Kondisi	RF	RF + <i>Grid Search</i>
TP	546	554
TN	174	173
FP	97	98
FN	51	43

Hasil Tabel 4.14 menunjukkan bahwa model dengan optimasi *hyperparameter* mampu meningkatkan jumlah True Positive (TP) dari 546 menjadi 554, yang berarti model lebih baik dalam mengenali ulasan positif dengan benar. Selain itu, False Negative (FN) berkurang dari 51 menjadi 43, menunjukkan peningkatan dalam menangkap ulasan positif yang sebelumnya diklasifikasikan salah. Namun, False Positive (FP) mengalami sedikit peningkatan dari 97 menjadi 98, menandakan adanya sedikit kesalahan dalam mengklasifikasikan ulasan negatif sebagai positif. True Negative (TN) mengalami sedikit penurunan dari 174 menjadi 173, menunjukkan bahwa model masih memiliki tantangan dalam mempertahankan akurasi klasifikasi ulasan negatif. Secara keseluruhan, optimasi *hyperparameter* menunjukkan perbaikan terutama dalam mengidentifikasi ulasan positif dengan lebih baik.

Tabel 4.15 Metrik Evaluasi Model RF dan (RF + *Grid Search*) Split 70:30

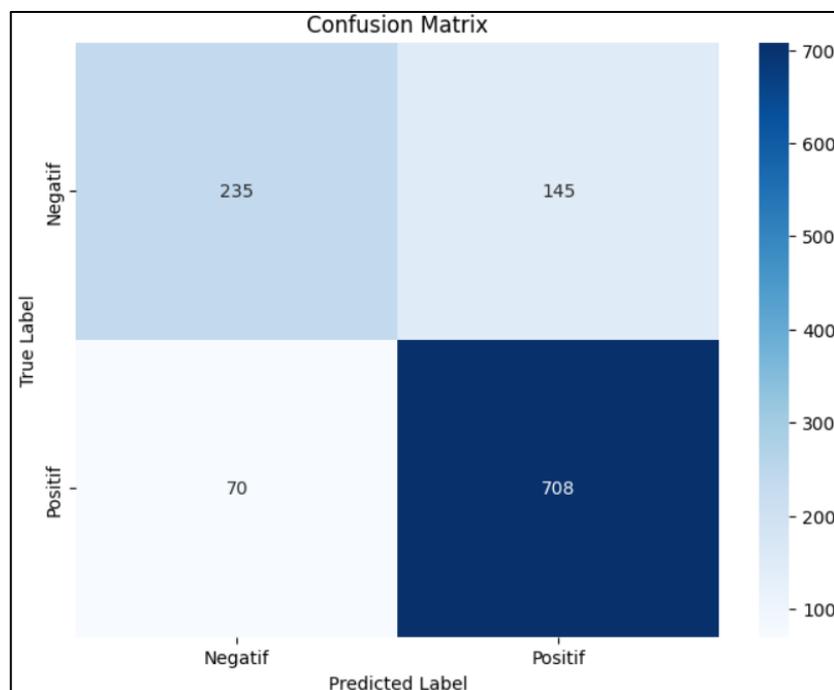
Metrik	RF	RF + <i>Grid Search</i>
<i>Precision</i> (%)	84.91	84.97
<i>Recall</i> (%)	91.46	92.80
<i>F1-score</i> (%)	88.06	88.71
Akurasi (%)	82.59	83.76

Hasil Tabel 4.15 menunjukkan bahwa model dengan optimasi *hyperparameter* mengalami peningkatan pada hampir semua metrik evaluasi. *Precision* sedikit meningkat dari 84.91% menjadi 84.97%, menunjukkan perbaikan kecil dalam akurasi klasifikasi ulasan positif, yang membantu mengurangi kesalahan dalam mengidentifikasi ulasan yang benar. *Recall* juga mengalami peningkatan dari 91.46% menjadi 92.80%, yang berarti model lebih efektif dalam mengenali dan menangkap lebih banyak ulasan positif dengan benar, sehingga

mengurangi kemungkinan kesalahan klasifikasi. *F1-score* naik dari 88.06% menjadi 88.71%, menandakan keseimbangan yang lebih baik antara *precision* dan *recall*, memastikan model dapat memberikan klasifikasi yang lebih stabil dan konsisten. Selain itu, akurasi keseluruhan meningkat dari 82.59% menjadi 83.76%, menunjukkan bahwa optimasi *hyperparameter* tidak hanya meningkatkan performa pada aspek tertentu tetapi juga memberikan dampak positif terhadap hasil prediksi model secara keseluruhan.

4.5.3 Split 60:40

Bagian ini menyajikan hasil evaluasi model *random forest* sebelum dan sesudah optimasi *hyperparameter* pada skenario pembagian data 60:40. Gambar 4.20 evaluasi model sebelum optimasi *hyperparameter*.



Gambar 4.20 *Confusion Matrix* Model Split 60:40 Sebelum Optimasi

Perbandingan True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), serta evaluasi menggunakan akurasi, presisi, *recall*, dan *F1-score* dilakukan untuk memperoleh gambaran yang lebih menyeluruh mengenai perbedaan performa kedua model.

Tabel 4.16 Confusion Matrix Model RF dan (RF + *Grid Search*) Split 60:40

Kondisi	RF	RF + <i>Grid Search</i>
TP	708	708
TN	235	237
FP	145	143
FN	70	70

Hasil Tabel 4.16 menunjukkan bahwa model dengan optimasi *hyperparameter* mengalami peningkatan pada True Negative (TN) dari 235 menjadi 237, yang berarti lebih banyak ulasan negatif diklasifikasikan dengan benar dibandingkan model default. True Positive (TP) tetap sama pada 708, menunjukkan bahwa optimasi *hyperparameter* tidak mengubah kemampuan model dalam mengenali ulasan positif. Selain itu, False Positive (FP) mengalami sedikit penurunan dari 145 menjadi 143, yang berarti ada sedikit lebih sedikit kesalahan dalam mengklasifikasikan ulasan negatif sebagai positif. Namun, False Negative (FN) tetap sama pada 70, menunjukkan bahwa masih ada beberapa ulasan positif yang salah dikategorikan sebagai negatif. Secara keseluruhan, model dengan *hyperparameter* terbaik menunjukkan peningkatan kecil dalam klasifikasi ulasan negatif, sementara performa dalam mengenali ulasan positif tetap stabil.

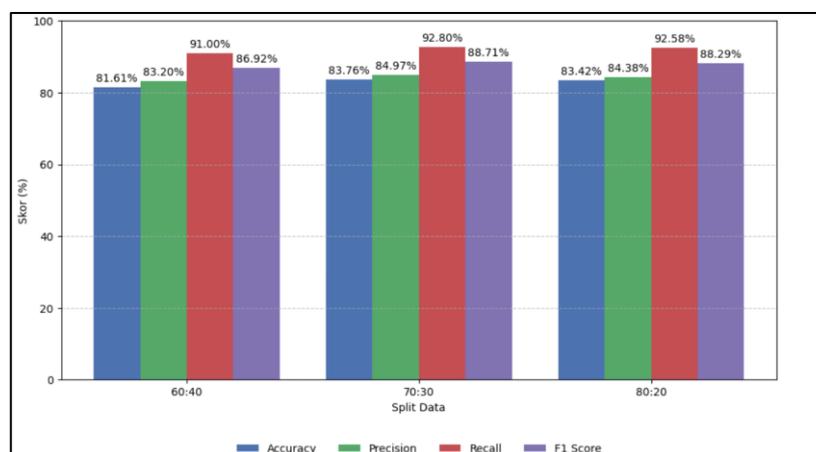
Tabel 4.17 Metrik Evaluasi Model RF dan (RF + *Grid Search*) Split 60:40

Metrik	RF	RF + <i>Grid Search</i>
<i>Precision</i> (%)	83.00	83.20
<i>Recall</i> (%)	91.00	91.00
<i>F1-score</i> (%)	86.82	86.92
Akurasi (%)	81.43	81.61

Hasil Tabel 4.17 menunjukkan bahwa model dengan optimasi *hyperparameter* mengalami peningkatan pada *precision* dari 83.00% menjadi 83.20%, menunjukkan perbaikan kecil dalam akurasi klasifikasi ulasan positif. *Recall* tetap stabil di 91.00%, mengindikasikan bahwa kedua model memiliki performa yang sama dalam mengenali ulasan positif dengan benar. *F1-score* meningkat dari 86.82% menjadi 86.92%, menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall* setelah optimasi. Selain itu, akurasi keseluruhan naik dari 81.43% menjadi 81.61%, menandakan bahwa optimasi *hyperparameter* memberikan peningkatan kecil namun positif terhadap performa model.

4.6 Perbandingan Performa Model Teroptimasi Berdasarkan Split Data

Bab ini membahas perbandingan performa model teroptimasi berdasarkan berbagai skenario pembagian data. Analisis dilakukan untuk melihat bagaimana perubahan proporsi data latih dan data uji memengaruhi akurasi, *precision*, *recall*, dan *F1-score*, serta sejauh mana optimasi *hyperparameter* memberikan peningkatan performa pada setiap split.



Gambar 4.21 Performa Model Berdasarkan Split Data

Berdasarkan Gambar 4.21, hasil evaluasi dari berbagai skenario pembagian data (80:20, 70:30, dan 60:40), terlihat bahwa jumlah data pelatihan yang digunakan memengaruhi performa model *Random forest* secara signifikan. Model dengan split 70:30 menunjukkan hasil terbaik, dengan akurasi tertinggi sebesar 83.76%, *precision* meningkat menjadi 84.97%, *recall* mencapai 92.80%, dan *F1-score* 88.71%. Hal ini menunjukkan bahwa pembagian data 70% untuk pelatihan dan 30% untuk pengujian memberikan keseimbangan optimal, memungkinkan model mempelajari pola dengan cukup baik tanpa mengalami overfitting atau kehilangan kemampuan generalisasi. Split 80:20 tetap menunjukkan performa yang kompetitif, dengan akurasi 83.42%, *precision* 84.38%, *recall* 92.58%, dan *F1-score* 88.29%. Meskipun sedikit lebih rendah dibandingkan 70:30, hasil ini tetap menunjukkan bahwa model dapat bekerja dengan baik dengan jumlah data pelatihan yang lebih sedikit, tetapi mungkin memiliki sedikit kesalahan dalam mengklasifikasikan ulasan negatif sebagai positif.

Sebaliknya, split 60:40 menunjukkan penurunan akurasi menjadi 81.61%, dengan *precision* 83.20%, *recall* tetap tinggi di 91.00%, dan *F1-score* turun menjadi 86.92%. Meskipun jumlah data pelatihan lebih besar dibandingkan split lainnya, hasil ini mengindikasikan kemungkinan overfitting, di mana model terlalu terlatih pada data pelatihan sehingga kurang mampu mengenali pola baru dalam data uji. Penurunan *precision* mengindikasikan bahwa model mengalami lebih banyak kesalahan dalam mengklasifikasikan ulasan negatif sebagai positif,

sementara *recall* tetap tinggi, menunjukkan bahwa model masih cukup baik dalam mengenali ulasan positif.

Dari analisis ini, dapat diketahui bahwa split 70:30 memberikan keseimbangan terbaik antara jumlah data pelatihan dan data uji, sehingga menghasilkan performa yang optimal dalam akurasi, *precision*, *recall*, dan *F1-score*. Split 80:20 tetap kompetitif, tetapi memiliki tantangan dalam *precision* yang sedikit lebih rendah. Split 60:40 mengalami sedikit penurunan performa, kemungkinan karena model menjadi terlalu spesifik pada data pelatihan sehingga kurang fleksibel dalam menghadapi data baru. Dengan demikian, jika tujuan utama adalah mendapatkan stabilitas dan akurasi optimal, maka split 70:30 menjadi pilihan terbaik, karena memberikan hasil terbaik tanpa mengalami penurunan performa yang signifikan.

4.7 Integrasi Penelitian

Dalam Islam, setiap proses yang melibatkan pencarian ilmu, pengambilan keputusan yang bijak, serta upaya perbaikan harus dilandasi dengan prinsip ketelitian, keseimbangan, dan tanggung jawab. Hal ini sejalan dengan tujuan dari penelitian ini yang berfokus pada optimasi *hyperparameter* algoritma *Random forest* melalui *Grid Search* untuk meningkatkan akurasi dalam klasifikasi sentimen publik terhadap ulasan produk iPhone. Penelitian ini mengedepankan proses evaluatif dan selektif dalam pemilihan parameter terbaik, sebagaimana Allah SWT telah menciptakan segala sesuatu dengan penuh hikmah dan ukuran yang presisi. Hal ini tercermin dalam firman-Nya dalam QS. Al-Qamar ayat 49:

بِقَدْرِ خَلْقَانَاهُ شَيْءٌ كُلِّ إِنَّا

"Sesungguhnya Kami menciptakan segala sesuatu menurut ukuran." (Q.S. Al-Qamar: 49)

Menurut penafsiran dari Kementerian Agama RI, ayat ini menegaskan bahwa segala sesuatu yang terjadi pada makhluk telah ditentukan oleh Allah. Sesungguhnya, Kami menciptakan segala sesuatu dengan takaran, yakni berdasarkan sistem dan ketetapan yang telah ditentukan sebelumnya (Tafsir Al-Qamar Ayat 49 Kemenag RI, 2023). Dalam konteks penelitian ini, proses optimasi parameter melalui *Grid Search* mencerminkan upaya untuk menemukan konfigurasi yang paling sesuai agar hasil klasifikasi mencapai performa terbaik, sebagaimana segala ciptaan Allah memiliki keseimbangan dan ukuran yang sempurna.

Selain itu, prinsip evaluasi terhadap performa model dan pengujian berulang melalui validasi juga mencerminkan ajaran Islam tentang pentingnya berpikir kritis dan tidak mengikuti sesuatu tanpa dasar yang jelas, sebagaimana tercantum dalam QS. Al-Isra' ayat 36:

مَسْئُولًا عَنْهُ كَانَ أُولَئِكَ كُلُّهُم أَعْمَى وَالْبَصِيرَ أَلْسَمِعَ إِنَّ عِلْمَ بِهِ لَكَ لَيْسَ مَا تَقْفُ وَلَا

"Dan janganlah kamu mengikuti sesuatu yang tidak kamu ketahui, karena pendengaran, penglihatan, dan hati, semuanya akan dimintai pertanggungjawaban." (Q.S. Al-Isra': 36)

Tafsir Kemenag RI menyatakan bahwa ayat ini melarang keras mengikuti sesuatu tanpa dasar ilmu yang sah. Allah mengingatkan bahwa setiap informasi, ucapan, atau tindakan harus berdasarkan pengetahuan yang benar. Pendengaran, penglihatan, dan hati disebutkan secara khusus karena ketiganya merupakan alat

utama manusia dalam memperoleh ilmu dan informasi. Maka, penggunaannya harus dijaga dan akan dimintai pertanggungjawaban di akhirat kelak (Tafsir Al Isra Ayat 36 Kemenag RI, 2024), yang dalam penelitian ini sepadan dengan proses validasi model dan evaluasi kinerja menggunakan metrik klasifikasi seperti akurasi, presisi, dan *recall*. Dengan melakukan analisis berbasis data dan pengujian yang sistematis, hasil penelitian dapat dipertanggungjawabkan dan tidak hanya didasarkan pada asumsi atau dugaan semata.

Penelitian ini juga sejalan dengan nilai Islam dalam menggunakan ilmu pengetahuan sebagai sarana untuk memberikan manfaat kepada masyarakat. Dalam QS. Al-Mujadilah ayat 11, Allah SWT berfirman:

اللَّهُ يَرْفَعُ فَاَنْشُرُوْا اَنْشُرُوْا قِيْلَ وَاِذَا لَكُمْ اللّٰهُ يَفْسَحُ فَاْفْسَحُوْا الْمَجْلِسِ فِي تَفْسَحُوْا لَكُمْ قِيْلَ اِذَا اٰمَنُوْا الدِّيْنَ يٰۤاَيُّهَا

حَبِيْبٌ تَعْمَلُوْنَ بِمَا وَاللّٰهُ دَرَجَتٍ الْعِلْمِ اَوْتُوْا وَالَّذِيْنَ مِنْكُمْ اٰمَنُوْا الدِّيْنَ

" Wahai orang-orang yang beriman, apabila dikatakan kepadamu "Berilah kelapangan di dalam majelis-majelis," lapangkanlah, niscaya Allah akan memberi kelapangan untukmu. Apabila dikatakan, "Berdirilah," (kamu) berdirilah. Allah niscaya akan mengangkat orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat. Allah Mahateliti terhadap apa yang kamu kerjakan." (Q.S. Al-Mujadilah: 11)

Menurut (Tafsir Al Mujadilah Ayat 11 Kemenag RI, 2024), ayat ini mengajarkan pentingnya menciptakan suasana yang harmonis dan penuh penghormatan dalam setiap pertemuan, serta menegaskan bahwa Allah akan mengangkat derajat orang-orang beriman karena keimanan mereka, dan meninggikan derajat orang-orang berilmu karena ilmu mereka menjadi hujah (dalil) yang menerangi umat. Allah Mahateliti terhadap niat, cara, dan tujuan dari setiap

perbuatan manusia. Dalam konteks penelitian ini, penerapan ilmu dalam bidang kecerdasan buatan dan analisis data menjadi salah satu bentuk aktualisasi nilai keislaman, yakni menjadikan ilmu sebagai alat untuk membawa manfaat sosial, membantu masyarakat memahami opini publik, dan mendukung pengambilan keputusan yang lebih baik berdasarkan data yang valid.

Akhirnya, semangat untuk terus belajar, mengembangkan metode yang lebih baik, dan memanfaatkan teknologi secara bertanggung jawab juga sejalan dengan QS. An-Nahl ayat 43:

تَعْلَمُونَ لَا كُنْتُمْ إِنْ الذِّكْرِ أَهْلٍ فَاسْأَلُوا إِلَيْهِمْ تُؤْحِي رِجَالًا إِلَّا قَبْلِكَ مِنْ أَرْسَلْنَا وَمَا

" Kami tidak mengutus sebelum engkau (Nabi Muhammad), melainkan laki-laki yang Kami beri wahyu kepadanya. Maka, bertanyalah kepada orang-orang yang mempunyai pengetahuan jika kamu tidak mengetahui." (Q.S. An-Nahl: 43)

Menurut (Tafsir An Nahl Ayat 43 Kemenag RI, 2024), ayat ini menegaskan bahwa para rasul yang diutus kepada umat manusia adalah laki-laki dari kalangan manusia yang dipilih oleh Allah dan diberi wahyu, bukan dari kalangan malaikat. Maka apabila ada yang meragukan atau tidak mengetahui ajaran Allah, hendaknya bertanya kepada orang-orang yang memiliki pengetahuan, yaitu ahli ilmu dan kitab. Dalam konteks penelitian ini, prinsip ini tercermin dalam pendekatan ilmiah yang dilakukan berdasarkan literatur dan penelitian terdahulu yang valid, serta bimbingan dari para ahli dalam bidang terkait. Dengan demikian, optimasi model melalui *Grid Search* bukan sekadar eksperimen acak, melainkan dilandasi oleh pengetahuan yang sistematis dan bisa dipertanggungjawabkan. Penelitian ini tidak

hanya bernilai secara akademis, tetapi juga mencerminkan nilai-nilai Islam dalam menegakkan kebenaran, tanggung jawab, dan kemanfaatan ilmu pengetahuan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini dilakukan untuk menganalisis efektivitas metode *Random forest* dalam mengidentifikasi sentimen publik terhadap ulasan pengguna iPhone, serta menganalisis pengaruh optimasi *hyperparameter* terhadap hasil klasifikasi. Berdasarkan hasil analisis, beberapa kesimpulan utama dapat disampaikan sebagai berikut:

1. Performa Metode *Random forest* dalam Klasifikasi Sentimen Publik

Berdasarkan hasil penelitian, metode *Random forest* menunjukkan performa yang baik dalam mengklasifikasikan sentimen publik terhadap ulasan pengguna iPhone. Evaluasi menggunakan confusion matrix menunjukkan bahwa model mampu memberikan hasil yang stabil pada berbagai skenario pembagian data. Akurasi terbaik dicapai pada split 70:30, yaitu 83.76%. Selain itu, model yang teroptimasi juga menunjukkan *precision* sebesar 84.97%, *recall* 92.80%, dan *F1-score* 88.71%, yang mencerminkan keseimbangan antara prediksi positif dan negatif yang akurat.

2. Pengaruh Optimasi *Hyperparameter* terhadap Performa Model

Optimasi *hyperparameter* dalam model dengan split 70:30 terbukti meningkatkan performa. Dengan kombinasi terbaik $n_estimators = 200$, $max_features = sqrt$, $max_depth = 600$, $min_samples_split = 5$, $min_samples_leaf = 1$, dari 162 kombinasi *hyperparameter*, rata-rata akurasi validasi silang tertinggi

yang diperoleh dari kombinasi ini adalah 81.53%, menunjukkan bahwa model lebih stabil dan memiliki generalisasi yang lebih baik. Dengan demikian, optimasi *hyperparameter* memberikan peningkatan signifikan pada kualitas prediksi, menjadikan model lebih akurat dalam klasifikasi sentimen publik terhadap ulasan produk iPhone.

5.2 Saran

Berdasarkan proses dan hasil penelitian ini, peneliti menyadari bahwa masih terdapat beberapa keterbatasan yang perlu diperbaiki. Oleh karena itu, diharapkan penelitian selanjutnya dapat melakukan pengembangan dan penyempurnaan agar menghasilkan analisis yang lebih optimal. Beberapa saran untuk penelitian mendatang adalah sebagai berikut:

1. Penggunaan dataset yang lebih beragam, baik dari segi jumlah maupun sumber, untuk menguji kemampuan generalisasi model terhadap data yang memiliki karakteristik berbeda.
2. Pengembangan klasifikasi sentimen menjadi tiga kategori, yaitu positif, netral, dan negatif, agar analisis sentimen menjadi lebih komprehensif.
3. Eksplorasi metode klasifikasi lain seperti *Support Vector Machine* (SVM) atau *Naïve Bayes* sebagai pembanding untuk mengevaluasi keunggulan relatif model yang digunakan.

Pengoptimalan *hyperparameter* dengan pendekatan lain, seperti *Random Search* atau *Bayesian Optimization*, guna memperoleh hasil *tuning* yang lebih efisien.

DAFTAR PUSTAKA

- Abdulkareem, N. M., & Abdulazeez, A. M. (2021). *Machine Learning Classification Based on Radom Forest Algorithm: A Review*. <https://doi.org/10.5281/ZENODO.4471118>
- Aftab, F., Bazai, S. U., Marjan, S., Baloch, L., Aslam, S., Amphawan, A., & Neo, T. K. (2023). A Comprehensive Survey on Sentiment Analysis Techniques. *International Journal of Technology*, 14(6), 1288. <https://doi.org/10.14716/ijtech.v14i6.6632>
- Arham, A., Swedia, E. R., Cahyanti, M., & Septian, M. R. D. (2022). IMPLEMENTASI SENTIMENT ANALYSIS PADA OPINI MASYARAKAT INDONESIA DI TWITTER TERHADAP VIRUS COVID-19 VARIAN OMICRON DENGAN ALGORITMA NAÏVE BAYES, *DECISION TREE*, DAN SUPPORT VECTOR MACHINE. *Sebatik*, 26(2), 565–572. <https://doi.org/10.46984/sebatik.v26i2.1961>
- Artama, M., Sukajaya, I. N., & Indrawan, G. (2020). Classification of official letters using TF-IDF method. *Journal of Physics: Conference Series*, 1516(1), 012001. <https://doi.org/10.1088/1742-6596/1516/1/012001>
- Belharar, O., & Chakor, A. (2021). Internet technologies as a source of consumer information on their consumption. *SHS Web of Conferences*, 119, 03006. <https://doi.org/10.1051/shsconf/202111903006>
- Bichri, H., Chergui, A., & Hain, M. (2024). Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets. *International Journal of Advanced Computer Science and Applications*, 15(2). <https://doi.org/10.14569/IJACSA.2024.0150235>
- Che, C., Hu, H., Zhao, X., Li, S., & Lin, Q. (2023). Advancing Cancer Document Classification with R andom Forest. *Academic Journal of Science and Technology*, 8(1), 278–280. <https://doi.org/10.54097/ajst.v8i1.14333>
- Chicco, D., & Jurman, G. (2022). An Invitation to Greater Use of Matthews Correlation Coefficient in Robotics and Artificial Intelligence. *Frontiers in Robotics and AI*, 9. <https://doi.org/10.3389/frobt.2022.876814>
- Contreras, P., Orellana-Alvear, J., Muñoz, P., Bendix, J., & Célleri, R. (2021). Influence of *Random forest Hyperparameterization* on Short-Term Runoff

Forecasting in an Andean Mountain Catchment. *Atmosphere*, 12(2), Article 2. <https://doi.org/10.3390/atmos12020238>

Dana, A. R., Kristananda, R. V., Wibowo, M. B. S., & Prasetya, D. A. (2024). Perbandingan Algoritma *Decision Tree* dan *Random forest* dengan *Hyperparameter* Tuning dalam Mendeteksi Penyakit Stroke. *Prosiding Seminar Nasional Informatika Bela Negara*, 4, 66–75.

Darioshi, R., & Lahav, E. (2021). The impact of technology on the human DECISION-MAKING process. *Human Behavior and Emerging Technologies*, 3(3), 391–400. <https://doi.org/10.1002/hbe2.257>

Desika. (2024, November 29). *iPhone 13 Masih Jadi Smartphone Terlaris pada 2024*. <https://chatnews.id/read/iphone-13-masih-jadi-smartphone-terlaris-pada-2024>

Fahmy Amin, M. (2022). Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial. *Journal of Engineering Research*, 6(5), 0–0. <https://doi.org/10.21608/erjeng.2022.274526>

Fajri, M., & Primajaya, A. (2023). Komparasi Teknik *Hyperparameter* Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan *Grid Search* dan *Random Search*. *Journal of Applied Informatics and Computing*, 7(1), 14–19. <https://doi.org/10.30871/jaic.v7i1.5004>

G, S. G. C., & B.Sumathi. (2020). *Grid Search* Tuning of *Hyperparameters* in *Random forest* Classifier for Customer Feedback Sentiment Prediction. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(9), Article 9. <https://doi.org/10.14569/IJACSA.2020.0110920>

Gifari, O. I., Adha, Muh., Freddy, F., & Durrand, F. F. S. (2022). Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine. *Journal of Information Technology*, 2(1), 36–40. <https://doi.org/10.46229/jifotech.v2i1.330>

Harjadi, C. N. (2024, Februari 12). *Raih Total \$119.6 Miliar, iPhone Masih Jadi Produk Andalan Apple di Pasar Tekno*. GoodStats. <https://goodstats.id/article/raih-total-1196-miliar-iphone-masih-jadi-produk-andalan-apple-di-pasar-tekno-xM4dO>

Hidayat, F. M., & Sanjaya, H. (2025). ANALISIS SENTIMEN PUBLIK TERHADAP PENJUALAN IPHONE 16 DAN KEBIJAKAN TKDN DI

INDONESIA. *INFOTECH Journal*, 11(1), 74–80.
<https://doi.org/10.31949/infotech.v11i1.13159>

Hien, N. L. H., Tien, T. Q., & Hieu, N. V. (2020). Web Crawler: Design And Implementation For Extracting Article-Like Contents. *Cybernetics and Physics, Issue Volume 9, 2020, Number 3*, 144–151.
<https://doi.org/10.35470/2226-4116-2020-9-3-144-151>

Jackins, V., Vimal, S., Kaliappan, M., & Lee, M. Y. (2021). AI-based smart prediction of clinical disease using *random forest* classifier and Naive Bayes. *The Journal of Supercomputing*, 77(5), 5198–5219.
<https://doi.org/10.1007/s11227-020-03481-x>

Jennifer, D. P., Ningrum, N. C., Pinasty, S., Edma, S. M. N., Andini, W. N., & Widodo, E. (2024). Analisis Sentimen Pada Pengguna Tiktok Menggunakan Metode *Random forest* (Studi Kasus: Jessica-Mirna). *Innovative: Journal Of Social Science Research*, 4(3), Article 3.
<https://doi.org/10.31004/innovative.v4i3.12240>

Kurnianto, E., & Febriawan, D. (2023). Analisis Sentimen Perbedaan Pendapat Netizen Indonesia Terhadap Penutupan Tiktok Shop Menggunakan Algoritma Naïve Bayes. *Jurnal Sistem Komputer dan Informatika (JSON)*, 5(2), 404. <https://doi.org/10.30865/json.v5i2.7170>

Miftahusalam, A., Nuraini, A. F., Khoirunisa, A. A., & Pratiwi, H. (2022). Perbandingan Algoritma *Random forest*, Naïve Bayes, dan Support Vector Machine Pada Analisis Sentimen Twitter Mengenai Opini Masyarakat Terhadap Penghapusan Tenaga Honorer. *Seminar Nasional Official Statistics*, 2022(1), Article 1.
<https://doi.org/10.34123/semnasoffstat.v2022i1.1410>

Miftahusalam, A., Pratiwi, H., & Slamet, I. (2023). Perbandingan Metode *Random forest* dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA Mobile. *SIPTEK: Seminar Nasional Inovasi Dan Pengembangan Teknologi Pendidikan*, 1(1).
<https://proceeding.unesa.ac.id/index.php/siptek/article/view/184>

Nasalsabila, N. (t.t.). *Kamus Alay Github*. <https://github.com/nasalsabila/kamus-alay/blob/master/colloquial-indonesian-lexicon.csv>

Novikova, T., Evdokimova, S., & Wu, G. (2022). Development of a quantitative investment algorithm based on *Random forest*. *Modeling of Systems and Processes*, 15(4), 53–60. <https://doi.org/10.12737/2219-0767-2022-15-4-53-60>

- Palomino, M. A., & Aider, F. (2022). Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis. *Applied Sciences*, 12(17), 8765. <https://doi.org/10.3390/app12178765>
- Rohman, F. N., Farikhin, F., & Surarso, B. (2025). *Hyperparameter* Tuning of *Random forest* Algorithm for Diabetes Classification. *International Journal of Current Science Research and Review*, 08(01). <https://doi.org/10.47191/ijcsrr/V8-i1-31>
- Saepudin, A., Faqih, A., & Dwilestari, G. (2024). Perbandingan Algoritma Klasifikasi Support Vector Machine, *Random forest* dan *Logistic Regression* Pada Ulasan Shopee. *Jurnal Tekno Kompak*, 18(1), Article 1. <https://doi.org/10.33365/jtk.v18i1.3764>
- Sage, A. J., Genschel, U., & Nettleton, D. (2020). Tree aggregation for *random forest* class probability estimation. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13(2), 134–150. <https://doi.org/10.1002/sam.11446>
- Salungweni, B., Weku, W., & Ketaren, E. (2024). ANALISIS PENGARUH FILM “ICE COLD” KASUS KOPI SIANIDA TERHADAP SENTIMEN PENGGUNA YOUTUBE DENGAN SVM DAN *RANDOM FOREST*. *Jurnal TIMES*, 13(2), Article 2. <https://doi.org/10.51351/jtm.13.2.2024755>
- Shams, M. Y., Elshewey, A. M., El-kenawy, E.-S. M., Ibrahim, A., Talaat, F. M., & Tarek, Z. (2023). Water quality prediction using machine learning models based on *Grid Search* method. *Multimedia Tools and Applications*, 83(12), 35307–35334. <https://doi.org/10.1007/s11042-023-16737-4>
- Siahaan, M. (2024, Oktober 29). *Indonesia: Apple share in mobile phone market 2024*. Statista. <https://www.statista.com/statistics/1258390/indonesia-apple-share-in-mobile-phone-market/>
- Siemers, F. M., & Bajorath, J. (2023). Differences in learning characteristics between support vector machine and *random forest* models for compound classification revealed by Shapley value analysis. *Scientific Reports*, 13(1), 5983. <https://doi.org/10.1038/s41598-023-33215-x>
- Singh, K. R., Neethu, K. P., Madhurekaa, K., Harita, A., & Mohan, P. (2021). Parallel SVM model for forest fire prediction. *Soft Computing Letters*, 3, 100014. <https://doi.org/10.1016/j.socl.2021.100014>
- Sipper, M., & Moore, J. H. (2020). Conservation machine learning. *BioData Mining*, 13(1), 9, s13040-020-00220-z. <https://doi.org/10.1186/s13040-020-00220-z>

- 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>
- Syafia, A. N., Hidayattullah, M. F., & Suteddy, W. (2023). Studi Komparasi Algoritma SVM Dan *Random forest* Pada Analisis Sentimen Komentar Youtube BTS. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(3), Article 3. <https://doi.org/10.30591/jpit.v8i3.5064>
- Tafsir Al Hujurat Ayat 6 Kemenag RI. (2024). *Tafsir Al Hujurat Ayat 6*. <https://quran.kemenag.go.id/quran/per-ayat/surah/49?from=1&to=18>
- Tafsir Al Isra Ayat 36 Kemenag RI. (2024). *Tafsir Al Isra Ayat 36*. <https://quran.kemenag.go.id/quran/per-ayat/surah/17?from=36&to=111>
- Tafsir Al-Mujādalah Ayat 11 Kemenag RI. (2024). *Tafsir Al-Mujādalah Ayat 11*. <https://quran.kemenag.go.id/quran/per-ayat/surah/58?from=11&to=22>
- Tafsir Al-Qamar Ayat 49 Kemenag RI. (2023). *Tafsir Al Qamar Ayat 49*. <https://quran.nu.or.id/al-qamar/49>
- Tafsir An-Nahl Ayat 43 Kemenag RI. (2024). *Tafsir An-Nahl Ayat 43*. <https://quran.kemenag.go.id/quran/per-ayat/surah/16?from=43&to=128>
- Toha, A., Purwono, P., & Gata, W. (2022). Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi *Hyperparameter* GridSearch CV. *Buletin Ilmiah Sarjana Teknik Elektro*, 4(1), 12–21. <https://doi.org/10.12928/biste.v4i1.6079>
- Yan, T., Shen, S.-L., Zhou, A., & Chen, X. (2022). Prediction of geological characteristics from shield operational parameters by integrating *Grid Search* and K-fold cross validation into stacking classification algorithm. *Journal of Rock Mechanics and Geotechnical Engineering*, 14(4), 1292–1303. <https://doi.org/10.1016/j.jrmge.2022.03.002>
- Yonatan, A. Z. (2024, Mei 11). *10 Smartphone Paling Laku di 2024, iPhone Nomor Satu!* GoodStats. <https://goodstats.id/article/10-smartphone-paling-laku-di-2024-iphone-nomor-satu-3LNMW>
- Zhu, N., Zhu, C., Zhou, L., Zhu, Y., & Zhang, X. (2022). Optimization of the *Random forest Hyperparameters* for Power Industrial Control Systems Intrusion Detection Using an Improved *Grid Search* Algorithm. *Applied Sciences*, 12(20), Article 20. <https://doi.org/10.3390/app122010456>

- Zhu, T. (2020). Analysis on the Applicability of the *Random forest*. *Journal of Physics: Conference Series*, 1607(1), 012123. <https://doi.org/10.1088/1742-6596/1607/1/012123>
- Zulqornain, J. A., Indriati, I., & Adikara, P. P. (2021). Analisis sentimen tanggapan masyarakat aplikasi tiktok menggunakan metode naïve bayes dan categorial propotional difference (cpd). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(7), 2886–2890.