# IMPLEMENTASI ALGORITMA RANDOM FOREST PADA MODEL KLASIFIKASI SEKUENS DNA MANUSIA DALAM PENYAKIT DIABETES

**SKRIPSI**

**OLEH:**
**MOH NURUL CHOLIL**
**NIM. 210601110048**

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

# IMPLEMENTASI ALGORITMA RANDOM FOREST PADA MODEL KLASIFIKASI SEKUENS DNA MANUSIA DALAM PENYAKIT DIABETES

**SKRIPSI**

**Diajukan Kepada**
**Fakultas Sains dan Teknologi**
**Universitas Islam Negeri Maulana Malik Ibrahim Malang**
**untuk Memenuhi Salah Satu Persyaratan dalam**
**Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh**
**Moh Nurul Cholil**
**NIM. 210601110048**

**PROGRAM STUDI MATEMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAIM MALANG**
**2025**

# IMPLEMENTASI ALGORITMA RANDOM FOREST PADA MODEL KLASIFIKASI SEKUENS DNA MANUSIA DALAM PENYAKIT DIABETES

**SKRIPSI**

**Oleh**
**Moh Nurul Cholil**
**NIM. 210601110048**

Telah Disetujui Untuk Diuji

Malang, 23 Mei 2025

| Dosen Pembimbing I | Dosen Pembimbing II |
|---|---|
| Ari Kusumastuti, M.Pd., M.Si. | Evawati Alisah, M.Pd. |
| NIP. 19770521 200501 2 004 | NIP. 19720604 199903 2 001 |

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

# IMPLEMENTASI ALGORITMA RANDOM FOREST PADA MODEL KLASIFIKASI SEKUENS DNA MANUSIA DALAM PENYAKIT DIABETES

**SKRIPSI**

**Oleh**
**Moh Nurul Cholil**
**NIM. 210601110048**

Telah Dipertahankan di Depan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana (S.Mat)

Malang, 13 Juni 2025

| | | | |
|---|---|---|---|
| Ketua Penguji | : | Dr. Mohammad Jamhuri, M.Si | ..... |
| Anggota Penguji 1 | : | Juhari M.Si | ..... |
| Anggota Penguji 2 | : | Ari Kusumastuti, M.Pd., M.Si | ..... |
| Anggota Penguji 3 | : | Evawati Alisah M.Pd. | ..... |

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

## PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Moh Nurul Cholil

NIM : 2106101110048

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Algoritma Random Forest pada Model Klasifikasi Sekuens DNA Manusia dalam Penyakit Diabetes

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Juni 2025
Yang Membuat Pernyataan,


Moh Nurul Cholil
NIM. 210601110048

## MOTTO

“Tetaplah optimis,
dengan ketidak optimisan itu sendiri”

# PERSEMBAHAN

Dengan rasa syukur yang mendalam, karya ini penulis persembahkan kepada orang tua dan keluarga tercinta yang senantiasa memberikan dukungan penuh, baik secara moral maupun material. Doa, kasih sayang, dan bimbingan yang tulus dari mereka telah menjadi sumber kekuatan dalam setiap langkah dan pencapaian penulis.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada para guru yang telah memberikan ilmu pengetahuan, bimbingan spiritual, serta teladan dalam bersikap dan berpikir. Segala ilmu dan nilai yang diberikan menjadi bekal yang sangat berarti dalam proses penulisan karya ini.

Karya ini juga penulis persembahkan kepada rekan-rekan dan sahabat yang senantiasa memberikan dorongan semangat, motivasi, serta kebersamaan yang berharga. Kehadiran mereka telah memberikan warna dan makna tersendiri dalam perjalanan akademik penulis.

# KATA PENGANTAR

Ucapan terima kasih dituliskan sebagai berikut:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Prof. Dr. Hj. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Elly Susanti, S.Pd., M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Ari Kusumastuti, M.Pd., M.Si., selaku dosen pembimbing I.
5. Evawati Alisah, M.Pd. selaku dosen pembimbing II.
6. Dr. Mohammad Jamhuri, M.Si, selaku penguji utama dalam seminar proposal.
7. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim
8. Orang tua dan seluruh keluarga
9. Seluruh mahasiswa Angkatan 2021.
10. Seluruh guru penulis yang memberikan bimbingan spiritual maupun intelektual.
11. Muhammad Kristover Armand, M. Wildan Nuril Akmal dan Fitria Susanti selaku teman diskusi dalam penelitian ini.

Akhir kata, penulis mengucapkan terima kasih yang sebesar-besarnya atas segala perhatian, bantuan, dan dukungan dari berbagai pihak yang telah berkontribusi dalam penyusunan karya ini. Penulis menyadari bahwa tanpa bantuan dan dukungan tersebut, karya ini tidak akan dapat terselesaikan dengan baik.

Semoga tulisan ini dapat memberikan manfaat bagi pembaca dan menjadi kontribusi positif dalam pengembangan ilmu pengetahuan.

Malang, 13 Juni 2025

Moh Nurul Cholil

# DAFTAR ISI

## DAFTAR TABEL

## DAFTAR GAMBAR

# DAFTAR SIMBOL

| | |
|---|---|
| $T$ | : Banyak pohon keputusan |
| $p_i$ | : Proporsi sampel dari kelas ke-$i$ |
| $c$ | : Jumlah total kelas |
| $n_i$ | : Jumlah sampel pada *child node* ke-$i$ setelah *split* |
| $n$ | : Total jumlah sampel pada *node* induk sebelum *split* |
| $Gini(S_i)$ | : Nilai *gini impurity* dari *child node* ke-$i$ |
| $h_i(x)$ | : Hasil prediksi pada pohon keputusan ke-$i$ |
| $H(x)$ | : Hasil klasifikasi akhir untuk data input $x$ berdasarkan hasil dari seluruh pohon keputusan $h_1(x)$ hingga $h_i(x)$ |

# ABSTRAK

Cholil, Moh Nurul. 2025. Implementasi Algoritma Random Forest pada Model Klasifikasi Sekuens DNA Manusia dalam Penyakit Diabetes. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Ari Kusumastuti, M.Si, M.Pd (2) Evawati Alisah, M.Pd.

Kata Kunci: Random Forest; Sekuens DNA; Diabetes Mellitus

Penelitian ini membahas tentang kalsifikasi sekuens DNA diabetes mellitus dalam machine learning berguna untuk mendeteksi pola penyakit. Dataset yang digunakan berasal dari sekuens gen insulin, yaitu protein penting yang diproduksi tubuh manusia untuk mengatur kadar glukosa darah yang dapat memicu diabetes melitus. Dalam penelitian ini, model klasifikasi dibangun menggunakan algoritma Random Forest, yang dikenal mampu menangani data dengan jumlah fitur besar serta menghasilkan performa yang stabil. Hasil evaluasi menunjukkan bahwa model mencapai akurasi optimal sebesar 93%, dengan menggunakan representasi 3-mers, proporsi data uji sebesar 20%, serta 100 pohon keputusan. Model ini berhasil membedakan dengan baik antara diabetes melitus tipe 1, tipe 2, serta kondisi non-diabetes. Kontribusi utama penelitian ini adalah pengembangan metode prediksi berbasis sekuens DNA yang dapat dimanfaatkan sebagai alat bantu untuk deteksi dini dan penanganan diabetes melitus. Dengan akurasi yang tinggi, model yang dihasilkan berpotensi diterapkan dalam praktik klinis guna meningkatkan kualitas diagnosis dan mendukung perencanaan pengobatan pasien.

# ABSTRACT

Cholil, Moh Nurul. 2025. Implementation of Random Forest Algorithm on Human DNA Sequence Classification Model in Diabetes Disease. Thesis, Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Maulana Malik Ibrahim Malang. Advisors: (1) Ari Kusumastuti, M.Si M.Pd, (2) Evawati Alisah M.Pd.

Keywords: Random Forest; DNA Sequence; Diabetes Mellitus

This study discusses the classification of DNA sequences related to diabetes mellitus using machine learning to help detect disease patterns. The dataset used comes from the insulin gene sequence, which is an important protein produced by the human body to regulate blood glucose levels, where disruption of this function can trigger diabetes mellitus. In this research, the classification model was built using the Random Forest algorithm, which is known for its ability to handle large numbers of features and produce stable performance. The evaluation results showed that the model achieved an optimal accuracy of 93%, using 3-mer representations, 20% test data, and 100 decision trees. The model successfully distinguished between type 1 diabetes, type 2 diabetes, and non-diabetic conditions. The main contribution of this study is the development of a DNA sequence-based prediction method that can be used as a tool to assist in the early detection and treatment of diabetes mellitus. With its high accuracy, the resulting model has the potential to be applied in clinical practice to improve diagnostic quality and support patient treatment planning.

**مستخلص البحث**

خليل، محمد نورول. ٢٠٢٤. تطبيق خوارزمية الغابة العشوائية على نموذج تصنيف تسلسل الحمض النووي البشري في مرض السكري. رسالة بكالوريوس، برنامج دراسة الرياضيات، كلية العلوم والتكنولوجيا، جامعة ،مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرفون: (١) أري كوسوماستيتي، ماجستير في العلوم .ماجستير في التربية؛ (٢) إفاواتي أليسة، ماجستير في التربية

الكلمات المفتاحية: الغابة العشوائية؛ تسلسل الحمض النووي؛ مرض السكري

تناقش هذه الدراسة تصنيف تسلسلات الحمض النووي المرتبطة بمرض السكري باستخدام التعلم الآلي ،للمساعدة في الكشف عن أنماط المرض. وتأتي مجموعة البيانات المستخدمة من تسلسل جين الأنسولين وهو بروتين مهم ينتجه جسم الإنسان لتنظيم مستويات الجلوكوز في الدم، حيث يمكن أن يؤدي تعطيل هذه الوظيفة إلى الإصابة بمرض السكري. تم في هذا البحث بناء نموذج التصنيف باستخدام خوارزمية الغابة العشوائية المعروفة بقدرتها على التعامل مع أعداد كبيرة من الميزات وإنتاج أداء مستقر. وأظهرت نتائج ،٪التقييم أن النموذج حقق دقة مثالية بلغت ٩٣٪، باستخدام تمثيلات مير-٣، وبيانات اختبار ٢٠ و١٠٠ شجرة قرار. نجح النموذج في التمييز بين مرض السكري من النوع الأول، ومرض السكري من النوع الثاني، والحالات غير السكرية. المساهمة الرئيسية لهذه الدراسة هي تطوير طريقة التنبؤ القائمة على تسلسل .الحمض النووي والتي يمكن استخدامها كأداة للمساعدة في الكشف المبكر عن مرض السكري وعلاجه بفضل دقته العالية، يتمتع النموذج الناتج بالقدرة على التطبيق في الممارسة السريرية لتحسين جودة التشخيص .ودعم تخطيط علاج المريض

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan teknologi *machine learning* yang cukup pesat dalam beberapa tahun terakhir telah memungkinkan para peneliti menganalisis data penyakit degenertif yang lebih kompleks dan akurat (Chakraborty dkk., 2024). Hal ini membuka peluang baru dalam penelitian di bidang kesehatan dan bioinformatika. Melalui berbagai teknik dalam *machine learning*, proses diagnosis terhadap berbagai macam penyakit dan gejala dapat dilakukan dengan lebih cepat, efisien, serta dengan biaya klinis yang lebih rendah (McLeish dkk., 2024). Selain itu, pendekatan ini memungkinkan para peneliti untuk melakukan analisis mendalam terhadap molekul biologis, seperti sekuens DNA, yang sering digunakan dalam identifikasi awal penyakit diabetes mellitus (Abnoosian dkk., 2023).

Diabetes melitus sebagai salah satu penyakit degenerative dipilih seebagai fokus pada penelitian ini. Penyakit yang ditandai oleh peningkatan kadar glukosa dalam darah, atau yang dikenal sebagai *hiperglikemia* (Hare & Topliss, 2022). Hal ini dikarenakan gejala eksternal sering kali tidak jelas, sehingga diperlukan pemeriksaan diagnostik yang lebih mendalam untuk memahami kondisi biologis penderita melalui data sekuens DNA-nya. Data tersebut dapat diakses melalui situs NCBI, yang memuat akses data genetik hasil penelitian. Sekuens DNA tersebut terdiri dari kombinasi empat jenis basa nukleotida, yaitu adenin (A), sitosin (C), guanin (G), dan timin (T) (T. Yang dkk., 2023). Data diunduh dalam format .fasta, dengan filter homosapien berjenis nukleotida.

Penelitian ini menggunakan pendekatan klasifikasi data sekuens DNA diabetes menggunakan algoritma *Random Forest*. Algoritma ini menjadi salah satu pilihan utama dalam klasifikasi data genetik, seperti sekuens DNA, karena kemampuannya menangani data berukuran besar dan kompleks. Algoritma ini bekerja dengan menggabungkan banyak *decision tree* dan memproses pemungutan suara mayoritas (*majority vote*) untuk meningkatkan akurasi prediksi, yang membuatnya sangat andal dalam klasifikasi data yang kompleks dan berdimensi tinggi (Parmar dkk., 2019).

Penelitian sebelumnya dari (Amaliah dkk., 2022) menggunakan *Random Forest* dalam klasifikasi varian minuman kopi. Penelitian tersebut menggunakan *Gini Impurity* dalam klasifikasi *Random Forest* seperti yang akan dilakukan pada penelitian ini. Namun, adanya tahapan baru seperti *pre-rpocessing* yang tidak dilakukan pada penelitian sebelumnya. Selain itu, ada penelitian dari (Pahlevi dkk., 2023) yang menggunakan *Random Forest* untuk mengklasifikasikan kelayakan kredit. Penlitian tersebut meneliti nilai kelayakan untuk melakukan sistem kredit. Penelitian dari (Indrayanti dkk., 2017) menggunakan KNN untuk melakukan klasifikasi penyakit diabetes, namun data penyakit yang digunakan bukan sekuens DNA melainkan data publik berupa glukosa, DBP, TSFT, dan sebagainya. Penelitian dari (Ridwan, 2020) menggunakan Naïve Bayes juga melakukan klasifikasi penyakit diabetes tipe 1, tipe 2 dan diabetes *gestasional*.

Metode yang digunakan dalam tahap klasifikasi sekuens DNA dengan *Random Forest* merujuk pada (Ariyadi dkk., 2023). Setiap pohon dalam *Random Forest* dibentuk dari subset data yang dipilih secara acak, dan keputusan akhir dihasilkan melalui voting mayoritas. Namun, terdapat celah dalam penelitian

sebelumnya, terutama pada penanganan *outlier* dan komposisi data latih dan uji yang belum dibahas secara mendalam. Penelitian ini memperkenalkan kebaruan dalam penggunaan *Gini impurity* sebagai alternatif dari *Information Gain* untuk menentukan kriteria pemisahan dalam algoritma *Random Forest*.

Selain aspek teknis dan ilmiah, penelitian ini juga mencerminkan ajaran nilai-nilai dalam Al-Qur'an, yang menekankan keseimbangan dan kemanfaatan. Dalam Surat Al-An’am ayat 141 (Kementerian Agama Republik Indonesia, 2019) yang berbunyi:

وَهُوَ الَّذِي أَنْشَأَ جَنَّاتٍ مَعْرُوشَاتٍ وَغَيْرَ مَعْرُوشَاتٍ وَالنَّخْلَ وَالزَّرْعَ مُخْتَلِفًا أُكُلُهُ وَالزَّيْتُونَ وَالرُّمَّانَ مُتَشَابِهًا وَغَيْرَ مُتَشَابِهٍ ۚ كُلُوا مِنْ ثَمَرِهِ إِذَا أَثْمَرَ وَآتُوا حَقَّهُ يَوْمَ حَصَادِهِ ۖ وَلَا تُسْرِفُوا ۚ إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ

*Artinya: “Dialah yang menumbuhkan tanaman-tanaman yang merambat dan yang tidak merambat, pohon kurma, tanaman yang beraneka ragam rasanya, serta zaitun dan delima yang serupa (bentuk dan warnanya) dan tidak serupa (rasanya). Makanlah buahnya apabila ia berbuah dan berikanlah haknya (zakatnya) pada waktu memetik hasilnya. Akan tetapi, janganlah berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang-orang yang berlebih-lebihan.” (Q.S Al-An’am ayat 141)*

Menurut Alu Syaikh, ayat di atas merupakan bukti bahwa Allah SWT menciptakan segala sesuatu di dunia ini, termasuk berbagai jenis tanaman dan buah-buahan. Allah menciptakan setiap jenis tanaman berdasarkan karakteristiknya sendiri, ada tanaman yang tumbuh merambat dan ada pula yang tidak. Meskipun zaitun dan delima memiliki bentuk yang serupa, rasa yang dihasilkan oleh keduanya sangatlah berbeda. Hal ini mencerminkan keanekaragaman ciptaan Allah dan pentingnya memahami setiap aspek ciptaan-Nya. Dalam konteks penelitian kesehatan, pengetahuan ini dapat diterapkan untuk memahami variasi biologis,

termasuk klasifikasi sekuens DNA manusia, yang berperan dalam pengidentifikasian dan penanganan penyakit seperti diabetes (Alu Syaikh, 2008).

Hal ini menunjukkan bahwa klasifikasi, atau pengkategorian berdasarkan kesamaan ciri dan sifat, merupakan bagian integral dari proses penciptaan yang terjadi di alam semesta. Klasifikasi memiliki peran penting dalam berbagai aspek kehidupan, termasuk dalam bidang ilmu pengetahuan dan kesehatan. Dalam ilmu pengetahuan, klasifikasi membantu para peneliti untuk memahami dan menganalisis berbagai fenomena, termasuk klasifikasi sekuens DNA dan molekul biologis. Pendekatan ini sangat penting dalam konteks kesehatan, di mana pemahaman tentang variasi genetik dapat mendukung identifikasi dan penanganan penyakit, seperti diabetes. Dengan demikian, implementasi algoritma *Random Forest* dalam klasifikasi sekuens DNA manusia berpotensi memberikan kontribusi signifikan terhadap pengembangan solusi medis yang lebih efektif (Govender dkk., 2022).

Penelitian ini bertujuan untuk mengembangkan model klasifikasi sekuens DNA manusia menggunakan algoritma *Random Forest* dengan pendekatan *Gini impurity* untuk prediksi risiko diabetes mellitus. Melalui pendekatan ini, diharapkan penelitian dapat memberikan kontribusi nyata dalam meningkatkan akurasi deteksi dini diabetes dan mendorong pengembangan metode diagnostik berbasis genetik. Pendekatan ini juga diharapkan dapat membuka wawasan baru dalam penggunaan *machine learning* untuk kesehatan dengan tetap memegang prinsip keseimbangan dan kemanfaatan, sesuai dengan ajaran Al-Qur'an.

## 1.2 Rumusan Masalah

Bedasarkan uraian pada latar belakang sebelumnya, rumusan masalah yang diambil dalam penelitian ini adalah sebagai berikut: (berikan tujuan kebaruan, tentang model prediksi mesin)

1. Bagaimana tahapan dalam mengembangkan model klasifikasi sekuens DNA manusia gen insulin menggunakan algoritma *Random Forest*?
2. Bagaimana hasil dan evaluasi dari model klasifikasi sekuens DNA pada penderita diabetes mellitus menggunakan algoritma *Random Forest*?

## 1.3 Tujuan Penelitian

Tujuan penelitian adalah hal yang penting dalam penelitian. Oleh karena itu, berdasarkan uraian mengenai rumusan masalah yang telah disampaikan sebelumnya, tujuan dari penelitian ini dapat dinyatakan sebagai berikut:

1. Mengembangkan model klasifikasi sekuens DNA manusia terkait diabetes mellitus menggunakan algoritma *Random Forest* untuk mengidentifikasi tahap-tahap utama dalam proses klasifikasi dan menganalisis pola genetik yang berhubungan dengan diabetes.
2. Mengevaluasi efektivitas model klasifikasi sekuens DNA manusia pada penderita diabetes mellitus menggunakan *Random Forest*.

## 1.4 Manfaat Penelitian

Melaksanakan sebuah penelitian harus selaras dengan manfaat yang akan diperoleh di masa mendatang. Hal ini menunjukkan bahwa penelitian yang dilakukan tidak hanya bersifat teoritis, tetapi juga diharapkan dapat memberikan

dampak positif dalam kehidupan nyata. Oleh karena itu, berdasarkan tujuan yang telah diuraikan di atas, penelitian ini diharapkan dapat memberikan kontribusi dan manfaat kepada pembaca dalam hal-hal sebagai berikut:

1. Dengan memahami dan menerapkan tahapan proses klasifikasi sekuens DNA manusia pada penderita diabetes mellitus menggunakan algoritma *Random Forest*, diharapkan dapat dihasilkan data yang terstruktur dan berkualitas, sehingga meningkatkan akurasi dalam pembuatan model klasifikasi untuk penderita diabetes mellitus.
2. Dengan mengetahui hasil evaluasi dari model klasifikasi sekuens DNA manusia pada penderita diabetes mellitus yang menggunakan algoritma *Random Forest*, penelitian ini diharapkan dapat memberikan gambaran mengenai akurasi dan efektivitas model dalam mengidentifikasi dan mengklasifikasikan DNA penderita diabetes mellitus.

## 1.5 Batasan Masalah

Untuk mencegah terjadinya perluasan masalah, maka penelitian ini menggunakan batasan-batasan masalah sebagai berikut:

1. Data yang digunakan adalah data sekuens DNA manusia gen insulin yang telah terlabeli dalam diabetes mellitus tipe 1 (DMT 1), tipe 2 (DMT 2), dan non-diabetes mellitus (NON-DM).
2. Nilai k-*mers* yang digunakan dalam tahap pemotongan DNA menjadi substring adalah 3-*mers*, 4-*mers*, 5-*mers*.
3. Perancangan model klasifikasi menggunakan algoritma *machine learning Random Forest* dengan nilai *n estimator* = 10, 50 dan 100.

4. Proses validasi dilakukan menggunakan *Repeated Stratified K-Fold Cross Validation*, dimana nilai *fold* yang digunakan adalah 5.

## 1.6 Definisi Istilah

Terdapat beberapa istilah yang digunakan pada penelitian ini diantaranya:

| | |
|---|---|
| *Machine learning* | : Ilmu yang mempelajari pembuatan sistem komputer yang dapat belajar dari data. |
| *Random Forest* | : algoritma *machine learning* berbasis *ensemble* yang menggunakan beberapa pohon keputusan. |
| *Decision Tree* | : struktur pohon yang digunakan untuk membuat Keputusan. |
| Data *Training* | : Kumpulan data yang digunakan untuk melatih model klasifikasi. |
| Data *Testing* | : Kumpulan data yang digunakan untuk melatih model klasifikasi. |
| K-*mers* | : Kombinasi pasangan nukleotida yang menyusun DNA manusia |
| *Outlier* | : Data yang memiliki nilai atau pola yang jauh berbeda dari mayoritas data dalam sebuah kumpulan data. |
| *Overfitting* | : Kondisi di mana model terlalu menyesuaikan diri dengan data latih, sehingga performa model pada data baru atau data uji menjadi kurang baik. |

# BAB II

# KAJIAN TEORI

## 2.1 Teori Pendukung

### 2.1.1 *Ensemble Learning*

*Ensemble learning* merupakan metode yang menggabungkan beberapa model pembelajaran untuk meningkatkan akurasi dan kestabilan hasil prediksi. Konsep dasar dari *ensemble learning* adalah bahwa kombinasi dari berbagai model dapat menghasilkan kinerja yang lebih baik dibandingkan dengan model tunggal. Metode ini berfungsi dengan memanfaatkan kekuatan masing-masing model untuk mengatasi kelemahan yang ada, sehingga menghasilkan prediksi yang lebih akurat (Mahajan dkk., 2023).

Ada beberapa teknik *ensemble learning* yang umum digunakan, antara lain *bagging*, *boosting*, dan *stacking*. *Bagging*, atau *bootstrap aggregating*, bekerja dengan mengambil sampel acak dari dataset dan melatih model secara terpisah pada setiap sampel. Hasil dari model-model ini kemudian digabungkan untuk memberikan prediksi akhir. Teknik ini sangat efektif dalam mengurangi varians dan meningkatkan stabilitas model (Tüysüzoğlu & Birant, 2020).

Sementara itu, *boosting* merupakan metode yang berfokus pada peningkatan akurasi model dengan memberikan bobot lebih pada kesalahan yang dibuat oleh model sebelumnya (Ganie dkk., 2023). Dengan cara ini, model baru dilatih untuk memperbaiki kesalahan yang dilakukan oleh model sebelumnya, sehingga menghasilkan model akhir yang lebih kuat. *Stacking*, di sisi lain, melibatkan pelatihan beberapa model secara bersamaan dan menggunakan

prediksi dari model-model tersebut sebagai input untuk model lain, yang dikenal sebagai meta-learner (Park dkk., 2022).

Keunggulan *ensemble learning* telah terbukti dalam berbagai bidang, termasuk dalam mendeteksi interaksi gen-gen pada kanker kolorektal. Dalam penelitian yang dilakukan oleh, penerapan *ensemble learning* menunjukkan kemampuan yang lebih baik dalam mendeteksi interaksi gen yang relevan, dibandingkan dengan metode tunggal. Hal ini menunjukkan bahwa *ensemble learning* dapat menjadi alat yang efektif dalam analisis data kompleks, seperti yang sering ditemukan dalam studi genomik (Chan dkk., 2021).

### 2.1.2 *Decision Trees*

Pohon keputusan (*Decision Trees*) merupakan salah satu algoritma fundamental dalam *machine learning* yang digunakan untuk tugas klasifikasi dan regresi. Model ini bekerja dengan membagi dataset menjadi subset yang lebih kecil berdasarkan kriteria tertentu, membentuk struktur yang menyerupai pohon. Setiap *node* dalam pohon ini mewakili fitur, sedangkan cabang-cabangnya menggambarkan keputusan atau hasil yang diambil berdasarkan nilai dari fitur tersebut (Alpaydin, 2020).

Proses pembangunan pohon keputusan dimulai dengan pemilihan fitur yang paling relevan untuk memisahkan data. Kriteria pemilihan ini biasanya menggunakan ukuran *impurity*, seperti *Gini impurity* atau *entropy*, yang membantu menentukan seberapa baik suatu fitur dapat memisahkan kelas dalam dataset. Fitur yang dapat memberikan pemisahan terbaik akan dipilih sebagai *node* pada tingkat tertentu dari pohon. Proses ini diulang hingga kriteria penghentian

tertentu tercapai, misalnya, ketika semua data pada *node* tersebut termasuk dalam kelas yang sama atau ketika kedalaman pohon mencapai batas yang telah ditentukan (Kotu & Deshpande, 2018).

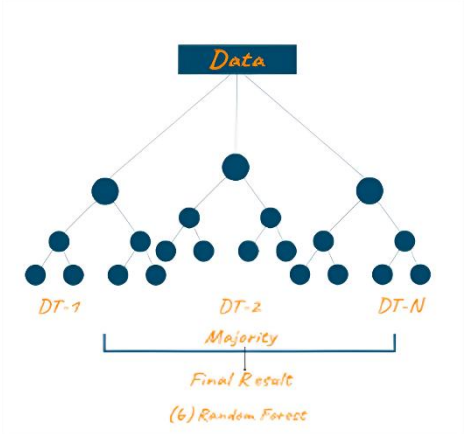
Keunggulan utama dari pohon keputusan adalah kemudahan dalam interpretasi dan visualisasi. Model ini memungkinkan pengguna untuk memahami bagaimana keputusan dibuat, sehingga sangat berguna dalam konteks di mana transparansi sangat penting, seperti dalam bidang kesehatan dan keuangan. Namun, pohon keputusan juga memiliki kelemahan, seperti kecenderungan untuk mengalami *overfitting*, terutama ketika pohon terlalu dalam atau kompleks. Untuk mengatasi hal ini, teknik pruning dapat diterapkan, yang bertujuan untuk memangkas cabang yang tidak diperlukan dan menyederhanakan model (Alpaydin, 2020).

Dalam praktiknya, pohon keputusan dapat digunakan dalam berbagai aplikasi, seperti diagnosis medis, penilaian risiko kredit, dan pengelompokan pelanggan. Kekuatan dan fleksibilitasnya dalam menangani data kategori dan numerik menjadikan pohon keputusan salah satu metode yang populer dalam analisis data (Kotu & Deshpande, 2018).

### 2.1.3 *Random Forest*

Teknik pembelajaran mesin yang dikenal dengan sebutan *Random Forest* merupakan metode yang memanfaatkan data dari berbagai pohon keputusan untuk mencapai hasil yang lebih akurat. Dibandingkan dengan pendekatan *Classification and Regression Tree* (CART), algoritma ini menawarkan sejumlah keunggulan yang signifikan. Sebagai sebuah pengembangan dari metode CART,

*Random Forest* beroperasi dengan cara membentuk beberapa pohon keputusan yang berbeda. Proses ini melibatkan pengambilan sampel acak dari dataset untuk menghasilkan pohon-pohon yang beragam, yang selanjutnya akan digunakan untuk menguji data baru. Hasil akhir dari prediksi *Random Forest* ditentukan melalui suara mayoritas yang diberikan oleh semua pohon keputusan yang ada, sehingga meningkatkan akurasi hasil klasifikasi.




Gambar 2.1 Pohon Keputusan Pada Algoritma *Random Forest*
Sumber: Ariyadi dkk., 2023

Langkah awal dalam algoritma *Random Forest* adalah membentuk subset data pelatihan untuk setiap pohon keputusan. Dari dataset awal dengan $N$ sampel, akan dibangun sejumlah $T$ pohon keputusan. Setiap pohon dibangun menggunakan subset data yang diambil secara acak dan beberapa pohon keputusan bisa dipilih lebih dari sekali.

Setelah subset data terbentuk, selanjutnya menghitung setiap pohon keputusan untuk dilakukan klasifikasi menggunakan model *Random Forest* dengan *gini impurity*. Rumus untuk menghitung *Gini Impurity* pada suatu himpunan $S$ dengan $c$ kelas adalah:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{2.1}$$

Di mana $p_i$ adalah proporsi sampel dari kelas ke-$i$ di dalam himpunan $S$ dan $c$ adalah jumlah total kelas. Nilai *Gini* mendekati 0 jika semua sampel dalam *node* termasuk dalam satu kelas (*node* murni atau *leaf node*), dan mendekati 0.5 atau lebih tinggi jika sampel dari berbagai kelas terdistribusi secara merata di dalam *node* (Menze dkk., 2009). Setelah dilakukan *split* terhadap suatu *node* berdasarkan fitur tertentu, maka nilai rata-rata setelah *split* dapat dihitung menggunakan *Gini Split* dengan rumus:

$$Gini_{split} = \sum_{i=1}^{c} \left(\frac{n_i}{n}\right) \times Gini(S_i) \tag{2.2}$$

Di mana $n_i$ adalah jumlah sampel pada *child node* ke-$i$ setelah pemisahan (*split*), $n$ adalah total jumlah sampel pada *node* induk sebelum *split* dan $Gini(S_i)$ adalah nilai *Gini Impurity* dari *child node* ke-$i$. Fitur dengan nilai *Gini Split* terendah akan dipilih untuk pemisahan *node*, karena menunjukkan pemisahan terbaik untuk klasifikasi (Lestari dkk., 2024). Setelah semua pohon dalam *Random Forest* terbentuk, algoritma menggabungkan hasil prediksi dari setiap pohon dengan menggunakan teknik voting mayoritas. Setiap pohon memberikan prediksi kelas untuk suatu sampel, dan kelas yang memperoleh suara terbanyak dari seluruh pohon akan menjadi prediksi akhir untuk sampel tersebut. Dalam formulasi matematis, jika terdapat $T$ pohon dalam hutan, dan setiap pohon $T_i$ memberikan prediksi $h_i(x)$ untuk sampel $x$, maka prediksi akhir $H(x)$ dihitung sebagai:

$$H(x) = mode\{h_1(x), h_2(x), \dots, h_i(x)\} \tag{2.3}$$

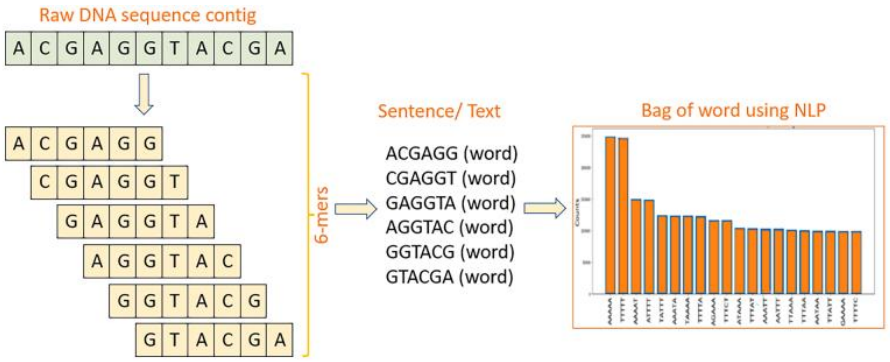
Di mana $H(x)$ merupakan hasil klasifikasi akhir untuk data input $x$ berdasarkan hasil dari seluruh pohon keputusan $h_1(x)$ hingga $h_i(x)$ dalam *Random* Forest. *Mode* adalah nilai yang paling sering muncul di antara prediksi semua pohon untuk sampel tersebut. Teknik voting mayoritas ini membantu meningkatkan akurasi model dengan memanfaatkan kontribusi kolektif dari banyak pohon yang dilatih pada data yang berbeda-beda (Triyana dkk., 2024). Dengan menggabungkan teknik bagging, pemilihan fitur acak, *Gini Index*, dan voting mayoritas, *Random Forest* menjadi model yang andal untuk klasifikasi yang akurat dan stabil. Algoritma ini terbukti efektif dalam aplikasi yang memerlukan analisis data besar dan kompleks, seperti klasifikasi sekuens DNA untuk mendeteksi risiko diabetes mellitus.

### 2.1.4 K-*Mers Encoding*

K-*mers encoding* adalah teknik yang digunakan dalam bioinformatika untuk mengkodekan struktur dari sekuens biologis menjadi substring dengan panjang k. Setiap substring ini merepresentasikan pasangan atau struktur tertentu dalam sekuens tersebut (Alshayeji dkk., 2023). Penggunaan k-*mers* menjadi sangat relevan dalam analisis genom, di mana teknik ini dapat membantu dalam mengidentifikasi pola-pola yang terdapat dalam data sekuens dan memberikan wawasan yang lebih dalam tentang interaksi genetik.

Dalam contoh praktis, pada sekuens DNA yang sederhana seperti ACGT, dapat dihasilkan berbagai jenis k-*mers*. Terdapat empat monomer (A, C, G, dan T), serta kombinasi 2-*mer* (seperti AC, CG, dan GT), 3-*mer* (seperti ACG dan CGT), dan 4-*mer* (yaitu ACGT). Meskipun konsep ini tampak sederhana,

penghitungan k-*mers* dalam dataset yang besar dapat menjadi tantangan, terutama terkait kapasitas memori komputer, karena jumlah substring yang dihasilkan bisa sangat besar.




Gambar 2.2 Ilustrasi K-*mers Encoding* pada sekuens DNA
Sumber: (Alshayeji dkk., 2023)

### 2.1.5 *Count Vectorizer*

*Count Vectorizer* merupakan salah satu teknik yang digunakan dalam pemrosesan teks untuk mengubah teks mentah menjadi representasi numerik yang dapat digunakan dalam model *machine learning*. Teknik ini bekerja dengan menghitung frekuensi kemunculan kata-kata dalam suatu dokumen teks dan kemudian menyusun representasi vektor berdasarkan jumlah tersebut. Setiap elemen dalam vektor mewakili sebuah kata, dan nilainya merefleksikan seberapa sering kata tersebut muncul dalam dokumen yang diolah (Kulkarni & Shivananda, 2019).

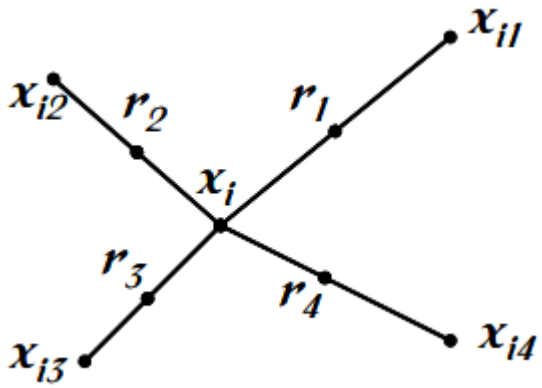
Pendekatan ini sering digunakan dalam berbagai aplikasi, termasuk deteksi ujaran kebencian, di mana *Count Vectorizer* digunakan untuk menganalisis teks dan mengidentifikasi pola tertentu yang berhubungan dengan ujaran kebencian. Misalnya, dalam penelitian oleh (Turki & Roy, 2022), *Count Vectorizer* diterapkan

untuk mengubah teks menjadi fitur numerik yang kemudian diproses oleh algoritma ensemble learning dalam mendeteksi ujaran kebencian secara lebih efektif. Dalam kasus ini, teknik ini tidak hanya memungkinkan pemodelan data teks yang efisien, tetapi juga memungkinkan visualisasi data melalui pembuatan word cloud untuk memahami distribusi kata-kata yang paling sering muncul dalam teks.

Proses *Count Vectorizer* umumnya melibatkan beberapa tahapan, mulai dari tokenisasi kata-kata dalam teks, pemrosesan token seperti penghilangan kata-kata umum (*stopwords*), hingga pembuatan matriks istilah-dokumen (*term-document matrix*). Matriks ini nantinya digunakan untuk menganalisis dokumen teks dan mengubahnya menjadi input yang dapat diolah oleh model *machine learning*. Dengan demikian, teknik ini menjadi alat yang sangat berguna dalam tugas-tugas klasifikasi teks, analisis sentimen, dan deteksi pola linguistik dalam data teks (Turki & Roy, 2022).

### 2.1.6 SMOTE

SMOTE (*Synthetic Minority Oversampling Technique*) merupakan metode yang dirancang untuk mengatasi masalah ketidakseimbangan kelas dalam dataset, yang sering menjadi kendala dalam penerapan algoritma pembelajaran mesin. Ketidakseimbangan kelas dapat mengakibatkan model lebih cenderung untuk memprediksi kelas mayoritas, sehingga mengabaikan kelas minoritas yang sebenarnya mungkin lebih signifikan (Fernandez dkk., 2018). SMOTE bertujuan untuk meningkatkan representasi kelas minoritas dengan menciptakan titik data sintetik, sehingga membantu model dalam belajar dari data yang lebih beragam.




Gambar 2.3 Pembuatan Data Sintesis menggunakan metode SMOTE
Sumber: Fernandez dkk., 2018

Sebuah kelas minoritas dilambangkan sebagai $x_i$, dipilih sebagai basis untuk pembuatan titik data sintetis baru. Berdasarkan metrik jarak, beberapa lingkungan (*neighbors*) terdekat dari kelas yang sama (ditetapkan sebagai titik $x_{i1}$ hingga $x_{i4}$) dipilih dari set pelatihan. Selanjutnya, proses interpolasi acak dijalankan untuk mendapatkan *instance* baru yaitu $r_1$ hingga $r_4$.

Dengan cara ini, SMOTE tidak hanya menambah jumlah contoh dalam kelas minoritas, tetapi juga menciptakan variasi baru yang dapat membantu model dalam belajar dari pola yang lebih representatif. Metode ini telah terbukti efektif dalam meningkatkan akurasi model dalam berbagai aplikasi, termasuk dalam bidang pengenalan pola dan klasifikasi, serta dapat memberikan kinerja yang lebih baik pada dataset yang tidak seimbang (Fernandez dkk., 2018; A. O. Widodo dkk., 2024).

### 2.1.7 *Confusion Matrix*

*Confusion matrix* adalah alat evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan memberikan gambaran mengenai hasil prediksi

dibandingkan dengan nilai aktual dari data uji. Menurut (Sammut & Webb, 2011), *confusion matrix* menyajikan informasi penting tentang sejauh mana model dapat membedakan antara kelas-kelas yang berbeda dalam dataset, sehingga membantu dalam memahami akurasi dan kesalahan model.

Tabel 2.1 Tabel *Confusion Matrix*

| | | Data Prediksi | |
|---|---|---|---|
| | | Positif | Negatif |
| Data Aktual | Positif | *True Positive* (TP) | *False Positive* (FP) |
| | Negatif | *False Negative* (FN) | *True Negative* (TN) |

Komponen utama dari *confusion matrix* diatas adalah:

1. *True Positives* (TP): Jumlah kasus yang diprediksi positif oleh model dan benar-benar positif. Ini menunjukkan kemampuan model dalam mengidentifikasi kelas positif dengan benar.
2. *False Positives* (FP): Jumlah kasus yang diprediksi positif oleh model tetapi sebenarnya negatif. Hal ini mencerminkan kesalahan model dalam mengklasifikasikan kelas negatif sebagai positif.
3. *True Negatives* (TN): Jumlah kasus yang diprediksi negatif oleh model dan benar-benar negatif. Ini menunjukkan seberapa baik model dapat mengenali kelas negatif.
4. *False Negatives* (FN): Jumlah kasus yang diprediksi negatif oleh model tetapi sebenarnya positif. Ini menunjukkan kesalahan model dalam mendeteksi kelas positif.

Melalui komponen utama dari *confusion matrix* yang telah dijelaskan, performa dan kinerja model klasifikasi dapat dievaluasi dengan menghitung beberapa metrik performa. Pertama, Presisi mengukur seberapa baik model dalam memprediksi data kelas positif, yaitu proporsi prediksi positif yang benar dari total prediksi positif. Selanjutnya, *Recall* atau sensitivitas menggambarkan seberapa baik model dapat mengidentifikasi kelas positif dengan benar, yaitu proporsi prediksi positif yang benar dari total aktual positif. Selain itu, Akurasi menggambarkan proporsi prediksi yang benar dari total prediksi, memberikan gambaran umum mengenai kinerja model secara keseluruhan. Terakhir, F1-*Score* merupakan metrik yang menggabungkan presisi dan *recall*, sehingga memberikan informasi yang lebih seimbang mengenai kinerja model, terutama dalam konteks dataset yang tidak seimbang. Keempat metrik tersebut dapat dilihat pada tabel di bawah:

Tabel 2.2 Rumus Metriks Performa pada *Confusion Matrix*

| | |
|---|---|
| Presisi | $\frac{TP}{TP + FP}$ |
| $Akurasi$ | $\frac{TP + TN}{TP + TN + FP + FN}$ |
| *Recall* | $\frac{TP}{TP + FN}$ |
| F1-*Score* | $2 \times \frac{Presisi \times Recall}{Presisi + Recall}$ |

### 2.1.8 ***Stratified* K-*Fold Cross Validation***

*Stratified K-Fold Cross-Validation* merupakan teknik validasi yang efektif dalam pengujian model klasifikasi dengan membagi dataset menjadi $K$ lipatan (*fold*) sambil mempertahankan proporsi kelas di setiap lipatan agar tetap seimbang

seperti dalam dataset asli. Metode ini sangat bermanfaat ketika menghadapi data yang tidak seimbang, sehingga mampu meningkatkan keandalan model yang dihasilkan (Mayangsari dkk., 2023; S. Widodo dkk., 2022). Dalam penerapannya, setiap lipatan digunakan sebagai data uji, sementara $K-1$ lipatan lainnya digunakan untuk pelatihan, dan proses ini diulang $K$ kali untuk memastikan setiap lipatan berfungsi sebagai data uji setidaknya sekali. Agregasi hasil dari $K$ iterasi ini akan memberikan gambaran yang lebih stabil dan representatif mengenai kinerja model secara keseluruhan. Dengan demikian, *Stratified K-Fold Cross-Validation* menjadi pilihan yang lebih baik dibandingkan *K-Fold Cross-Validation* biasa, terutama dalam situasi dengan ketidakseimbangan kelas.

### 2.1.9 Diabetes Mellitus

Diabetes melitus adalah penyakit kronis yang disebabkan oleh ketidakmampuan tubuh dalam memproduksi atau menggunakan insulin secara efektif, sehingga menyebabkan peningkatan kadar glukosa dalam darah (American Diabetes Association, 2021). Menurut (Ogurtsova dkk., 2017), diabetes merupakan salah satu masalah kesehatan global yang paling mendesak, dengan prevalensi yang terus meningkat, diperkirakan akan mencapai lebih dari 642 juta orang di seluruh dunia pada tahun 2040.

Berdasarkan tipe dan penyebabnya, diabetes melitus terbagi menjadi beberapa kategori, yaitu diabetes melitus tipe 1 (DMT1), diabetes melitus tipe 2 (DMT2), diabetes melitus gestasional, dan diabetes monogenik (American Diabetes Association, 2021). Tipe 1 umumnya terjadi akibat kerusakan sel beta pankreas yang bersifat autoimun, yang mengakibatkan ketidakmampuan untuk

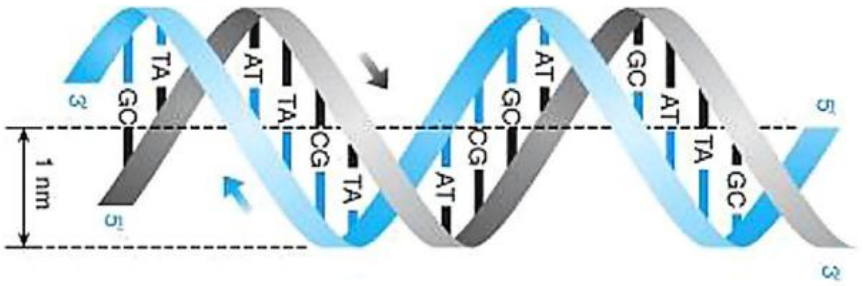
memproduksi insulin. Diabetes tipe 2 adalah tipe yang paling umum, ditandai dengan resistensi insulin dan disfungsi sel beta pankreas (Saeedi dkk., 2019). Diabetes gestasional muncul selama kehamilan dan biasanya hilang setelah melahirkan, tetapi dapat meningkatkan risiko diabetes tipe 2 di kemudian hari. Sementara itu, diabetes monogenik disebabkan oleh mutasi gen tunggal yang memengaruhi pengaturan glukosa.

Gejala diabetes melitus meliputi peningkatan rasa haus (*polidipsia*), sering buang air kecil (*poliuria*), kaburnya penglihatan, dan penurunan berat badan yang signifikan (American Diabetes Association, 2021). Namun, pada diabetes tipe 2, gejalanya sering kali lebih ringan dan bahkan dapat tidak terdeteksi selama bertahun-tahun. Hal ini dapat menyebabkan komplikasi serius jika tidak terdiagnosis, termasuk penyakit jantung, kerusakan saraf, dan gangguan fungsi ginjal (Saeedi dkk., 2019). Untuk mendiagnosis diabetes, beberapa tes dapat dilakukan, termasuk pengukuran glukosa darah puasa, tes toleransi glukosa oral (OGTT), dan pengukuran Hemoglobin A1c (HbA1c) (American Diabetes Association, 2021). Pengelolaan diabetes melitus memerlukan pendekatan multidisipliner yang melibatkan perubahan gaya hidup, manajemen diet, dan terkadang terapi insulin atau obat antidiabetes.

### 2.1.10 Sekuens DNA

Sekuens DNA merujuk pada urutan spesifik nukleotida dalam molekul DNA yang berfungsi sebagai instruksi genetik untuk sintesis protein dan pengaturan berbagai fungsi biologis dalam sel. DNA (asam *deoksiribonukleat*) terdiri dari dua untai yang membentuk heliks ganda, di mana setiap untai terdiri

dari rangkaian nukleotida yang terdiri dari empat jenis, yaitu adenina (A), timina (T), sitosina (C), dan guanina (G) (A. Yang dkk., 2020). Struktur ini memudahkan pengkodean informasi genetik, di mana setiap kombinasi spesifik dari nukleotida dapat mewakili berbagai asam amino yang diperlukan untuk membentuk protein. DNA memiliki struktur dalam heliks ganda seperti pada ilustrasi dibawah.




Gambar 2.4 Rantai Sekuens DNA
Sumber: Ahmad dkk., 2017

Berdasarkan fungsinya, sekuens DNA dapat dibagi menjadi beberapa kategori, termasuk sekuens koding yang berfungsi sebagai template untuk sintesis protein, sekuens non-koding yang tidak menghasilkan protein tetapi memiliki peran penting dalam pengaturan gen, sekuens pengatur yang mengontrol ekspresi gen, serta sekuens ulangan yang terdiri dari pengulangan unit nukleotida. Pemahaman mengenai sekuens DNA sangat penting dalam bidang genetika dan bioteknologi. Dengan mempelajari dan menganalisis sekuens DNA, para peneliti dapat mengidentifikasi variasi genetik yang dapat menyebabkan penyakit, serta mengembangkan terapi gen yang dapat memperbaiki atau mengganti gen yang rusak.

Selain itu, penelitian terbaru dalam pemrosesan data sekuens DNA menggunakan algoritma *machine learning* menunjukkan potensi besar untuk

meningkatkan akurasi analisis sekuens serta memberikan wawasan baru dalam bidang genomik (A. Yang dkk., 2020). Dengan pemanfaatan teknologi ini, proses identifikasi dan klasifikasi sekuens DNA dapat dilakukan dengan lebih efisien, memungkinkan pengembangan metode diagnostik yang lebih cepat dan akurat. Hal ini menunjukkan bahwa pemahaman mendalam tentang sekuens DNA dan aplikasinya di bidang kesehatan dan bioteknologi akan semakin penting di masa depan.

## 2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits

DNA adalah molekul yang sangat penting dalam biologi, karena berfungsi sebagai pengatur sifat dan karakteristik organisme hidup. Dalam hal ini, Al-Qur'an juga memberikan penjelasan yang mendalam tentang penciptaan manusia serta pewarisan sifat melalui firman-Nya. Salah satu ayat yang mengisyaratkan proses penciptaan manusia adalah Surat Al-Qiyamah ayat 37-39 (Kementerian Agama Republik Indonesia, 2019) yang berbunyi:

أَلَمْ يَكُ نُطْفَةً مِنْ مَنِيٍّ يُمْنَى(٣٧) ثُمَّ كَانَ عَلَقَةً فَخَلَقَ فَسَوَّى(٣٨) فَجَعَلَ مِنْهُ الزَّوْجَيْنِ الذَّكَرَ وَالأُنْثَى(٣٩)

*Artinya: "Bukankah Dia (manusia) dahulu setetes mani yang ditumpahkan (ke dalam rahim), 38. kemudian mani itu menjadi segumpal darah, lalu Allah menciptakannya, dan menyempurnakannya, 39. lalu Allah menjadikan dari padanya sepasang: laki-laki dan perempuan. 40." (QS al-Qiyamah: 37-39)*

Menurut Shihab dalam buku tafsirnya menyatakan bahwa penciptaan manusia dimulai saat air mani memasuki rahim dan bertemu dengan indung telur. Proses ini mengarah pada pembentukan 'alaqah, yang terus membelah dan menempel pada dinding rahim. Penjelasan ini menggambarkan betapa Allah SWT,

dengan segala kuasa-Nya, menciptakan manusia dan menyempurnakan proses penciptaan tersebut hingga menjadi bentuk manusia yang sempurna, yaitu terciptanya sepasang laki-laki dan perempuan (Shihab, 2002).

Firman Allah tersebut mengingatkan kita bahwa manusia diciptakan dari setetes air mani yang kemudian berkembang menjadi segumpal darah, hingga akhirnya menjadi janin yang hidup. Proses ini mengisyaratkan bahwa sifat-sifat yang dimiliki oleh orang tua akan diwariskan kepada anaknya. Dalam perspektif genetika, hal ini berkaitan dengan transfer materi genetik (DNA) dari orang tua kepada anak selama proses reproduksi dan kehamilan. Ilmu genetika modern menunjukkan bahwa setiap manusia akan mewarisi informasi genetik dari kedua orang tuanya, meskipun salah satu sifat mungkin bersifat resesif dan tidak terlihat. Materi genetik ini mengandung informasi yang mengatur berbagai sifat fisik, biologis, dan bahkan kondisi terhadap suatu penyakit tertentu.

Proses pewarisan sifat yang dijelaskan dalam ayat tersebut sangat relevan dengan penelitian yang dilakukan dalam klasifikasi DNA, khususnya dalam konteks penyakit diabetes mellitus. Dengan memahami bahwa setiap individu memiliki informasi genetik yang unik, kita dapat lebih baik memahami hubungan antara sekuens DNA dan predisposisi terhadap penyakit. Dalam konteks penelitian ini, pemahaman mengenai hubungan antara genetika dan penyakit dapat membantu kita dalam melakukan analisis dan pengklasifikasian risiko penyakit berdasarkan data sekuens DNA.

Sebagaimana dijelaskan pada ayat tersebut, Al-Qur'an tidak hanya memberikan penjelasan tentang penciptaan dan pewarisan sifat, tetapi juga menunjukkan relevansinya dengan ilmu pengetahuan modern. Penemuan dalam

bidang genetika mendukung dan sejalan dengan ajaran-ajaran yang terdapat dalam Al-Qur'an. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam memahami hubungan antara sekuens DNA dan penyakit diabetes mellitus, serta membuka jalan bagi pengembangan pendekatan yang lebih efektif dalam pengobatan dan pencegahan penyakit ini. Integrasi antara ilmu pengetahuan dan ajaran agama menunjukkan bahwa keduanya dapat saling melengkapi dalam memahami kompleksitas kehidupan manusia.

## 2.3 Kajian Topik dengan Teori Pendukung

Penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasikan sekuens DNA dari individu sehat serta penderita diabetes mellitus tipe 1 dan tipe 2. Sekuens DNA terdiri dari empat macam nukleotida yang membentuk urutan yang kompleks, dengan panjang yang dapat mencapai 32.000 nukleotida. Setiap individu memiliki karakteristik unik pada sekuens DNA mereka, yang dapat memberikan informasi penting tentang predisposisi genetik terhadap berbagai kondisi kesehatan, termasuk diabetes mellitus. Melalui analisis dan klasifikasi sekuens DNA, diharapkan dapat ditemukan pola-pola yang berhubungan dengan penyakit diabetes, yang selanjutnya dapat mendukung diagnosis dan pengobatan yang lebih efektif.

Untuk proses klasifikasi, penelitian ini menggunakan algoritma Random Forest, yang merupakan metode *ensemble* yang menggabungkan banyak pohon keputusan untuk memberikan hasil yang lebih stabil dan akurat. *Random Forest* mampu menangani dataset dengan dimensi tinggi, seperti data sekuens DNA, dengan baik. Algoritma ini tidak hanya meningkatkan akurasi, tetapi juga

membantu dalam mengurangi risiko *overfitting* yang sering terjadi pada model tunggal. Dalam penelitian ini, model *Random Forest* digunakan untuk mengklasifikasikan sekuens DNA berdasarkan karakteristik dari k-mer yang diekstraksi dari sekuens tersebut.

Proses analisis diawali dengan pengumpulan data sekuens DNA dari sumber NCBI, yang terdiri dari tiga kelas: sekuens dari individu sehat, penderita diabetes mellitus tipe 1, dan penderita diabetes mellitus tipe 2. Setelah pengumpulan data, tahap selanjutnya adalah *preprocessing*, yang mencakup pembersihan data, penghapusan outlier menggunakan perulangan for untuk mencari apakah ada nilai yang menyimpang dari A, C, G dan T pada data sekuens DNA manusia, serta transformasi sekuens menjadi k-mer untuk mempermudah analisis. Penggunaan k-mer merupakan metode yang efektif dalam mengidentifikasi pola-pola dalam data sekuens DNA, di mana setiap k-mer mewakili kombinasi dari tiga nukleotida. Setelah proses ekstraksi k-mer, dilakukan representasi fitur menggunakan teknik *bag-of-words* melalui *Count Vectorizer* untuk menghasilkan data numerik yang siap digunakan dalam model *Random Forest*. Mengingat adanya ketidakseimbangan dalam distribusi kelas, *oversampling* dilakukan dengan metode SMOTE untuk menciptakan keseimbangan antara kelas-kelas yang ada. Hal ini penting untuk mencegah model terlatih bias terhadap kelas yang lebih dominan.

Model *Random Forest* kemudian dilatih dengan menggunakan data yang telah diolah, dan evaluasi model dilakukan dengan membagi data menjadi data latih dan data uji. Evaluasi meliputi perhitungan metrik performa seperti akurasi, presisi, recall, dan nilai F1, yang diharapkan memberikan gambaran yang jelas tentang kinerja model dalam mengklasifikasikan sekuens DNA. Selain itu, validasi silang

dengan *Repeated Stratified K-Fold Cross Validation* dilakukan untuk memastikan bahwa model tidak hanya memiliki kinerja yang baik pada satu set data, tetapi juga dapat digeneralisasi dengan baik ke data lain. Melalui penelitian ini, diharapkan dapat diperoleh wawasan yang lebih mendalam mengenai hubungan antara sekuens DNA dan penyakit diabetes mellitus, serta menghasilkan model klasifikasi yang handal untuk mendukung diagnosis yang lebih akurat dan efektif.

# BAB III

# METODE PENELITIAN

## 3.1 Jenis Penelitian

Jenis penelitian yang dilakukan oleh peneliti adalah penelitian kuantitatif yang memfokuskan pada analisis sekuens DNA manusia dengan memanfaatkan algoritma *Random Forest*. Dalam penelitian ini, peneliti akan melakukan perhitungan matematis terhadap data sekuens DNA, di mana hasil yang diperoleh akan disimpulkan berdasarkan nilai numerik. Diharapkan, nilai-nilai ini dapat memudahkan pemahaman, perbandingan, dan penjelasan mengenai akurasi model klasifikasi yang dibuat. Proses penelitian akan mengikuti tahapan yang sistematis dan terstruktur. Data yang digunakan adalah sekuens DNA yang akan dikonversi dari format string menjadi data numerik, sebelum akhirnya diklasifikasikan menggunakan algoritma *Random Forest*.

## 3.2 Sumber Data

Data yang digunakan dalam penelitian ini diperoleh dari situs NCBI dan telah tersedia dalam format .csv, yang dapat diakses melalui tautan Github berikut: https://github.com/NurlChl/dna-sequence-skripsi. Data ini mencakup tiga kelompok sekuens DNA, yaitu DNA manusia sehat yang tidak menderita diabetes mellitus, DNA penderita diabetes mellitus tipe 1, dan DNA penderita diabetes mellitus tipe 2. Setiap entri data terdiri dari informasi sekuens DNA lengkap dengan panjang tertentu serta kelas yang mengidentifikasi kelompok data, yaitu sehat, diabetes tipe 1, atau diabetes tipe 2. Data final yang siap diolah terdiri dari 989

sekuens DNA manusia sehat, 145 sekuens DNA penderita diabetes mellitus tipe 1, dan 443 sekuens DNA penderita diabetes mellitus tipe 2.

## 3.3 Perancangan Sistem

Sistem yang akan dirancang dalam penelitian ini adalah sistem yang mampu mengidentifikasi berbagai sekuens DNA manusia, baik yang mengidap diabetes mellitus maupun yang tidak. Sistem ini akan melakukan pengelompokan berdasarkan analisis sekuens DNA mentah yang dimasukkan ke dalam model klasifikasi. Proses identifikasi akan menggunakan algoritma *Random Forest*, yang dirancang untuk melakukan perhitungan matematis guna menentukan klasifikasi yang tepat. Selanjutnya, sistem ini akan memproses dan mengolah data sekuens DNA sehingga dapat menghasilkan informasi yang akurat mengenai kelompok sekuens DNA yang telah dianalisis.

## 3.4 *Preprocessing* Data

1. Deteksi *Oulier*

   Deteksi *outlier* pada sekuens DNA dalam penelitian ini dilakukan untuk memastikan bahwa data yang digunakan bersih dan valid sebelum masuk ke tahap analisis lebih lanjut. *Outlier* dalam konteks ini didefinisikan sebagai sekuens DNA yang mengandung karakter selain empat basa nitrogen penyusun DNA, yaitu Adenin (A), Sitosin (C), Guanin (G), dan Timin (T). Proses deteksi dilakukan menggunakan metode pemrograman dengan perulangan *for*, di mana setiap sekuens DNA diperiksa satu per satu. Setiap karakter dalam sekuens akan dicek apakah termasuk dalam himpunan karakter

valid. Jika ditemukan karakter selain A, C, G, atau T, maka sekuens tersebut dianggap sebagai *outlier* dan tidak digunakan dalam proses analisis. Metode ini bersifat deterministik dan sederhana namun efektif dalam menjaga kualitas dan integritas data sebelum digunakan dalam proses klasifikasi atau pemodelan.

2. K-*mers Encoding*

   Setelah tahap pengecekan outlier, sekuens DNA yang telah dipastikan valid kemudian diproses menggunakan teknik pemotongan k-*mers*, yaitu metode yang membagi sekuens DNA menjadi potongan-potongan kecil (substring) berukuran tetap, dalam hal ini 3 nukleotida atau disebut 3-*mers*. Teknik ini bertujuan untuk mengubah data sekuens DNA menjadi format yang lebih terstruktur dan seragam, sehingga lebih mudah diolah oleh algoritma pemodelan dan klasifikasi. Setiap 3-*mers* dihasilkan dengan cara menggeser satu posisi ke kanan dari awal hingga akhir sekuens, tanpa melewati batas panjang sekuens. Misalnya, untuk sekuens AACGTAAGTTCGATGAA, hasil encoding 3-*mers*-nya adalah AAC, ACG, CGT, GTA, TAA, AAG, AGT, GTT, TTC, TCG, CGA, GAT, ATG, TGA, dan GAA. Representasi 3-*mers* ini tidak hanya mempertahankan urutan lokal dari nukleotida dalam DNA, tetapi juga membantu model mengenali pola-pola biologis penting dalam data, menjadikannya teknik yang umum dan efektif dalam bioinformatika..

3. *Count Vectorizer*

   Tahap selanjutnya dalam proses pengolahan data adalah transformasi sekuens DNA ke dalam bentuk vektor numerik menggunakan teknik *Count Vectorizer*. Setelah sekuens DNA dipotong menjadi potongan-potongan kecil berukuran

tiga nukleotida (3-*mers*), potongan-potongan tersebut kemudian diubah menjadi representasi numerik dengan bantuan kelas *CountVectorizer* dari *library Scikit-learn*. *Count Vectorizer* bekerja melalui dua tahap utama, yaitu pembentukan kamus dan pembentukan matriks fitur. Pertama, seluruh 3-*mers* yang terdapat dalam dataset dikumpulkan untuk membentuk sebuah kamus berisi kumpulan 3-*mers* unik, di mana setiap 3-*mers* diberikan indeks numerik yang bersifat unik. Kedua, dilakukan pembentukan matriks fitur, di mana setiap baris pada matriks merepresentasikan satu sekuens DNA dan setiap kolom merepresentasikan salah satu 3-*mers* dari kamus. Nilai pada setiap sel dalam matriks menunjukkan frekuensi kemunculan 3-*mers* tertentu dalam sekuens tersebut. Dengan demikian, seluruh sekuens DNA dapat direpresentasikan sebagai vektor numerik berdimensi tetap, yang selanjutnya dapat digunakan sebagai input dalam proses pemodelan menggunakan algoritma *machine learning*.

Tabel 3.1 Contoh Data Sekuens DNA Manusia

| Kelas | Sekuens |
|---|---|
| DMT 1 | ATC, TCG, CGA, GAT, ATC, TCG |
| DMT 2 | CGA, GAT, ATC, TCG, GAT |
| Non-DM | ATC, TCG, CGA, GAT, ATC, TCG, CGA, GAT, ATC, TCG |

Dari Tabel 3.1 dimisalkan sekuens DNA telah dilakukan pemotongan k-*mers* sebanyak 3-*mers* dengan kelas DMT 1, DMT 2 dan Non-DM. Lalu akan dilakukan tahap *Count Vectorizer* dengan menghitung banyak tiap fitur dalam setiap sekuens. Seperti pada sekuens pertama pada fitur ATC ada sebanyak 2

fitur, sehingga hasil *Count Vectorizer* pada fitur ATC di sekuens pertama adalah 2. maka diperoleh hasil dari tahap *Count Vectorizer* nya yaitu:

Tabel 3.2 Contoh Proses Count Vectorizer

| Sekuens | ATC | TCG | GAT | CGA |
|---|---|---|---|---|
| 1 | 2 | 2 | 1 | 1 |
| 2 | 1 | 1 | 2 | 1 |
| 3 | 3 | 3 | 2 | 2 |

Pada Tabel 3.2 kolom sekuens menunjukkan banyak sekuens pada data. Dan setiap sekuens memiliki fitur dari hasil pemotongan k-*mers* yaitu ATC, TCG, GAT dan CGA. Dan nilai masing-masing kolom pada fitur menunjukkan berapa banyak fitur ATC, TCG, GAT dan CGA dalam sekuens tersebut. Sehingga nilai pada setiap fitur inilah yang akan digunakan dalam model *Random Forest*.

4. Label *Encoding*

Tahap selanjutnya dalam proses prapemrosesan data adalah label *encoding*, yaitu proses mengubah label kelas dari sekuens DNA manusia ke dalam bentuk numerik. Label *encoding* diperlukan karena algoritma *machine learning* umumnya hanya dapat mengolah data dalam bentuk angka, bukan data kategorikal berupa teks. Dalam penelitian ini, setiap sekuens DNA memiliki label kelas yang menunjukkan kategori diabetes, yaitu DMT1, DMT2, dan Non-DM. Melalui proses label *encoding*, ketiga label tersebut akan dikonversi menjadi nilai numerik, sehingga DMT1 dikodekan menjadi 0, DMT2 menjadi 1, dan Non-DM menjadi 2. Proses ini dilakukan menggunakan *LabelEncoder* dari *library Scikit-learn*, yang secara otomatis mengubah nilai

kategorikal menjadi angka berdasarkan urutan kemunculannya. Hasil dari tahap ini adalah sebuah vektor label dalam format numerik yang sesuai untuk digunakan sebagai target (output) dalam proses pelatihan model *machine learning*.

5. *Oversampling*

   Tahap *oversampling* dilakukan untuk mengatasi masalah ketidakseimbangan jumlah data antar kelas dalam dataset sekuens DNA manusia, khususnya antara kelas DMT1, DMT2, dan Non-DM. Ketidakseimbangan ini dapat menyebabkan model pembelajaran mesin menjadi bias terhadap kelas mayoritas dan mengabaikan kelas minoritas. Oleh karena itu, dalam penelitian ini digunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) sebagai metode oversampling. SMOTE bekerja dengan membandingkan kelas minoritas terhadap kelas mayoritas, kemudian menghasilkan data sintetik baru yang serupa dengan data asli kelas minoritas. Data sintetik ini dibuat dengan menginterpolasi antar titik-titik data minoritas terdekat dalam ruang fitur, bukan dengan melakukan duplikasi langsung. Kelebihan dari metode ini adalah mampu memperluas representasi data kelas minoritas secara lebih alami dan bervariasi, sehingga mengurangi risiko *overfitting*. Dengan diterapkannya SMOTE, distribusi jumlah data antar kelas menjadi lebih seimbang, yang memungkinkan model untuk belajar secara adil dari ketiga kelas dan meningkatkan performa klasifikasi secara keseluruhan..

## 3.5 Teknik Analisis Data dan Klasifikasi

### 3.5.1 Implementasi Matematika *Random Forest*

Klasifikasi dilakukan menggunakan metode *Random Forest* dengan membentuk pohon acak untuk memprediksi setiap sekuens DNA manusia dari data yang dipakai. Sekuens DNA akan diambil beberapa sampel DNA dari data yang dipakai untuk dilakukan menggunakan perhitungan secara manual. Berikut langkah-langkah klasifikasi *Random Forest* secara manual:

1. Membentuk pohon keputusan $T_i$

   Setelah melakukan *pre-processing* tahap selanjutnya dalam melakukan klasifikasi menggunakan *Random Forest* secara manual dilakukan dengan membentuk pohon keputusan $T_i$. Setiap pohon keputusan berisi lebih dari 1 sekuens DNA manusia yang dipilih secara acak.

2. Melakukan *split* fitur dan prediksi

   Setelah pohon keputusan dibentuk, selanjutnya setiap pohon keputusan akan dihitung menggunakan Persamaan (2.1) dan Persamaan (2.2) untuk mendapatkan nilai *split* pada setiap fitur yang ada di pohon keputusan $T_i$. Dalam menghitung $Gini_{split}$ diperlukan nilai threshold dengan rumus rata-rata dari dua nilai yang sudah di sorting secara berurutan dari yang terkecil hingga terbesar pada setiap sekuens DNA dalam fitur ke-$i$ di pohon keputusan $T_i$. Selanjutnya akan dipilih fitur dengan nilai *split* paling rendah untuk digunakan sebagai fitur utama pada klasifikasi pohon $T_i$. Proses klasifikasi dilakukan dengan langkah yang sama yaitu menggunakan Persamaan (2.1) dan Persamaan (2.2). Jika *node* kiri dan *node* kanan mencapai *leaf node* (bernilai 0), maka proses klasifikasi selesai dan dilanjut

pada pohon keputusan selanjutnya. Jika salah satu *node* belum mencapai *leaf node* maka dilanjutkan perhitungan selanjutnya menggunakan fitur terkecil lain setelah fitur utama dan akan dihitung menggunakan Persamaan (2.1) dan Persamaan (2.2) hingga mencapai kondisi *leaf node*. Hasil dari klasifikasi setiap fitur pada masing-masing pohon keputusan akan memberikan prediksi untuk penyakit DMT1, DMT2 dan Non-DM.

3. Voting Mayoritas

   Setelah tahap klasifikasi selesai dengan memperoleh hasil prediksi pada setiap pohon keputusan, selanjutnya akan dilakukan voting maoritas untuk menentukan hasil prediksi setiap sekuens DNA manusia menggunakan voting terbanyak dari masing-masing pohon keputusan.

### 3.5.2 Implementasi Program *Random Forest*

Klasifikasi pada penelitian ini, selain diimplementasikan perhitungan matematika, lebih lengkapnya akan dilakukan dengan menggunakan program *Python*. Klasifikasi *Random Forest* ini dilakukan dengan membagi data *training* dan data *testing* menjadi $80:20$, $70:30$ atau $90:10$. Setelah data dibagi menjadi dua, selanjutnya data *training* akan dilatih menggunakan model *Random Forest* pada *library Sklearn* yaitu *Random Forest Classifier* dengan memberikan parameter banyak pohon keputusan yang akan dipakai. Selanjutnya model akan melakukan prediksi menggunakan data *testing*. Performa model akan ditunjukkan pada tahap evaluasi menggunakan *Confusion Matrix*.

### 3.5.3 Proses Evaluasi

Proses evaluasi bertujuan untuk mengukur kinerja model *Random Forest* dalam mengklasifikasikan sekuens DNA pada kasus diabetes mellitus. Evaluasi dilakukan menggunakan *Confusion Matrix*, yang mencakup metrik utama seperti akurasi, presisi, *recall*, dan F1-*score*. Akurasi dihitung untuk menentukan persentase prediksi yang benar terhadap seluruh data uji, sementara presisi mengukur sejauh mana prediksi kelas positif yang dihasilkan oleh model benar adanya. *Recall* digunakan untuk menilai sejauh mana model dapat mengenali sampel yang benar-benar termasuk dalam kelas positif, sedangkan F1-*score*, sebagai rata-rata harmonis antara presisi dan *recall*, memberikan evaluasi yang lebih seimbang terhadap performa model terutama dalam kondisi data yang tidak seimbang.

Evaluasi ini dilakukan setelah proses pengujian, di mana data uji yang tidak digunakan dalam pelatihan diterapkan pada model untuk menghasilkan prediksi. Performa model kemudian diukur berdasarkan rumus yang telah dirangkum dalam Tabel 2.2, yaitu: Presisi = $TP \;/\; (TP + FP)$, Akurasi = $(TP + TN) \;/\; (TP + TN + FP + FN)$, *Recall* = $TP \;/\; (TP + FN)$ dan F1-*Score* = $2 \times (Presisi \times Recall) \;/\; (Presisi + Recall)$.

Dengan menggunakan metrik-metrik tersebut, kemampuan model dalam mengenali pola sekuens DNA pada data baru dapat dinilai secara objektif. Hasil evaluasi ini memberikan gambaran menyeluruh tentang efektivitas model *Random Forest* dalam mengklasifikasikan sekuens DNA yang terkait dengan diabetes mellitus, serta membantu dalam mengidentifikasi potensi perbaikan untuk meningkatkan performa model.

### 3.5.4 Proses Validasi

Proses validasi dilakukan untuk memastikan keandalan model *Random Forest* dalam mengklasifikasikan sekuens DNA pada berbagai subset data. Validasi ini bertujuan untuk mengurangi risiko *overfitting* dan meningkatkan generalisasi model terhadap data baru. Dalam penelitian ini, digunakan metode *Repeated Stratified K-Fold Cross Validation* untuk memperoleh hasil evaluasi yang lebih stabil. Pada metode ini, dataset dibagi menjadi $K$ bagian (*folds*) dengan proporsi kelas yang seimbang dalam setiap fold, dan proses validasi diulang sebanyak $n$ kali. Setiap kali iterasi, model dilatih pada $K - 1$ *fold* dan diuji pada *fold* yang tersisa, sehingga semua data mendapat kesempatan untuk menjadi data uji. Metrik performa seperti akurasi, presisi, *recall*, dan F1-*score* kemudian dihitung rata-ratanya dari seluruh iterasi, memberikan gambaran performa model yang lebih konsisten dan mengurangi variabilitas hasil. Dengan menggunakan *Repeated Stratified K-Fold Cross Validation*, model *Random Forest* dapat dievaluasi dengan lebih komprehensif, sehingga meningkatkan kepercayaan terhadap kemampuan model dalam mengklasifikasikan data sekuens DNA yang berhubungan dengan risiko diabetes mellitus.

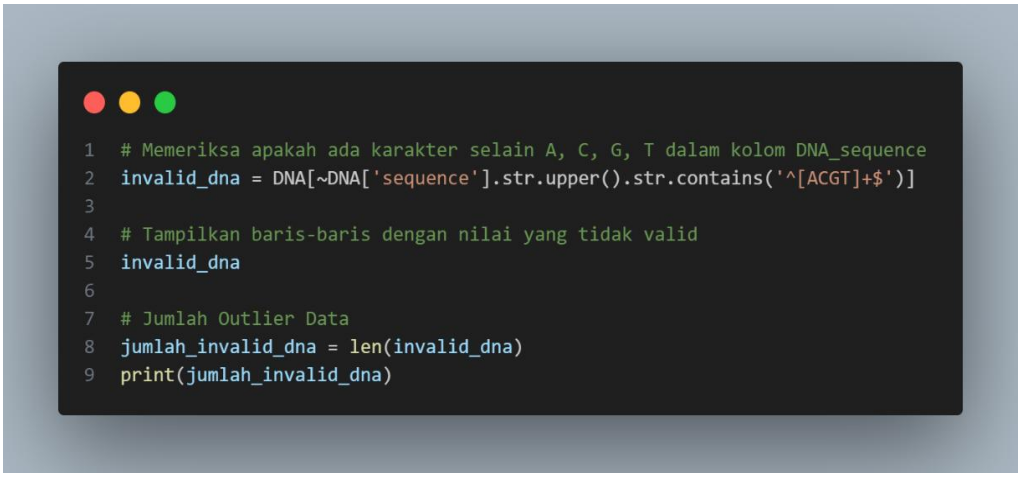
# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1 Preprocessing Data

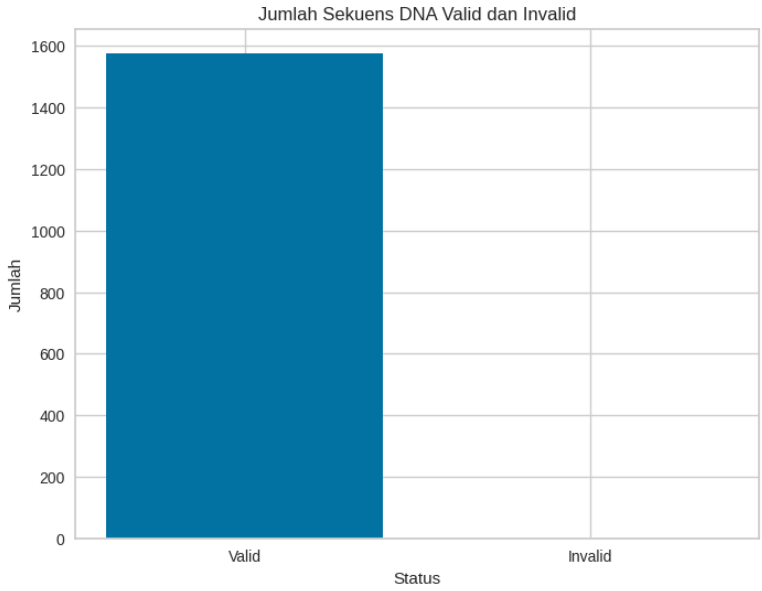
### 4.1.1 Pengecekan *Outlier*

Pengcekan *outlier* dilakukan untuk mengidentifikasi apakah ada titik yang menyimpang pada dataset.




Gambar 4.1 Pengecekaan *Outlier*

Pada Gambar 4.1 dapat dilihat tahapan pengecekan *outlier* dilakukan dengan menggunakan *str.upper* untuk mengubah kolom *sequence* menjadi *uppercase* dan *str.contains* untuk mengambil nilai A, C, G dan T. Jika ada nilai diluar A, C, G dan T di dalam sekuens DNA makan akan ditampilkan di output *invalid_dna* dan jika tidak ada maka output tidak menampilkan apapun.




Gambar 4.2 Hasil Pengecekan Data *Outlier*

Pada Gambar 4.2 Hasil dari pengecekan data *oulier* menunjukkan bahwa tidak ada titik yang menyimpang, sehingga dataset sudah bersih dari data *outlier*.

### 4.1.2 Pemotongan K-*mers*

Tahap preprocessing selanjutnya adalah pemotongan k-mers, dimana setiap sekuens DNA manusia akan dipotong menjadi 3-*mer*, 4-*mer*, dan 5-*mer*.

```
# define k-mers function to split the sequence into group of k-mers words
def getKmers(sequence, size=3):
  return [sequence[x:x+size].upper() for x in range(len(sequence) - size + 1)]

# apply the k-mers function
DNA['k-mers'] = DNA.apply(lambda x: getKmers(x['sequence']), axis=1)
DNA[['sequence', 'k-mers']]
```

Gambar 4.3 Pemotongan K-*mers*

Pada Gambar 4.3 proses pemotongan sekuens DNA menjadi k-mers dilakukan dengan menggunakan fungsi getKmers(). Fungsi ini bertujuan untuk memecah setiap sekuens DNA menjadi potongan substring berukuran tetap sebanyak k nukleotida. Parameter size=3 menunjukkan bahwa pemotongan dilakukan dalam bentuk 3-*mers*, yaitu setiap potongan terdiri dari tiga *nukleotida* berurutan. Pemotongan dilakukan dengan cara iteratif, yaitu dimulai dari indeks pertama hingga akhir sekuens dengan langkah maju satu nukleotida, sehingga menghasilkan seluruh kombinasi 3-mers yang mungkin dari sekuens tersebut. Fungsi ini kemudian diterapkan ke setiap baris data menggunakan apply() pada kolom *sequence*, dan hasil potongan disimpan dalam kolom baru k-*mers*. Dengan demikian, setiap sekuens DNA diubah menjadi daftar 3-*mers* yang merepresentasikan pola lokal dalam sekuens tersebut, yang selanjutnya akan digunakan pada tahap *Count Vectorizer*.

Tabel 4.1 Pemototngan Sekuens DNA 3-*mer*

| | Sekuens | 3-*mers* |
|---|---|---|
| 0 | AATAATTTGTGCACTTCAGAATA… | [AAT, ATA, TAA, AAT, …] |
| 1 | AGTCCTTCGCCGTCCCTCGCCG… | [AGT, GTC, TCC, CCT, …] |
| 2 | AGTCCTTCGCCGTCCCTCGCCG… | [AGT, GTC, TCC, CCT, …] |
| 3 | AGTTGGAGTCTCCAGGGATCAG… | [AGT, GTT, TTG, TGG, …] |
| 4 | AGTCCTTCGCCGTCCCTCGCCG… | [AGT, GTC, TCC, CCT, …] |
| … | … | … |
| 1572 | ATGTGTCGCCTTTTTTTGTTTGC… | [ATG, TGT, GTG, TGT, …] |
| 1573 | AGTAGCAGAAAGTGAGGCTGG… | [AGT, GTA, TAG, AGC, …] |
| 1574 | GGCGGCCAGGGAAGGTCTCTG… | [GGC, GCG, CGG, GGC,…] |
| 1575 | GCTTAGTGCTGAGCACATCCAG… | [GCT, CTT, TTA, TAG, …] |
| 1576 | GGGCTTGGCCTCTGCCCGGCCA… | [GGG, GGC, GCT, CTT, …] |

Pada Tabel 4.1 menunjukkan hasil dari pemotongan sekuens DNA 3-*mer*. Dimana setiap sekuens pada data akan dilakukan pemotongan secara berurutan satu persatu, mulai dari posisi pertama pada sekuens, diambil tiga *nukleotida* secara berurutan sebagai satu unit 3-*mer*. Kemudian posisi pengambilan digeser satu *nukleotida* ke kanan untuk mengambil tiga *nukleotida* berikutnya, dan proses ini diulang terus hingga mencapai akhir sekuens. Panjang dari sekuens yaitu sekitar 1000 hingga 9000 an. Seperti pada sekuens ke 0 panjang sebelum pemotongan k-*mers* 1010 dan setelah pemotongan 3-*mer* panjangnya 1008. Sekuens ke 1576 panjang sebelum pemotongan k-*mers* 9205 dan setelah pemotongan 3-*mer* menjadi 9203.

Tabel 4.2 Pemototngan Sekuens DNA 4-*mer*

| | Sekuens | 4-*mers* |
|---|---|---|
| 0 | AATAATTTGTGCACTTCAGAATA… | [AATA, ATAA, TAAT, …] |
| 1 | AGTCCTTCGCCGTCCCTCGCCG… | [AGTC, GTCC, TCCT, …] |
| 2 | AGTCCTTCGCCGTCCCTCGCCG… | [AGTC, GTCC, TCCT, …] |
| 3 | AGTTGGAGTCTCCAGGGATCAG… | [AGTT, GTTG, TTGG, …] |
| 4 | AGTCCTTCGCCGTCCCTCGCCG… | [AGTC, GTCC, TCCT, …] |
| … | … | … |
| 1572 | ATGTGTCGCCTTTTTTTGTTTGC… | [ATGT, TGTG, GTGT, …] |
| 1573 | AGTAGCAGAAAGTGAGGCTGG… | [AGTA, GTAG, TAGC, …] |
| 1574 | GGCGGCCAGGGAAGGTCTCTG… | [GGCG, GCGG, CGGC, …] |
| 1575 | GCTTAGTGCTGAGCACATCCAG… | [GCTT, CTTA, TTAG, …] |
| 1576 | GGGCTTGGCCTCTGCCCGGCCA… | [GGGC, GGCT, GCTT, …] |

Pada Tabel 4.2 menunjukkan hasil dari pemotongan k-*mers* sebanyak 4-*mer*. Dimana setiap sekuens pada data akan dilakukan pemotongan secara berurutan satu persatu seperti pada pemotongan 3-*mer* sebelumnya. Panjang dari

sekuens yaitu sekitar 1000 hingga 9000 an. Seperti pada sekuens ke 0 panjang sebelum pemotongan k-*mers* 1010 dan setelah pemotongan 4-mer panjangnya 1007. Sekuens ke 1576 panjang sebelum pemotongan k-*mers* 9205 dan setelah pemotongan 4-*mer* menjadi 9202.

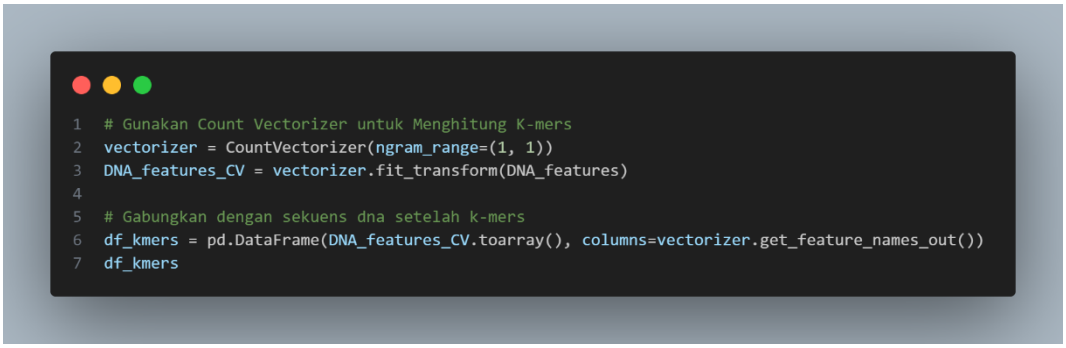
Tabel 4.3 Pemotongan Sekuens DNA 5-*mer*

| | Sekuens | 5-*mers* |
|---|---|---|
| 0 | AATAATTTGTGCACTTCAGAAT… | [AATAA, ATAAT, TAATT, …] |
| 1 | AGTCCTTCGCCGTCCCTCGCC… | [AGTCC, GTCCT, TCCTT, …] |
| 2 | AGTCCTTCGCCGTCCCTCGCC… | [AGTCC, GTCCT, TCCTT, …] |
| 3 | AGTTGGAGTCTCCAGGGATCA… | [AGTTG,GTTGG, TTGGA,…] |
| 4 | AGTCCTTCGCCGTCCCTCGCC… | [AGTC, GTCC, TCCT, …] |
| … | … | … |
| 1572 | ATGTGTCGCCTTTTTTTGTTTG… | [AGTCC, GTCCT, TCCTT, …] |
| 1573 | AGTAGCAGAAAGTGAGGCTG… | [ATGTG, TGTGT, GTGTC,…] |
| 1574 | GGCGGCCAGGGAAGGTCTCT… | [GGCG, GCGG, CGGC, …] |
| 1575 | GCTTAGTGCTGAGCACATCCA… | [GCTTA, CTTAG, TTAGT, …] |
| 1576 | GGGCTTGGCCTCTGCCCGGCC… | [GGGCT, GGCTT, GCTTG,…] |

Pada Tabel 4.3 menunjukkan hasil dari pemotongan k-*mers* sebanyak 5-*mer*. Dimana setiap sekuens pada data akan dilakukan pemotongan secara berurutan satu persatu seperti pada pemotongan 4-*mer* sebelumnya. Panjang dari sekuens yaitu sekitar 1000 hingga 9000 an. Seperti pada sekuens ke 0 panjang sebelum pemotongan k-*mers* 1010 dan setelah pemotongan 5-mer panjangnya 1006. Sekuens ke 1576 panjang sebelum pemotongan k-*mers* 9205 dan setelah pemotongan 5-*mer* menjadi 9201.

### 4.1.3 *Count vectorizer*

Setelah pemotongan k-*mers* dataset akan diubah ke dalam bentuk string, karena pada hasil pemotongan k-*mers* masih berbentuk array. Setelah dibentuk string dataset akan transformasi ke dalam bentuk numerik menggunakan kelas pada *library Scikit-learn* yaitu *Count Vectorizer*.




Gambar 4.4 *Count Vectorizer*

Pada Gambar 4.4 menunjukkan tahap *Count Vectorizer* setelah pemotongan k-*mers*. Pada proses ini, digunakan CountVectorizer dari *library Scikit-learn* dengan parameter ngram_range=(1, 1) yang berarti bahwa setiap k-*mer* dianggap sebagai satu unit kata (*unigram*). Metode ini bekerja dengan menghitung frekuensi kemunculan setiap k-*mer* unik dalam seluruh dataset, kemudian menyusunnya ke dalam bentuk matriks fitur. Setiap baris dalam matriks merepresentasikan satu sekuens DNA, dan setiap kolom merepresentasikan satu jenis k-*mer* unik. Nilai pada setiap sel matriks menunjukkan berapa kali k-*mer* tersebut muncul dalam sekuens tersebut. Hasil transformasi kemudian dikonversi ke dalam bentuk DataFrame agar lebih mudah dianalisis dan digunakan dalam proses selanjutnya, yaitu label *encoding* dan pelatihan model *machine learning*.

Tabel 4.4 Hasil *Count Vectorizer* 3-*mer*

| | AAA | AAC | AAG | … | TTC | TTG | TTT |
|---|---|---|---|---|---|---|---|
| 0 | 40 | 19 | 27 | … | 18 | 17 | 30 |
| 1 | 2 | 2 | 13 | … | 15 | 12 | 16 |
| 2 | 6 | 3 | 17 | … | 15 | 14 | 18 |
| 3 | 15 | 13 | 23 | … | 15 | 16 | 17 |
| 4 | 6 | 3 | 17 | … | 15 | 14 | 18 |
| … | … | … | … | … | … | … | … |
| 1572 | 288 | 119 | 164 | … | 170 | 155 | 345 |
| 1573 | 256 | 89 | 179 | … | 147 | 128 | 183 |
| 1574 | 45 | 44 | 88 | … | 66 | 66 | 35 |
| 1575 | 385 | 134 | 235 | … | 180 | 148 | 218 |
| 1576 | 365 | 136 | 209 | … | 189 | 214 | 346 |

Pada Tabel 4.4 menunjukkan hasil dari tahap *Count Vectorizer* menggunakan pemotongan k-*mers* sebanyak 3-*mer*. Kolom pertama pada tabel menunjukkan urutan sekuens. Baris pertama pada tabel menunjukkan fitur pada setiap sekuens. Kolom pada setiap fitur menunjukkan hasil dari *Count Vectorizer* yang diperoleh dari jumlah fitur pada setiap sekuens, seperti pada fitur AAA pada sekuens ke-0 jumlahnya ada 40, sehingga hasil dari *Count Vectorizer* adalah 40.

Tabel 4.5 Hasil *Count Vectorizer* 4-*mer*

| | AAAA | AAAC | AAAT | … | TTTC | TTTG | TTTT |
|---|---|---|---|---|---|---|---|
| 0 | 11 | 12 | 6 | … | 4 | 8 | 8 |
| 1 | 0 | 1 | 0 | … | 2 | 4 | 7 |
| 2 | 0 | 2 | 2 | … | 3 | 5 | 7 |
| 3 | 2 | 3 | 3 | … | 3 | 3 | 7 |
| 4 | 0 | 2 | 2 | … | 3 | 5 | 7 |
| … | … | … | … | … | … | … | … |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1572 | 128 | 46 | 63 | … | 69 | 63 | 157 |
| 1573 | 120 | 34 | 64 | … | 46 | 50 | 58 |
| 1574 | 9 | 12 | 19 | … | 10 | 13 | 10 |
| 1575 | 141 | 55 | 99 | … | 59 | 52 | 61 |
| 1576 | 139 | 38 | 81 | … | 68 | 84 | 124 |

Pada Tabel 4.5 menunjukkan hasil dari tahap *Count Vectorizer* menggunakan pemotongan k-*mers* sebanyak 4-*mer*. Kolom pertama pada tabel menunjukkan urutan sekuens. Baris pertama pada tabel menunjukkan fitur pada setiap sekuens. Kolom pada setiap fitur menunjukkan hasil dari *Count Vectorizer* yang diperoleh dari jumlah fitur pada setiap sekuens, seperti pada fitur AAAA pada sekuens ke-0 jumlahnya ada 11, sehingga hasil dari *Count Vectorizer* adalah 11.

Tabel 4.6 Hasil *Count Vectorizer* 5-*mer*

| | AAAAA | AAAAC | AAAAG | … | TTTTG | TTTTT |
|---|---|---|---|---|---|---|
| 0 | 2 | 6 | 1 | … | 1 | 0 |
| 1 | 0 | 0 | 0 | … | 1 | 3 |
| 2 | 0 | 0 | 0 | … | 1 | 3 |
| 3 | 0 | 0 | 1 | … | 2 | 1 |
| 4 | 0 | 0 | 0 | … | 1 | 3 |
| … | … | … | … | … | … | … |
| 1572 | 60 | 20 | 25 | … | 30 | 72 |
| 1573 | 67 | 9 | 21 | … | 7 | 18 |
| 1574 | 1 | 1 | 7 | … | 2 | 3 |
| 1575 | 51 | 26 | 31 | … | 13 | 16 |
| 1576 | 53 | 14 | 27 | … | 32 | 38 |

Pada Tabel 4.6 menunjukkan hasil dari tahap *Count Vectorizer* menggunakan pemotongan k-*mers* sebanyak 5-*mer*. Kolom pertama pada tabel menunjukkan urutan sekuens. Baris pertama pada tabel menunjukkan fitur pada setiap sekuens. Kolom pada setiap fitur menunjukkan hasil dari *Count Vectorizer* yang diperoleh dari jumlah fitur pada setiap sekuens, seperti pada fitur AAAAA pada sekuens ke-0 jumlahnya ada 2, sehingga hasil dari *Count Vectorizer* adalah 2.

### 4.1.4 Label *Encoder*

Pada tahap label *encoder*, kolom *class* pada dataset sekuens DNA manusia akan diubah menjadi bentuk numerik menggunakan kelas Label *Encoder*.

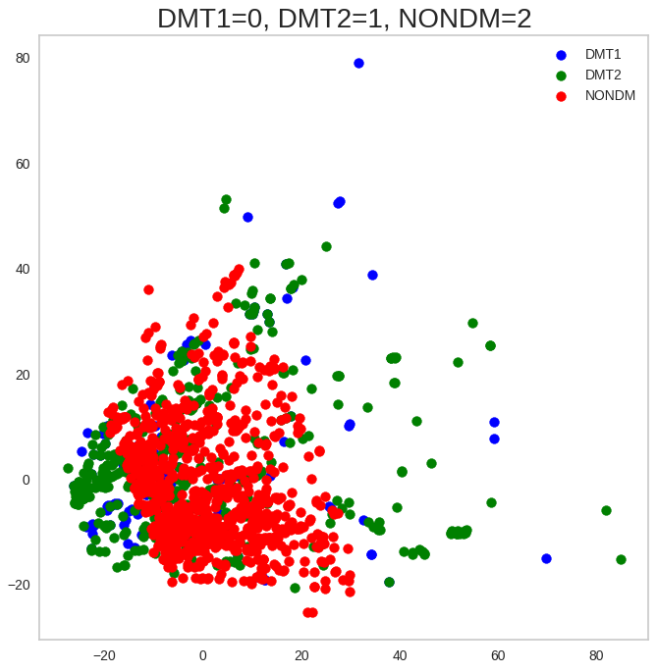
```
# Ubah Kelas menjadi Numerik dengan Label Encoder
encoder = LabelEncoder()
y_encoded = encoder.fit_transform(DNA_targets)

# Buat DataFrame dengan kelas dan numeriknya
class_mapping = pd.DataFrame({
    'Class': encoder.classes_,
    'Numeric': encoder.transform(encoder.classes_)
})

# Tampilkan DataFrame dalam bentuk tabel
print(class_mapping.to_string())
```

Gambar 4.5 Label *Encoder*

Pada Gambar 4.5 tahap label *encoder* mengubah label kelas sekuens DNA yang semula berupa data kelas (seperti DMT1, DMT2, dan Non-DM) menjadi representasi numerik menggunakan metode Label *Encoding*. Pada potongan kode ini, proses dilakukan dengan memanfaatkan kelas LabelEncoder dari *library Scikit-learn*. Pertama, objek encoder didefinisikan, kemudian digunakan untuk melakukan fit dan transform terhadap label kelas yang tersimpan dalam variabel DNA_targets. Hasil transformasi disimpan dalam variabel y_encoded, yang berisi representasi numerik dari masing-masing label. Untuk memberikan gambaran yang jelas terhadap konversi tersebut, dibuat sebuah DataFrame class_mapping yang berisi dua kolom, yaitu kolom *Class* yang berisi label asli, dan kolom *Numeric* yang berisi nilai numerik hasil *encoding*. Proses ini sangat penting agar algoritma machine learning dapat memahami dan memproses variabel target dalam bentuk numerik yang sesuai.

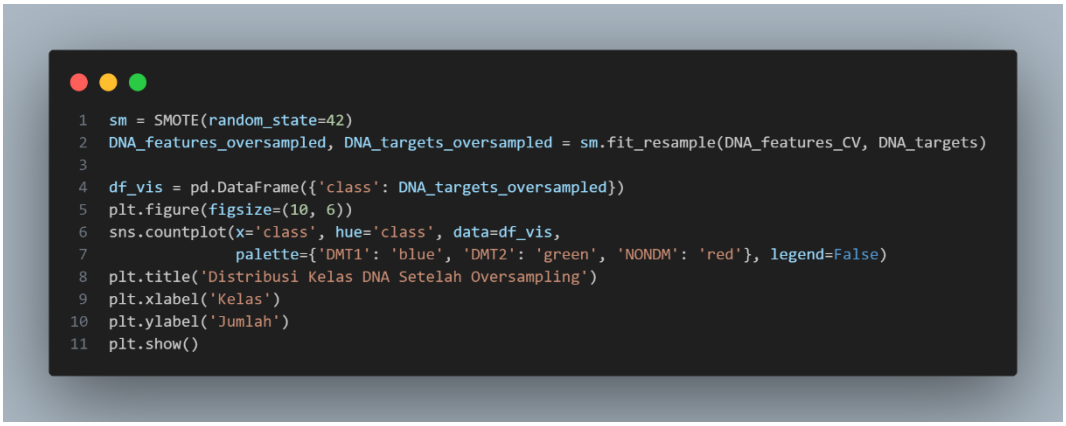



Gambar 4.6 Hasil Label *Encoder*

Pada Gambar 4.6 menunjukkan hasil dari label *encoder*. Hasil dari perubahan label *class* kebentuk numerik dapat diliat pada gambar, dimana DMT 1 merepresentasikan 0, DMT 2 menjadi 1 dan Non-DM menjadi 2.

### 4.1.5 *Oversampling*

Tahap *oversampling* dilakukan untuk menangani data yang tidak seimbang. Seperti pada dataset penelitian ini sekuens DNA manusia jenis DMT 1, DMT 2 dan Non-DM tidak seimbang.



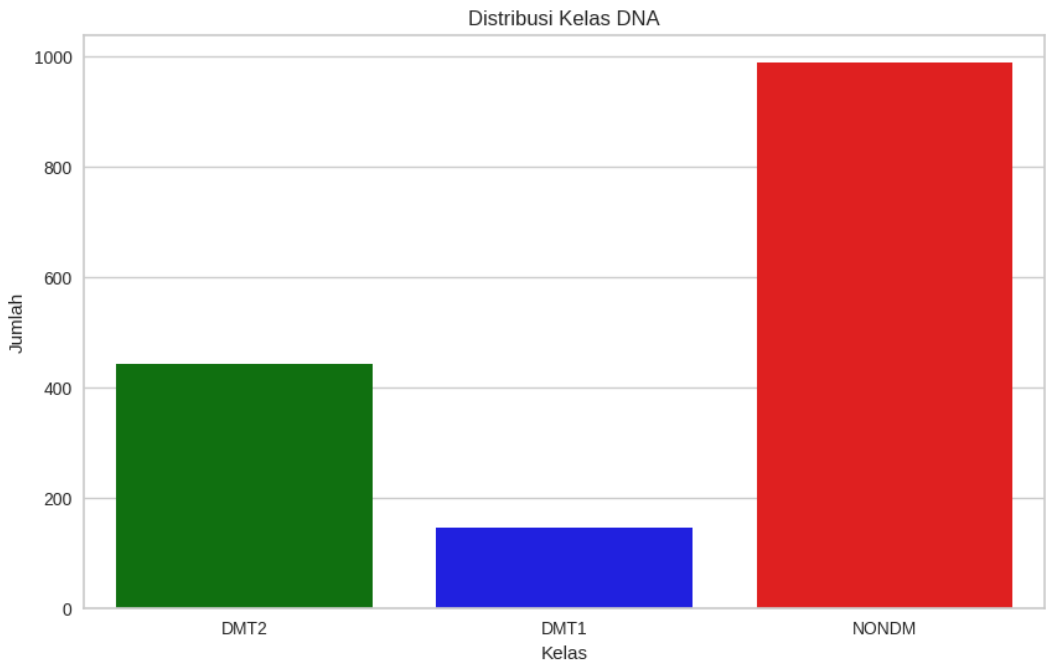
```
sm = SMOTE(random_state=42)
DNA_features_oversampled, DNA_targets_oversampled = sm.fit_resample(DNA_features_CV, DNA_targets)

df_vis = pd.DataFrame({'class': DNA_targets_oversampled})
plt.figure(figsize=(10, 6))
sns.countplot(x='class', hue='class', data=df_vis,
              palette={'DMT1': 'blue', 'DMT2': 'green', 'NONDM': 'red'}, legend=False)
plt.title('Distribusi Kelas DNA Setelah Oversampling')
plt.xlabel('Kelas')
plt.ylabel('Jumlah')
plt.show()
```

Gambar 4.7 *Oversampling*

Pada Gambar 4.7 tahapan oversampling dilakukan untuk mengatasi ketidakseimbangan kelas pada dataset sekuens DNA, di mana jumlah data pada kelas DMT1, DMT2, dan Non-DM tidak merata. Dalam penelitian ini, digunakan teknik *Synthetic Minority Oversampling Technique* (SMOTE) yang tersedia pada *library imblearn*. Melalui kode sm = SMOTE(random_state=42), objek SMOTE didefinisikan dengan parameter acak yang dikontrol untuk memastikan reprodusibilitas hasil. Selanjutnya, data hasil ekstraksi fitur k-mers yang telah ditransformasikan menggunakan *Count Vectorizer* (DNA_features_CV) serta

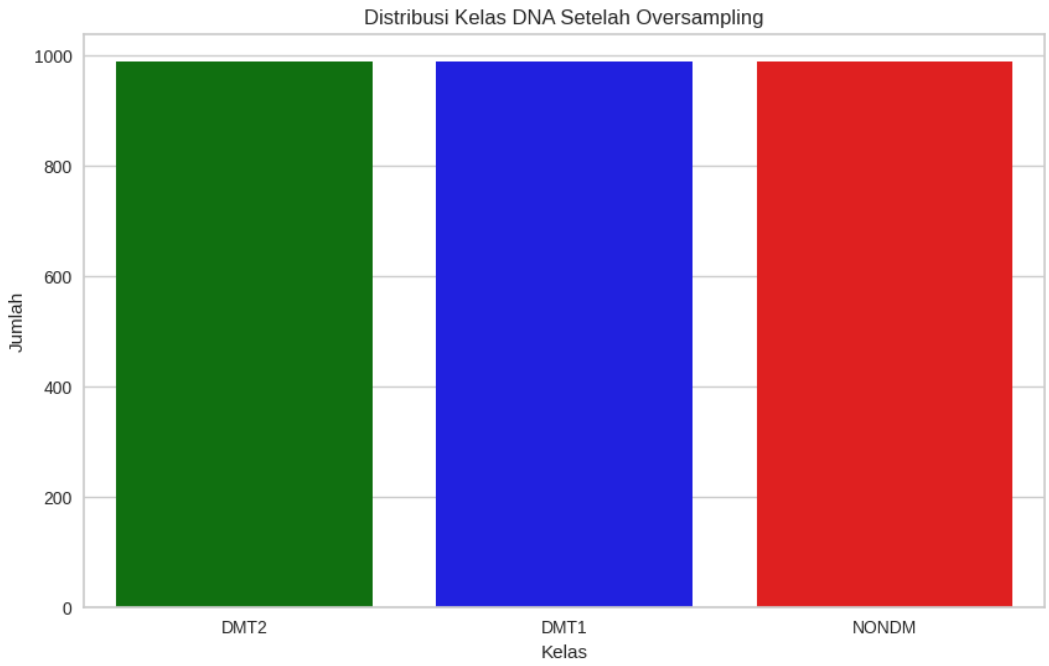
label kelas (DNA_targets) digunakan sebagai input ke fungsi fit_resample() untuk menghasilkan data yang sudah di *oversample*. SMOTE bekerja dengan membuat data sintetis baru pada kelas minoritas berdasarkan kedekatan jarak antar titik dalam ruang fitur, sehingga memperbanyak jumlah data minoritas tanpa melakukan duplikasi langsung. Hasilnya disimpan ke dalam variabel DNA_features_oversampled dan DNA_targets_oversampled. Untuk memvisualisasikan distribusi kelas setelah dilakukan *oversampling*, dibuat grafik batang menggunakan seaborn.countplot yang menunjukkan bahwa jumlah data pada masing-masing kelas telah seimbang. Langkah ini penting agar model pembelajaran mesin tidak bias terhadap kelas mayoritas dan mampu mempelajari ketiga kelas secara adil.




Gambar 4.8 Distribusi Kelas DNA

Dari Gambar 4.8 dapat dilihat bahwa Ketika jenis penyakit diabetes pada dataset tidak seimbang. Dataset Non-DM memiliki jumlah sekuens DNA manusia

sebanyak 989, sedangkan DMT 1 sebanyak 145 dan DMT 2 sebanyak 443. Sehingga perlu dilakukan oversampling agar data yang akan digunakan seimbang. Pada penelitian ini digunakan kelas SMOTE untuk melakukan oversampling data. SMOTE bekerja dengan mengidentifikasi tetangga terdekat dari setiap data di kelas minoritas, lalu membuat data baru di antara data tersebut dan tetangganya. Data baru ini kemudian ditambahkan ke dataset asli, sehingga jumlah data di kelas minoritas meningkat dan data menjadi lebih seimbang.




Gambar 4.9 Hasil Tahap *Oversampling*

Pada Gambar 4.9 dapat dilihat setelah melakukan oversampling menggunakan SMOTE dataset pada ketiga jenis penyakit diabetes sudah seimbang dengan jumlah sekuens pada setiap sebanyak 989.

## 4.2 Implementasi Matematika Algoritma *Random Forest*

Penerapan algoritma *Random Forest* akan diambil sedikit dari data sekuens DNA manusia untuk dilakukan perhitungan secara manual. Berikut adalah data yang akan dipakai:

Tabel 4.7 Sekuens DNA Manusia

| ID/Fitur | AAA | AAC | AAG | AAT | ACA | ACC | ACG | ACT | Kelas |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 40 | 19 | 27 | 28 | 5 | 4 | 14 | DMT2 |
| 2 | 2 | 2 | 13 | 11 | 9 | 21 | 6 | 9 | DMT2 |
| 3 | 44 | 21 | 25 | 19 | 26 | 17 | 5 | 17 | DMT1 |
| 4 | 30 | 13 | 24 | 18 | 19 | 21 | 4 | 13 | DMT1 |
| 5 | 25 | 23 | 33 | 20 | 34 | 36 | 19 | 31 | Non-DM |
| 6 | 39 | 21 | 32 | 30 | 28 | 27 | 9 | 28 | Non-DM |

Data pada Tabel 4.7 merupakan bagian dari data sekuens DNA manusia yang sudah melewati tahap pre-processing. Selanjutnya akan dilakukan penerapan algoritma Random Forest mengunakan cara manual. Pertama, yaitu membentuk pohon keputusan secara acak dari data pada tabel di atas. Disini akan dipakai pohon keputusan sebanyak 5. Berikut tabel pohon keputusan:

Tabel 4.8 Pembentukan Pohon Keputusan

| Pohon Keputusan ($T_i$) | Sekuens (ID) |
|---|---|
| 1 | ID1, ID3, ID4, ID2, ID5, ID6 |
| 2 | ID2, ID5, ID1, ID2, ID4, ID6 |
| 3 | ID3, ID4, ID4, ID1, ID2 ID1 |
| 4 | ID6, ID2, ID3, ID6, ID4, ID5 |

| | |
|---|---|
| 5 | ID3, ID2, ID1, ID4, ID5, ID6 |

Pada Tabel 4.8 menunjukkan hasil pembentukan pohon keputusan. Pohon keputusan dibentuk dengan memilih secara acak sekuens pada data untuk dimasukkan pada setiap pohon keputusan. Seperti pada pohon keputusan pertama $T_1$ memiliki sekuens dengan ID1, ID3, ID4, ID2, ID5 dan ID6 yang dipilih secara acak. Setiap pohon keputusan juga boleh memiliki sekuens yang sama seperti pada pohon $T_2$ terdapat 2 ID2.

### 4.2.1 Pohon Keputusan $T_1$

Setelah membentuk pohon keputusan, Langkah selanjutnya yaitu menghitung setiap pohon keputusan yang sudah dibentuk, dimulai dengan menghitung $Gini(S)$ dan $Gini_{split}$. Data sekuens DNA manusia yang akan dipakai pada pohon keputusan pertama adalah 2 sekuens DMT1, 2 sekuens DMT2 dan 2 sekuens Non-DM. Sehingga untuk meghitung $Gini(S)$ dan $Gini_{split}$ akan digunakan rumus *Gini Impurity* pada persamaan (2.1) dan persamaan (2.2) untuk mencari nilai terkecil pada pohon keputusan $T_1$ dan mendapatkan hasil prediksi.

1. Fitur $AAA$ ($split = 1.5$):

   Nilai threshold diperoleh dari rata-rata dari dua nilai pada sekuens DNA dalam fitur $AAA = [1, 2, 44, 30, 25, 39]$. Dari pohon $T_1$ pada fitur $AAA$ akan di *sorting* dari yang terkecil hingga terbesar menjadi $[1, 2, 25, 30, 39, 40]$. Setelah di sorting nilai threshold diperoleh dengan mencari rata rata dari nilai 1 dan 2, 2 dan 25, 25 dan 30, 30 dan 39 serta 39 dan 40. Sehingga diperoleh $\frac{1+2}{2} = 1.5$ dengan nilai gini split 0.4, $\frac{2+25}{2} = 13.5$ dengan nilai gini split

0.417, $\frac{25+30}{2} = 27.5$ dengan nilai gini split 0.5, $\frac{30+39}{2} = 34.5$ dengan nilai gini split 0.58 dan $\frac{39+40}{2} = 41.5$ dengan nilai gini split 0.47. nilai threshold 1.5 memperoleh nilai gini terkecil, sehingga pada perhitungan gini impurity akan digunakan 1.5 sebagai nilai split.

*Node* kiri ($AAA \leq 1.5$): data ID1

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

*Node* kanan ($AAA > 1.5$): data ID2, ID3, ID4, ID5, ID6

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2$$

$$Gini_{right} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.437$$

$$Gini_{split} = \sum_{i=1}^{c} \left(\frac{n_i}{n}\right) \times Gini(S_i)$$

$$Gini_{AAA} = \left(\frac{\dot{2}}{6}\right) Gini_{left} + \left(\frac{\dot{4}}{6}\right) Gini_{right} = 0.29 + 0 = 0.29$$

2. Fitur $AAC$ ($split = 7.5$):

   *Node* kiri ($AAC \leq 7.5$): data ID2

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

   *Node* kanan ($AAC > 7.5$): data ID1, ID3, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.64$$

$$Gini_{AAC} = \left(\frac{1}{6}\right) \cdot 0 + \left(\frac{5}{6}\right) \cdot 0.64 = 0.533$$

3. Fitur $AAG$ ($split = 21.5$):

   *Node* kiri ($AAG \leq 21.5$): data ID1, ID2

$$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan ($AAG > 21.5$): data ID3, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

$$Gini_{AAG} = \left(\frac{2}{6}\right) \cdot 0 + \left(\frac{4}{6}\right) \cdot 0.5 = 0.333$$

4. Fitur $AAT$ ($split = 19.5$):

   *Node* kiri ($AAT \leq 19.5$): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($AAT > 19.5$): data ID1, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$Gini_{AAT} = \left(\frac{3}{6}\right) \cdot 0.444 + \left(\frac{3}{6}\right) \cdot 0.444 = 0.444$$

5. Fitur $ACA$ ($split = 27$):

   *Node* kiri ($ACA \leq 27$): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($ACA > 27$): data ID1, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.444$$

$$Gini_{ACA} = \left(\frac{3}{6}\right) \cdot 0.444 + \left(\frac{3}{6}\right) \cdot 0.444 = 0.444$$

6. Fitur $ACC$ $(split = 21.2)$

   *Node* kiri $(ACC \leq 21.2)$: data ID1, ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.5$$

   *Node* kanan $(ACC > 21.2)$: data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACC} = \left(\frac{4}{6}\right) \cdot 0.5 + \left(\frac{2}{6}\right) \cdot 0 = 0.333$$

7. Fitur $ACG$ $(split = 7.5)$

   *Node* kiri $(ACG \leq 7.5)$: data ID1, ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.5$$

   *Node* kanan $(ACG > 7.5)$: data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACG} = \left(\frac{4}{6}\right) \cdot 0.5 + \left(\frac{2}{6}\right) \cdot 0 = 0.333$$

8. Fitur $ACT$ $(split = 22.5)$

   *Node* kiri $(ACT \leq 22.5)$: data ID1, ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.5$$

   *Node* kanan $(ACT > 22.5)$: data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACG} = \left(\frac{4}{6}\right) \cdot 0.5 + \left(\frac{2}{6}\right) \cdot 0 = 0.333$$

Dari fitur yang telah diperoleh dan memiliki nilai terkecil adalah fitur $AAA$, $AAG$, $ACC$, $ACG$ dan $ACT$. Sehingga yang akan dipakai ditahap selanjutnya adalah fitur $AAA$.

Dipilih fitur $AAA$ sebagai fitur utama, selanjutnya akan dihitung untuk *node* kiri dan *node* kanan hingga mencapai kondisi *leaf node*.

Fitur $AAA$ ($split = 1.5$):

*Node* kiri ($AAA \leq 1.5$): data ID1

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

*Node* kanan ($AAA > 1.5$): data ID3, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.437$$

$$Gini_{AAA} = \left(\frac{\dot{2}}{6}\right) Gini_{left} + \left(\frac{\dot{4}}{6}\right) Gini_{right} = 0.29 + 0 = 0.29$$

Dari perhitungan di atas, dapat dilihat bahwa *node* kiri sudah mencapai kondisi *leaf node* atau 0 dengan hasil prediksi 1 DMT2, sementara *node* kanan bernilai 0.437 sehingga belum mencapai kondisi *leaf node* dan akan dilakukan perhitungan *gini impurity* Kembali menggunakan fitur yang lainnya. Fitur $AAG$ akan digunakan di iterasi selanjutnya pada *node* kanan.

Fitur $AAG$ (*split* = 28.5):

*Node* kiri ($AAG \leq 28.5$): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

*Node* kanan ($AAG > 28.5$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

Fitur menunjukkan bahwa *node* kiri belum mencapai *leaf node* dengan nilai 0.444, sedangkan untuk *node* kanan sudah mencapai *leaf node* dengan hasil prediksi 2 Non-DM untuk *node* kanan. Sehingga akan dilakukan perhitungan *gini impurity* Kembali menggunakan fitur yang lainnya. Fitur $ACC$ akan digunakan di iterasi selanjutnya pada *node* kiri.

Fitur $ACC$ (*split* = 19):

*Node* kiri ($ACC \leq 19$): data ID3

$$Gini_{left} = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

*Node* kanan ($ACC > 19$): ID2, ID4

$$Gini_{right} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0.5$$

Fitur menunjukkan bahwa *node* kiri sudah mencapai *leaf node* dengan prediksi 1 DMT1, sedangkan untuk *node* kanan belum mencapai *leaf node* dengan nilai 0.5. Sehingga akan dilakukan perhitungan *gini impurity* Kembali menggunakan fitur yang lainnya. Fitur $ACG$ akan digunakan di iterasi selanjutnya pada *node* kiri.

Fitur $ACG$ ($split = 5$):

*Node* kiri ($ACG \leq 5$): data ID4

$$Gini_{left} = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

*Node* kanan $ACG > 5$): data ID2

$$Gini_{right} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

Fitur menunjukkan bahwa semua *node* sudah dalam kondisi *leaf node* dengan hasil prediksi 1 DMT 1 untuk *node* kiri dan 1 DMT 2 untuk *node* kanan. Selanjutnya akan dihitung untuk pohon keputusan yang kedua $T_2$.

### 4.2.2 Pohon Keputusan $T_2$

Setelah pohon keputusan pertama sudah mencapai kondisi *leaf node*, selanjutnya akan dihitung untuk pohon keputusan $T_2$. Data sekuens DNA manusia yang akan dipakai pada pohon keputusan $T_2$ adalah 1 sekuens DMT1, 3 sekuens DMT2 dan 2 sekuens Non-DM. Sehingga untuk meghitung $Gini(S)$ dan $Gini_{split}$ akan digunakan rumus *Gini Impurity* pada persamaan (2.1) dan persamaan (2.2) untuk mencari nilai terkecil pada pohon keputusan $T_2$ dan mendapatkan hasil prediksi.

1. Fitur $AAA$ ($Split = 2$):

   *Node* kiri ($AAA \leq 2$): data ID1, ID2, ID2

   $$Gini_{left} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

   *Node* kanan ($AAA > 2$): data ID4, ID5, ID6

   $$Gini_{right} = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

   $$Gini_{AAA} = \left(\frac{3}{6}\right) \cdot 0 + \left(\frac{3}{6}\right) \cdot 0.444 = 0.222$$

2. Fitur $AAC$ ($Split = 2$):

   *Node* kiri ($AAC \leq 2$): data ID2, ID2

   $$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan ($AAC > 2$): data ID1, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.625$$

$$Gini_{AAC} = \left(\frac{2}{6}\right) \cdot 0 + \left(\frac{4}{6}\right) \cdot 0.4167$$

3. Fitur $AAG$ ($Split$ =21.5):

   *Node* kiri ($AAG \leq 21.5$): data ID1, ID2, ID2

$$Gini_{left} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

   *Node* kanan ($AAG > 21.5$): data ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$Gini_{AAG} = \left(\frac{3}{6}\right) \cdot 0 + \left(\frac{3}{6}\right) \cdot 0.444 = 0.222$$

4. Fitur $AAT$ ($Split = 11$):

   *Node* kiri ($AAT \leq 11$): data ID2, ID2

$$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan ($AAT > 11$): data ID1, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.625$$

$$Gini_{AAT} = \left(\frac{2}{6}\right) \cdot 0 + \left(\frac{4}{6}\right) \cdot 0.625 = 0.4167$$

5. Fitur $ACA$ ($Split = 9$):

   *Node* kiri ($ACA \leq 9$): data ID2, ID2

$$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan ($ACA > 9$): data ID1, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.625$$

$$Gini_{ACA} = \left(\frac{2}{6}\right) \cdot 0 + \left(\frac{4}{6}\right) \cdot 0.625 = 0.4167$$

6. Fitur $ACC$ ($Split = 21$):

   *Node* kiri ($ACC \leq 21$): data ID1, ID2, ID2, ID4

$$Gini_{left} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.375$$

   *Node* kanan ($ACC > 21$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACC} = \left(\frac{4}{6}\right) \cdot 0.375 + \left(\frac{2}{6}\right) \cdot 0 = 0.25$$

7. Fitur $ACG$ ($Split = 6$):

   *Node* kiri ($ACG \leq 6$): data ID1, ID2, ID2, ID4

$$Gini_{left} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.375$$

   *Node* kanan ($ACG > 6$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACG} = \left(\frac{4}{6}\right) \cdot 0.375 + \left(\frac{2}{6}\right) \cdot 0 = 0.25$$

8. Fitur $ACT$ ($Split = 21$):

   *Node* kiri $ACT \leq 21$): data ID1, ID2, ID2, ID4

$$Gini_{left} = 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.375$$

   *Node* kanan ($ACT > 21$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACT} = \left(\frac{4}{6}\right) \cdot 0.375 + \left(\frac{2}{6}\right) \cdot 0 = 0.25$$

Dari fitur yang telah diperoleh dan memiliki nilai terkecil adalah fitur $AAA$ dan $AAG$. Sehingga yang akan dipakai ditahap selanjutnya adalah fitur $AAA$. Dipilih fitur $AAA$ sebagai fitur utama, selanjutnya akan dihitung untuk *node* kiri dan *node* kanan hingga mencapai kondisi *leaf node*.

Fitur $AAA$ ($Split = 2$):

*Node* kiri ($AAA \leq 2$): data ID1, ID2, ID2

$$Gini_{left} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

*Node* kanan ($AAA > 2$): data ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

Dari perhitungan di atas, dapat dilihat bahwa *node* kiri sudah mencapai kondisi *leaf node* atau 0 dengan hasil prediksi 3 DMT2, sementara *node* kanan bernilai 0.5 sehingga belum mencapai kondisi *leaf node* dan akan dilakukan perhitungan *gini impurity* Kembali menggunakan fitur yang lainnya. Fitur $AAG$ akan digunakan di iterasi selanjutnya pada *node* kanan.

Fitur $AAG$ (*split* = 28):

*Node* kiri ($AAG \leq 28$): data ID4

$$Gini_{left} = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

*Node* kanan ($AAG > 28$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

Fitur menunjukkan bahwa semua *node* sudah dalam kondisi *leaf node* dengan hasil prediksi 1 DMT 1 untuk *node* kiri dan 2 Non-DM untuk *node* kanan. Selanjutnya akan dihitung untuk pohon keputusan yang kedua $T_3$.

### 4.2.3 Pohon Keputusan $T_3$

Setelah pohon keputusan $T_2$ sudah mencapai kondisi *leaf node*, selanjutnya akan dihitung untuk pohon keputusan $T_3$. Data sekuens DNA manusia yang akan dipakai pada pohon keputusan $T_3$ adalah 3 sekuens DMT1 dan 3 sekuens DMT2. Sehingga untuk meghitung $Gini(S)$ dan $Gini_{split}$ akan digunakan rumus *Gini Impurity* pada persamaan (2.1) dan persamaan (2.2) untuk mencari nilai terkecil pada pohon keputusan $T_2$ dan mendapatkan hasil prediksi.

1. Fitur $AAA$ ($split = 16$):

   *Node* kiri ($AAA \leq 16$): data ID1, ID1, ID2

   $$Gini_{left} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

   *Node* kanan ($AAA > 16$): data ID3, ID4, ID4

   $$Gini_{right} = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

   $$Gini_{AAA} = \left(\frac{3}{6}\right)0 + \left(\frac{3}{6}\right)0 = 0 + 0 = 0$$

2. Fitur $AAC$ ($split = 30.5$):

   *Node* kiri ($AAC \leq 30.5$): data ID3, ID4, ID4, ID2

   $$Gini_{left} = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{0}{4}\right)^2 0 = 0.375$$

*Node* kanan ($AAC > 30{,}5$): data ID1, ID1

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

$$Gini_{AAC} = \left(\frac{\dot{4}}{6}\right) 0{,}375 + \left(\frac{\dot{2}}{6}\right) 0 = 0.25$$

3. Fitur AAG ($split = 19$ ):

   *Node* kiri ($AAG \leq 19$ ): data ID1, ID2 ID1

$$Gini_{left} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

   *Node* kanan ($AAG > 19$): data ID3, ID4, ID4

$$Gini_{right} = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

$$Gini_{AAG} = \left(\frac{3}{6}\right) 0 + \left(\frac{3}{6}\right) 0 = 0$$

4. Fitur *AAT* ($split = 23$):

   *Node* kiri ($AAT \leq 23$ ): data ID3, ID4, ID4, ID2

$$Gini_{left} = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{0}{4}\right)^2 0 = 0.375$$

   *Node* kanan ($AAT > 23$): data ID1, ID1

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

$$Gini_{AAT} = \left(\frac{4}{6}\right) 0.375 + \left(\frac{2}{6}\right) 0 = 0.25$$

5. Fitur $ACA$ ($split = 27$):

   *Node* kiri ($ACA \leq 27$ ): data ID3, ID4, ID4, ID2

$$Gini_{left} = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.375$$

   *Node* kanan ($ACA > 27$): data ID1, ID1

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

$$Gini_{ACA} = \left(\frac{4}{6}\right) 0.375 + \left(\frac{2}{6}\right) 0 = 0.25$$

6. Fitur $ACC$ $(split = 5)$:

   *Node* kiri $(ACC \leq 5)$: data ID1, ID1

$$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan $(ACC > 5)$: data ID3, ID4, ID4, ID2

$$Gini_{right} = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.375$$

$$Gini_{ACC} = \left(\frac{2}{6}\right) 0 + \left(\frac{4}{6}\right) 0.375 = 0.25$$

7. Fitur $ACG$ $(split = 5.5)$:

   *Node* kiri $(ACG \leq 5.5)$: data ID3, ID4, ID4 ID1, ID1

$$Gini_{left} = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 - \left(\frac{0}{5}\right)^2 = 0.48$$

   *Node* kanan $(ACG > 5.5)$: data ID2

$$Gini_{right} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

$$Gini_{ACG} = \left(\frac{5}{6}\right) 0.48 + \left(\frac{1}{6}\right) 0 = 0.4$$

8. Fitur $ACT$ $(split = 11)$:

   *Node* kiri $(ACT \leq 11)$: data ID2

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

   *Node* kanan $(ACT > 11)$: data ID3, ID4, ID4 ID1, ID1

$$Gini_{right} = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 - \left(\frac{0}{5}\right)^2 = 0.48$$

$$Gini_{ACT} = \left(\frac{1}{6}\right)0 + \left(\frac{5}{6}\right)0.48 = 0.4$$

Dari fitur yang telah diperoleh dan memiliki nilai terkecil adalah fitur $AAA$ dan $AAG$. Sehingga yang akan dipakai ditahap selanjutnya adalah fitur $AAA$. Dipilih fitur $AAA$ sebagai fitur utama, selanjutnya akan dihitung untuk *node* kiri dan *node* kanan hingga mencapai kondisi leaf *node*.

Fitur $AAA$ ($split = 13.5$):

*Node* kiri ($AAA \leq 16$): data ID1, ID1, ID2

$$Gini_{left} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

*Node* kanan ($AAA > 16$): data ID3, ID4, ID4

$$Gini_{right} = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

Dari perhitungan di atas, dapat dilihat bahwa *node* kiri dan *node* kanan sudah mencapai *leaf node* dengan hasil prediksi *node* kiri 3 DMT2 dan *node* kanan 3 DMT1. Sehingga tidak perlu dilanjutkan ke perhitungan selanjutnya.

### 4.2.4 Pohon Keputusan $T_4$

Setelah pohon keputusan $T_3$ sudah mencapai kondisi *leaf node*, selanjutnya akan dihitung untuk pohon keputusan $T_4$. Data sekuens DNA manusia yang akan dipakai pada pohon keputusan $T_4$ adalah 2 sekuens DMT1, 1 sekuens DMT2 dan 3 sekuens Non-DM. Sehingga untuk meghitung $Gini(S)$ dan $Gini_{split}$ akan digunakan rumus *Gini Impurity* pada persamaan (2.1) dan persamaan (2.2) untuk mencari nilai terkecil pada pohon keputusan $T_4$ dan mendapatkan hasil prediksi.

1. Fitur $AAA$ $(split = 13.5)$:

   *Node* kiri ($AAA \leq 13.5$): data ID2

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

   *Node* kanan ($AAA > 13.5$): data ID6, ID3, ID6, ID4, ID5

$$Gini_{right} = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{0}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$$

$$Gini_{AAA} = \left(\frac{1}{6}\right)0 + \left(\frac{5}{6}\right)0{,}48 = 0{,}4$$

2. Fitur $AAC$ $(split = 7.5)$:

   *Node* kiri ($AAC \leq 7.5$): data ID2

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 0 = 0$$

   *Node* kanan ($AAC > 7.5$): data ID6, ID2, ID3, ID6, ID4, ID5

$$Gini_{right} = 1 - \left(\frac{1}{5}\right)^2 - \left(\frac{0}{5}\right)^2 - \left(\frac{3}{5}\right)^2 = 0.48$$

$$Gini_{AAC} = \left(\frac{1}{6}\right)0 + \left(\frac{5}{6}\right)0.48 = 0.4$$

3. Fitur $AAG$ $(split = 28.5)$:

   *Node* kiri ($AAG \leq 28.5$ ): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($AAG > 28.5$): data ID6, ID6, ID5

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{AAG} = \left(\frac{3}{6}\right)0{,}444 + \left(\frac{3}{6}\right)0 = 0.222$$

4. Fitur $AAT$ $(split = 19.5)$:

   *Node* kiri ($AAT \leq$ 19.5): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($AAT >$ 19,5): data ID6, ID6, ID5

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{AAT} = \left(\frac{3}{6}\right) 0.444 + \left(\frac{3}{6}\right) 0 = 0.222$$

5. Fitur $ACA$ $(split = 27)$:

   *Node* kiri ($ACA \leq$ ): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($ACA >$): data ID6, ID6, ID5

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{ACA} = \left(\frac{3}{6}\right) 0{,}444 + \left(\frac{3}{6}\right) 0 = 0{,}222$$

6. Fitur $ACC$ $(split = 21)$:

   *Node* kiri ($ACC \leq$ 21): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($ACC >$ 21): data ID6, ID6, ID5

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{ACC} = \left(\frac{3}{6}\right) 0.444 + \left(\frac{3}{6}\right) 0 = 0.222$$

7. Fitur $ACG$ $(split = 7.5)$:

   *Node* kiri ($ACG \leq$ 7,5): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($ACG >$): data ID6, ID6, ID5

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{ACG} = \left(\frac{3}{6}\right)0.444 + \left(\frac{3}{6}\right)0 = 0.222$$

8. Fitur $ACT$ $(split = 22.5)$:

   *Node* kiri ($ACT \leq$ 22,5): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan ($ACT >$): data ID6, ID6, ID5

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

$$Gini_{ACT} = \left(\frac{3}{6}\right)0.444 + \left(\frac{3}{6}\right)0 = 0.222$$

Dari fitur yang telah diperoleh dan memiliki nilai terkecil adalah fitur $AAG$, $AAT$, $ACA$, $ACC$, $ACG$ dan $ACT$. Sehingga yang akan dipakai ditahap selanjutnya adalah fitur $AAG$.

Dipilih fitur $AAG$ sebagai fitur utama, selanjutnya akan dihitung untuk *node* kiri dan *node* kanan hingga mencapai kondisi *leaf node*.

Fitur $AAG$ $(split = 28.5)$

*Node* kiri ($AAG \leq 28{,}5$ ): data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

*Node* kanan ($AAG > 26{,}5$): data ID5, ID6, ID6

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$$

Dari hasil di atas dapat dilihat bahwa *node* kanan sudah mencapai *leaf node* dengan hasil prediksi 3 Non-DM, sedangkan *node* kiri masih bernilai 0.444 sehingga belum mencapai *leaf node*. Selanjutnya akan dilanjutkan perhitungan pada *node* kiri menggunakan fitur $AAT$.

Fitur $AAT$ ($split = 14.5$):

*Node* kiri ($AAT \leq 14.5$): data ID2

$$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

*Node* kanan ($AAT > 14.5$): data ID3, ID4

$$Gini_{right} = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

Hasil di atas menunjukkan bahwa semua *node* sudah mencapai *node* murni atau *leaf node* dengan prediksi *node* kiri 1 DMT2 dan *node* kanan 2 DMT1. Selanjutnya akan dihitung untuk pohon keputusan $T_5$.

### 4.2.5 Pohon Keputusan $T_5$

Setelah pohon keputusan $T_4$ sudah mencapai kondisi *leaf node*, selanjutnya akan dihitung untuk pohon keputusan $T_5$. Data sekuens DNA manusia yang akan dipakai pada pohon keputusan $T_5$ adalah 2 sekuens DMT1, 2 sekuens DMT2 dan 2 sekuens Non-DM. Sehingga untuk meghitung $Gini(S)$ dan $Gini_{split}$ akan digunakan rumus *Gini Impurity* pada persamaan (2.1) dan persamaan (2.2) untuk mencari nilai terkecil pada pohon keputusan $T_5$ dan mendapatkan hasil prediksi.

1. Fitur $AAA$ ($split = 13.5$):

   *Node* kiri ($AAA \leq 13.5$): data ID2, ID1

   $$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan ($AAA >$): data ID3, ID4, ID5, ID6

   $$Gini_{right} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

   $$Gini_{AAA} = \left(\frac{2}{6}\right)0 + \left(\frac{4}{6}\right)0.5 = 0.333$$

2. Fitur $AAC$ ($split = 7.5$):

   *Node* kiri ($AAC \leq 7.5$): data ID2

   $$Gini_{left} = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$$

   *Node* kanan ($AAC > 7.5$): data ID3, ID1, ID4, ID5, ID6

   $$Gini_{right} = 1 - \left(\frac{2}{5}\right)^2 - \left(\frac{1}{5}\right)^2 - \left(\frac{2}{5}\right)^2 = 0.64$$

   $$Gini_{AAC} = \left(\frac{1}{6}\right)0 + \left(\frac{5}{6}\right)0.64 = 0.533$$

3. Fitur $AAG$ ($split = 21.5$):

   *Node* kiri ($AAG \leq 21.5$ ): data ID1, ID2

   $$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

   *Node* kanan ($AAG > 21.5$): data ID3, ID4, ID5, ID6

   $$Gini_{right} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

$$Gini_{AAG} = \left(\frac{2}{6}\right)0 + \left(\frac{4}{6}\right)0.5 = 0.333$$

4. Fitur $AAT$ $(split = 19.5)$:

   *Node* kiri $(AAT \leq 19.5)$: data ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan $(AAT > 19.5)$: data ID1, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$Gini_{AAT} = \left(\frac{3}{6}\right)0.444 + \left(\frac{3}{6}\right)0.444 = 0.444$$

5. Fitur $ACA$ $(split = 27)$:

   *Node* kiri $(ACA \leq 27)$: data ID2,ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0.444$$

   *Node* kanan $(ACA > 27)$: data ID1, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.444$$

$$Gini_{ACA} = \left(\frac{3}{6}\right)0.444 + \left(\frac{3}{6}\right)0.444 = 0.444$$

6. Fitur $ACC$ $(split = 21)$:

   *Node* kiri $(ACC \leq 21)$: data ID3, ID2, ID1, ID4

$$Gini_{left} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.5$$

   *Node* kanan $(ACC > 21)$: data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACC} = \left(\frac{4}{6}\right)0.5 + \left(\frac{2}{6}\right) = 0.333$$

7. Fitur $ACG$ $(split = 7.5)$:

   *Node* kiri ($ACG \leq 7.5$): data ID1, ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.5$$

   *Node* kanan ($ACG > 7.5$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACG} = \left(\frac{4}{6}\right)0.5 + \left(\frac{2}{6}\right) = 0.333$$

8. Fitur $ACT$ $(split = 22.5)$:

   *Node* kiri ($ACT \leq 22.5$ ): data ID1, ID2, ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.5$$

   *Node* kanan ($ACT > 22.5$ ): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

$$Gini_{ACT} = \left(\frac{4}{6}\right)0.5 + \left(\frac{2}{6}\right) = 0.333$$

Dari fitur yang telah diperoleh dan memiliki nilai terkecil adalah fitur $AAA$, $AAG$, $ACG$ dan $ACT$. Sehingga yang akan dipakai ditahap selanjutnya adalah fitur $AAA$.

Dipilih fitur $AAA$ sebagai fitur utama, selanjutnya akan dihitung untuk *node* kiri dan *node* kanan hingga mencapai kondisi *leaf node*.

Fitur $AAA$ $(split = 13.5)$:

*Node* kiri ($AAA \leq 13.5$): data ID1, ID2

$$Gini_{left} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

*Node* kanan ($AAA > 13.5$): data ID3, ID4, ID5, ID6

$$Gini_{right} = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{0}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = 0.5$$

Dari hasi di atas menunjukkan bahwa *node* kiri sudah mencapai *leaf node* dengan hasil prediksi 2 DMT2, sedangkan *node* kanan masih belum mencapai *leaf node* dengan nilai 0.5. Selanjutnya akan dihitung untuk *node* kanan menggunakan fitur $AAG$.

Fitur $AAG$ ($split = 28.5$):

*Node* kiri ($AAG =\leq 28.5$): data ID3, ID4

$$Gini_{left} = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 = 0$$

*Node* kanan ($AAG > 28.5$): data ID5, ID6

$$Gini_{right} = 1 - \left(\frac{0}{2}\right)^2 - \left(\frac{0}{2}\right)^2 - \left(\frac{2}{2}\right)^2 = 0$$

Dari hasil perhitungan fitur kedua menunjukkan bahwa *node* kiri dan *node* kanan sudah mencapai *leaf node* dengan prediksi 2 DMT1 untuk *node* kiri dan 2 Non-DM untuk *node* kanan. Semua pohon keputusan telah memperoleh hasil dengan semua pohon sudah mencapai kondisi *leaf node*. Selanjutnya akan dilakukan voting mayoritas untuk hasil prediksi pada setiap pohon keputusan.

### 4.2.6 Voting Mayoritas

Setelah menghitung split dan prediksi pada setiap pohon keputusan, selanjutnya akan dilakukan voting mayoritas dari hasil prediksi setiap pohon keputusan pada Tabel 4.9 menggunakan rumus pada Persamaan (2.3).

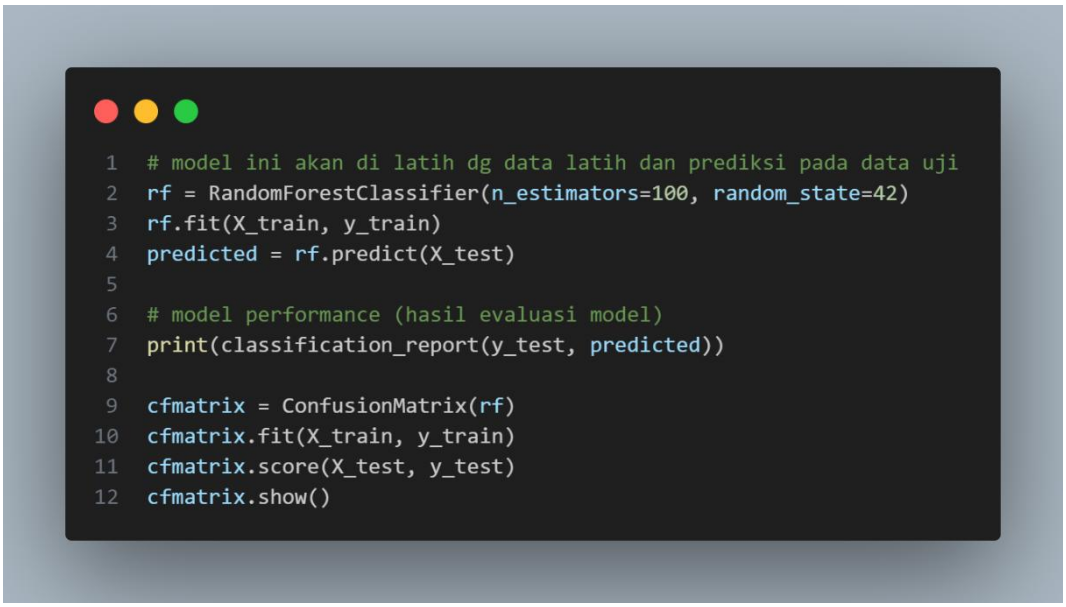
Tabel 4.9 Voting Mayoritas

| ID | Pohon | Prediksi | Voting Mayoritas | Kelas Sebenarnya |
|---|---|---|---|---|
| 1 | $T_1$ | 1 DMT2 | $H(1) = mode\{DMT2, DMT2, DMT2, DMT2\} = DMT2$ | DMT2 |
| | $T_2$ | 3 DMT2 | | |
| | $T_3$ | 3 DMT2 | | |
| | $T_5$ | 2 DMT2 | | |
| 2 | $T_1$ | 1 DMT2 | $H(2) = DMT2$ | DMT2 |
| | $T_2$ | 3 DMT2 | | |
| | $T_3$ | 3 DMT2 | | |
| | $T_4$ | 1 DMT2 | | |
| | $T_5$ | 2 DMT2 | | |
| 3 | $T_1$ | 1 DMT1 | $H(3) = DMT1$ | DMT2 |
| | $T_3$ | 3 DMT1 | | |
| | $T_4$ | 2 DMT1 | | |
| | $T_5$ | 2 DMT1 | | |
| 4 | $T_1$ | 1 DMT1 | $H(4) = DMT1$ | DMT1 |
| | $T_2$ | 1 DMT1 | | |
| | $T_3$ | 3 DMT1 | | |
| | $T_4$ | 2 DMT1 | | |
| | $T_5$ | 2 DMT1 | | |
| 5 | $T_1$ | 2 Non-DM | $H(5) = Non - DM$ | Non-DM |
| | $T_2$ | 2 Non-DM | | |
| | $T_4$ | 3 Non-DM | | |
| | $T_5$ | 2 Non-DM | | |
| 6 | $T_1$ | 2 Non-DM | $H(6) = Non - DM$ | Non-DM |
| | $T_2$ | 2 Non-DM | | |
| | $T_4$ | 3 Non-DM | | |
| | $T_5$ | 2 Non-DM | | |

Tabel 4.9 menunjukkan hasil dari voting mayoritas dari implementasi matematika *Random Forest*. ID 1 pada perhitungan diperoleh hasil klasifikasi yaitu ID1 prediksinya adalah DMT 2, ID 2 prediksinya adalah DMT 2, ID 3 prediksinya adalah DMT 1, ID 4 prediksi adalah DMT 1, ID 5 prediksinya adalah Non-DM dan ID 6 prediksinya adalah Non-DM.

## 4.3 Implementasi Program Algoritma *Random Forest*

Tahap klasifikasi *Random Forest* dilakukan secara manual dengan mengambil sedikit dari data sekuens DNA manusia pada penelitian ini. Namun, secara lengkapnya klasifikasi akan dilakukan menggunakan program *Python*. Langkah selanjutnya data sekuens DNA manusia yang sudah melewati tahap *pre-processing* akan dibagi menjadi data *training* dan data *testing* dengan rasio 80:20, dimana 80 untuk data *training* dan 20 untuk data *testing*.



```
# model ini akan di latih dg data latih dan prediksi pada data uji
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
predicted = rf.predict(X_test)

# model performance (hasil evaluasi model)
print(classification_report(y_test, predicted))

cfmatrix = ConfusionMatrix(rf)
cfmatrix.fit(X_train, y_train)
cfmatrix.score(X_test, y_test)
cfmatrix.show()
```

Gambar 4.10 Model *Random Forest*

Setelah data sekuens DNA dibagi menjadi dua bagian, selanjutnya akan dilakukan pelatihan model dengan data *training* dan diuji dengan data *testing* menggunakan salah satu kelas dari *library Sklearn* yaitu *Randon Forest Classifier*. Gambar 4.10 Menunjukkan proses pelatihan dan pengujian model *Random Forest* untuk tahap klasifikasi. Tabel 4.10 menunjukkan hasil dari klasifikasi *Random Forest* dengan berbagai macam k-*mer* dan $T$ pohon keputusan.

Tabel 4.10 Hasil Klasifikasi

| *K-mers* | Rasio | $T = 10$ | $T = 50$ | $T = 100$ |
|---|---|---|---|---|
| 3-*mer* | 80: 20 | 91% | 91% | 93% |
| 4-*mer* | 80: 20 | 92% | 92% | 92% |
| 5-*mer* | 80: 20 | 90% | 92% | 93% |

Pada Tabel 4.10 diperoleh hasil klasifikasi dengan menggunakan beberapa percobaan. Nilai akurasi tertinggi diperoleh pada pemotongan 3-*mer* dengan banyak pohon keputusan $T = 100$ dan 5-*mer* dengan banyak pohon keputusan $T = 100$ yaitu sebesar 93%.

Tabel 4.11 Perbandingan Metode

| Metode | Data | Akurasi (%) |
|---|---|---|
| *Random Forest* | Sekuens DNA diabetes mellitus tipe 1, tipe 2 dan non diabetes | 93 |
| SVM | Sekuens DNA diabetes mellitus tipe 1, tipe 2 dan non diabetes | 83 |
| KNN | Sekuens DNA diabetes mellitus tipe 1, tipe 2 dan non diabetes | 86 |
| *Decision Tree* | Sekuens DNA diabetes mellitus tipe 1, tipe 2 dan non diabetes | 85 |

| | | |
|---|---|---|
| MLP | Sekuens DNA diabetes mellitus tipe 1, tipe 2 dan non diabetes | 84 |

Berdasarkan Tabel 4.11 Diperoleh hasil akurasi dari beberapa metode, untuk membandingkan metode *Random Forest* dengan metode yang lain dengan data yang sama. Metode *Random Forest* memperoleh nilai akurasi sebesar 93%, metode SVM memperoleh akurasi sebesar 83%, metode KNN memperoleh akurasi sebesar 86%, metode *Decision Tree* memperoleh akurasi sebesar 85% dan MLP memperoleh akurasi sebesar 84%.

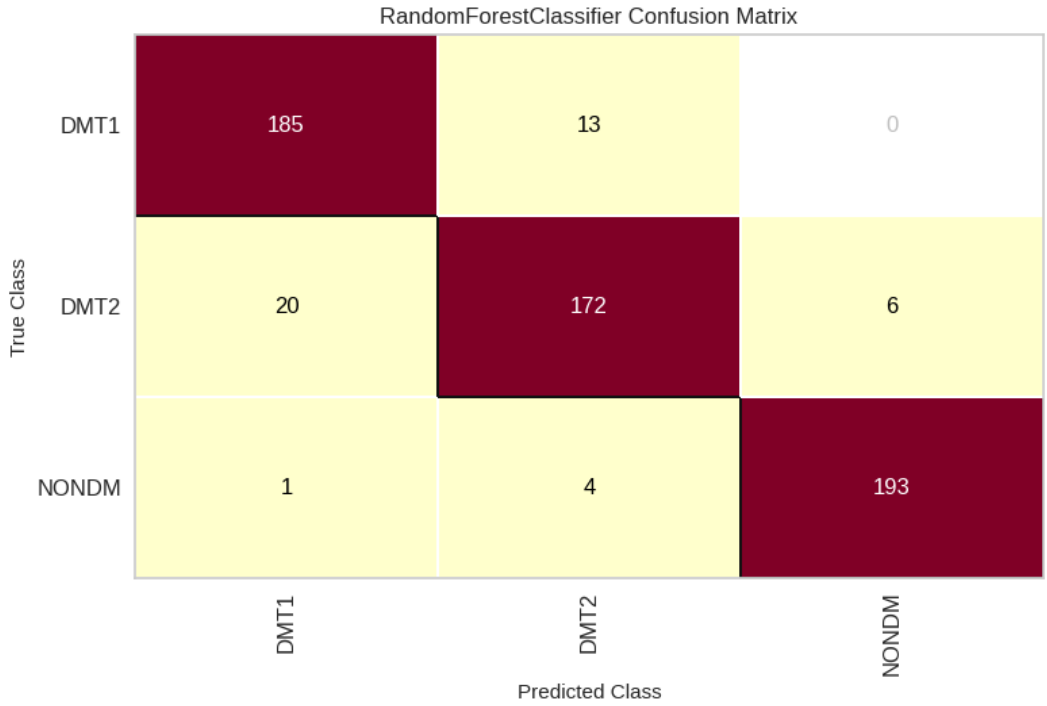
Tabel 4.12 Perbandingan Akurasi

| Metode | Data | Akurasi (%) |
|---|---|---|
| *Random Forest* | Sekuens DNA diabetes mellitus tipe 1, tipe 2 dan non diabetes | 93 |
| *Random Forest* (Ariyadi dkk., 2023) | Jenis kelamin, usia, berat dan tinggi pada balita stunting | 90.1 |
| *Random Forest* (Amaliah dkk., 2022) | Varian minuman kopi | 86.96 |
| *Random Forest* (Pahlevi dkk., 2023) | Informasi kendaraan bermotor (status pernikahan, data tanggungan, usia, status tempat tinggal, pendapatan, status perusahaan, uang muka, Pendidikan terakhir, lama tinggal dan kondisi rumah) | 78.6 |
| KNN (Indrayanti dkk., 2017) | Data publik berupa *pregnant*, *glucose*, DBP, TSFT, INS, BMI, DPF, *age* | 75.14 |
| Naïve Bayes (Ridwan, 2020) | Penderita diabetes tipe 1, tipe 2 dan diabetes *gestasional* | 90.2 |

Pada Tabel 4.12 dijelaskan bahwa *Random Forest* digunakan untuk klasifikasi sekuens DNA manusia pada diabetes mellitus tipe 1, tipe 2 dan non diabetes memperoleh nilai akurasi sebesar 93%. *Random Forest* (Ariyadi dkk., 2023) diterapkan klasifikasi pada balita stunting diperoleh nilai akurasi sebesar 90.1%. *Random Forest* (Amaliah dkk., 2022) diterapkan pada klasifikasi varian minuman kopi diperoleh akurasi sebesar 86.86%. *Random Forest* (Pahlevi dkk., 2023) diterapkan pada klasifikasi kelayakan kredit diperoleh akurasi sebesar 78.6%. KNN (Indrayanti dkk., 2017) diterapkan pada klasifikasi penyakit diabetes mellitus memperoleh akurasi sebesar 75.14% dan Naïve Bayes (Ridwan, 2020) diterapkan pada klasifikasi penyakit diabetes mellitus memperoleh akurasi sebesar 90.2%.

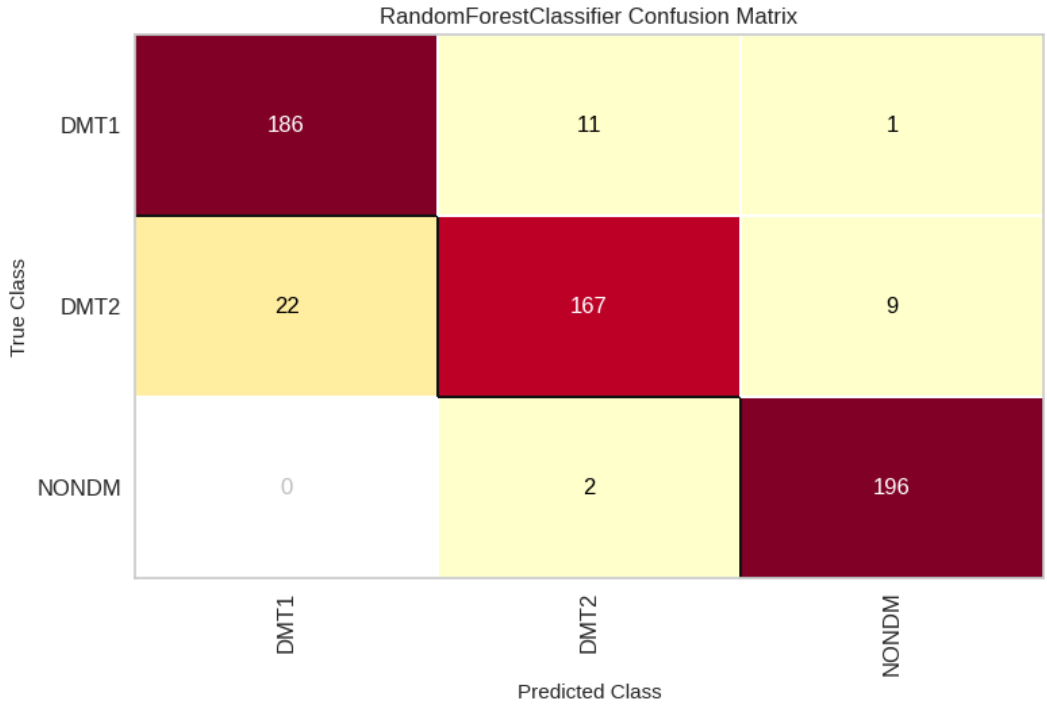
## 4.4 Tahap Evaluasi

Setelah pelatihan dan pengujian model selesai, selanjutnya menampilkan hasil performa dan evaluasi dari model *Random Forest* sebelumnya. Evaluasi model akan dilakukan dengan menggunakan *Confusion Matrix*. Menggunakan salah satu kelas dari *yellowbrick* yaitu *Confusion Matrix*, diperoleh hasil evaluasi sebagai berikut.



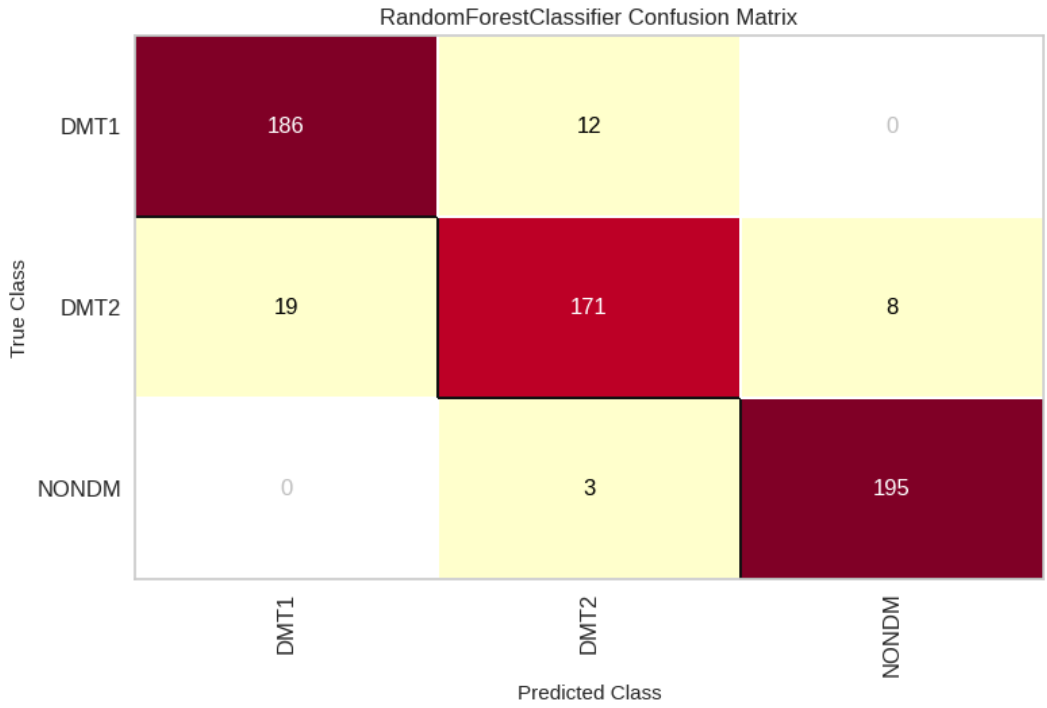

Gambar 4.11 *Confusion Matrix* Model *Random Forest* 3-*mer*

Pada Gambar 4.11 menunjukkan hasil evaluasi model *Random Forest* yang divisualisasi menggunakan *Confusion Matrix* pada pemotongan DNA 3-*mer* pada 100 pohon keputusan. *Confusion matrix* ini merepresentasikan kinerja klasifikasi terhadap tiga kelas target, yaitu DMT1, DMT2, dan NONDM. Baris pada matriks menunjukkan label aktual, sedangkan kolom menunjukkan label prediksi. Berdasarkan hasil tersebut, model berhasil mengklasifikasikan kelas DMT1 dengan benar sebanyak 185 data, meskipun terdapat 13 data yang salah diklasifikasikan sebagai DMT2. Untuk kelas DMT2, sebanyak 172 data diklasifikasikan dengan benar, sedangkan sisanya salah diklasifikasikan ke DMT1 dan NONDM sebanyak 20 dan 6 data. Sementara itu, model menunjukkan performa sangat baik dalam mengklasifikasikan kelas NONDM, dengan 193 data diklasifikasikan dengan benar dan hanya sedikit yang salah diklasifikasikan.




Gambar 4.12 *Confusion Matrix* Model *Random Forest* 4-*mer*

Pada Gambar 4.12 menunjukkan hasil evaluasi model *Random Forest* yang divisualisasi menggunakan *Confusion Matrix* pada pemotongan DNA 4-*mer* pada 100 pohon keputusan. Berdasarkan hasil tersebut, model mampu mengklasifikasikan kelas DMT1 dengan akurasi tinggi, yakni 186 data diklasifikasikan dengan benar, hanya 11 data yang salah diklasifikasikan sebagai DMT2, dan 1 data sebagai NONDM. Untuk kelas DMT2, sebanyak 167 data berhasil diklasifikasikan dengan benar, namun masih terdapat kesalahan prediksi ke kelas DMT1 sebanyak 22 data dan ke kelas NONDM sebanyak 9 data. Sementara itu, performa klasifikasi untuk kelas NONDM sangat baik, dengan 196 data diklasifikasikan dengan benar dan hanya 2 data yang salah diklasifikasikan sebagai DMT2.




Gambar 4.13 *Confusion Matrix* Model *Random Forest* 5-*mer*

Pada Gambar 4.13 menunjukkan hasil evaluasi model *Random Forest* yang divisualisasi menggunakan *Confusion Matrix* pada pemotongan DNA 5-*mer* pada 100 pohon keputusan. Model mampu mengklasifikasikan kelas DMT1 dengan sangat baik, di mana sebanyak 186 data diklasifikasikan dengan benar, dan hanya 12 data yang salah diklasifikasikan sebagai DMT2, tanpa ada kesalahan ke kelas NONDM. Untuk kelas DMT2, sebanyak 171 data berhasil diklasifikasikan dengan benar, sementara 19 data salah diprediksi sebagai DMT1 dan 8 data sebagai NONDM. Pada kelas NONDM, performa model juga sangat baik, dengan 195 data diklasifikasikan dengan tepat, dan hanya 3 data yang keliru diprediksi sebagai DMT2, tanpa ada kesalahan ke kelas DMT1.

Selanjutnya untuk menghitung presisi, *recall*, f1-*score* dan akurasi di gunakan rumus pada Tabel 2.2 yaitu sebagai berikut:

1. Presisi

$$\text{Presisi}_{\text{DMT1}} = \frac{TP_{DMT1}}{TP_{DMT1} + FP_{DMT1}} = \frac{185}{185 + 20 + 1} \times 100\% = 89.8\%$$

$$\text{Presisi}_{\text{DMT2}} = \frac{TP_{DMT2}}{TP_{DMT2} + FP_{DMT2}} = \frac{172}{172 + 13 + 4} \times 100\% = 91\%$$

$$\text{Presisi}_{\text{NONDM}} = \frac{TP_{NONDM}}{TP_{NONDM} + FP_{NONDM}} = \frac{193}{193 + 0 + 6} \times 100\%$$

$$= 96.98\%$$

$$\text{Presisi}_{3mer} = \frac{\text{Presisi}_{\text{DMT1}} + \text{Presisi}_{\text{DMT2}} + \text{Presisi}_{\text{NONDM}}}{3} = 92.5\%$$

$$\text{Presisi}_{\text{DMT1}} = \frac{186}{186 + 22 + 0} \times 100\% = 89.4\%$$

$$\text{Presisi}_{\text{DMT2}} = \frac{167}{167 + 11 + 2} \times 100\% = 92.8\%$$

$$\text{Presisi}_{\text{NONDM}} = \frac{196}{196 + 1 + 9} \times 100\% = 95.1\%$$

$$\text{Presisi}_{4mer} = \frac{\text{Presisi}_{\text{DMT1}} + \text{Presisi}_{\text{DMT2}} + \text{Presisi}_{\text{NONDM}}}{3} = 92.4\%$$

$$\text{Presisi}_{\text{DMT1}} = \frac{186}{186 + 19} \times 100\% = 90.7\%$$

$$\text{Presisi}_{\text{DMT2}} = \frac{171}{171 + 12 + 3} \times 100\% = 91.9\%$$

$$\text{Presisi}_{\text{NONDM}} = \frac{195}{195 + 8} \times 100\% = 96\%$$

$$\text{Presisi}_{5mer} = \frac{\text{Presis}i_{\text{DMT1}} + \text{Presisi}_{\text{DMT2}} + \text{Presisi}_{\text{NONDOM}}}{3} = 93\%$$

2. Akurasi

$$\text{Akurasi}_{3mer} = \frac{185 + 172 + 193}{185 + 172 + 193 + 13 + 20 + 1 + 4 + 6} \times 100\%$$

$$= 92.6\%$$

$$\text{Akurasi}_{4mer} = \frac{186 + 167 + 196}{186 + 167 + 196 + 11 + 22 + 1 + 9 + 2} \times 100\%$$

$$= 92.4\%$$

$$\text{Akurasi}_{5mer} = \frac{186 + 171 + 195}{186 + 171 + 195 + 12 + 19 + 8 + 3} \times 100\% = 93\%$$

3. *Recall*

$$Recall_{DMT1} = \frac{TP_{DMT1}}{TP_{DMT1} + FN_{DMT1}} = \frac{185}{185 + 13 + 0} \times 100\% = 93.4\%$$

$$Recall_{DMT2} = \frac{TP_{DMT2}}{TP_{DMT2} + FN_{DMT2}} = \frac{172}{172 + 20 + 6} \times 100\% = 86.8\%$$

$$Recall_{NONDM} = \frac{TP_{NONDM}}{TP_{NONDM} + FN_{NONDM}} = \frac{193}{193 + 1 + 4} \times 100\%$$

$$= 97.47\%$$

$$Recall_{3mer} = \frac{Recall_{DMT1} + Recall_{DMT2} + Recall_{NONDM}}{3} = 92.5\%$$

$$Recall_{DMT1} = \frac{186}{186 + 11 + 1} \times 100\% = 93.9\%$$

$$Recall_{DMT2} = \frac{167}{167 + 22 + 9} \times 100\% = 84.3\%$$

$$Recall_{NONDM} = \frac{196}{196 + 0 + 2} \times 100\% = 98.99\%$$

$$Recall_{4mer} = \frac{Recall_{DMT1} + Recall_{DMT2} + Recall_{NONDOM}}{3} = 92.4\%$$

$$Recall_{DMT1} = \frac{186}{186 + 12} \times 100\% = 93.9\%$$

$$Recall_{DMT2} = \frac{171}{171 + 19 + 8} \times 100\% = 86.4\%$$

$$Recall_{NONDM} = \frac{195}{195 + 3} \times 100\% = 98.5\%$$

$$Recall_{5mer} = \frac{Recall_{DMT1} + Recall_{DMT2} + Recall_{NONDM}}{3} = 93\%$$

4. F1-*Score*

$$F1_{DMT1} = \frac{2 \times Presisi_{DMT1} \times Recall_{DMT1}}{Presisi_{DMT1} + Recall_{DMT1}} = \frac{2 \times 0.898 \times 0.934}{0.989 + 0.934} \times 100\%$$

$$= 91.56\%$$

$$F1_{DMT2} = \frac{2 \times Presisi_{DMT2} \times Recall_{DMT2}}{Presisi_{DMT2} + Recall_{DMT2}} = \frac{2 \times 0.91 \times 0.934}{0.91 + 0.934} \times 100\%$$

$$= 92.18\%$$

$$F1_{NONDM} = \frac{2 \times Presisi_{NONDM} \times Recall_{NONDM}}{Presisi_{NONDM} + Recall_{NONDM}}$$

$$= \frac{2 \times 0.9698 \times 0.9747}{0.9698 + 0.9747} \times 100\% = 97.2\%$$

$$F1_{3mer} = \frac{F1_{DMT1} + F1_{DMT2} + F1_{NONDM}}{3} = 93.6\%$$

$$F1_{DMT1} = \frac{2 \times 0.894 \times 0.939}{0.894 + 0.939} \times 100\% = 91.6\%$$

$$F1_{DMT2} = \frac{2 \times 0.928 \times 0.843}{0.928 + 0.843} \times 100\% = 88.3\%$$

$$F1_{NONDM} = \frac{2 \times 0.951 \times 0.9899}{0.951 + 0.9899} \times 100\% = 97\%$$

$$F1_{4mer} = \frac{F1_{DMT1} + F1_{DMT2} + F1_{NONDM}}{3} = 92.3\%$$

$$F1_{DMT1} = \frac{2 \times 0.907 \times 0.939}{0.907 + 0.939} \times 100\% = 92.3\%$$

$$F1_{DMT2} = \frac{2 \times 0.919 \times 0.864}{0.919 + 0.864} \times 100\% = 89.1\%$$

$$F1_{NONDM} = \frac{2 \times 0.96 \times 0.985}{0.96 + 0.985} \times 100\% = 97.2\%$$

$$F1_{5mer} = \frac{F1_{DMT1} + F1_{DMT2} + F1_{NONDM}}{3} = 93\%$$

Tabel 4.13 Performa Model *Random Forest*

| | Akurasi (%) | Presisi (%) | *Recall* (%) | *F1-Score* (%) |
|---|---|---|---|---|
| 3-*mer* | 93 | 92.5 | 92.5 | 93.6 |
| 4-*mer* | 92 | 92.4 | 92.3 | 92.3 |
| 5-*mer* | 93 | 93 | 93 | 93 |

Tabel 4.13 menunjukkan hasil dari performa model *Random Forest* pada tahap klasifikasi sebelumnya pada pemotong DNA 3-*mer*, 4-*mer* dan 5-*mer*. Dapat dilihat model *Random Forest* dengan 3-*mer* dan pembagian data *training* dan *testing* dengan rasio 80:20 dengan 100 pohon keputusan, mendapatkan akurasi 93%. Model dengan 4-*mer* dan pembagian data *training* dan *testing* dengan rasio 80:20 mendapatkan akurasi 92%. Sedangkan untuk 5-*mer* memperoleh akurasi sebesar 93%.

## 4.5 Tahap Validasi

Setelah tahap evaluasi selesai, untuk memastikan hasil prediksi yang lebih akurat. Proses validasi dilakukan dengan menggunakan salah satu kelas pada *library Sklearn* yaitu *Reapeated Stratified Field KFold*. Setiap iterasi digunakan 4 *fold* sebagai data pelatihan dan 1 *fold* sebagai pengujian. Proses ini dilakukan sebanyak 5 iterasi.

Tabel 4.14 *Cross Validation* Model *Random Forest* 3-*mer*

| K | Presisi (%) | *Recall* (%) | *F-1 Score* (%) | Akurasi (%) |
|---|---|---|---|---|
| 1 | 93 | 93 | 93 | 93 |
| 2 | 89 | 89 | 89 | 89 |
| 3 | 91 | 91 | 90 | 91 |
| 4 | 92 | 92 | 92 | 92 |

| | | | | |
|---|---|---|---|---|
| 5 | 90 | 90 | 90 | 90 |
| Rata-rata | 91 | 91 | 91 | 91 |

Berdasarkan Tabel 4.14 pada pemotongan DNA 3-*mer* dengan 100 pohon keputusan diperoleh rata-rata presisi sebesar 91%, *recall* sebesar 91%, *f1-score* sebesar 91% dan rata-rata akurasi sebesar 91%.

Tabel 4.15 *Cross Validation* Model *Random Forest* 4-*mer*

| K | Presisi (%) | *Recall* (%) | *F-1 Score* (%) | Akurasi (%) |
|---|---|---|---|---|
| 1 | 92 | 92 | 92 | 92 |
| 2 | 90 | 90 | 90 | 90 |
| 3 | 92.7 | 92.7 | 92 | 92.7 |
| 4 | 90.8 | 90.8 | 90.8 | 90.8 |
| 5 | 91 | 91 | 91 | 91 |
| Rata-rata | 91.5 | 91.5 | 91.5 | 91 |

Berdasarkan Tabel 4.15 pada pemotongan DNA 4-*mer* dengan 100 pohon keputusan diperoleh rata-rata presisi sebesar 91.5%, *recall* sebesar 91.5%, *f1-score* sebesar 91.5% dan rata-rata akurasi sebesar 91%.

Tabel 4.16 *Cross Validation* Model *Random Forest* 5-*mer*

| K | Presisi (%) | *Recall* (%) | *F-1 Score* (%) | Akurasi (%) |
|---|---|---|---|---|
| 1 | 91.7 | 91.7 | 91.7 | 91.7 |
| 2 | 93 | 93 | 93 | 93 |
| 3 | 92 | 92 | 92 | 92 |
| 4 | 91.6 | 91.7 | 91.6 | 91.7 |
| 5 | 91.8 | 91.8 | 91.8 | 91.9 |
| Rata-rata | 92 | 92 | 92 | 92 |

Berdasarkan Tabel 4.16 pada pemotongan DNA 5-*mer* dengan 100 pohon keputusan diperoleh rata-rata presisi sebesar 92%, *recall* sebesar 92%, *f1-score* sebesar 92% dan rata-rata akurasi sebesar 92%.

Hasil ini menunjukkan bahwa model klasifikasi sekuens DNA manusia menggunakan algoritma *Random Forest* memiliki performa yang sangat baik dan akurat dalam mengklasifikasikan data sekuens DNA manusia.

## 4.6 Kajian Keislaman dengan Hasil Penelitian

Penelitian ini bertujuan untuk mengembangkan metode klasifikasi sekuens DNA manusia menggunakan algoritma *Random Forest* guna membantu mendeteksi potensi penyakit diabetes lebih awal dan akurat. Dalam perspektif Islam, menjaga kesehatan adalah bagian dari kewajiban manusia dalam memelihara amanah yang diberikan oleh Allah SWT atas tubuh dan kehidupan. Allah SWT berfirman dalam Surah Al-Ma'idah ayat 32:

مِنْ أَجْلِ ذَٰلِكَ كَتَبْنَا عَلَىٰ بَنِيٓ إِسْرَٰٓءِيلَ أَنَّهُۥ مَن قَتَلَ نَفْسًۢا بِغَيْرِ نَفْسٍ أَوْ فَسَادٍ فِى ٱلْأَرْضِ
فَكَأَنَّمَا قَتَلَ ٱلنَّاسَ جَمِيعًا وَمَنْ أَحْيَاهَا فَكَأَنَّمَآ أَحْيَا ٱلنَّاسَ جَمِيعًا ۚ وَلَقَدْ جَآءَتْهُمْ رُسُلُنَا بِٱلْبَيِّنَٰتِ ثُمَّ
إِنَّ كَثِيرًا مِّنْهُم بَعْدَ ذَٰلِكَ فِى ٱلْأَرْضِ لَمُسْرِفُونَ

*Artinya: "Oleh karena itu, Kami menetapkan (suatu hukum) bagi Bani Israil bahwa siapa yang membunuh seseorang bukan karena (orang yang dibunuh itu) telah membunuh orang lain atau karena telah berbuat kerusakan di bumi, maka seakan-akan dia telah membunuh semua manusia. Sebaliknya, siapa yang memelihara kehidupan seorang manusia, dia seakan-akan telah memelihara kehidupan semua manusia. Sungguh, rasul-rasul Kami benar-benar telah datang kepada mereka dengan (membawa) keterangan-keterangan yang jelas. Kemudian, sesungguhnya banyak di antara mereka setelah itu melampaui batas di bumi. (Q.S Al-Mai'dah ayat 32)*

Ayat ini menunjukkan bahwa menyelamatkan satu jiwa manusia memiliki nilai yang sangat tinggi dalam Islam. Dalam konteks penelitian ini, pengembangan teknologi untuk deteksi dini penyakit seperti diabetes merupakan bentuk nyata dari upaya memelihara kehidupan manusia. Dengan mendeteksi risiko penyakit lebih awal melalui analisis sekuens DNA, individu dapat mengambil langkah pencegahan untuk menjaga kesehatannya, sehingga potensi komplikasi serius dapat dihindari.

# BAB V

# PENUTUP

## 5.1 Kesimpulan

Berikut kesimpulan dari hasil penelitian kalsifikasi sekuens DNA manusia menggunakan *Random Forest*:

1. Hasil dari klasifikasi dengan pemotongan sebanyak 3-*mer* menggunakan *split* data *training* dan *testing* dengan rasio 80: 20 dan 100 pohon keputusan memperoleh nilai akurasi sebesar 93%, sedangkan pemotongan sebanyak 4-mer menggunakan *split* data dengan rasio 80: 20 memperoleh akurasi sebesar 92% dan untuk pemotongan 5-mer memperoleh hasil akurasi sebesar 93%. Hal ini menunjukkan bahwa pemotongan 3-*mer* dan 4-*mer* memperoleh akurasi yang lebih tinggi dibandingkan dengan pemotongan 4-*mer*. Dengan nilai akurasi sebesar 93% membuktikan bahwa performa dan akurasi model *Random Forest* dalam mengklasifikasi sekuens DNA manusia sangat baik.
2. Berdasarkan hasil perolehan skor pada proses validasi, menunjukkan bahwa pemotongan 3-*mer* memperoleh akurasi sebesar 91%, Pemotongan 4-*mer* memperoleh akurasi sebesar 91% dan pemotongan 5-*mer* memperoleh akurasi sebesar 92%. Pemotongan 5-*mer* memperoleh skor akurasi yang lebih tinggi daripada 3-*mer* dan 4-*mer*. Dapat disimpulkan bahwa model klasifikasi menggunakan pemotongan 3-*mer* memperoleh akurasi yang lebih tinggi.
3. Berdasarkan pada Tabel 4.11 dan Tabel 4.12 diperoleh bahwa klasifikasi sekuens DNA manusia pada penyakit diabetes dalam penelitian ini memiliki nilai akurasi yang lebih tinggi dari pada metode lain dan penelitian-penelitian

sebelumnya yaitu sebesar 93%. Dapat disimpulkan bahwa hasil klasifikasi menggunakan algoritma *Random Forest* memperoleh hasil yang sangat bagus.

## 5.2 Saran

Berdasarkan penelitian ini, untuk lebih menyempurnakan penelitian selanjutnya terkait topik klasifikasi sekuens DNA manusia, penulis memberikan saran sebagai berikut:

1. Pada tahap *split* data untuk pembagian data *training* dan *testing*, penelitian selanjutnya dapat menggunakan beberapa opsi rasio seperti $90:10, 70:30$ atau yang lainnya untuk membandingkan skor akurasi mana yang lebih tinggi.
2. Penelitian selanjutnya dapat dilakukan dengan menggunakan algoritma yang berbeda seperti LSTM dan algoritma *Deep Learning* lainnya sehingga dapat mengetahui algoritma yang lebih optimal dengan akurasi tinggi dalam pembuatan model kalsifikasi sekuens DNA manusia yang lebih kompleks.

# DAFTAR PUSTAKA


Abnoosian, K., Farnoosh, R., & Behzadi, M. H. (2023). Prediction of diabetes disease using an ensemble of machine learning multi-classifier models. *BMC bioinformatics*, *24*(1), 337.

Ahmad, M., Jung, L. T., & Bhuiyan, A.-A. (2017). From DNA to protein: Why genetic code context of nucleotides for DNA signal processing? A review. *Biomedical Signal Processing and Control*, *34*, 44–63. https://doi.org/10.1016/j.bspc.2017.01.004

Alpaydin, E. (2020). *Introduction to Machine Learning, fourth edition*. MIT Press. https://books.google.co.id/books?id=uZnSDwAAQBAJ

Alshayeji, M. H., Sindhu, S. C., & Abed, S. (2023). Viral genome prediction from raw human DNA sequence samples by combining natural language processing and machine learning techniques. *Expert Systems with Applications*, *218*, 119641. https://doi.org/10.1016/j.eswa.2023.119641

Alu Syaikh, A. bin M. bin A. bin I. (2008). *Tafsir Ibnu Katsir Jilid 8*.

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

American Diabetes Association. (2021). 6. Glycemic Targets: *Standards of Medical Care in Diabetes—2021*. *Diabetes Care*, *44*(Supplement_1), S73–S84. https://doi.org/10.2337/dc21-S006

Ariyadi, M. R. A., Lestanti, S., & Kirom, S. (2023). *Klasifikasi Balita Stunting Menggunakan Random Forest Classifier di Kabupaten Blitar*. *7*(6).

Chakraborty, C., Bhattacharya, M., Pal, S., & Lee, S.-S. (2024). From machine learning to deep learning: Advances of the recent data-driven paradigm shift in medicine and healthcare. *Current Research in Biotechnology*, *7*, 100164. https://doi.org/10.1016/j.crbiot.2023.100164

Chan, H.-C., Chattopadhyay, A., Chuang, E. Y., & Lu, T.-P. (2021). Development of a gene-based prediction model for recurrence of colorectal cancer using an ensemble learning algorithm. *Frontiers in Oncology*, *11*, 631056.

Fernandez, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905. https://doi.org/10.1613/jair.1.11192

Ganie, S. M., Dutta Pramanik, P. K., Mallik, S., & Zhao, Z. (2023). Chronic kidney disease prediction using boosting techniques based on clinical parameters. *Plos one*, *18*(12), e0295234.

Govender, P., Fashoto, S. G., Maharaj, L., Adeleke, M. A., Mbunge, E., Olamijuwon, J., Akinnuwesi, B., & Okpeku, M. (2022). The application of machine learning to predict genetic relatedness using human mtDNA hypervariable region I sequences. *Plos one*, *17*(2), e0263790.

Hare, M. J., & Topliss, D. J. (2022). Classification and laboratory diagnosis of diabetes mellitus. *Endocrinology and Diabetes: A Problem Oriented Approach*, 303–313.

Indrayanti, I., Sugianti, D., & Al Karomi, A. (2017). Optimasi Parameter K Pada Algoritma K-Nearest Neighbour Untuk Klasifikasi Penyakit Diabetes Mellitus. *Prosiding SNATIF*, 823–829.

Kementerian Agama Republik Indonesia. (2019). *Al-Qur'an dan Terjemahannya Edisi Penyempurnaan 2019*. Lajnah Pentashihan Mushaf Al-Qur'an.

Kotu, V., & Deshpande, B. (2018). *Data Science: Concepts and Practice*. Morgan Kaufmann. https://books.google.co.id/books?id=-nt8DwAAQBAJ

Kulkarni, A., & Shivananda, A. (2019). Converting Text to Features. Dalam A. Kulkarni & A. Shivananda (Ed.), *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python* (hlm. 67–96). Apress. https://doi.org/10.1007/978-1-4842-4267-4_3

Lestari, A., Santoso, R., & Suparti, S. (2024). Analisis Sentimen Pengguna Online Travel Agent (OTA) pada Perusahaan Pegipegi.com Menggunakan Random Forest. *Jurnal Gaussian*, *12*(4), 616–624.

Mahajan, P., Uddin, S., Hajati, F., & Moni, M. A. (2023). *Ensemble learning for disease prediction: A review*. *11*(12), 1808.

Mayangsari, M. K., Syarif, I., & Barakbah, A. (2023). Evaluation of Stratified K-Fold Cross Validation for Predicting Bug Severity in Game Review Classification. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*. https://doi.org/10.22219/kinetik.v8i3.1740

McLeish, E., Slater, N., Mastaglia, F. L., Needham, M., & Coudert, J. D. (2024). From data to diagnosis: How machine learning is revolutionizing biomarker discovery in idiopathic inflammatory myopathies. *Briefings in bioinformatics*, *25*(1), bbad514.

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection

and classification of spectral data. *BMC Bioinformatics*, *10*(1), 213. https://doi.org/10.1186/1471-2105-10-213

Ogurtsova, K., Da Rocha Fernandes, J. D., Huang, Y., Linnenkamp, U., Guariguata, L., Cho, N. H., Cavan, D., Shaw, J. E., & Makaroff, L. E. (2017). IDF Diabetes Atlas: Global estimates for the prevalence of diabetes for 2015 and 2040. *Diabetes Research and Clinical Practice*, *128*, 40–50. https://doi.org/10.1016/j.diabres.2017.03.024

Pahlevi, O.-, Amrin, A.-, & Handrianto, Y.-. (2023). Implementasi Algoritma Klasifikasi Random Forest Untuk Penilaian Kelayakan Kredit. *Jurnal Infortech*, *5*(1), 71–76. https://doi.org/10.31294/infortech.v5i1.15829

Park, U., Kang, Y., Lee, H., & Yun, S. (2022). A stacking heterogeneous ensemble learning method for the prediction of building construction project costs. *Applied sciences*, *12*(19), 9729.

Parmar, A., Katariya, R., & Patel, V. (2019). *A review on random forest: An ensemble classifier*. 758–763.

Ridwan, A. (2020). Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Penyakit Diabetes Mellitus. *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, *4*(1), 15–21. https://doi.org/10.47970/siskom-kb.v4i1.169

Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., Colagiuri, S., Guariguata, L., Motala, A. A., Ogurtsova, K., Shaw, J. E., Bright, D., & Williams, R. (2019). Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas, 9th edition. *Diabetes Research and Clinical Practice*, *157*, 107843. https://doi.org/10.1016/j.diabres.2019.107843

Sammut, C., & Webb, G. I. (2011). *Encyclopedia of Machine Learning*. Springer US. https://books.google.co.id/books?id=i8hQhp1a62UC

Shihab, M. Q. (2002). *Tafsir al-mishbah: Pesan, kesan dan keserasian al-Qur'an* (Vol. 9). Lentera Hati.

Triyana, D., Muharrom Al Haromainy, M., & Maulana, H. (2024). Implementasi Metode Ensemble Majority Vote pada Algritma Naive Bayes dan Random Forest untuk Analisis Sentimen Twitter Harga Tiket Pesawat Domestik. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(4), 7885–7894. https://doi.org/10.36040/jati.v8i4.10475

Turki, T., & Roy, S. S. (2022). Novel Hate Speech Detection Using Word Cloud Visualization and Ensemble Learning Coupled with Count Vectorizer. *Applied Sciences*, *12*(13), 6611. https://doi.org/10.3390/app12136611

Tüysüzoğlu, G., & Birant, D. (2020). Enhanced bagging (eBagging): A novel approach for ensemble learning. *International Arab Journal of Information Technology*, *17*(4).

Widodo, A. O., Setiawan, B., & Indraswari, R. (2024). Machine Learning-Based Intrusion Detection on Multi-Class Imbalanced Dataset Using SMOTE. *Procedia Computer Science*, *234*, 578–583. https://doi.org/10.1016/j.procs.2024.03.042

Widodo, S., Brawijaya, H., & Samudi, S. (2022). Stratified K-fold cross validation optimization on machine learning for prediction. *Sinkron*, *7*(4), 2407–2414. https://doi.org/10.33395/sinkron.v7i4.11792

Yang, A., Zhang, W., Wang, J., Yang, K., Han, Y., & Zhang, L. (2020). Review on the Application of Machine Learning Algorithms in the Sequence Data Mining of DNA. *Frontiers in Bioengineering and Biotechnology*, *8*, 1032. https://doi.org/10.3389/fbioe.2020.01032

Yang, T., Yan, C., Yang, L., Tan, J., Jiang, S., Hu, J., Gao, W., Wang, Q., & Li, Y. (2023). Identification and validation of core genes for type 2 diabetes mellitus by integrated analysis of single-cell and bulk RNA-sequencing. *European Journal of Medical Research*, *28*(1), 340.

# LAMPIRAN

Lampiran 1: Dataset dan *Source Code*

## RIWAYAT HIDUP

Moh Nurul Cholil, lahir di Bangkalan pada 23 Juli 2003, menempuh Pendidikan dasar di SD Negeri Lembung Paseser Sepulu Bangkalan dari tahun 2010-2016, kemudian melanjutkan studi ke SMP Negeri 1 Sepulu pada periode 2016-2019. Setelah itu, penulis melanjutkan Pendidikan Tingkat menengah di MAN Bangkalan pada tahun 2019-2021 dan terpilih dalam program akselerasi. Pada tahun 2021, penulis diterima di Universitas Islam Negeri Maulana Malik Ibrahim Malang untuk melanjutkan studi di program studi Matematika. Selama menjalani perkuliahan, penulis aktif berpartisipasi dalam kepanitiaan Kompetisi matematika (KOMET) dari tahun 2021-2022, yang memberikan pengalman berharga dalam pengembangan keterampilan *editing* foto dan video. Penulis juga aktif berpartisipasi dalam organisasi intra kampus DEMA Fakultas Sains dan Teknologi (DEMA FST) dari tahun 2022-2023 dalam departement akademik. Selain itu, penulis juga aktif mengikuti *bootcamp* di luar kampus seperti Sanbercode dan Bangkit Academy untuk meningkatkan skill dalam dunia teknologi, diantaranya yaitu UI/UX *Designer*, *Web Developer* dan *Cloud Computing* untuk mendukung kemampuan dalam dunia kerja. Penulis juga membuka jasa sebagai *freelancer* dalam bidang desain grafis, UI/UX desain dan pembuatan website dari sebagai pengalaman sebelum masuk dunia kerja.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

# BUKTI KONSULTASI SKRIPSI

Nama : Moh Nurul Cholil
NIM : 210601110048
Fakultas / Program Studi : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Algoritma Random Forest pada Model Klasifikasi Sekuens DNA Manusia dalam Penyakit Diabetes
Pembimbing I : Ari Kusumastuti, M.Pd., M.Si.
Pembimbing II : Evawati Alisah, M.Pd.

| No | Tanggal | Hal | Tanda Tangan |
|---|---|---|---|
| 1. | 02 September 2024 | Konsultasi Topik | 1. |
| 2. | 11 September 2024 | Konsultasi Bab I, II, dan III | 2. |
| 3. | 27 September 2024 | Konsultasi Bab I, II, dan III | 3. |
| 4. | 02 Oktober 2024 | Konsultasi Bab I, II, dan III | 4. |
| 5. | 03 Oktober 2024 | Konsultasi Kajian Integrasi Bab I dan II | 5. |
| 6. | 04 Oktober 2024 | Konsultasi Kajian Integrasi Bab I dan II | 6. |
| 7. | 07 Oktober 2024 | Konsultasi Kajian Integrasi Bab I dan II | 7. |
| 8. | 08 Oktober 2024 | Persetujuan untuk maju Seminar Proposal oleh Pembimbing I | 8. |
| 9. | 08 Oktober 2024 | Persetujuan untuk maju Seminar Proposal oleh Pembimbing II | 9. |
| 10. | 05 Mei 2025 | Konsultasi Kajian Integrasi Bab IV | 10. |
| 11. | 08 Mei 2025 | Konsultasi dan Revisi Seminar Proposal, Konsultasi Bab IV dan V, dan Persetujuan untuk maju Seminar Hasil oleh Pembimbing I | 11. |

**KEMENTERIAN AGAMA RI**
**UNIVERSITAS ISLAM NEGERI**
**MAULANA MALIK IBRAHIM MALANG**
**FAKULTAS SAINS DAN TEKNOLOGI**
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

| | | | |
|---|---|---|---|
| 12. | 08 Mei 2025 | Konsultasi Kajian Integrasi Bab IV, Persetujuan untuk maju Seminar Hasil oleh Pembimbing II | 12. |
| 13. | 29 Mei 2025 | Konsultasi dan Revisi Seminar Hasil, Persetujuan untuk maju Sidang Skripsi oleh Pembimbing I | 13. |
| 14. | 02 Juni 2025 | Persetujuan untuk maju Keseluruhan oleh Pembimbing I | 14. |
| 15. | 02 Juni 2025 | Persetujuan untuk maju Keseluruhan oleh Pembimbing II | 15. |

Malang, 13 Juni 2025
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005