KLASIFIKASI SEKUENS DNA DIABETES MELITUS DENGAN ALGORITMA K-NEAREST NEIGHBOR

SKRIPSI

OLEH:
MUHAMMAD KRISTOVER ARMAND
NIM. 210601110051

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

KLASIFIKASI SEKUENS DNA DIABETES MELITUS DENGAN ALGORITMA K-NEAREST NEIGHBOR

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:
Muhammad Kristover Armand
NIM. 210601110051

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

KLASIFIKASI SEKUENS DNA DIABETES MELITUS DENGAN ALGORITMA *K-NEAREST NEIGHBOR*

SKRIPSI

Oleh:
Muhammad Kristover Armand
NIM. 210601110051

Telah Disetujui Untuk Diuji

Malang, 31 Januari 2025

Dosen Pembimbing I	Dosen Pembimbing II
Ari Kusumastuti, M.Pd., M.Si	Ach. Nashichuddin, M.A
NIP. 19770521 200501 2 004	NIP. 19730705 200003 1 002

Mengetahui,
Ketua Program Studi Matematika

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

KLASIFIKASI SEKUENS DNA DIABETES MELITUS DENGAN ALGORITMA *K-NEAREST NEIGHBOR*

SKRIPSI

Oleh:
Muhammad Kristover Armand
NIM. 210601110051

Telah Dipertahankan di Depan Penguji Seminar Hasil Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Mengikuti Sidang Skripsi

Malang, 14 Maret 2025

Ketua Penguji	: Dr. Usman Pagalay, M.Si	.....
Anggota Penguji 1	: Dr. Mohammad Jamhuri, M.Si	.....
Anggota Penguji 2	: Ari Kusumastuti, M.Pd., M.Si	.....
Anggota Penguji 3	: Ach. Nashichuddin, M.A	.....

Mengetahui,
Ketua Program Studi Matematika

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

PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Kristover Armand

NIM : 2106101110051

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Klasifikasi Sekuens DNA Diabetes Melitus dengan Algoritma K-Nearest Neighbor

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, 14 Maret 2025

Yang Membuat Pernyataan,

Muhammad Kristover Armand

NIM. 210601110051

MOTTO

“Jika hanya mempunyai 1% Kesempatan,
Maka harus mempunyai 99% Keyakinan”
(Neymar)

\

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

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala limpahan rahmat, taufik, serta hidayah-Nya yang tiada henti, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul *"Klasifikasi Sekuens DNA Diabetes Melitus dengan Algoritma K-Nearest Neighbor"* ini dengan lancar dan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Matematika pada Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Proses penyusunan skripsi ini tidaklah mudah dan penuh dengan tantangan. Namun berkat doa, dukungan, semangat, serta bimbingan dari berbagai pihak, penulis dapat melewati setiap tahap dengan penuh keikhlasan dan ketekunan. Oleh karena itu, dengan segala kerendahan hati dan rasa hormat, penulis ingin menyampaikan ucapan terima kasih yang setulus-tulusnya kepada semua pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung, dalam proses penyelesaian skripsi ini.

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. Ach. Nashichuddin, M.A, selaku Dosen Pembimbing II.
6. Dr. Usman Pagalay, M.Si, selaku Ketua Penguji.
7. Dr. Mohammad Jamhuri, M.Si, selaku Anggota Penguji 1.
8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, selaku pengajar dan pembimbing akademik.
9. Sayyed Aamir Hassan, Muhammad Fathun Nuha, M. Wildan Nuril Akmal, dan Moh. Nurul Cholil selaku teman diskusi skripsi.
10. Kedua orang tua, Nur Affan dan Ainur Rusiyati, selaku pemberi dukungan moral dan material.

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, 14 Maret 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL ..... **i**
HALAMAN PENGAJUAN ..... **ii**
HALAMAN PERSETUJUAN ..... **iii**
HALAMAN PENGESAHAN ..... **iv**
PERNYATAAN KEASLIAN ..... **v**
MOTTO ..... **vi**
PERSEMBAHAN ..... **vii**
KATA PENGANTAR ..... **viii**
DAFTAR ISI ..... **x**
DAFTAR TABEL ..... **xii**
DAFTAR GAMBAR ..... **xiii**
DAFTAR SIMBOL ..... **xiv**
DAFTAR LAMPIRAN ..... **xv**
ABSTRAK ..... **xvi**
ABSTRACT ..... **xvii**
مستخلص البحث ..... **xviii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 7
1.3 Tujuan Penelitian 7
1.4 Manfaat Penelitian 7
1.5 Batasan Masalah 8
BAB II KAJIAN TEORI **9**
2.1 Teori Pendukung 9
2.1.1 K-Nearest Neighbor 9
2.1.2 Distance Metrics 12
2.1.3 K-Mers 16
2.1.4 Count vectorizer 17
2.1.5 SMOTE Oversampling 18
2.1.6 Label encoder 20
2.1.7 Min-max scaler 21
2.1.8 Confusion matrix 22
2.1.9 Sekuens DNA 24
2.1.10 Diabetes Melitus 25
2.2 Proses Penciptaan Manusia Menurut Al-Quran 26
BAB III METODE PENELITIAN **29**
3.1 Jenis Penelitian 29
3.2 Data dan Sumber Data 29
3.3 Teknik Analisis Data 30
3.3.1 *Preprocessing* Data 30
3.3.2 Klasifiakasi Model KNN 32
3.3.3 Evaluasi Model 33
BAB IV PEMBAHASAN **35**
4.1 Preprocessing Data 35
4.1.1 Cleaning Data 35
4.1.2 3-Mers 37

4.1.3 *Count vectorizer* 38
4.1.4 *SMOTE Oversampling* 40
4.1.5 *Min-max scaler* 43
4.1.6 Label encoder 45
4.2 Klasifikasi Model KNN 46
4.2.1 Deskripsi Model KNN pada Sampel Data 46
4.2.2 Hasil Klasifikasi Model KNN 51
4.2.3 Prediksi Sekuens DNA Baru dengan model KNN 54
4.3 Evaluasi Model dengan Confusion Matrix 56
4.4 Klasifikasi DNA Diabetes Melitus dalam Prespektif Islam 64
BAB V PENUTUP 66
5.1 Kesimpulan 66
5.2 Saran 68
DAFTAR PUSTAKA 69
DAFTAR LAMPIRAN 72
RIWAYAT HIDUP 73

DAFTAR TABEL

Tabel 2.1 Confusion Matrix 23
Tabel 4.1 Data Sekuens DNA Diabetes Melitus 36
Tabel 4.2 Perbandingan *Accuracy* K-Mers 37
Tabel 4.3 Sekuens DNA K-Mers..... 38
Tabel 4.4 Sekuens DNA *Count vectorizer* 39
Tabel 4.5 Kelas Setelah Oversampling 40
Tabel 4.6 DNA Count Vectorizer setelah Oversampling..... 42
Tabel 4.7 Tabel setelah Normalisasi Data 44
Tabel 4.8 Label Encoder 45
Tabel 4.9 Count Vectorizer dengan Sampel Data..... 46
Tabel 4.10 Sampel Data Setelah SMOTE 48
Tabel 4.11 Sampel Data *Min-max scaler* 49
Tabel 4.12 Hasil Jarak Sampel Sekuens DNA 50
Tabel 4.13 Perbandingan Data Test..... 52
Tabel 4.14 Perbandingan Nilai K-Nearest Neighbor 53
Tabel 4.15 Perbandingan Bobot 53
Tabel 4.16 Perbandingan Jarak 54
Tabel 4.17 Confusion Matrix 57
Tabel 4.18 Hasil Klasifikasi 60
Tabel 4.19 Perbandingan dengan Penelitian Terdahulu 61

DAFTAR GAMBAR

Gambar 2.1 Teori KNN 10
Gambar 2.2 K-Mers 16
Gambar 2.3 *Count vectorizer* 18
Gambar 2.4 *Oversampling* 19
Gambar 2.5 *Label encoder* 21
Gambar 4.1 Script Code Cleaning Data 35
Gambar 4.2 Script Code K-mers 37
Gambar 4.3 Script Code K-Mers 39
Gambar 4.4 Script Code SMOTE Oversampling 41
Gambar 4.5 Script Code *Min-max scaler* 43
Gambar 4.6 Script Code Label Encoder 45
Gambar 4.7 Script Code Split Data 52
Gambar 4.8 Script Code Prediksi DNA 55
Gambar 4.9 Prediksi DNA 55
Gambar 4.10 Confusion Matrix 56

DAFTAR SIMBOL

W	: Jarak perhitungan *distance weight KNN*
D	: Jarak baris x terhadap baris y
x_i	: Jarak dalam baris pertama yang akan dihitung
y_i	: Jarak dalam baris kedua yang akan dihitung
z	: data sintetis baru yang dihasilkan
w	: bilangan acak dari distribusi uniform dalam rentang [0, 1]
x_0	: titik data asli dari kelas minoritas yang dipilih secara acak
x_1	: salah satu dari *tetangga terdekat* dari x_0
$x_{i,j}$	: Nilai fitur pada baris ke j kolom ke i
$x'_{i,j}$	: Nilai fitur pada baris ke j kolom ke i setelah *min-max scaler*
$\min(\vec{x}_i)$	: Fitur terkecil dalam kolom i
$\max(\vec{x}_i)$	: Fitur terbesar dalam kolom i

DAFTAR LAMPIRAN

Lampiran 1 Data dan Coding 72

ABSTRAK

Armand, Muhammad Kristover 2025. Klasifikasi Sekuens DNA Diabetes Melitus dengan Algoritma K-Nearest Neighbor. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Ari Kusumastuti, M.Pd, M.Si (2) Ach. Nashichuddin, M.A

Kata Kunci: K-Nearest Neighbor; Sekuens DNA; Diabetes Melitus

Penelitian ini meneliti tentang klasifikasi sekuens DNA diabetes melitus dalam machine learning berguna untuk mendeteksi pola penyakit. Sekuens DNA pada dataset penelitian berasal dari gen insulin yaitu protein yang diproduksi oleh tubuh manusia untuk mengatur kadar glukosa yang menyebabakn diabetes melitus. Model klasifikasi untuk mendeteksi diabetes melitus berdasarkan sekuens DNA dengan menggunakan algoritma K-Nearest Neighbor (KNN). Hasil evaluasi model menunjukkan bahwa *accuracy* terbaik tercapai sebesar 93,26%, dengan menggunakan 3-mers, 10% data uji, 2-Nearest Neighbor, bobot jarak, dan jarak Minkowski ($p = 3$). Model ini dapat membedakan dengan baik antara diabetes melitus tipe 1, tipe 2, dan gestasional. Kontribusi utama dari penelitian ini adalah pengembangan prediksi sekuens DNA baru, yang dapat digunakan untuk membantu deteksi dini dan penanganan diabetes melitus. Dengan *accuracy* yang tinggi, model ini memiliki potensi untuk digunakan dalam aplikasi klinis untuk meningkatkan diagnosis dan pengobatan penyakit diabetes melitus.

ABSTRACT

Armand, Muhammad Kristover. 2025. Classification of Diabetes Mellitus DNA Sequences with K-Nearest Neighbor Algorithm. Thesis, Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (1) Ari Kusumastuti, M.Pd, M.Si, (2) Ach. Nashichuddin, M.A.

Keywords**:** K-Nearest Neighbor; DNA Sequences; Diabetes Mellitus

This study investigates the classification of DNA sequences related to diabetes mellitus in machine learning, aimed at detecting disease patterns. The DNA sequences in the research dataset originate from the insulin gene, a protein produced by the human body to regulate glucose levels, which is linked to diabetes mellitus. The classification model detects diabetes mellitus based on DNA sequences using the K-Nearest Neighbor (KNN) algorithm. The evaluation results show that the best accuracy achieved is 93,26%, with 3-mers, 10% test data, 2-Nearest Neighbor, distance weight, and the Minkowski distance ($p = 3$). The model effectively distinguishes between type 1 diabetes, type 2 diabetes, and gestational diabetes. The main contribution of this research is the development of a novel DNA sequence prediction model that can assist in the early detection and management of diabetes mellitus. With high accuracy, this model has the potential for clinical applications to improve the diagnosis and treatment of diabetes mellitus.

مستخلص البحث

أرماند، محمد كريستوفر، ٢٠٢٥. مرض السكري تصنيف تسلسل الحمض النووي باستخدام خوارزمية أقرب الجيران. البحث الجامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف: (1) آري كوسوماستوتي، الماجستير في التربية و العلوم. (2) أحمد ناصح الدين، ماجستير في العلوم الدينية.

الكلمات المفتاحية: أقرب جار *(K-Nearest Neighbor)*؛ تسلسلات الحمض النووي؛ مرض السكري.

بحثت هذه الدراسة في تصنيف تسلسلات الحمض النووي المرتبطة بمرض السكري في التعلم الآلي، يرجى بهدف للكشف عن أنماط المرض. تنشأ تسلسلات الحمض النووي في مجموعة البيانات البحثية من جين الأنسولين، وهو بروتين ينتجه جسم الإنسان لتنظيم مستويات الجلوكوز، وهو مرتبط بمرض السكري. يكتشف نموذج التصنيف داء السكري بناءً على تسلسل الحمض النووي باستخدام خوارزمية *K-Nearest Neighbor (KNN)*. أظهرت نتائج التقييم أن أفضل دقة تم تحقيقها هي 93,26%، مع *3-mers*، و10% بيانات اختبار، و *2-Nearest Neighbor*، ووزن المسافة، ومسافة مينكوفسكي .($p = 3$) يميز النموذج بشكل فعال بين مرض السكري من النوع الأول، ومرض السكري من النوع الثاني، وسكري الحمل. تتمثل المساهمة الرئيسية لهذا البحث في تطوير نموذج جديد للتنبؤ بتسلسل الحمض النووي يمكن أن يساعد في الكشف المبكر عن مرض السكري وإدارته. بفضل دقته العالية، يتمتع هذا النموذج بإمكانية التطبيقات السريرية لتحسين تشخيص وعلاج مرض السكري.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Diabetes melitus adalah penyakit kronis yang terjadi ketika pankreas tidak menghasilkan cukup insulin atau ketika tubuh tidak dapat menggunakan insulin secara efektif (WHO, 2024). Penyakit ini disebabkan oleh gangguan pada kadar gula darah (glukosa) yang berkaitan dengan produksi atau penggunaan insulin, yaitu hormon yang bertanggung jawab dalam mengatur metabolisme gula darah. Diabetes tipe 1 disebabkan oleh ketidakmampuan tubuh menghasilkan insulin, sedangkan diabetes tipe 2 terjadi ketika tubuh tidak dapat menggunakan insulin dengan efektif. Selain berdampak pada kesehatan fisik, seperti risiko komplikasi pada jantung, ginjal, mata, dan saraf, diabetes juga memengaruhi kualitas hidup secara keseluruhan. Di Indonesia, penyakit ini menjadi salah satu penyebab utama kematian, dengan jumlah kasus yang terus meningkat seiring perkembangan urbanisasi, perubahan pola makan, dan gaya hidup modern. Oleh karena itu, upaya pencegahan dan penanganan diabetes melalui pola hidup sehat, deteksi dini, serta pengelolaan yang tepat sangat diperlukan untuk mencegah komplikasi yang lebih serius di masa depan (El-Attar dkk., 2022).

Penyakit adalah kondisi yang mengganggu kesehatan fisik, mental, atau spiritual seseorang, dan setiap manusia pasti akan mengalaminya dalam berbagai bentuk dan tingkatan. Namun, Islam mengajarkan bahwa setiap penyakit pasti memiliki obatnya, sebagaimana disampaikan dalam sabda Nabi Muhammad SAW (NU Online, 2024).

"Sesungguhnya Allah menurunkan penyakit dan obatnya dan menjadikan bagi setiap penyakit ada obatnya. Maka berobatlah kalian, dan jangan kalian berobat dengan yang haram." (HR. Abu Dawud dari Abu Darda)

Prinsip ini memberikan harapan dan dorongan bagi setiap orang yang sedang sakit bahwa kesembuhan adalah sesuatu yang mungkin dicapai. Sebagaimana juga dijelaskan dalam Al-Quran dalam surat Yunus ayat 57 (Kemenag, 2024).

يٰٓاَيُّهَا النَّاسُ قَدْ جَاۤءَتْكُمْ مَّوْعِظَةٌ مِّنْ رَّبِّكُمْ وَشِفَاۤءٌ لِّمَا فِى الصُّدُوْرِۙ وَهُدًى وَّرَحْمَةٌ لِّلْمُؤْمِنِيْنَ ﴿٥٧﴾

"Wahai manusia, sungguh telah datang kepadamu pelajaran (Al-Qur'an) dari Tuhanmu, penyembuh bagi sesuatu (penyakit) yang terdapat dalam dada, dan petunjuk serta rahmat bagi orang-orang mukmin." (QS. Yunus: 57)

Menurut M. Quraish Shihab (2001) dalam karya tafsirnya, wahyu yang diturunkan kepada Nabi Muhammad SAW memuat ajaran yang sarat dengan peringatan agar manusia senantiasa beriman dan taat kepada Allah SWT. Al-Qur'an memberikan berbagai nasihat untuk mengajak umat manusia melakukan perbuatan baik serta menjauhkan diri dari keburukan. Di dalamnya juga terdapat kisah-kisah umat terdahulu yang dapat dijadikan pelajaran dan bahan perenungan. Selain itu, Al-Qur'an mengajak manusia untuk mengamati tanda-tanda kebesaran Allah di alam semesta sebagai bukti keagungan-Nya. Kitab suci ini juga berfungsi sebagai penyembuh bagi berbagai penyakit hati seperti kemusyrikan dan kemunafikan. Sebagai pedoman hidup yang menunjukkan jalan kebenaran, dan menjadi rahmat bagi siapa pun yang menerimanya dengan keimanan. Tidak hanya menyentuh aspek spiritual, Al-Qur'an juga memuat petunjuk yang berkaitan dengan penyembuhan penyakit fisik dan psikologis.

Kesehatan mental memiliki hubungan yang erat dan saling memengaruhi dengan kesehatan fisik, di mana gangguan kejiwaan seperti depresi, kecemasan, dan stres kronis dapat memicu atau memperburuk berbagai penyakit fisik seperti

penyakit jantung, diabetes melitus, gangguan pencernaan, dan gangguan kekebalan tubuh. Sebaliknya, penyakit fisik yang berkepanjangan juga dapat menimbulkan atau memperparah masalah kesehatan jiwa. Hubungan dua arah ini menunjukkan bahwa kesehatan jiwa tidak bisa dipisahkan dari kesehatan secara keseluruhan, sehingga penting untuk menangani keduanya secara bersamaan guna mencegah komplikasi dan meningkatkan kualitas hidup individu (Kemenkes, 2024).

DNA (*Deoxyribonucleic Acid*) adalah molekul utama yang menyimpan informasi genetik pada setiap makhluk hidup. Informasi genetik ini tersusun dalam urutan empat basa nitrogen, yaitu adenin (A), timin (T), sitosin (C), dan guanin (G). Setiap urutan basa ini membentuk kode genetik yang mengatur karakteristik dan fungsi biologis suatu organisme. Data genetika manusia mencakup informasi mengenai struktur dan fungsi genom, termasuk urutan DNA, variasi genetik, serta pola pewarisan sifat dari satu generasi ke generasi berikutnya. Informasi genetik ini tersimpan dalam molekul DNA yang berperan sebagai penyimpan informasi genetik di dalam sel-sel tubuh manusia. (Famuji dkk., 2024). Dalam beberapa tahun terakhir, teknologi *sequencing* dan kemampuan komputasi telah berkembang pesat, memungkinkan analisis data urutan DNA dalam jumlah besar. Hal ini memungkinkan penggunaan teknik seperti klasifikasi sekuens DNA, yang sangat berguna untuk mendeteksi pola penyakit. Teknik ini dapat digunakan untuk membandingkan pola urutan DNA individu yang terinfeksi dengan data yang ada, memungkinkan identifikasi penyakit lebih awal. (Yang dkk., 2020).

Machine learning adalah teknologi yang memungkinkan komputer untuk menganalisis sejumlah besar data dan menemukan pola-pola tersembunyi yang dapat digunakan untuk membuat prediksi dan keputusan yang lebih baik. *Machine*

learning memungkinkan komputer untuk secara mandiri mempelajari data yang dimilikinya dan mengambil keputusan secara cerdas. Dalam era big data, di mana volume data terus meningkat, *machine learning* memainkan peran penting dalam membantu menyelesaikan permasalahan data yang kompleks dengan cara yang cepat dan efektif (Retnoningsih dan Pramudita, 2020).

Machine learning mampu mengatasi berbagai tantangan di berbagai sektor seperti kesehatan, pendidikan, industri, dan keuangan. Algoritma *machine learning* dapat mengumpulkan dan menganalisis data dalam jumlah besar untuk memprediksi hasil dengan akurat atau mengelompokkan data berdasarkan pola-pola tertentu. Dalam metode supervised learning, teknik ini digunakan untuk tugas-tugas seperti regresi, klasifikasi, serta pengelompokan data. Jika klasifikasi hanya melibatkan dua kategori, maka disebut binary classification, sedangkan jika lebih dari dua kategori, dikenal sebagai multiple classification. Klasifikasi *machine learning* ini telah diterapkan dengan sukses dan menunjukkan efektivitasnya dalam berbagai bidang (Zhang, 2022).

Klasifikasi *K-Nearest Neighbor* (KNN) dikenal sebagai solusi yang efektif untuk pendekatan data. Seiring perkembangan teknologi dan meningkatnya volume data yang harus dianalisis, KNN semakin banyak digunakan di berbagai bidang, seperti *data mining*. Algoritma KNN sudah banyak diterapkan di berbagai bidang, termasuk untuk klasifikasi sekuens DNA. Keunggulan utama KNN adalah kemampuannya dalam bekerja tanpa memerlukan asumsi tertentu terkait distribusi data. Namun, agar hasil klasifikasinya optimal, penting untuk memilih parameter k yang sesuai dan melakukan *preprocessing* yang baik, terutama pada dataset sekuens DNA yang berukuran besar dan rumit (Zhang, 2022).

Klasifikasi urutan DNA dapat membantu mempelajari dan menemukan fungsi baru dari suatu protein. Insulin adalah contoh protein yang diproduksi oleh tubuh manusia untuk mengatur kadar glukosa. Diabetes adalah salah satu penyakit kronis yang banyak tersebar dan, jika diagnosis serta pengobatannya tidak tepat, dapat mengakibatkan efek serius dalam jangka Panjang. Klasifikasi DNA merupakan peran penting, terutama dalam bidang diagnosis penyakit genetik yang berkaitan dengan penyakit seperti diabetes melitus, kanker, dan lainnya. Kemajuan teknologi sekuensing DNA yang dipadukan dengan metode *machine learning*, seperti KNN, membuka peluang besar dalam meningkatkan *accuracy* deteksi dan klasifikasi berbagai penyakit. Penelitian lebih lanjut mengenai penerapan *Machine learning* dalam analisis sekuens DNA sangat penting untuk membantu diagnosis dini dan pengelolaan penyakit dengan lebih efisien, serta menghadapi berbagai tantangan di dunia kesehatan secara global. (El Attar er al., 2022).

Najar dkk. (2022) telah melakukan penelitian untuk mengidentifikasi sekuens DNA jenis kanker darah (leukimia) menggunakan berbagai algoritma klasifikasi dengan *accuracy* tertinggi 83,3% untuk Decision Tree Classifier. Beberapa penelitian sebelumnya (Zou dkk., 2018) menggunakan teknik *machine learning* seperti jaringan saraf, dan *random forest* untuk memprediksi diabetes mellitus, dengan indikator kadar *fasting* glukosa dengan algoritma *random forest* memberikan hasil *accuracy* 80,48%. Sementara itu, Hathaway dkk. (2019) menerapkan algoritma *machine learning Classification and Regression Tree* pada data sekuensing biokimia dan fisiologis untuk mengidentifikasi biomarker yang mempengaruhi status diabetes dengan *accuracy* sebesar 84%. Di sisi lain, Faruqui dkk., (2019) menggunakan pembelajaran mendalam berbasis *Long Short-Term*

Memory (LSTM) untuk memprediksi kadar glukosa harian pasien diabetes tipe 2, berdasarkan data kesehatan harian dari ponsel seperti aktivitas fisik, berat badan, pola makan, dan kadar glukosa, dengan *accuracy* prediksi mencapai 84,12%. El-Attar dkk., (2022) menerapkan algoritma LSTM dengan data sekuens DNA diabetes melitus didapatkan *accuracy* sebesar 95%.

Beberapa algoritma yang menunjukkan *accuracy* yang cukup baik, hasil ini masih menunjukkan adanya peluang untuk meningkatkan *accuracy* klasifikasi dengan melakukan pembaruan pada pendekatan pemrosesan data dan optimasi algoritma. Penelitian terdahulu tersebut memiliki beberapa kekurangan seperti tidak menggunakan teknik k-mers untuk ekstraksi fitur dari sekuens DNA. Selain itu, tidak adanya normalisasi data juga menjadi faktor yang mempengaruhi hasil klasifikasi. Normalisasi data seperti penerapan *Min-max scaler*, dapat membantu menstandarisasi fitur dan mengurangi bias yang diakibatkan oleh skala variabel yang berbeda. Penerapan *Min-max scaler* dan *distance weight* pada KNN menunjukkan bahwa *accuracy* klasifikasi sekuens DNA dapat meningkat dibandingkan hasil sebelumnya, sehingga menghasilkan model yang lebih akurat dan andal dalam mendeteksi jenis kanker darah berdasarkan sekuens DNA. Penambahan. Penelitian ini dapat digunakan untuk memprediksi sekuens DNA baru apakah termasuk diabetes melitus tipe 1, diabetes melitus tipe 2, dan gestasional. Pembaruan ini akan menambah kontribusi signifikan dalam bidang sekuens DNA. Dengan melakukan perbaikan ini, diharapkan penelitian selanjutnya dapat mencapai hasil yang lebih optimal dan memberikan pemahaman yang lebih baik tentang klasifikasi sekuens DNA pada penderita diabetes melitus.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijabarkan, maka rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana penerapan algoritma *k-nearest neighbor* (KNN) untuk mengklasifikasikan sekuens DNA diabetes melitus?
2. Bagaimana hasil evaluasi algoritma *k-nearest neighbor* (KNN) dalam hal *accuracy*, *precision, recall,* dan *f1-score* pada klasifikasi sekuens DNA ?

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah yang dijabarkan, maka tujuan penelitian penelitian ini adalah sebagai berikut:

1. Menganalisis penerapan algoritma *k-nearest neighbor* (KNN) untuk mengklasifikasikan sekuens DNA yang terkait dengan diabetes melitus.
2. Mengevaluasi performa algoritma *k-nearest neighbor* (KNN) dalam klasifikasi sekuens DNA, dengan mengukur *accuracy*, *precision, recall,* dan *f1-score.*

1.4 Manfaat Penelitian

Berdasarkan perumusan masalah yang dijabarkan, maka tujuan penelitian penelitian ini adalah sebagai berikut:

1. Penelitian ini diharapkan dapat memberikan pemahaman lebih mendalam tentang penerapan algoritma *k-nearest neighbor* (KNN) dalam klasifikasi sekuens DNA yang berkaitan dengan diabetes melitus. Selain itu, penelitian ini juga bisa menjadi referensi untuk mengembangkan metode klasifikasi yang lebih akurat dalam mengidentifikasi penyakit berdasarkan data genetik.

2. Penelitian ini adalah untuk memberikan informasi yang komprehensif mengenai performa algoritma *k-nearest neighbor* (KNN) dalam klasifikasi sekuens DNA, khususnya dengan mengukur *accuracy*, *precision, recall,* dan *f1-score.* Hasil evaluasi ini diharapkan dapat menjadi acuan dalam memilih metode klasifikasi yang paling tepat dan efisien dalam analisis data genetik, serta membantu pengembangan teknik yang lebih akurat untuk deteksi dini penyakit berbasis sekuens DNA.

1.5 Batasan Masalah

Agar penelitian ini tetap terarah pada tujuan yang diinginkan, perlu ditetapkan beberapa pembatasan masalah sebagai berikut:

1. Data berjumlah 1577.
2. Menggunakan 3-mers yang digunakan untuk membagi DNA.
3. Data *test* berjumlah 10%.
4. Nilai k pada nearest neighbor berjumlah 2.
5. Perhitungan KNN menggunakan bobot jarak.
6. Perhitungan KNN menggunakan jarak minkowski dengan $p = 3$

.

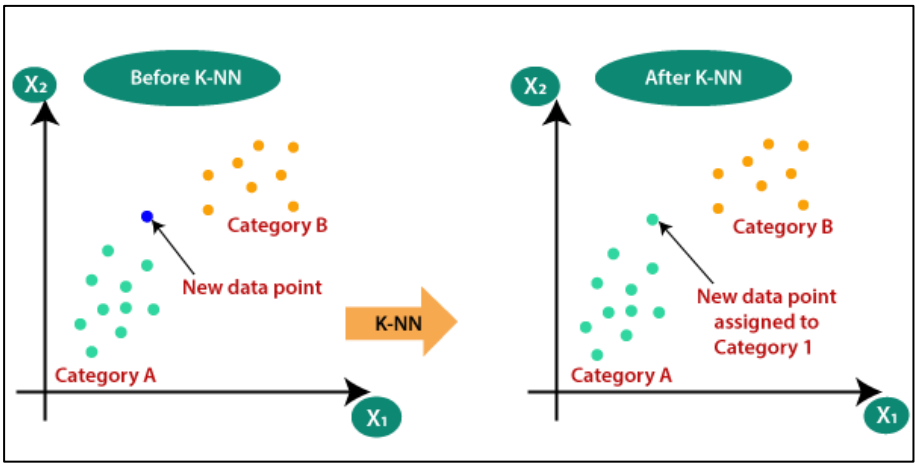
BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 K-Nearest Neighbor

K-Nearest Neighbor (KNN) adalah algoritma klasifikasi yang sangat populer di dunia *Machine learning*, data mining, dan statistik karena kesederhanaan penerapannya serta performanya yang bagus dalam klasifikasi. Algoritma ini bekerja dengan mencari data baru yang paling dekat dengan data dalam dataset pelatihan, berdasarkan prinsip bahwa data yang berdekatan biasanya memiliki label yang sama. KNN menggunakan parameter k untuk menentukan jumlah tetangga terdekat yang dipertimbangkan, kemudian mengklasifikasikan data baru berdasarkan mayoritas label dari k tetangga tersebut. Tantangan terbesar dalam menggunakan KNN adalah menentukan nilai k yang optimal, karena menetapkan nilai k yang tetap untuk semua sampel uji sering kali tidak ideal. Dalam praktik umum, nilai k biasanya dipilih secara manual oleh ahli atau berdasarkan eksperimen, namun pendekatan ini tidak selalu efektif untuk semua jenis data. Nilai k yang optimal dapat bervariasi tergantung pada karakteristik data, sehingga solusi terbaru mencoba menetapkan nilai k yang berbeda untuk setiap sampel uji untuk meningkatkan *accuracy* klasifikasi (Zhang, 2022).



Gambar 2.1 Teori KNN
Sumber: Nandan (2020)

Pada gambar 2.1 dijelaskan bahwa algoritma K-Nearest Neighbor (KNN) melakukan proses klasifikasi berdasarkan prinsip jarak terdekat. Algoritma ini menentukan kelas dari suatu data baru dengan mengidentifikasi sejumlah data tetangga terdekatnya dalam ruang fitur, biasanya berdasarkan metrik jarak. Data baru tersebut kemudian diklasifikasikan ke dalam kelas mayoritas dari tetangga-tetangganya, sehingga hasil klasifikasi sangat bergantung pada distribusi data dan jumlah tetangga yang dipilih.

Algoritma KNN mengukur jarak antara data baru dan data pelatihan menggunakan metrik jarak seperti *euclidean distance, manhattan distance, minkowski distance, dan chebyshev distance* yang dipilih berdasarkan kebutuhan dataset. KNN diterapkan dalam berbagai bidang, seperti pengenalan pola, sistem rekomendasi, analisis medis, dan banyak lagi. Keberhasilan algoritma ini sangat dipengaruhi oleh pemilihan nilai k yang tepat dan kualitas *preprocessing* data, agar model dapat menghasilkan prediksi yang akurat dan relevan. Salah satu keunggulan KNN adalah fleksibilitasnya, yang memungkinkan penyesuaian lebih lanjut dalam proses klasifikasi, terutama jika digabungkan dengan teknik modern untuk penentuan nilai k secara dinamis (Muhammad Ali, 2022).

Distance Weight K-Nearest Neighbor adalah pengembangan dari algoritma KNN yang digunakan untuk mengurangi tingkat kesalahan dalam klasifikasi. Pada algoritma KNN, proses klasifikasi dilakukan dengan menghitung jarak antara data baru dan data dalam dataset pelatihan, kemudian memilih k tetangga terdekat berdasarkan kelas mereka. Algoritma ini menggunakan berbagai metode untuk mengukur jarak, seperti *euclidean distance, manhattan distance, minkowski distance, dan chebyshev distance*. Namun, kelemahan dari KNN adalah jika semua tetangga memiliki bobot yang sama dalam menentukan hasil klasifikasi yaitu jika memilih k yang berjumlah genap. Hal ini sering kali mengakibatkan hasil klasifikasi kurang akurat, terutama ketika distribusi data tidak merata atau terdapat *outlier*.

Distance weight KNN digunakan untuk mengatasi masalah tersebut dengan memberikan bobot pada setiap tetangga berdasarkan jaraknya dengan data yang akan diklasifikasikan. Tetangga yang lebih dekat dengan data baru diberikan bobot lebih besar, sementara tetangga yang lebih jauh diberi bobot lebih kecil. untuk setiap tetangga (W) dihitung berdasarkan jarak (D) dengan persamaan berikut:

$$W = \frac{1}{D} \tag{2.1}$$

Keterangan:

W : Jarak perhitungan *distance weight KNN*

D : Jarak baris x terhadap baris y

Pada persamaan (2.1), *distance weight* KNN memberikan prioritas lebih kepada tetangga yang lebih relevan, yang berjarak lebih dekat dengan sampel, sehingga menghasilkan klasifikasi yang lebih akurat. Selain itu, *distance weight*

KNN juga memperbaiki kelemahan *uniform weight* di mana semua fitur dianggap sama penting dalam menentukan tetangga terdekat. Dengan memberikan bobot pada setiap tetangga, algoritma ini menjadi lebih efektif dalam mengklasifikasikan data, terutama dalam kasus di mana relevansi fitur-fitur tertentu lebih dominan untuk tugas klasifikasi tertentu (Zhang, 2022).

2.1.2 Distance Metrics

Jarak minkowski adalah metrik yang digunakan untuk mengukur kedekatan atau kesamaan antara dua titik dalam ruang vektor, yang merupakan generalisasi dari beberapa jenis metrik lainnya, seperti jarak euclidean dan jarak manhattan. Metrik ini pertama kali diperkenalkan oleh matematikawan Hermann Minkowski dan banyak digunakan dalam *machine learning* dan *data*. Dalam konteks *machine learning*, jarak minkowski berguna untuk mengklasifikasikan data dan mengelompokkan objek berdasarkan kemiripan geometrisnya. (Jin dkk., 2023).

Secara matematis, jarak minkowski antara dua titik $x = (x_1, x_2 \dots x_n)$ dan $y = (y_1, y_2 \dots y_n)$ dalam ruang *n* dimensi diberikan oleh persamaan:

$$D = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{2.2}$$

Keterangan:

D : Jarak baris x terhadap baris y

x_i : Jarak dalam baris pertama yang akan dihitung

y_i : Jarak dalam baris kedua yang akan dihitung

Parameter p yang ada pada persamaan (2.2) berfungsi untuk mengatur jenis jarak yang digunakan. ketika $p = 1$, jarak minkowski berubah menjadi jarak manhattan, yang menghitung jarak absolut antara titik-titik. Sementara itu, untuk $p = 2$, metrik ini berubah menjadi jarak euclidean, yang merupakan jarak garis lurus terpendek antara dua titik, jika $p = \infty$ menjadi jarak chebyshev. Dengan fleksibilitas ini, jarak minkowski dapat disesuaikan untuk berbagai jenis data dan situasi, yang membuatnya sangat berguna dalam analisis data multidimensi. Pada persamaan tersebut, parameter p menentukan seberapa besar dampak perbedaan pada setiap dimensi terhadap total jarak. Hal ini berarti bahwa, untuk nilai p yang lebih tinggi, perbedaan pada dimensi yang paling dominan akan lebih berpengaruh terhadap total jarak.

Jarak manhattan adalah metrik yang paling umum digunakan untuk mengukur jarak antara dua titik dalam ruang vektor. Jarak manhattan mengukur total jarak yang ditempuh untuk mencapai suatu titik dari titik lainnya dengan menjumlahkan perbedaan absolut antara setiap dimensi. Jarak antara dua titik dihitung sebagai jumlah dari perbedaan absolut antara koordinat kartesian mereka sehingga jarak manhattan dapat dipahami sebagai total penjumlahan perbedaan antara koordinat x dan koordinat y dari dua titik tersebut (Uyanik dan Orman, 2023).

Secara matematis, jarak manhattan antara dua titik $x = (x_1, x_2 \dots x_n)$ dan $y = (y_1, y_2 \dots y_n)$ dalam ruang *n* dimensi diberikan oleh persamaan:

$$D = \sum_{i=1}^{n} |x_i - y_i| \tag{2.3}$$

Keterangan:

D : Jarak baris x terhadap baris y

x_i : Jarak dalam baris pertama yang akan dihitung

y_i : Jarak dalam baris kedua yang akan dihitung

Pada persamaan (2.3) di mana $|x_i - y_i|$ adalah jarak absolut antara komponen *i* dari kedua titik. Pada persamaan ini, setiap dimensi berkontribusi secara langsung terhadap total jarak, sehingga setiap perubahan pada satu dimensi akan berpengaruh pada hasil akhir.

Jarak euclidean adalah salah satu metrik yang paling umum digunakan untuk mengukur jarak antara dua titik dalam ruang vektor, dan sering kali dianggap sebagai jarak garis lurus atau lintasan terpendek antara titik-titik tersebut. Nama metrik ini diambil dari matematikawan Yunani, Euclid, yang dikenal sebagai bapak geometri dan mengembangkan dasar-dasar geometri yang digunakan hingga saat ini. Dalam konteks analisis data, jarak euclidean sangat penting karena memberikan cara yang intuitif untuk menghitung kedekatan antara dua objek berdasarkan posisi mereka dalam ruang multidimensi. Sebagai contoh, dalam *Machine learning*, jarak euclidean sering digunakan dalam algoritma pengelompokan seperti k-means dan dalam algoritma klasifikasi seperti *KNN*. Dengan menggunakan jarak euclidean, model dapat mengevaluasi kemiripan antara titik data dan menentukan kelompok atau kelas yang paling relevan. (Harjanti dkk., 2022).

Secara matematis, jarak euclidean antara dua titik $x = (x_1, x_2 \dots x_n)$ dan $y = (y_1, y_2 \dots y_n)$ dalam ruang *n* dimensi diberikan oleh persamaan:

$$D = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.4}$$

Keterangan:

D : Jarak baris x terhadap baris y

x_i : Jarak dalam baris pertama yang akan dihitung

y_i : Jarak dalam baris kedua yang akan dihitung

Pada persamaan (2.4), $(x_i - y_i)^2$adalah selisih antara komponen i dari kedua titik, dan penjumlahan dari kuadrat selisih tersebut dihitung sebelum diambil akar kuadratnya.

Jarak chebyshev adalah ukuran penting yang digunakan untuk mengukur ketidaksamaan antara entitas, di mana semakin besar jaraknya, semakin besar pula perbedaan yang ditunjukkan. Metrik ini memiliki banyak aplikasi dalam tugas *Machine learning*, seperti clustering, klasifikasi. Jarak ini, yang juga dikenal sebagai jarak $p = \infty$ yang mengukur jarak antara dua vektor berdimensi n, dengan fokus pada perbedaan maksimum di seluruh dimensi (Nayak dkk., 2022).

Secara matematis, jarak chebyshev antara dua titik $x = (x_1, x_2 \dots x_n)$ dan $y = (y_1, y_2 \dots y_n)$ dalam ruang *n* dimensi diberikan oleh persamaan:

$$D = max(|x_i - y_i|) \tag{2.5}$$

Keterangan:

D : Jarak baris x terhadap baris y

x_i : Jarak dalam baris pertama yang akan dihitung

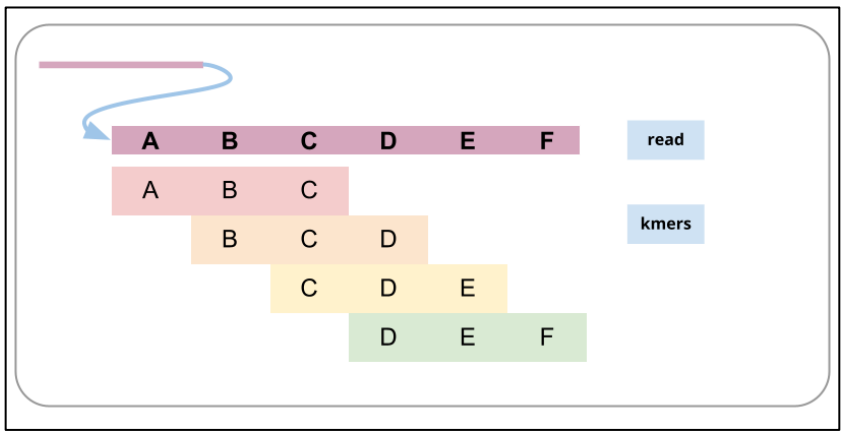
y_i : Jarak dalam baris kedua yang akan dihitung

Persamaan (2.5) digunakan untuk menghitung jarak Chebyshev antara dua titik atau vektor dalam ruang berdimensi-n, yaitu dengan mengambil selisih

mutlak dari setiap pasangan elemen pada posisi yang sama dari vektor x dan y, lalu memilih nilai yang paling besar di antara selisih-selisih tersebut.

2.1.3 K-Mers

K-mers adalah subsekuens unik dari panjang tertentu, k, yang diambil dari urutan DNA, RNA, atau protein yang lebih besar.K-mers memiliki peran penting dalam berbagai algoritma *Machine learning* dan analisis data biologis, karena mereka memungkinkan peneliti untuk mengubah urutan biologis menjadi representasi numerik yang dapat diproses oleh algoritma komputasi. Penggunaan k-mers yang berbeda, seperti k=3 (tri-mers), memberikan informasi yang berbeda terkait konteks biologis dari urutan tersebut. Analisis k-mers juga membantu menangani masalah terkait noise dalam data genomik, karena memberikan pendekatan yang lebih terfokus untuk memahami interaksi genetik. Dengan demikian, k-mers menjadi alat yang sangat penting dalam pemrosesan dan analisis data biologis modern (Pflug dkk., 2020).



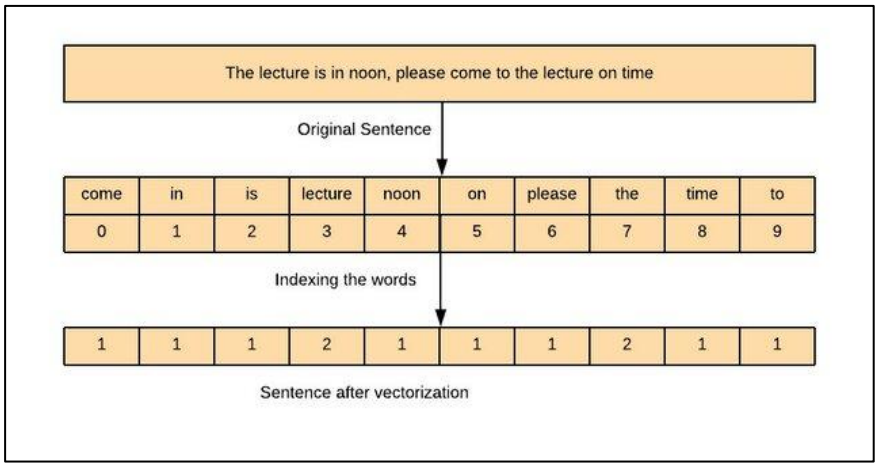
Gambar 2.2 K-Mers
Sumber: Syme (2023)

Pada gambar 2.2 dijelaskan bahwa k-mers adalah teknik untuk memecah data teks menjadi potongan-potongan kecil berukuran k karakter, yang diambil

secara berurutan dari teks asli. Misalnya, jika teks asli adalah "ABCDEF" dan ukuran k yang dipilih adalah 3, maka hasil k-mers-nya adalah "ABC", "BCD", "CDE", DAN "DEF". Teknik ini berguna dalam berbagai analisis data, termasuk pengolahan teks dan bioinformatika, untuk mengidentifikasi pola atau fitur penting dalam data tersebut.

2.1.4 Count vectorizer

Count vectorizer adalah teknik dalam pemrosesan bahasa alami (*Natural Language Processing*/NLP) yang digunakan untuk mengubah kumpulan teks menjadi representasi numerik, dengan cara menghitung frekuensi kemunculan kata dalam setiap dokumen. Metode ini mengonversi dokumen teks menjadi matriks vektor, di mana setiap baris mewakili sebuah dokumen dan setiap kolom mewakili kata unik yang ditemukan dalam seluruh koleksi teks. Nilai dalam matriks tersebut menunjukkan jumlah kemunculan setiap kata dalam dokumen tertentu. Dengan menggunakan *Count vectorizer*, yang juga dikenal sebagai term frequency, analisis teks menjadi lebih mudah dilakukan, terutama untuk aplikasi seperti klasifikasi teks, pengelompokan (*clustering*), dan analisis sentimen. *Count vectorizer* mencatat nilai frekuensi setiap token yang berbeda dalam dokumen; semakin besar nilai token, semakin sering token tersebut muncul dalam teks. Dengan demikian, *Count vectorizer* menjadi alat yang efektif untuk mengevaluasi dan menganalisis teks berdasarkan frekuensi kemunculan kata-kata kunci (Sajid dkk., 2023).



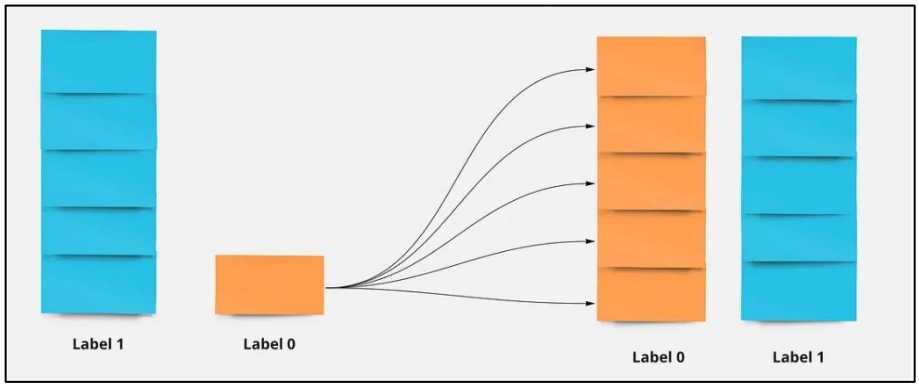
Gambar 2.3 *Count vectorizer*
Sumber: Shah dkk., (2020)

Pada gambar 2.3 dijelaskan bahwa *Count Vectorizer* adalah salah satu metode dalam pemrosesan teks yang digunakan untuk mengubah teks menjadi representasi numerik berupa vektor. Metode ini menghitung frekuensi kemunculan setiap kata dalam teks dan menyusunnya dalam bentuk vektor atau matriks. Seperti kata "*come*" dihitung kemunculannya sebanyak 1 kali, kata "*lecture*" dihitung kemunculannya sebanyak 2 kali.

2.1.5 SMOTE Oversampling

Oversampling adalah teknik yang digunakan dalam pemrosesan data untuk mengatasi masalah ketidakseimbangan kelas dalam dataset. Ketidakseimbangan kelas terjadi ketika jumlah contoh dari satu kelas jauh lebih sedikit dibandingkan dengan kelas lainnya, yang dapat menyebabkan model *machine learning* cenderung bias terhadap kelas mayoritas. *Oversampling* berfungsi untuk meningkatkan jumlah contoh pada kelas minoritas dengan cara menduplikasi data yang sudah ada atau menciptakan data baru berdasarkan pola yang ada dalam kelas tersebut. Teknik ini bertujuan untuk memberikan representasi yang lebih baik bagi kelas minoritas dalam dataset, sehingga model yang dilatih dapat lebih

akurat dan generalis dalam pengenalan pola, beberapa Teknik *oversampling* dalam *machine learning* yaitu random over sampler, SMOTE, ADASYN (Viloria dan Mecardo, 2020).



Gambar 2.4 *Oversampling*
Sumber: Alves (2021)

Berdasarkan gambar 2.4, *oversampling* adalah metode yang digunakan untuk menyeimbangkan data dengan cara menambahkan jumlah data pada kelas minoritas sehingga jumlahnya menjadi seimbang atau setara dengan jumlah data pada kelas mayoritas. Seperti contoh pada gambar, label 1 berjumlah 5, dan label 0 berjumlah 1. Oversampling bekerja dengan cara menambahkan label 0 sampai sama dengan label 1 yaitu berjumlah 5. Teknik ini diterapkan dalam masalah klasifikasi dengan data yang tidak seimbang untuk memastikan model dapat belajar dengan baik dari kedua kelas.

SMOTE (Synthetic Minority Over-sampling Technique) adalah metode *oversampling* yang berbeda dari pendekatan tradisional yang hanya menggandakan data kelas minoritas secara langsung. SMOTE menciptakan sampel baru secara sintetis dengan memanfaatkan interpolasi antara titik-titik data asli dalam kelas minoritas. Dengan cara ini, SMOTE tidak hanya menambah jumlah data kelas minoritas tetapi juga memperluas ruang data sehingga distribusi

data menjadi lebih representatif dan beragam Dengan menggunakan SMOTE, penelitian ini dapat menghasilkan dataset yang seimbang dan representatif, sehingga memungkinkan analisis yang lebih akurat dan bermakna dalam pengolahan data genetik. (Wongvorachan dkk., 2023).

$$z = x_0 + w(x_1 - x_0) \tag{2.6}$$

Keterangan:

z : data sintetis baru yang dihasilkan

w : bilangan acak dari distribusi uniform dalam rentang [0, 1]

x_0 : titik data asli dari kelas minoritas yang dipilih secara acak

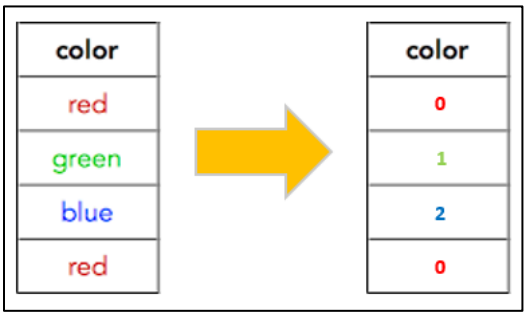
x_1 : salah satu dari *tetangga terdekat* dari x_0

Pada persamaan (2.6) Algoritma SMOTE pertama-tama memilih salah satu sampel minoritas x_0 secara acak. Misalkan untuk saat ini titik x_0 tersebut dianggap tetap atau sudah diberikan. Selanjutnya, kita memilih salah satu tetangga terdekat x_k secara acak dari K tetangga terdekat dari x_0 . Kemudian, kita memilih satu titik z secara acak dari garis yang menghubungkan tetangga tersebut x_k dan x_0 (Elreedy dkk., 2024).

2.1.6 Label encoder

Label encoder adalah teknik dalam pemrosesan data yang digunakan untuk mengubah label kategorikal menjadi representasi numerik. Teknik ini sangat penting dalam konteks *machine learning*, di mana banyak algoritma memerlukan input dalam bentuk angka untuk dapat berfungsi secara efektif. *Label encoder* mengonversi setiap kategori dalam variabel kategorikal menjadi bilangan bulat unik, di mana setiap label kategori diwakili oleh angka tertentu. Proses ini

memungkinkan model untuk memahami dan memanfaatkan informasi dari data kategorikal dalam analisis dan prediksi. *Label encoder* adalah metode pengkodean yang sederhana dan langsung, di mana setiap label diberikan angka unik. Dengan menggunakan *label encoder*, data kategorikal dapat diolah menjadi format numerik yang lebih mudah dipahami oleh algoritma *machine learning* (Ren dkk., 2022).



Gambar 2.5 *Label encoder*
Sumber: Jawle (2020)

Berdasarkan gambar 2.5, dalam dataset yang memiliki kategori seperti "Merah", "Hijau", dan "Biru", label encoder akan mengubahnya menjadi angka seperti 0, 1, dan 2. Proses ini memastikan bahwa model machine learning dapat memahami data kategori dalam bentuk numerik tanpa kehilangan informasi tentang hubungan antar kategori. Perlu diperhatikan bahwa nilai numerik yang dihasilkan tidak merepresentasikan urutan atau peringkat tertentu, melainkan hanya sebagai kode unik untuk masing-masing kategori.

2.1.7 Min-max scaler

Min-max scaler adalah teknik normalisasi data yang digunakan untuk mengubah rentang nilai dari fitur (variabel) dalam dataset sehingga berada dalam skala tertentu, biasanya antara 0 dan 1. *Min-max scaler* adalah metode normalisasi

yang dilakukan dengan melakukan transformasi linier pada data asli untuk menghasilkan keseimbangan perbandingan nilai antara data sebelum dan setelah proses. Teknik ini sangat penting dalam konteks *machine learning* dengan algoritma seperti KNN, bekerja lebih baik ketika fitur-fitur memiliki skala yang seragam. *Min-max scaler* berfungsi dengan cara mengurangi nilai minimum dari data dan membaginya dengan rentang (selisih antara nilai maksimum dan minimum) dari data tersebut. Formula yang digunakan untuk melakukan normalisasi dengan *Min-max scaler* adalah:

$$x'_{i,j} = \frac{x_{i,j} - \min(\vec{x}_i)}{\max(\vec{x}_i) - \min(\vec{x}_i)} \tag{2.7}$$

Keterangan:

$x_{i,j}$: Nilai fitur pada baris ke j kolom ke i

$x'_{i,j}$: Nilai fitur pada baris ke j kolom ke i setelah *min-max scaler*

$\min(\vec{x}_i)$: Fitur terkecil dalam kolom i

$\max(\vec{x}_i)$: Fitur terbesar dalam kolom i

Dengan menggunakan metode ini, semua nilai dalam dataset akan terdistribusi dalam rentang yang sama, sehingga menghindari masalah skala yang dapat mempengaruhi hasil analisis dan performa model (Henderi, 2021).

2.1.8 Confusion matrix

Confusion matrix adalah alat evaluasi yang digunakan dalam *Machine learning* untuk menganalisis kinerja model klasifikasi. Matrik ini menyajikan ringkasan visual dari hasil prediksi model dibandingkan dengan nilai aktual dari data. *Confusion matrix* terdiri dari empat elemen utama, yaitu *true positive* (TP),

true negative (TN), *false positive* (FP), dan *false negative* (FN). *Confusion matrix* adalah presentasi yang berguna dan komprehensif dari kinerja classifier untuk menghitung berbagai metrik evaluasi penting seperti *accuracy*, *precision*, *recall*, dan *f1-score*. *Accuracy* diukur sebagai proporsi prediksi yang benar, *precision* mengukur seberapa banyak dari prediksi positif yang benar-benar positif, sementara *recall* mengukur seberapa banyak dari kasus positif yang berhasil diidentifikasi oleh model. *f1-score* adalah rata-rata dari *precision* dan *recall*, memberikan keseimbangan antara keduanya. Selain merepresentasikan hasil klasifikasi, *confusion matrix* juga alat yang sangat untuk mengevaluasi dan memperbaiki model *machine learning* agar lebih efektif dalam tugas klasifikasi (Heydarian dkk., 2022).

Tabel 2.1 Confusion Matrix

Data	Prediksi Positif	Prediksi Negatif
Aktual Positif	*True positive* (TP)	*False negative* (FN)
Aktual Negatif	*False positive* (FP)	*True negative* (TN)

Berdasarkan tabel 2.1, *confusion matrix* terdapat:

1. *True positive* (TP): Jumlah prediksi positif yang benar.
2. *False negative* (FN): Jumlah prediksi positif yang salah (seharusnya positif tetapi diprediksi negatif).
3. *False positive* (FP): Jumlah prediksi negatif yang salah (seharusnya negatif tetapi diprediksi positif).
4. *True negative* (TN): Jumlah prediksi negatif yang benar.

Dari tabel *confusion matrix*, dapat dihitung berbagai metrik evaluasi yang digunakan untuk menilai kinerja model klasifikasi. Metrik-metrik ini mencakup *accuracy*, *precision*, *recall*, dan *f1-score*.

Accuracy adalah proporsi prediksi yang benar dari total prediksi yang dilakukan. Ini mengukur seberapa banyak prediksi model yang tepat dibandingkan dengan total data yang ada.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

Precision adalah proporsi contoh yang diprediksi positif yang benar-benar positif. Ini mengukur seberapa akurat prediksi positif dari model.

$$Precision = \frac{TP}{TP + FP} \tag{2.9}$$

Recall, juga dikenal sebagai sensitivitas, adalah proporsi contoh positif yang berhasil diidentifikasi oleh model. Ini mengukur kemampuan model untuk menangkap semua contoh positif

$$Recall = \frac{TP}{TP + FN} \tag{2.10}$$

F1-score adalah rata-rata dari *precision* dan *recall*. Ini memberikan gambaran seimbang dari kedua metrik tersebut, dan sangat berguna ketika memiliki kelas yang tidak seimbang (misalnya, jumlah positif jauh lebih sedikit dibandingkan jumlah negatif).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.11}$$

2.1.9 Sekuens DNA

Penelitian tentang DNA menjadi sangat penting dalam biologi molekuler karena kemampuan untuk membaca dan memahami urutan DNA memiliki pengaruh besar dalam berbagai bidang, mulai dari penelusuran evolusi hingga pengembangan pengobatan yang lebih baik. Metode untuk menentukan urutan

keempat pasangan basa dalam molekul DNA disebut DNA *sequencing* atau sekuens DNA. Pengetahuan tentang teknik pengurutan DNA semakin penting, baik dalam penelitian biologi dasar maupun dalam berbagai bidang lain seperti bioteknologi, diagnosis medis, virologi, biologi forensik, dan sistematika biologi. Dengan kemajuan teknologi, proses membaca urutan DNA menjadi jauh lebih mudah dan cepat, dan jumlah data urutan DNA juga meningkat dengan pesat (Mathur dkk., 2023).

Sekuens DNA adalah rangkaian nukleotida yang membentuk gen dan menentukan informasi genetik suatu organisme. Dalam konteks DNA, terdapat empat jenis basa nukleotida yang menjadi penyusun sekuens tersebut, yaitu Adenin (A), Sitosin (C), Guanin (G), dan Timin (T), yang secara kolektif sering disingkat sebagai ACGT. Urutan dan kombinasi dari keempat basa ini membentuk kode genetik yang mengatur berbagai fungsi biologis dalam tubuh, seperti sintesis protein dan regulasi gen. Setiap kombinasi tertentu dari basa-basa ini dapat mengkodekan informasi yang spesifik, yang penting untuk proses-proses biologis yang kompleks.

2.1.10 Diabetes Melitus

Diabetes melitus adalah penyakit endokrin yang diakui sebagai kondisi inflamasi kronis, yang dipicu oleh ketidakseimbangan sistem imun jangka panjang dan sindrom metabolik. Penyakit ini telah menjadi masalah kesehatan global dan salah satu penyebab utama kematian pada orang dewasa di seluruh dunia. Gangguan metabolisme ini ditandai dengan kadar glukosa darah yang tinggi akibat gangguan produksi insulin, fungsi insulin, atau kombinasi keduanya.

Diabetes dapat dibagi menjadi beberapa tipe, dengan dua yang paling umum adalah diabetes tipe 1 dan diabetes tipe 2, yang memiliki penyebab, karakteristik, dan pendekatan pengelolaan yang berbeda (Panua dkk., 2021).

Diabetes tipe 1, juga dikenal sebagai diabetes juvenil atau diabetes tergantung insulin, adalah kondisi autoimun di mana sistem kekebalan tubuh menyerang sel-sel beta di pankreas yang bertanggung jawab untuk memproduksi insulin, sehingga menyebabkan tubuh tidak dapat memproduksi insulin yang cukup. Biasanya muncul pada masa kanak-kanak atau remaja, diabetes tipe 1 memerlukan terapi insulin seumur hidup dan pemantauan kadar glukosa darah secara teratur. Di sisi lain, diabetes tipe 2 adalah bentuk diabetes yang lebih umum dan biasanya berkembang seiring bertambahnya usia, meskipun semakin banyak ditemukan pada anak-anak dan remaja akibat peningkatan obesitas. Pada diabetes tipe 2, tubuh tidak menggunakan insulin dengan efektif atau tidak memproduksi cukup insulin. Faktor risiko untuk diabetes tipe 2 termasuk kelebihan berat badan, kurangnya aktivitas fisik, dan pola makan yang tidak sehat, dengan gejala yang seringkali lebih halus. Pengelolaan diabetes tipe 2 dapat mencakup perubahan gaya hidup dan penggunaan obat-obatan untuk mengatur kadar glukosa darah.

2.2 Proses Penciptaan Manusia Menurut Al-Quran

Konsep penciptaan manusia yang detail dan bertahap sebenarnya telah dijelaskan dalam Al-Qur'an jauh sebelum ilmu pengetahuan modern berkembang. Dalam Surah Al-Mu'minun (23:12-14), Allah menggambarkan proses penciptaan manusia dari substansi biologis yang kompleks (Kemenag, 2024):

وَلَقَدْ خَلَقْنَا الْاِنْسَانَ مِنْ سُلٰلَةٍ مِّنْ طِيْنٍۚ ﴿١٢﴾

"Sungguh, Kami telah menciptakan manusia dari sari pati (yang berasal) dari tanah." (QS. Al-Mu'minun:12)

ثُمَّ جَعَلْنٰهُ نُطْفَةً فِيْ قَرَارٍ مَّكِيْنٍۖ ﴿١٣﴾

"Kemudian, Kami menjadikannya air mani di dalam tempat yang kukuh (rahim)." (QS. Al-Mu'minun:13)

ثُمَّ خَلَقْنَا النُّطْفَةَ عَلَقَةً فَخَلَقْنَا الْعَلَقَةَ مُضْغَةً فَخَلَقْنَا الْمُضْغَةَ عِظٰمًا فَكَسَوْنَا الْعِظٰمَ لَحْمًا ثُمَّ اَنْشَأْنٰهُ خَلْقًا اٰخَرَۗ فَتَبٰرَكَ اللّٰهُ اَحْسَنُ الْخٰلِقِيْنَ ﴿١٤﴾

"Kemudian, air mani itu Kami jadikan sesuatu yang menggantung (darah). Lalu, sesuatu yang menggantung itu Kami jadikan segumpal daging. Lalu, segumpal daging itu Kami jadikan tulang belulang. Lalu, tulang belulang itu Kami bungkus dengan daging. Kemudian, Kami menjadikannya makhluk yang (berbentuk) lain. Maha Suci Allah sebaik-baik pencipta." (QS. Al-Mu'minun:14)

Beberapa ulama menafsirkan ayat ini sebagai penjelasan mengenai proses penciptaan manusia melalui mekanisme reproduksi biologis. M. Quraish Shihab (2001), dalam bukunya, menekankan pentingnya pengamatan terhadap asal mula penciptaan manusia, karena hal tersebut merupakan salah satu bukti kekuasaan Allah yang semestinya mendorong keyakinan akan keberadaan Allah dan hari akhir. Allah menciptakan manusia dari saripati tanah, dan setelah melalui berbagai tahap perkembangan, segumpal darah tersebut dibentuk menjadi tulang belulang, yang kemudian dibalut dengan daging. Pada akhirnya, penciptaan disempurnakan, dan setelah roh ditiupkan, sebagian di antaranya menjadi makhluk yang menyimpang dari fitrahnya. Maha Agunglah Allah dalam kekuasaan dan keagungan-Nya, tiada yang dapat menandingi-Nya dalam hal kemampuan mencipta.

Penciptaan manusia dalam Al-Qur'an bukan hanya menggambarkan proses biologis, tetapi juga menekankan keistimewaan manusia sebagai makhluk ciptaan Allah SWT. Beberapa keistimewaan penciptaan manusia menurut Al-Qur'an dalam surat At-Tin ayat 4 (Kemenag, 2024)

لَقَدْ خَلَقْنَا الْاِنْسَانَ فِيْٓ اَحْسَنِ تَقْوِيْمٍۖ ﴿٤﴾

"sungguh, Kami benar-benar telah menciptakan manusia dalam bentuk yang sebaik-baiknya." (QS At-Tin: 4)

Dalam tafsir kemenag RI (2024), Allah menegaskan bahwa manusia diciptakan dalam kondisi fisik dan psikis terbaik. Dari segi fisik, manusia memiliki kemampuan berdiri tegak yang memungkinkan otak untuk berpikir, menghasilkan ilmu, dan menggerakkan tangan untuk menerapkan ilmu tersebut, yang pada akhirnya melahirkan teknologi. Bentuk tubuh manusia adalah yang paling indah di antara semua makhluk. Dari segi psikis, manusia memiliki pikiran dan perasaan yang sempurna, serta satu-satunya makhluk yang beragama. Banyak keistimewaan lainnya dari segi fisik dan psikis yang tidak dapat dijelaskan secara rinci.

Penegasan bahwa manusia diciptakan dengan kondisi fisik dan psikis terbaik juga mengandung makna bahwa keduanya perlu dijaga dan dikembangkan. Fisik dipelihara melalui pemberian gizi yang cukup dan menjaga kesehatan, sedangkan psikis dipelihara dengan pendidikan dan agama yang baik. Jika kedua aspek tersebut dijaga dan dikembangkan dengan baik, manusia akan mampu memberikan manfaat besar bagi alam dan dengan demikian menjadi makhluk yang paling mulia.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini termasuk dalam jenis penelitian kuantitatif komparatif yang bertujuan untuk mengklasifikasikan sekuens DNA yang terkait dengan penyakit diabetes melitus menggunakan algoritma *k-nearest neighbor* (KNN). Melalui pendekatan kuantitatif, penelitian ini menekankan pada pengukuran numerik seperti *accuracy* dan performa algoritma dalam melakukan klasifikasi data DNA. Komparatif dari penelitian ini terdapat pada untuk membandingkan hasil klasifikasi yang dihasilkan oleh model KNN berdasarkan variasi jumlah *nearest neighbor* (K) dan metrik jarak minkowski, manhattan, euclidean, chebyshev yang digunakan dalam KNN. Penelitian ini dilakukan untuk menentukan nilai K dan jarak yang paling optimal dalam memisahkan sekuens DNA yang menunjukkan indikasi Diabetes Melitus menggunakan bahasa pemrograman *python*.

3.2 Data dan Sumber Data

Data yang digunakan dalam penelitian ini berupa sekuens DNA yang terkait dengan penyakit diabetes melitus. Data ini diambil dari National Center for Biotechnology Information (NCBI), yang merupakan sumber data publik internasional yang menyediakan beragam informasi biologis tentang sekuens DNA, dan penelitian terkait lainnya. Sekuens DNA dalam penelitian ini berasal dari gen insulin yaitu, protein yang diproduksi oleh tubuh manusia untuk mengatur kadar glukosa yang menyebabkan diabetes melitus. Sekuens DNA ini mencakup

informasi genetik dari individu yang mengidap Diabetes Melitus. Data ini berisi DMT1 (Diabetes Melitus tipe 1) berjumlah 145, DMT2 (Diabetes Melitus tipe 2) berjumlah 443, dan NONDM (Diabetes Gestasional) berjumlah 989. sekuens DNA terdiri dari empat jenis nukleotida yang berpasangan dan dapat mencapai panjang hingga 9000 nukleotida. Data yang diambil dari NCBI dengan format fasta kemudian diupload di Kaggle dengan format csv. Link sumber data terdapat pada lampiran 1.

3.3 Teknik Analisis Data

3.3.1 *Preprocessing* Data

Tahap pertama dalam teknik analisis data adalah *preprocessing* data, yaitu proses mempersiapkan data sekuens DNA yang telah diambil dari NCBI agar siap digunakan dalam algoritma KNN. *Preprocessing* dilakukan untuk memastikan bahwa data dalam kondisi bersih, lengkap, dan sesuai dengan format input yang dibutuhkan untuk analisis lebih lanjut. Langkah-langkah *preprocessing* meliputi:

1. Cleaning Data: Cleaning data adalah langkah awal dalam proses preprocessing yang bertujuan untuk memastikan data yang digunakan dalam analisis bebas dari kesalahan, ketidaksesuaian, atau elemen yang tidak relevan. Dalam konteks sekuens DNA, cleaning data melibatkan pemeriksaan dan penghapusan elemen-elemen yang dapat mengganggu *accuracy* hasil analisis seperti selain elemen A, C, G, T.
2. K-mers: Metode K-mers mengubah sekuens DNA menjadi urutan fragmen kecil yang disebut "k-mers." Setiap fragmen berisi sejumlah nukleotida sepanjang nilai k yang telah ditentukan seperti pada gambar (2.2). Teknik ini

membantu menangkap pola spesifik dalam sekuens DNA, yang dapat menjadi indikator penting dalam klasifikasi. Dengan membagi sekuens menjadi k-mers, model dapat mengenali motif atau pola tertentu yang relevan dengan kondisi Diabetes Melitus.

3. *Count vectorizer*: *Count vectorizer* digunakan untuk menghitung frekuensi kemunculan setiap k-mers di dalam sekuens DNA, dan mengonversi frekuensi ini ke dalam bentuk matriks seperti pada gambar (2.3). Matriks hasil *Count vectorizer* ini mengindikasikan berapa kali pola k-mers tertentu muncul dalam sampel DNA, yang kemudian digunakan sebagai fitur dalam model klasifikasi.
4. *Oversampling:* Metode *oversampling* yang digunakan dalam penelitian ini adalah Synthetic Minority Over-sampling Technique (SMOTE) supaya tidak ada duplikasi data. SMOTE Oversampling memilih sampel dari kelas minoritas secara acak, lalu menggunakan algoritma *k-nearest neighbor* (KNN) untuk menemukan sejumlah tetangga terdekat dari sampel tersebut. Sampel baru dibuat dengan interpolasi, yaitu menghitung titik di antara sampel asli dan tetangganya menggunakan bilangan acak seperti pada gambar (2.4) dengan persamaan (2.6).
5. *Min-max scaler*: *Min-max scaler* adalah teknik normalisasi yang mengubah skala data sehingga seluruh fitur memiliki rentang nilai yang sama, yaitu antara 0 dan 1. Dengan *Min-max scaler*, setiap nilai dalam data berada dalam rentang yang seragam, sehingga algoritma K-Nearest Neighbour dapat melakukan perhitungan jarak antar sampel dengan lebih akurat. Persamaan *min-max scaler* terdapat pada persamaan (2.7).

6. *Label encoder*: *Label encoder* adalah teknik yang digunakan untuk mengonversi label kelas dari format kategori menjadi format numerik seperti pada gambar (2.5). Dalam penelitian ini, kelas yang menggambarkan kondisi kelas DMT1, DMT2, dan NONDM dikonversi ke dalam angka 0, 1, dan 2 agar dapat dikenali oleh model K-Nearest Neighbour yang memerlukan input numerik.

3.3.2 Klasifiakasi Model KNN

Proses kerja KNN terdiri dari beberapa langkah yang saling berkaitan. Setiap langkah memiliki peran penting dalam memastikan hasil klasifikasi atau regresi yang akurat. Berikut adalah urutan dan penjelasan dari setiap langkah dalam cara kerja KNN:

1. Pengukuran Jarak: Pengukuran jarak adalah menghitung jarak antara data yang akan diklasifikasikan dengan setiap data yang sudah ada dalam dataset (data pelatihan). Pengukuran jarak ini bisa menggunakan beberapa metode seperti *Minkowski Distance, Euclidean Distance, Manhattan Distance dan Chebyshev distance* yang terdapat pada persamaan (2.2), (2.3), (2.4), (2.5). Dalam penelitian ini mencari penggunaan jarak terbaik untuk Model KNN menggunakan Bahasa Pemrograman *python.*
2. Pemilihan Jumlah *Nearest Neighbor* (K): Parameter K dalam algoritma KNN adalah jumlah *nearest neighbor* yang akan digunakan dalam menentukan kelas data. Nilai K ini sangat penting karena mempengaruhi hasil klasifikasi. Bahasa Pemrograman *python* digunakan untuk mencari nilai K terbaik.

3. Voting Mayoritas: Pada tahap ini, dilakukan voting atau pemilihan mayoritas kelas dari K tetangga terdekat yang telah diidentifikasi sebelumnya. Setiap tetangga terdekat memiliki label kelas masing-masing. Algoritma akan menghitung jumlah kemunculan masing-masing kelas di antara tetangga-tetangga tersebut. Kelas yang paling banyak muncul akan dianggap sebagai kelas prediksi untuk data baru. Jika niali K genap, hasil klasifikasi KNN bisa terjasdi seri, untuk mencegah jika data tersebut seri, maka dapat dilakukan *distance weight* KNN dengan persamaan di persamaan (2.1).
4. Pengambilan Keputusan Akhir: Berdasarkan hasil voting mayoritas, kelas yang paling banyak muncul di antara K tetangga terdekat akan ditetapkan sebagai hasil akhir dari klasifikasi data baru tersebut seperti pada gambar (2.1). Jika nilai K genap, maka nilai *distance weight* KNN terbesar ditetapkan hasil akhir klasifikasi. Dengan kata lain, data baru akan diberi label kelas yang sama dengan kelas mayoritas dari tetangga terdekat.

3.3.3 Evaluasi Model

Evaluasi dilakukan untuk mengetahui seberapa baik algoritma KNN dapat mengklasifikasikan sekuens DNA. metrik yang digunakan untuk mengevaluasi model adalah *Confusion matrix,* berikut merupakan menghitung *confusion matrix* dari tabel (2.1):

1. *Accuracy*: Mengukur persentase klasifikasi yang benar terhadap total data yang diuji. Persamaan *accuracy* terdapat pada persamaan (2.8).

2. *Precision*: Mengukur seberapa akurat model dalam memprediksi kategori positif, yaitu berapa banyak prediksi positif yang benar seperti pada persamaan (2.9).
3. *Recall* (Sensitivitas): Mengukur seberapa baik model dalam mendeteksi semua kasus positif yang ada. Persamaan *recall* terdapat pada persamaan (2.10).
4. *F1-score*: Merupakan rata-rata harmonis dari *precision* dan *recall*, memberikan penilaian yang lebih seimbang antara keduanya. Persamaan *f1-score* terdapat pada persamaan (2.11).

BAB IV

PEMBAHASAN

4.1 Preprocessing Data

4.1.1 Cleaning Data

Tahap preprocessing data dilakukan dengan merujuk pada data (Nuha, 2024). Data pada penelitian ini terdapat DMT1 (Diabetes Melitus tipe 1) berjumlah 145, DMT2 (Diabetes Melitus tipe 2) berjumlah 443, dan NONDM (Diabetes Melitus Gestasional) berjumlah 989. sekuens DNA terdiri dari empat jenis nukleotida (A, C, G, T) yang berpasangan dan mencapai panjang hingga 9000 nukleotida. Penelitian ini diawali dengan melakukan pengecekan data dengan menggunakan kode python berikut:

```
# Import Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from imblearn.over_sampling import RandomOverSampler, SMOTE
from sklearn.neighbor import KNeighborClassifier
from sklearn.metrics import classification_report, accuracy_score,
confusion_matrix
from sklearn.model_selection import train_test_split, GridSearchCV

# Muat dataset dari file CSV
data = pd.read_csv('Complete_DM_DNA_Sequence.csv')
data
# Memeriksa apakah ada karakter selain A, C, G, T dalam kolom DNA_sequence
invalid_dna = data[~data['sequence'].str.upper().str.contains('^[ACGT]+$')]

# Tampilkan baris-baris dengan nilai yang tidak valid
invalid_dna

# Jumlah Outlier Data
jumlah_invalid_dna = len(invalid_dna)
print(jumlah_invalid_dna)
```

Gambar 4.1 Script Code Cleaning Data

Berdasarkan gambar 4.1 hasil pengecekan diperoleh bahwa 1577 sekuens DNA telah bersih dari komponen selain A, C, G, T dan tidak ada duplikasi.

Sehingga penelitian ini dikerjakan menggunakan data tersebut yang dapat disajikan pada tabel berikut:

Tabel 4.1 Data Sekuens DNA Diabetes Melitus

No.	Sekuens DNA	Kelas	Panjang
1	AATAATTTGTGCACTTCAGAATATTCT...	DMT2	1010
2	AGTCCTTCGCCGTCCCTCGCCGTCCT…	DMT2	1010
3	AGTCCTTCGCCGTCCCTCGCCGTCCT…	DMT2	1013
4	AGTTGGAGTCTCCAGGGATCAGGGT…	DMT2	1013
…	…	…	…
160	CCCGCGCTCCTCCGGGCAGAGCGCG…	NONDM	1997
…	…	…	…
1574	AGTAGCAGAAAGTGAGGCTGGCAGG…	DMT2	8088
1575	GGCGGCCAGGGAAGGTCTCTGCCGC…	DMT1	8416
1576	GCTTAGTGCTGAGCACATCCAGTGGG…	DMT2	9032
1577	GGGCTTGGCCTCTGCCCGGCCACAG…	DMT2	9205

Berdasarkan tabel 4.1 dapat disimpulkan bahwa dataset DNA yang digunakan telah bebas dari karakter yang tidak valid dan hanya berisi nukleotida yang sah (A, C, G, T). Oleh karena itu, data dapat digunakan langsung untuk analisis lanjutan seperti klasifikasi *machine learning*. Pemeriksaan ini penting dilakukan untuk memastikan bahwa data yang digunakan akurat dan sesuai dengan standar yang akan meningkatkan kemampuan dan *accuracy* dari model *machine learning*. Dari data tersebut, maka akan dilakukan proses memotong sekuens DNA dengan pendekatan 3-mers. Tujuan menggunakan 3-mers dalam klasifikasi DNA adalah langkah pertama supaya sekuens DNA bisa diubah menjadi bentuk numerik.

4.1.2 3-Mers

K-mers adalah metode analisis urutan DNA yang memecah rangkaian nukleotida panjang menjadi segmen-segmen kecil dengan panjang *k*. Nilai *k* ini menentukan jumlah basa nukleotida dalam setiap segmen yang dianalisis. Pada penelitian ini dibandingkan beberapa nilai *k* untuk memecah sekuens DNA yang disajikan pada tabel berikut:

Tabel 4.2 Perbandingan *Accuracy* K-Mers

K-Mers	*Accuracy*
2-Mers	93,00%
3-Mers	93,26%
4-Mers	93,00%

Berdasarkan tabel 4.2 diperoleh *accuracy* tertinggi dengan *k=3*. Artinya pemotongan nukleotida dilakukan pemotongan dengan Panjang 3, yaitu pemecahan urutan DNA menjadi segmen tiga basa berturut-turut yang dikenal sebagai 3-mers. Dengan *accuracy* mencapai 93,26%, 3-mers menunjukkan performa yang sedikit lebih baik dibandingkan 2-mers (93%) dan 4-mers (93%). Implementasi *k=3* pada data tabel 4.1 mendapatkan hasil pada tabel 4.3 dengan script kode berikut.

```
# Buat fungsi untuk mengonversi sequence DNA menjadi K-mer
def get_kmers(sequence, size=3):
    return [sequence[i:i+size].upper() for i in range(len(sequence)-size+1)]

# Terapkan K-mer ke kolom 'sequence'
data['kmers'] = data['sequence'].apply(lambda x: get_kmers(x))
data['kmers']

# Gabungkan semua K-mers menjadi satu string per baris
data['kmers_joined'] = data['kmers'].apply(lambda x: ' '.join(x))
data['kmers_joined']
```

Gambar 4.2 Script Code K-mers

Berdasarkan kode pada gambar 4.2 akan mendapatkan hasil pada tabel berikut:

Tabel 4.3 Sekuens DNA K-Mers

No.	Sekuens DNA	K-Mers	Kelas
1	AATAATT…	AAT, ATA, TAA, AAT, ATT, ...	DMT2
2	AGTCCTT...	AGT, GTC, TCC, CCT, CTT, ...	DMT2
3	AGTCCTT...	AGT, GTC, TCC, CCT, CTT, ...	DMT2
4	AGTTGGA...	AGT, GTT, TTG, TGG, GGA, ...	DMT2
…	…	…	…
160.	CCCGCGC…	CCC, CCG, CGC, GCG, CGC, …	NONDM
…	…	…	…
1574	AGTAGCA…	AGT, GTA, TAG, AGC, GCA, …	DMT2
1575	GGCGGCC…	GGC, GCG, CGG, GGC, GCC, ...	DMT1
1576	GCTTAGT…	GCT, CTT, TTA, TAG, AGT, …	DMT2
1577	GGGCTTG…	GGG, GGC, GCT, CTT, TGG, …	DMT2

Pada tabel 4.3 terdapat proses 3-mers, sebagai contoh sekuens DNA nomor 1 “AATAATTT” dilakukan pemotongan 3-mers menjadi “AAT”, “ATA”, “TAA”, “AAT”, “ATT”, dan seterusnya. Setelah proses 3-mers, CountVectorizer digunakan untuk menghitung frekuensi kemunculan setiap 3-mer, sehingga menghasilkan representasi numerik dalam bentuk vektor fitur yang dapat digunakan dalam algoritma klasifikasi.

4.1.3 *Count vectorizer*

Langkah selanjutnya dalam analisis adalah merepresentasikan fiftur tersebut dalam bentuk numerik yang dapat diproses oleh algoritma. *Count vectorizer* adalah teknik yang memungkinkan penghitungan frekuensi setiap segmen 3-mers dan dalam urutan DNA yang dianalisis. *Count vectorizer* bekerja dengan cara membuat representasi jumlah dari kumpulan segmen DNA yang telah dipecah menjadi 3-mers. Proses ini diawali dengan membangun semua kemungkinan

kombinasi 3-mers sebanyak 64 kombinasi, seperti AAA, AAC, AAG. Berikut merupakan tabel DNA setelah *count vectorizer* yang diklakukan denga script kode berikut:

```
# Gunakan Count vectorizer untuk Menghitung K-mers
vectorizer = CountVectorizer(ngram_range=(1, 1))
x_kmers = vectorizer.fit_transform(data['kmers_joined'])

# Tampilkan DNA Setelah Count Vectorizer
x_kmers_df = pd.DataFrame(x_kmers.toarray(),
columns=vectorizer.get_feature_names_out())
x_kmers_df
```

Gambar 4.3 Script Code K-Mers

Berdasarkan kode pada gambar 4.3 menghasilkan data seperti pada tabel berikut:

Tabel 4.4 Sekuens DNA *Count vectorizer*

No.	Sekuens DNA								Kelas
	AAA	AAC	AAG	AAT	ACA	ACC	…	TTT	
1	40	19	27	28	24	5	…	30	DMT2
2	2	2	13	11	9	21	…	16	DMT2
3	6	3	17	10	11	14	…	18	DMT2
4	15	13	23	15	25	12	…	17	DMT2
…	…	…	…	…	…	…	…	…	…
160	26	25	33	20	34	36	…	50	NONDM
…	…	…	…	…	…	…	…	…	…
1574	256	89	179	112	145	110	…	183	DMT2
1575	45	44	88	19	112	40	…	35	DMT1
1576	385	134	235	225	201	20	…	218	DMT2
1577	365	136	209	251	115	10	…	346	DMT2

Data pada tabel 4.4 didapatkan matriks sebagai berikut.

$$x_{m,n} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \ldots & x_{m,n} \end{bmatrix}$$

Dengan:

$$m = 1577\ adalah\ batas\ akhir\ baris$$

$$n = 64\ adalah\ batas\ akhir\ kolom$$

Berdasarkan tabel 4.4 dapat diketahui bahwa sekuens DNA nomor 1 terdapat jumlah potongan DNA "AAA" berjumlah 40, potongan DNA "AAC" berjumlah 19, potongan DNA "AAG" berjumlah 27, dan seterusnya. Penggunaan *Count vectorizer* setelah 3-mers juga membantu mengurangi kompleksitas data dengan merepresentasikan informasi dalam format numerik yang lebih sederhana. Setelah diperoleh representasi numerik dari CountVectorizer, SMOTE digunakan untuk mengatasi ketidakseimbangan jumlah data antar kelas dengan cara membuat sampel sintetis, sehingga model klasifikasi dapat belajar dari data yang lebih seimbang.

4.1.4 *SMOTE Oversampling*

Data yang tidak seimbang sering menjadi tantangan ketika salah satu kelas memiliki jumlah data yang jauh lebih kecil dibandingkan kelas lainnya. Hal ini dapat menyebabkan model *machine learning* bias terhadap kelas mayoritas dan kurang akurat dalam mengenali kelas minoritas. Salah satu teknik yang efektif untuk mengatasi masalah ini adalah *Synthetic Minority Oversampling Technique* (SMOTE) supaya tidak ada duplikasi data.

Tabel 4.5 Kelas Setelah Oversampling

Kelas	Sebelum *Oversampling*	Setelah *Oversampling*
NONDM	989 Sekuens DNA	989 Sekuens DNA
DMT1	443 Sekuens DNA	989 Sekuens DNA
DMT2	145 Sekuens DNA	989 Sekuens DNA

Pada tabel 4.5 Sebelum penerapan *oversampling*, data penelitian menunjukkan distribusi kelas yang sangat tidak seimbang menyebabkan model *machine learning* cenderung bias terhadap kelas mayoritas (NONDM), sehingga *accuracy* prediksi untuk kelas minoritas (DMT1 dan DMT2) menjadi rendah. SMOTE Oversampling memilih sampel dari kelas minoritas secara acak menggunakan algoritma *k-nearest neighbor* (KNN) untuk menemukan sejumlah tetangga terdekat dari sampel tersebut. Sampel baru dibuat dengan interpolasi, yaitu menghitung titik di antara sampel asli dan tetangganya menggunakan bilangan acak. Proses ini diulangi hingga jumlah sampel sintetis yang dibutuhkan tercapai, kemudian sampel sintetis digabungkan ke dataset asli. Kode python dan hasilnya data SMOTE akan disajikan sebagai berikut:

```
# Pisahkan fitur dan label (kolom 'class')
y = data['class']
y.value_counts()

# Terapkan SMOTE untuk Oversampling
ros = SMOTE(random_state=42)
x_resampled, y_smote = ros.fit_resample(x_kmers_df, y)
y_smote.value_counts()

# Tampilkan Data setelah SMOTE
x_resampled_df = pd.DataFrame(x_resampled)
x_resampled_df

# Gabungkan data dengan kelas
x_resampled_with_class = pd.concat([pd.DataFrame(x_resampled_df), y_smote],
axis=1)

# Tampilkan Data
x_resampled_with_class
```

Gambar 4.4 Script Code SMOTE Oversampling

Berdasarkan script code pada gambar 4.4, akan disajikan data setelah SMOTE *oversampling* pada tabel berikut:

Tabel 4.6 DNA Count Vectorizer setelah Oversampling

No.	Sekuens DNA								Kelas
	AAA	AAC	AAG	AAT	ACA	ACC	…	TTT	
1	40	19	27	28	24	5	…	30	DMT2
2	2	2	13	11	9	21	…	16	DMT2
3	6	3	17	10	11	14	…	18	DMT2
4	15	13	23	15	25	12	…	17	DMT2
…	…	…	…	…	…	…	…	…	…
160	26	25	33	20	34	36	…	50	NONDM
…	…	…	…	…	…	…	…	…	…
1575	45	44	88	19	112	40	…	35	DMT1
1576	385	134	235	225	201	20	…	218	DMT2
…	…	…	…	…	…	…	…	…	…
2967	89	39	56	45	40	24	…	51	DMT2

Berdasarkan data pada tabel 4.6, didapatkan matrix sebagai berikut.

$$x_{m\times n} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}$$

Dengan:

$$m = 2967 \; adalah \; batas \; akhir \; baris$$

$$n = 64 \; adalah \; batas \; akhir \; kolom$$

Data yang sebelumnya berjumlah 1577, setelah dilakukan SMOTE *oversampling* data menjadi 2967 dengan menyeimbangkan jumlah NONDM, DMT1, DMT2. Setelah proses oversampling dengan SMOTE, Min-Max Scaler digunakan untuk menormalkan nilai fitur ke dalam rentang 0 sampai 1, agar setiap fitur memiliki skala yang sama.

4.1.5 *Min-max scaler*

Penggunaan *Min-max scaler* pada data DNA bertujuan untuk menstandarisasi skala fitur sehingga semua nilai berada dalam rentang yang seragam antara 0 dan 1. Normalisasi ini memastikan bahwa setiap fitur memiliki bobot yang setara dalam analisis, sehingga tidak ada fitur tertentu yang mendominasi karena memiliki skala yang lebih besar. *Min-max scaler* berfungsi dalam mencegah bias akibat perbedaan skala antar fitur serta mempercepat proses pelatihan model. Langkah ini membantu memastikan bahwa model dapat belajar secara adil tanpa terpengaruh oleh dominasi nilai tertentu.

Berdasarkan data pada tabel 4.6 akan dilakukan *min-max scaler*, pada persamaan tersebut didefinisikan $x'_{i,j}$ adalah nilai fitur pada baris ke i kolom ke j yang telah dinormalisasi. Berikut merupakan contoh perhitungan kolom AAA baris 1 berdasarkan persamaan (2.7):

$$x'_{i,j} = \frac{x_{i,j} - \min(\vec{x}_i)}{\max(\vec{x}_i) - \min(\vec{x}_i)}$$

$$i = 1, 2, \dots, m$$

$$j = 1, 2, \dots, n$$

$$x'_{i,j} = \frac{40 - 2}{385 - 2} = 0{,}099217$$

Semua data dilakukan *min-max scaler* dengan kode python berikut:

```
# Normalisasi data dengan Min-max scaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x_resampled_df)
x_scaled
# Menampilkan DNA Min-max scaler dengan Kelas
x_scaled_df = pd.DataFrame(x_scaled,
columns=x_resampled_with_class.columns[:-1])
x_scaled_with_class = pd.concat([x_scaled_df,
x_resampled_with_class['class']], axis=1)
x_scaled_with_class
```

Gambar 4.5 Script Code *Min-max scaler*

Berdasarkan kode pada gambar 4.5, didapatkan semua hasil data setelah dilakukan *min-max scaler* yang akan ditampilkan pada tabel berikut:

Tabel 4.7 Tabel setelah Normalisasi Data

No.	Sekuens DNA					Kelas
	AAA	AAC	AAG	…	TTT	
1	0,099217	0,126866	0,083700	…	0,073170	DMT2
2	0,000000	0,000000	0,022026	…	0,035230	DMT2
3	0,010444	0,007463	0,039648	…	0,025510	DMT2
4	0,033943	0,082090	0,066079	…	0,051020	DMT2
…	…	…	…	…	…	…
160	0,060052	0,156716	0,110132	…	0,127371	NONDM
…	…	…	…	…	…	…
2964	0,151436	0,380597	0,290749	…	0,149051	DMT2
2965	0,075718	0,373134	0,312775	…	0,089430	DMT1
2966	0,002611	0,000000	0,026432	…	0,035230	DMT2
2967	0,227154	0,276119	0,211454	…	0,130081	DMT2

Pada tabel 4.7 terdapat nilai 0,099217 pada kolom AAA baris 1, mengindikasikan bahwa proses normalisasi atau pembobotan telah diterapkan pada data DNA Count Vectorizer setelah oversampling. Sebelumnya, nilai yang ditampilkan adalah 40. Setelah dilakukan *min-max scaler*, nilai-nilai tersebut telah dinormalisasi menjadi nilai desimal yang lebih kecil, yang memungkinkan model untuk bekerja lebih baik. Setelah data fitur dinormalisasi dengan Min-Max Scaler, Label Encoder digunakan untuk mengubah label kelas dari format teks menjadi format numerik agar dapat diproses oleh algoritma klasifikasi.

4.1.6 Label encoder

Dalam pengolahan data untuk *machine learning*, representasi numerik dari data kategori seperti label kelas sangat penting agar dapat diproses oleh algoritma. metode yang digunakan dalam penelitian ini adalah *label encoder*, yang mengubah setiap kategori menjadi angka tanpa memberikan bobot antar kelas. Proses *label encoder* akan dilakukan dengan kode berikut:

```
Ubah Kelas menjadi Numerik dengan Label encoder
encoder = LabelEncoder()
y_encoded = encoder.fit_transform(y_smote)

# 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 4.6 Script Code Label Encoder

Berdasarkan gambar 4.6 akan didapatkan hasil pada tabel berikut:

Tabel 4.8 Label Encoder

Kelas	*Label encoder*
DMT 1	0
DMT 2	1
NONDM	2

Pada tabel 4.8 tiga kelas yang di-encode adalah DMT 1, DMT 2, dan NONDM, dengan hasil pengkodean sebagai berikut: DMT 1 menjadi 0, DMT 2 menjadi 1, dan NONDM menjadi 2. Kelas tersebut diurutkan berdasarkan abjad. Representasi ini membuat data dapat digunakan pada model KNN. Setelah fitur dan label dikonversi ke bentuk numerik, data tersebut kemudian dimasukkan ke dalam model K-Nearest Neighbor (KNN) untuk melakukan proses pelatihan dan klasifikasi berdasarkan kedekatan jarak antar data.

4.2 Klasifikasi Model KNN

4.2.1 Deskripsi Model KNN pada Sampel Data

Algoritma K-Nearest Neighbor (KNN) untuk mengklasifikasikan data berdasarkan kedekatannya dengan titik data lain. Perhitungan matematis yang digunakan dalam KNN berfokus pada pengukuran jarak dengan menggunakan persamaan jarak tertentu. Pada penelitian ini didapatkan bahwa jarak minkowski dengan $p = 3$ mendapatkan *accuracy* terbaik. Oleh karena itu, pada perhitungan matematis dengan sampel data ini akan menggunakan *Minkowski Distance* untuk menentukan kelas mana yang paling mungkin untuk data tersebut. Disini akan dijelaskan langkah-langkah matematis yang diperlukan untuk menghitung jarak antar data, serta bagaimana keputusan klasifikasi dibuat berdasarkan tetangga terdekat dalam hasil yang telah dibentuk.

Agar tidak terjadi perhitungan yang terlalu banyak, maka perhitungan matematika model KNN terdapat sampel data sebagai berikut:

1. Diambil 2 sekuens DNA setelah *count vectorizer* setiap kelas.
2. Diambil 8 potongan DNA k-mers.

Tabel 4.9 Count Vectorizer dengan Sampel Data

No.	Sekuens DNA								Kelas
	AAA	AAC	AAG	AAT	ACA	ACC	ACG	ACT	
1	6	3	17	10	11	14	6	9	DMT2
2	40	19	27	28	24	5	4	14	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	NONDM
6	31	22	31	18	26	29	11	21	NONDM
7	2	2	13	11	9	21	6	9	DMT2

Berdasarkan pada tabel 4.9, kelas DMT2 berjumlah 3, kelas DMT1 berjumlah 2, kelas NONDM berjumlah 2. Dikarenakan data tersebut tidak seimbang, maka dilakukan SMOTE oversampling supaya data tersebut seimbang dan penjadi sama 3 kelas

$$\vec{z} = \overrightarrow{x_0} + w(\overrightarrow{x_1} - \overrightarrow{x_0})$$

Berdasarkan persamaan, simbol w pada smote adalah nilai acak pada rentang 0 sampai 1. nilai pada w tersebut mempunyai nilai acak untuk menambah keragaman data dan menghindari duplikasi data. Berikut contoh perhitungan dalam sampel data DMT1.

$$\overrightarrow{x_0} = [44, 21, 25, 19, 26, 17, 5, 17]$$

$$\overrightarrow{x_1} = [30, 13, 24, 18, 19, 21, 4, 13]$$

$$z_1 = 44 + 0{,}5 \times (30 - 44) = 44 - 14 = 37$$

$$z_2 = 21 + 0{,}5 \times (13 - 21) = 21 - 8 = 17$$

$$z_3 = 25 + 0{,}99 \times (24 - 25) = 25 - 1 = 24$$

$$z_4 = 19 + 0{,}99 \times (18 - 19) = 19 - 1 = 18$$

$$z_5 = 26 - 0{,}99 \times (19 - 26) = 26 - 7 = 19$$

$$z_6 = 17 + 0{,}5 \times (21 - 17) = 17 + 2 = 19$$

$$z_7 = 5 + 0{,}99 \times (4 - 5) = 5 - 1 = 4$$

$$z_8 = 17 + 0{,}5 \times (13 - 17) = 17 - 2 = 15$$

$$\vec{z} = [37, 17, 24, 18, 19, 29, 4, 15]$$

Data NONDM juga dihitung dengan cara yang sama dengan DMT1. Data setelah SMOTE oversampling akan mendapatkan data sebagai berikut. Dan akan ditambahkan dara baru yang akan di klasifikasi dan akan menghasilkan tabel sebagai berikut:

Tabel 4.10 Sampel Data Setelah SMOTE

No.	Sekuens DNA								Kelas
	AAA	AAC	AAG	AAT	ACA	ACC	ACG	ACT	
1	6	3	17	10	11	14	6	9	DMT2
2	40	19	27	28	24	5	4	14	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	NONDM
6	31	22	31	18	26	29	11	21	NONDM
7	2	2	13	11	9	21	6	9	DMT2
8	37	17	24	18	19	29	4	15	DMT1
9	29	22	24	18	28	26	15	13	NONDM

Berdasarkan sampel data pada tabel 4.10, Data baris pertama akan dilakukan klasifikasi dan didapatkan matrix sebagai berikut.

$$x_{m\times n} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}$$

Dengan:

$$m = 9 \; adalah \; batas \; akhir \; baris$$

$$n = 8 \; adalah \; batas \; akhir \; kolom$$

$$i = 1, 2, \ldots, m$$

$$j = 1, 2, \ldots, n$$

Pada data tersebut akan dilakukal *min-max scaler* untuk mengubah nilai fitur didalam rentang 0 sampai 1 dengan persamaan. Berdasarkan persamaan (2.7) didefinisikan $x'_{i,j}$ adalah nilai fitur pada baris ke i kolom ke j yang telah dinormalisasi.

$$x'_{i,j} = \frac{x_{i,j} - \min(\vec{x}_i)}{\max(\vec{x}_i) - \min(\vec{x}_i)}$$

$$x'_{i,j} = \frac{6-2}{44-2} = 0{,}09$$

Setelah data DNA dinormalisasi menggunakan *Min-max scaler*, langkah berikutnya adalah menghitung jarak antar titik data menggunakan Minkowski Distance dengan parameter $p = 3$. *Min-max scaler* memastikan bahwa semua fitur memiliki skala yang seragam, sehingga jarak antar data dapat dihitung dengan lebih akurat tanpa adanya fitur yang mendominasi karena perbedaan skala. Hasil *min-max scaler* terdapat pada tabel berikut:

Tabel 4.11 Sampel Data *Min-max scaler*

No.	Sekuens DNA								Kelas
	AAA	AAC	AAG	AAT	ACA	ACC	ACG	ACT	
1	0,09	0,04	0,20	0	0,08	0,29	0,13	0	DMT2
2	0,90	0,80	0,70	1	0,60	0	0	0,22	DMT2
3	1	0,90	0,60	0,50	0,68	0,38	0,06	0,36	DMT1
4	0,66	0,52	0,55	0,44	0,40	0,51	0	0,18	DMT1
5	0,54	1	1	0,55	1	1	1	1	NONDM
6	0,69	0,95	0,90	0,44	0,68	0,77	0,46	0,54	NONDM
7	0	0	0	0,05	0	0,51	0,13	0	DMT2
8	0,83	0,71	0,55	0,44	0,40	0,77	0	0,27	DMT1
9	0,64	0,95	0,55	0,44	0,76	0,67	0,76	0,18	NONDM

Berdasarkan tabel 4.11 akan dihitung jarak menggunakan jarak minkowski dengan $p = 3$ berdasarkan persamaan (2.2). Berikut merukpakan contoh data baris 1 terhadap baris ke 2:

$$D = \left(\sum_{i=1}^{n} |\vec{x}_i - \vec{y}_i|^3 \right)^{\frac{1}{3}}$$

$$D_{1,2} = (|0{,}09 - 0{,}90|^3 + |0{,}04 - 0{,}8|^3 + |0{,}20 - 0{,}7|^3 |0 - 1|^3 + |0{,}08 - 0{,}6|^3 + |0{,}29 - 0|^3 + |0{,}13 - 0|^3 + |0 - 0{,}22|^3)^{\frac{1}{3}} = 1{,}31$$

Data pada baris 1 dilakukan perhitungan yang sama dengan data pada baris ke-3 dan seterusnya. Sehingga mendapatkan hasil pada tabel berikut.

Tabel 4.12 Hasil Jarak Sampel Sekuens DNA

Data	Jarak	Kelas
2	1,31	DMT2
3	1,22	DMT1
4	0,78	DMT1
5	1,64	NONDM
6	1,24	NONDM
7	0,27	DMT2
8	0,99	DMT1
9	1,17	NONDM

Pada tabel 4.12 setelah menghitung jarak antar sampel data menggunakan Minkowski Distance, dapat dilihat bahwa jarak antara data sampel berbeda-beda, dengan kelas yang terdeteksi sesuai dengan kedekatannya dengan data lainnya. Berdasarkan tabel 4.14, K-Nearest Neighbor (KNN) dengan nilai $k = 2$ memberikan *accuracy* terbaik, sampel data ini juga akan menggunakan $k = 2$. Namun, karena penggunaan $k = 2$ dapat menyebabkan terjadinya seri dalam prediksi kelas, langkah selanjutnya adalah mengimplementasikan Distance Weighted KNN.

Dalam sampel data diatas dengan $k = 2$, didapatkan $W_{1,4}$ dan $W_{1,7}$ mempunyai jarak terdekan, jika menggunakan Uniform Weight KNN terjadi seri

antara dua kelas terdekat, maka Distance Weight KNN akan mempertimbangkan jarak antar sampel, di mana kelas dengan jarak lebih dekat akan memiliki bobot lebih besar. Dalam hal ini, Distance Weight KNN digunakan untuk menentukan bobot, sehingga membantu menghindari seri dan menghasilkan keputusan yang lebih akurat dalam klasifikasi kelas berdasarkan persamaan (2.1).

$$W = \frac{1}{D}$$

$$W_{1,4} = \frac{1}{0,78} = 1,28$$

$$W_{1,7} = \frac{1}{0,27} = 3,7$$

Karena nilai $W_{1,7}$ lebih besar dibandingkan $W_{1,4}$ artinya tetangga dengan jarak 3,7 ($W_{1,7}$) memiliki bobot yang lebih besar dalam penentuan kelas dibandingkan tetangga dengan jarak 1,28 ($W_{1,4}$). Oleh karena itu, berdasarkan bobot yang lebih besar ini, sekuens DNA pada sampel data baris 1 diklasifikasikan sebagai Diabetes Melitus Tipe 2 (DMT2).

4.2.2 Hasil Klasifikasi Model KNN

Hasil *accuracy* yang diperoleh dari penerapan model K-Nearest Neighbor (KNN) dalam mengklasifikasikan sekuens DNA yang berkaitan dengan Diabetes Melitus menunjukkan efektivitas model ini dalam memprediksi jenis diabetes berdasarkan kedekatannya dengan data latih. Untuk eksperimen ini, sekuens DNA yang telah melalui proses 3-mers, *count vectorizer*, dan min-max scaling test 10% untuk melatih model KNN. Pada penelitian ini digunakan jarak minkowski dengan $p = 3$, nilai 2-nearest neighbor dengan distance weight. Hasil *accuracy*

yang diperoleh akan dianalisis untuk menilai performa model dalam konteks klasifikasi penyakit ini. Script code *accuracy* terdapat pada gambar berikut:

```
# Split data
x_train, x_test, y_train, y_test = train_test_split(
    x_scaled,
    y_encoded,
    test_size=0.1,
    random_state=42,
    stratify=y_encoded)

# Split data
x_train, x_test, y_train, y_test = train_test_split(
    x_scaled,
    y_encoded,
    test_size=0.1,
    random_state=42,
    stratify=y_encoded)

# Tentukan parameter grid untuk mencari nilai yang optimal
param_grid = {
    'n_neighbor': [2],  # Rentang nilai K
    'weights': ['distance'],  # Bobot distance
    'metric' : ['minkowski'], # Metric
    'p': [3] # P
}
# Inisialisasi model KNN
knn = KNeighborClassifier()

# Gunakan GridSearchCV untuk mencari nilai terbaik
grid_search = GridSearchCV(knn, param_grid, scoring='accuracy')

# Melakukan fit pada data training
grid_search.fit(x_train, y_train)

# Prediksi pada test set menggunakan model terbaik
best_knn = grid_search.best_estimator_
x_pred = best_knn.predict(x_test)
#Train
x_pred_tr = best_knn.predict(x_train)
```

Gambar 4.7 Script Code Split Data

Berdasarkan kode pada gambar 4.7 didapatkan hasil pada tabel berikut:

Tabel 4.13 Perbandingan Data Test

Data Test	*Accuracy*
10%	93,26%
20%	91,00%
30%	91,00%

Berdasarkan hasil pada tabel 4.13 diperoleh dari pengujian dengan berbagai proporsi data test, dapat disimpulkan bahwa data test 10% memberikan *accuracy* tertinggi, yaitu 93,26%. Sementara itu, pengujian dengan data test 20% dan 30%

menghasilkan *accuracy* masing-masing sebesar 91%. Meskipun perbedaan *accuracy* tidak terlalu besar, data test 10% menunjukkan performa terbaik dalam klasifikasi, yang menunjukkan bahwa ukuran data test yang lebih kecil dapat memberikan hasil yang lebih akurat dalam model ini.

Tabel 4.14 Perbandingan Nilai K-Nearest Neighbor

K-Nearest Neighbor	*Accuracy*
1-Nearest Neighbor	91,00%
2-Nearest Neighbor	93,26%
3-Nearest Neighbor	92,00%

Berdasarkan hasil pada tabel 4.14 diperoleh dari pengujian dengan berbagai nilai K pada K-Nearest Neighbor (KNN), dapat disimpulkan bahwa 2-Nearest Neighbor memberikan *accuracy* tertinggi sebesar 93,26%. Sementara itu, 1-Nearest Neighbor menghasilkan *accuracy* 91% dan 3-Nearest Neighbor menghasilkan *accuracy* 92%. Dengan demikian, 2-Nearest Neighbor menunjukkan performa terbaik dalam model ini, menghasilkan *accuracy* yang lebih tinggi dibandingkan dengan K yang lain.

Tabel 4.15 Perbandingan Bobot

Weight	*Accuracy*
Uniform	88,00%
Distance	93,26%

Berdasarkan hasil pada tabel 4.15 diperoleh dari pengujian dengan dua jenis weight pada K-Nearest Neighbor (KNN), dapat disimpulkan bahwa Distance weight memberikan *accuracy* tertinggi sebesar 93,26%, sementara Uniform weight menghasilkan *accuracy* 88%. Dengan demikian, penggunaan Distance

weight memberikan performa yang lebih baik dalam model ini, yang menunjukkan bahwa mempertimbangkan jarak antara titik data memberikan kontribusi lebih besar terhadap *accuracy* klasifikasi dibandingkan dengan menggunakan bobot yang seragam.

Tabel 4.16 Perbandingan Jarak

Jarak	*Accuracy*
Manhattan ($p = 1$)	91,00%
Euclidean ($p = 2$)	92,00%
Minkowski ($p = 3$)	93,26%
Chebyshev ($p = \infty$)	92,00%

Berdasarkan tabel 4.16 menunjukkan bahwa algoritma K-Nearest Neighbor (KNN) dengan jarak Minkowski ($p = 3$) memberikan *accuracy* tertinggi sebesar 93,26% dalam klasifikasi sekuens DNA untuk Diabetes Mellitus (DM). *Accuracy* ini lebih tinggi dibandingkan metode jarak lainnya seperti Euclidean dan Chebyshev yang mencapai 92%, serta Manhattan dengan *accuracy* 91%.

4.2.3 Prediksi Sekuens DNA Baru dengan model KNN

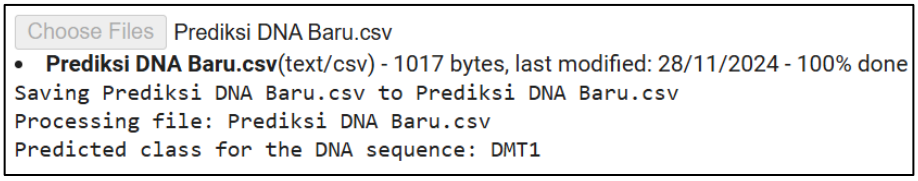
Prediksi sekuens DNA baru menggunakan model K-Nearest Neighbor (KNN) yang telah dilatih sebelumnya dilakukan dengan memanfaatkan teknik klasifikasi berbasis kedekatan antar data. Model ini mengidentifikasi kelas dari sekuens DNA yang belum pernah dilihat sebelumnya, berdasarkan kesamaan pola dengan data yang telah dilatih. Proses prediksi ini bertujuan untuk mengevaluasi kemampuan model dalam mengenali dan mengklasifikasikan sekuens DNA yang memiliki karakteristik serupa dengan data dalam kelas yang telah ditentukan,

yaitu Diabetes Melitus Tipe 1, Tipe 2, dan Gestasional. Berikut merupakan kode python untuk prediksi DNA baru:

```
# Prediksi DNA Baru dengan Model KNN
from google.colab import files
import re
import pandas as pd
def load_and_predict_dna(file, model, vectorizer, scaler, encoder, kmer_size=3):

    sequence = file.read().decode('utf-8').strip()  # Membaca file teks DNA sebagai
string

    # Cek apakah sequence hanya mengandung karakter A, C, G, T
    if not re.match("^[ACGT]+$", sequence.upper()):
        print("Invalid sequence: contains characters other than A, C, G, T")
        return

    # Buat K-mers dari sequence DNA
    kmers = get_kmers(sequence.upper(), size=kmer_size)
    kmers_joined = ' '.join(kmers)

    # Ubah ke bentuk numerik dengan CountVectorizer
    x_kmers_new = vectorizer.transform([kmers_joined])
    x_kmers_df_new = pd.DataFrame(x_kmers_new.toarray(),
columns=vectorizer.get_feature_names_out())

    # Normalisasi data
    x_scaled_new = scaler.transform(x_kmers_df_new)

    # Prediksi kelas
    y_pred_encoded = model.predict(x_scaled_new)
    y_pred = encoder.inverse_transform(y_pred_encoded)

    print(f"Predicted class for the DNA sequence: {y_pred[0]}")

# Unggah file DNA baru dan prediksi kelasnya
uploaded = files.upload()
for fn in uploaded.keys():
    print(f"Processing file: {fn}")
    with open(fn, 'rb') as file:
        load_and_predict_dna(file, best_knn, vectorizer, scaler, encoder)
```

Gambar 4.8 Script Code Prediksi DNA

Berdasarkan kode pada gambar 4.8, akan didapatkan hasil pada gambar berikut:



Gambar 4.9 Prediksi DNA

Berdasarkan gambar 4.9 model KNN menunjukkan kemampuannya dalam memprediksi sekuens DNA baru. Berdasarkan hasil *accuracy* 93.26%, dapat disimpulkan bahwa model *machine learning* ini mampu mengklasifikasikan data

baru yang mendekati sekuens DNA dalam kelas yang relevan secara tepat. Model ini efektif dalam mengenali pola dan memetakan sekuens DNA ke dalam kategori yang sesuai, baik itu untuk Diabetes Melitus Tipe 1, Tipe 2, atau NONDM, yang menunjukkan bahwa metode ini dapat diterapkan secara efisien untuk prediksi.

4.3 Evaluasi Model dengan Confusion Matrix

Evaluasi model klasifikasi sangat penting untuk mengukur sejauh mana performa model dalam melakukan prediksi yang tepat. Salah satu alat yang umum digunakan untuk menilai kinerja model adalah Confusion Matrix, yang memberikan gambaran jelas tentang bagaimana model mengklasifikasikan data ke dalam kelas yang benar atau salah. Pada bagian ini, evaluasi model dilakukan menggunakan Confusion Matrix untuk mengukur *accuracy* klasifikasi sekuens serta menghitung metrik evaluasi lainnya, seperti *precision*, *recall*, dan *F1-score*, untuk memberikan gambaran menyeluruh tentang kinerja model yang digunakan.

```
# Buat confusion matrix
conf_matrix = confusion_matrix(y_test, x_pred)

# Visualisasikan confusion matrix menggunakan heatmap
plt.figure(figsize=(3, 3))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',cbar=False,
xticklabels=best_knn.classes_, yticklabels=best_knn.classes_)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix Test')
plt.show()

# Buat confusion matrix
conf_matrix_tr = confusion_matrix(y_train, x_pred_tr)
# Visualisasikan confusion matrix menggunakan heatmap
plt.figure(figsize=(3, 3))
sns.heatmap(conf_matrix_tr, annot=True, fmt='d', cmap='Blues',cbar=False,
xticklabels=best_knn.classes_, yticklabels=best_knn.classes_)
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix Train')
plt.show()

# Evaluasi model dengan precision, recall, F1-score, dan support
print("Laporan Klasifikasi:\n", classification_report(y_test, x_pred))
```

Gambar 4.10 Confusion Matrix

Berdasarkan kode pada gambar 4.10, didapatkan hasil pada tabel berikut:

Tabel 4.17 Confusion Matrix

Data	Prediksi			
Aktual	Kelas	DMT1	DMT2	NONDM
	DMT1	93	6	0
	DMT2	8	90	1
	NONDM	2	3	94

Berdasarkan tabel 4.17 didapatkan interpretasi sebagai berikut:

1. Kelas DMT1 (Diabetes Melitus Tipe 2):
   a. 93 Data DMT1 diprediksi benar sebagai DMT1 (True Positive DMT1),
   b. 6 Data DMT1 diprediksi salah sebagai DMT2 (False Negative DMT1),
   c. 0 Data DMT1 diprediksi salah sebagai NONDM (False Negative DMT1),
2. Kelas DMT2 (Diabetes Melitus Tipe 1):
   a. 8 Data DMT2 diprediksi salah sebagai DMT1 (False Negative DMT2),
   b. 90 Data DMT2 diprediksi benar sebagai DMT2 (True Positive DMT2),
   c. 1 Data DMT2 diprediksi salah sebagai NONDM (False Negative DMT2),
3. Kelas NONDM (Diabetes Melitus Gestasional):
   a. 2 Data NONDM diprediksi salah sebagai DMT1 (False Negative NONDM),
   b. 3 Data NONDM diprediksi salah sebagai DMT2 (False Negative NONDM),
   c. 94 Data NONDM diprediksi benar sebagai NONDM (True Positive NONDM).

Dalam evaluasi ini, metrik yang digunakan meliputi *Precision*, *Recall*, dan *F1-score* untuk masing-masing kelas (DMT1, DMT2, dan NONDM), serta Accuracy untuk mengukur seberapa akurat model dalam mengklasifikasikan sekuens DNA. *Precision* mengukur seberapa tepat prediksi positif yang dilakukan

oleh model, *Recall* mengukur seberapa banyak prediksi positif yang benar-benar terdeteksi, dan *F1-score* memberikan gambaran keseimbangan antara *precision* dan *recall*. Selanjutnya, Accuracy memberikan informasi umum tentang seberapa banyak prediksi yang benar dibandingkan dengan total data yang diuji. Dengan menggunakan metrik ini, performa model dapat dievaluasi secara menyeluruh untuk setiap kelas yang ada. Berikut merupakan hasil evaluasinya:

1. Accuracy

$$Accuracy = \frac{Total\ True\ Positive}{Total\ Data}$$

$$= \frac{93 + 92 + 93}{93 + 6 + 0 + 92 + 7 + 0 + 93 + 2 + 4}$$

$$= 93\%$$

2. *Precision*

a. DMT1

$$Precision_{DMT1} = \frac{TP_{DMT1}}{TP_{DMT1} + FP_{DMT1}} = \frac{93}{93 + 8 + 2} = 90\%$$

b. DMT2

$$Precision_{DMT2} = \frac{TP_{DMT2}}{TP_{DMT2} + FP_{DMT2}} = \frac{90}{90 + 6 + 3} = 91\%$$

c. NONDM

$$Precision_{NONDM} = \frac{TP_{NONDM}}{TP_{NONDM} + FP_{NONDM}} = \frac{94}{94 + 1 + 0} = 99\%$$

3. *Recall*

a. DMT1

$$Recall_{DMT1} = \frac{TP_{DMT1}}{TP_{DMT1} + FN_{DMT1}} = \frac{93}{93 + 6 + 0} = 94\%$$

b. DMT2

$$Recall_{DMT2} = \frac{TP_{DMT2}}{TP_{DMT2} + FN_{DMT2}} = \frac{90}{90 + 8 + 1} = 91\%$$

c. NONDM

$$Recall_{NONDM} = \frac{TP_{NONDM}}{TP_{NONDM} + FN_{NONDM}} = \frac{94}{94 + 3 + 2} = 95\%$$

4. *F1-score*

a. DMT1

$$F1_{DMT1} = 2 \times \frac{Precision_{DMT1} \times Recall_{DMT1}}{Precision_{DMT1} + Recall_{DMT1}}$$

$$= \frac{2 \times 90\% \times 94\%}{90\% + 94\%} = 92\%$$

b. DMT1

$$F1_{DMT1} = 2 \times \frac{Precision_{DMT2} \times Recall_{DMT2}}{Precision_{DMT2} + Recall_{DMT2}}$$

$$= \frac{2 \times 91\% \times 91\%}{91\% + 91\%} = 91\%$$

c. NONDM

$$F1_{DMT1} = 2 \times \frac{Precision_{NONDM} \times Recall_{NONDM}}{Precision_{NONDM} + Recall_{NONDM}}$$

$$= \frac{2 \times 99\% \times 95\%}{99\% + 95\%} = 97\%$$

Berdasarkan perhitungan didapatkan tabel sebagai berikut:

Tabel 4.18 Hasil Klasifikasi

Kelas	*Precision*	*Recall*	*F1-score*
DMT1	90%	94%	92%
DMT2	91%	91%	91%
NONDM	99%	95%	97%
Accuracy	93,26%		

Berdasarkan hasil evaluasi model menggunakan confusion metrik pada tabel 4.18, berikut merupakan interpretasi hasil *Accuracy*, *Precision*, *Recall*, dan *F1-score*:

1. *Accuracy* mengukur seberapa sering model mengklasifikasikan data dengan benar. Dengan nilai 93,26%, model KNN menunjukkan bahwa sebagian besar prediksi yang dilakukan adalah benar. Hasil tersebut menunjukkan performa model yang baik.
2. *Precision* pada DMT1 sebesar 90% berarti bahwa dari semua prediksi yang menyebutkan DMT1, 90% di antaranya benar-benar merupakan DMT1. Pada NONDM, *precision* lebih tinggi yakni 99%, menunjukkan bahwa model sangat tepat dalam mengidentifikasi sekuens DNA diabetes gestasional.
3. *Recall* mengukur proporsi kasus positif yang berhasil dideteksi oleh model. Untuk DMT1, *recall* sebesar 94% yang berarti bahwa 94% dari semua sekuens DNA yang benar-benar termasuk dalam kategori DMT1 berhasil diprediksi dengan benar oleh model.
4. *F1-score* yang tinggi menandakan bahwa model berhasil mencapai keseimbangan yang baik antara *Recall* dan *Precision*. NONDM memiliki *F1-score* tertinggi, yaitu 97%, menunjukkan bahwa model sangat efektif dalam mengklasifikasikan sekuens DNA diabetes gestasional dengan sedikit kesalahan.

Tabel 4.19 Perbandingan dengan Penelitian Terdahulu

Metode	Data	Accuracy	Precision	Recall	F1-Score	Waktu Komputasi
K-Nearest Neighbor	Sekuens DNA diabetes melitus tipe 1, tipe 2, dan gestasional	93,26%	93,26%	93,26%	93,26%	2 Menit
Decision Tree (Najar dkk., 2022)	Sekuens DNA jenis kanker darah (leukimia)	83,30				
Random Forest (Zou dkk., 2018)	Diabetes melitus dengan indikator kadar *fasting* glukosa	80,84%				
Classification and Regression Tree (Hathaway dkk., 2019)	Sekuensing biokimia dan fisiologis untuk mengidentifikasi biomarker yang mempengaruhi status diabetes	84,00%				
Long Short-Term Memory (Faruqui dkk., 2019)	Kadar glukosa harian pasien diabetes tipe 2, berdasarkan data kesehatan harian dari ponsel seperti aktivitas fisik, berat badan, pola makan, dan kadar glukosa.	84,12%				
Long Short-Term Memory (El-Attar dkk., 2022)	Sekuens DNA Diabetes Melitus	95,00%	97,00%	98,00%	97,50%	

Pada tabel 4.19 dijelaskan bahwa K-Nearest Neighbor (KNN) digunakan untuk menganalisis sekuens DNA pada diabetes melitus tipe 1, tipe 2, dan gestasional. KNN adalah metode yang mengklasifikasikan data berdasarkan kedekatannya dengan data lainnya. dengan preprocessing menggungakan 3-mers, count vectorizer dan *min-max scaler*. Dengan *accuracy* 93,26 dan waktu komputasi 2 menit, metode ini terbukti efektif dalam klasifikasi jenis diabetes berdasarkan sekuens DNA menunjukkan bahwa KNN dapat mengenali pola dengan baik dalam data medis yang kompleks.

Decision Tree (Najar, 2022) diterapkan pada sekuens DNA jenis kanker darah (leukimia) untuk membedakan antara berbagai tipe kanker. Decision Tree membagi data menjadi serangkaian keputusan berdasarkan fitur yang relevan. Dengan *accuracy* 83,30%, metode ini memberikan hasil yang lebih rendah dikarenakan hanya memakai Teknik TF-IDF untuk preprocessing. Namun, metode ini tetap berguna dalam memberikan interpretasi yang mudah dipahami mengenai hasil klasifikasi. Random Forest (Zou, 2018) diterapkan pada analisis diabetes melitus, khususnya dengan indikator kadar fasting glukosa. Random Forest merupakan metode ensemble yang menggabungkan beberapa pohon keputusan untuk meningkatkan *accuracy*. Dengan *accuracy* 80,84%. Classification and Regression Tree (Hathaway dkk., 2019) digunakan untuk mengidentifikasi biomarker yang mempengaruhi status diabetes berdasarkan data biokimia dan fisiologis. Dengan *accuracy* 84,00%. Long Short-Term Memory (LSTM) (Faruqui dkk., 2019) digunakan untuk menganalisis kadar glukosa harian pada pasien diabetes tipe 2 dengan mempertimbangkan berbagai faktor seperti aktivitas fisik, berat badan, pola makan, dan kadar glukosa itu sendiri. LSTM adalah jenis jaringan saraf yang sangat

efektif dalam memproses data urutan waktu. Dengan *accuracy* 84,12%. Long Short-Term Memory (LSTM). Meskipun menunjukkan *accuracy* yang lebih rendah, ketiga metode tersebut dapat ditingkatkan dengan menambahkan dan mengubah teknik preprocessing tersebut

El-Attar dkk., (2022) juga menerapkan dalam analisis sekuens DNA untuk diabetes melitus dengan algoritma LSTM. Dengan teknik preprocessing one-hot encoding, LSTM menunjukkan *accuracy* yang sangat tinggi, yaitu 95,00%. LSTM dalam Sekuens DNA mendapatkan *accuracy* yang tinggi dikarenakan LSTM adalah jenis jaringan saraf dalam (deep learning) yang dirancang untuk memproses dan mempelajari urutan data panjang sekuens DNA dan membutuhkan waktu komputasi yang lebih lama daripada algoritma KNN.

Keunggulan penelitian ini terlihat dari fokus dataset yang berbasis sekuens DNA, berbeda dari penelitian sebelumnya yang lebih banyak menggunakan data seperti kadar glukosa, data biokimia, atau aktivitas harian. Meskipun jarak Minkowski (p = 3) memerlukan waktu komputasi lebih lama (2 menit) dibandingkan jarak lain (1 menit), *accuracy* yang dihasilkan menjadi bukti keunggulan metode ini. Temuan ini tidak hanya memberikan *accuracy* tertinggi tetapi juga memperkuat relevansi penggunaan algoritma KNN dalam mengolah data genetik, khususnya untuk identifikasi penyakit berbasis DNA seperti Diabetes Mellitus.

Secara keseluruhan, hasil ini menunjukkan bahwa model KNN memiliki kinerja yang sangat baik dalam mengklasifikasikan sekuens DNA untuk ketiga kelas, dengan *accuracy* tertinggi untuk NONDM dan kinerja yang cukup seimbang untuk DMT1 dan DMT2. Hal ini menunjukkan bahwa model berhasil mengenali

pola dalam sekuens DNA dan mengklasifikasikannya dengan *accuracy* yang tinggi, serta memiliki keunggulan dalam membedakan antara sekuens yang terkena diabetes melitus atau tidak. Dengan hasil ini, model KNN dapat diandalkan untuk klasifikasi jenis diabetes berdasarkan data sekuens DNA.

4.4 Klasifikasi DNA Diabetes Melitus dalam Prespektif Islam

Diabetes Melitus adalah penyakit kronis yang memengaruhi cara tubuh mengatur kadar gula darah. Penyakit ini terjadi akibat gangguan pada produksi atau fungsi insulin, hormon yang berperan dalam mengontrol kadar glukosa di dalam darah. Secara umum, diabetes dibedakan menjadi dua tipe utama, yaitu Diabetes Tipe 1 dan Tipe 2. Diabetes Tipe 1 terjadi ketika sistem kekebalan tubuh secara keliru menyerang sel-sel pankreas yang memproduksi insulin. Sementara itu, Diabetes Tipe 2 lebih sering disebabkan oleh resistensi insulin, yakni kondisi ketika tubuh tidak dapat menggunakan insulin dengan efektif, yang sering dipicu oleh gaya hidup tidak sehat, pola makan yang buruk, dan kurangnya aktivitas fisik.

Dalam ajaran Islam, menjaga kesehatan merupakan kewajiban yang sangat ditekankan, sebagaimana dijelaskan dalam berbagai ayat Al-Qur'an dan hadis Nabi. Tubuh manusia dianggap sebagai amanah dari Allah yang harus dijaga dengan penuh tanggung jawab. Dalam Surah Al-Mu'minun ayat 12-14, Allah menjelaskan proses penciptaan manusia dari segumpal darah hingga menjadi makhluk yang sempurna. Ayat ini menggambarkan betapa sempurnanya tubuh manusia sebagai hasil ciptaan Allah. Ajaran Islam mengajarkan bahwa menjaga kesehatan adalah bentuk syukur atas nikmat Allah, dan umat Islam dianjurkan untuk menerapkan pola hidup sehat, seperti makan dengan seimbang, rutin berolahraga, dan menjaga

ketenangan jiwa. Prinsip-prinsip ini selaras dengan anjuran medis untuk mencegah penyakit kronis, termasuk Diabetes Melitus tipe 1 dan tipe 2.

Surah At-Tin ayat 4 mengingatkan bahwa manusia diciptakan dalam bentuk yang sebaik-baiknya. Penciptaan ini mencakup kesempurnaan struktur genetik manusia, termasuk DNA, yang menjadi dasar kehidupan biologis. Teknologi klasifikasi DNA membantu manusia memahami lebih dalam bagaimana tubuh bekerja dan mendeteksi potensi risiko penyakit tertentu, seperti Diabetes Melitus. Dengan demikian, klasifikasi DNA tidak hanya menjadi alat medis yang canggih tetapi juga bukti ilmiah yang menguatkan keajaiban penciptaan manusia sebagaimana dinyatakan dalam Al-Qur'an. Teknologi ini menunjukkan bahwa tubuh manusia adalah sistem yang kompleks yang patut dijaga dengan baik sebagai wujud penghormatan kepada Allah SWT.

Pengetahuan tentang faktor genetik melalui teknologi DNA memungkinkan deteksi lebih awal terhadap risiko diabetes. Hal ini mengharuskan bagi seorang individu untuk mengambil langkah pencegahan dengan lebih baik, seperti mengatur pola makan, meningkatkan aktivitas fisik, atau berkonsultasi dengan tenaga medis. Dalam perspektif Islam, usaha menjaga tubuh melalui pencegahan penyakit juga termasuk bagian dari ibadah. Kombinasi antara ajaran agama dan ilmu pengetahuan modern menjadi cara yang efektif dalam mencegah dan mengelola penyakit kronis seperti Diabetes Melitus. Kolaborasi antara nilai-nilai agama dan pengetahuan ilmiah ini juga menguatkan pemahaman akan keagungan ciptaan Allah yang harus selalu disyukuri.

BAB V

PENUTUP

5.1 Kesimpulan

Kombinasi beberapa faktor memberikan *accuracy* tertinggi dalam model KNN. Pada Teknik *preprocessing data,* Penggunaan 3-mers, *count vectorizer*, *min-max scaler,* dan *label encoder* untuk representasi sekuens DNA kedalam numerik. Pada kasifikasi dengan model KNN dengan *test size* 10%, 2-nearest neighbor, distance weight, dan jarak Minkowski dengan $p = 3$ terbukti memberikan hasil yang optimal dalam algoritma k-nearest neighbor (KNN) dan memberikan *accuracy* 93,26%. Faktor-faktor ini berkontribusi pada model yang lebih sensitif terhadap pola-pola penting dalam data, serta mengurangi kemungkinan kesalahan dalam klasifikasi, terutama dalam membedakan kelas Data diabetes melitus tipe 1 (DMT1), diabetes melitus tipe 2 (DMT2), dan diabetes melitus gestasional (NONDM).

Model KNN menunjukkan kinerja yang baik dalam mengklasifikasikan data. Sebagai berikut:

1. DMT1, sebanyak 93 data berhasil diprediksi dengan tepat sebagai DMT1, sementara 6 data DMT1 salah diprediksi sebagai DMT2. Tidak ada data yang salah diprediksi sebagai NONDM.
2. Data DMT2, model berhasil memprediksi 90 data sebagai DMT2 dengan benar, tetapi ada 8 data DMT2 yang salah diprediksi sebagai DMT1 dan 1 data yang salah diprediksi sebagai NONDM.

3. Data NONDM, model dengan tepat memprediksi 94 data sebagai NONDM, sementara 2 data NONDM salah diprediksi sebagai DMT1 dan 3 data salah diprediksi sebagai DMT2.

Secara keseluruhan, model KNN memiliki hasil klasifikasi sebagai berikut:

1. Accuracy sebesar 93,26%, yang berarti mayoritas prediksi yang dibuat oleh model benar.
2. Rata-rata *Precision* yaitu 93% menunjukkan keakuratan model dalam mendeteksi diabetes melitus.
3. Rata-rata *recall* adalah 93%, yang menunjukkan model cukup sensitif dalam mendeteksi diabetes melitus.
4. Rata rata *f1-score* untuk mencapai 93%, menandakan model ini berhasil mencapai keseimbangan yang baik antara *accuracy* dan deteksi positif.
5. Secara keseluruhan, model ini efektif dalam meminimalkan kesalahan prediksi dan menghasilkan klasifikasi yang akurat.

Pengetahuan tentang faktor genetik melalui teknologi DNA memungkinkan deteksi lebih awal terhadap risiko diabetes. Dalam perspektif Islam, usaha menjaga tubuh melalui pencegahan penyakit juga termasuk bagian dari ibadah. Kombinasi antara ajaran agama dan ilmu pengetahuan modern menjadi cara yang efektif dalam mencegah dan mengelola penyakit kronis seperti Diabetes Melitus. Kolaborasi antara nilai-nilai agama dan pengetahuan ilmiah ini juga menguatkan pemahaman akan keagungan ciptaan Allah yang harus selalu disyukuri.

5.2 Saran

Berdasarkan penelitian terdahulu, penelitian lanjutan bisa dilakukan dengan mencoba model *machine learning* yang lebih modern seperti Random Forest, Support Vector Machine (SVM), atau Deep Learning untuk meningkatkan *accuracy*. Untuk melihat apakah hasilnya bisa lebih baik bisa menggunakan dataset yang lebih besar dan menambah variasi dataset, baik dari jumlah sampel maupun keragaman populasi, dapat membuat model lebih akurat dan dapat digunakan secara umum. Menggabungkan data genetik dengan data klinis, seperti riwayat kesehatan atau kadar gula darah, juga bisa membantu membuat prediksi yang lebih lengkap dan mendalam. Dengan langkah-langkah ini, penelitian di bidang ini diharapkan semakin berkembang dan lebih bermanfaat.

DAFTAR PUSTAKA

Alves L. M., (2021) Como lidar com Classes Desbalanceadas em *Machine learning* luis-miguel-code.medium.com

El-Attar, N., M. Moustafa, B., & A. Awad, W. (2022). Deep Learning Model to Detect Diabetes Melitus Based on DNA Sequence. Intelligent Automation & Soft Computing, 31(1), 325–338. https://doi.org/10.32604/iasc.2022.019970

Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, *113*(7), 4903–4923. https://doi.org/10.1007/s10994-022-06296-4

Famuji, T. S., Herman, H., & Sunardi, S. (2024). Smart Contract Penyimpanan Data Genetika Manusia Berbiaya Murah pada Blockchain Ethereum. Jurnal Teknologi Informasi Dan Ilmu Komputer, 11(3), 695–704. https://doi.org/10.25126/jtiik.1137558

Faruqui, S. H. A., Du, Y., Meka, R., Alaeddini, A., Li, C., Shirinkam, S., & Wang, J. (2019). Development of a Deep Learning Model for Dynamic Forecasting of Blood Glucose Level for Type 2 Diabetes Mellitus: Secondary Analysis of a Randomized Controlled Trial. *JMIR MHealth and UHealth*, *7*(11), e14452. https://doi.org/10.2196/14452

Ganesan. K. (2022). How to use *count vectorizer*. kavita-ganesan.com

Harjanti, T. W., Setiyani, H., Trianto, J., & Rahmanto, Y. (2022). Classification of Mint Leaf Types Based on the Image Using Euclidean Distance and K-Means Clustering with Shape and Texture Feature Extraction. Tech-E, 5(2), 115–124. https://doi.org/10.31253/te.v5i1.94

Hathaway, Q. A., Roth, S. M., Pinti, M. v., Sprando, D. C., Kunovac, A., Durr, A. J., Cook, C. C., Fink, G. K., Cheuvront, T. B., Grossman, J. H., Aljahli, G. A., Taylor, A. D., Giromini, A. P., Allen, J. L., & Hollander, J. M. (2019). Machine-learning to stratify diabetic patients using novel cardiac biomarkers and integrative genomics. *Cardiovascular Diabetology*, *18*(1), 78. https://doi.org/10.1186/s12933-019-0879-0

Henderi, H. (2021). Comparison of Min-Max normalization and Z-Score Normalization in the *K-nearest neighbor* (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. IJIIS: International Journal of Informatics and Information Systems, 4(1), 13–20. https://doi.org/10.47738/ijiis.v4i1.73

Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-Label *Confusion matrix*. IEEE Access, 10, 19083–19095. https://doi.org/10.1109/ACCESS.2022.3151048

Jawle C. (2020). Using *Label encoder* on Unbalanced Categorical Data in *Machine learning* Using Python. medium.com

Jin, W., Li, J., Zhang, S., Feng, M., & Feng, S. (2023). Weighted Minkowski distance-based differential protection for active distribution networks considering the uncertainty of frequency-domain characteristics. Frontiers in Energy Research, 11. https://doi.org/10.3389/fenrg.2023.1242325

Kemenag. (2024). Qur'an Kemenag. quran.kemenag.go.id

Kemenkes. (2024). Kesehatan Jiwa: Masalah Yang Sering Disepelekan Dan Dianggap Tidak Penting. ayosehat.kemkes.go.id

Mathur, G., Pandey, A., & Goyal, S. (2023). A comprehensive tool for rapid and accurate prediction of disease using DNA sequence classifier. Journal of Ambient Intelligence and Humanized Computing, 14(10), 13869–13885. https://doi.org/10.1007/s12652-022-04099-y

Muhammad Ali, P. J. (2022). Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements. *ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY*, *10*(1), 85–91. https://doi.org/10.14500/aro.10955

Nandan A. (2020). K-Nearest Neighbor. www.datasciencelovers.com/

Najar, A. M., Sudarsana, I. W., Albab, M. U., & Andhika, S. (2022). *Machine learning* untuk Identifikasi Jenis Kanker Darah (Leukemia). *Vygotsky*, *4*(1), 47. https://doi.org/10.30736/voj.v4i1.493

Nayak, S., Bhat, M., Subba Reddy, N. v, & Ashwath Rao, B. (2022). Study of distance metrics on k - nearest neighbor algorithm for star categorization. *Journal of Physics: Conference Series*, *2161*(1), 012004. https://doi.org/10.1088/1742-6596/2161/1/012004

NU Online (2024). Berobat dalam Pandangan Islam. islam.nu.or.id

Nuha, M. F. (2024). *Implementasi algoritma nu-support vector machine pada model klasifikasi sekuens DNA manusia: studi kasus penderita diabetes mellitus* [Skripsi]. Universitas Islam Negeri Maulana Malik Ibrahim.

Panua, A. A., Zainuddin, R., Ahmad, E. H., & Sangkala, F. (2021). Faktor Risiko Terjadinya Covid-19 Pada Penderita Diabetes Melitus Tipe 2. Jurnal Ilmiah Kesehatan Sandi Husada, 10(2), 624–634.

Pflug, J. M., Holmes, V. R., Burrus, C., Johnston, J. S., & Maddison, D. R. (2020). Measuring Genome Sizes Using Read-Depth, k-mers, and Flow Cytometry: Methodological Comparisons in Beetles (Coleoptera). G3; Genes|Genomes|Genetics, 10(9), 3047–3060.

Shihab, M. Q. (2001). Tafsir Al-Misbah. Lentera Hati.

Retnoningsih, E., & Pramudita, R. (2020). Mengenal *Machine learning* Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python. BINA INSANI ICT JOURNAL, 7(2), 156. https://doi.org/10.51211/biict.v7i2.1422

Sajid Khan, Mehmoon Anwar, Huma Qayyum, Farooq Ali, & Marriam Nawaz. (2023). Fake News Classification using *Machine learning*: *Count vectorizer* and Support Vector Machine. Journal of Computing & Biomedical Informatics, 4(01). https://doi.org/10.56979/401/2022/85

Shah, S. R., Kaushik, A., Sharma, S., & Shah, J. (2020). Opinion-Mining on Marglish and Devanagari Comments of YouTube Cookery Channels Using Parametric and Non-Parametric Learning Models. *Big Data and Cognitive Computing*, *4*(1), 3. https://doi.org/10.3390/bdcc4010003

Syme A. (2023). Large genome assembly and polishing. training.galaxyproject.org

Uyanik, B., & Orman, G. K. (2023). A Manhattan distancebased hybrid recommendation system. International Journal of Applied Mathematics Electronics and Computers, 11(1), 20–29. https://doi.org/10.18100/ijamec.1232090

Viloria, A., Pineda Lezama, O. B., & Mercado-Caruzo, N. (2020). Unbalanced data processing using *oversampling*: *Machine learning*. Procedia Computer Science, 175, 108–113. https://doi.org/10.1016/j.procs.2020.07.018

Wongvorachan, T., He, S., & Bulut, O. (2023). A Comparison of Undersampling, *Oversampling*, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*, *14*(1), 54. https://doi.org/10.3390/info14010054

World Health Organization (2024). Diabetes. www.who.int

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. https://doi.org/10.3389/fbioe.2020.01032

Zhang, S. (2022). Challenges in KNN Classification. IEEE Transactions on Knowledge and Data Engineering, 34(10), 4663–4675. https://doi.org/10.1109/TKDE.2021.3049250

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting Diabetes Mellitus with *Machine Learning* Techniques. *Frontiers in Genetics*, *9*. https://doi.org/10.3389/fgene.2018.00515

DAFTAR LAMPIRAN

Lampiran 1 Data dan Coding

RIWAYAT HIDUP

Muhammad Kristover Armand, lahir di Gresik pada 31 Januari 2003, menempuh pendidikan dasar di SD Mujahidin 2 Surabaya dari tahun 2009 hingga 2015, kemudian melanjutkan studi ke SMP Negeri 1 Kedamean Gresik pada tahun 2015 hingga 2018. Setelah itu, penulis melanjutkan pendidikan di SMA Progresif Bumi Shalawat Sidoarjo pada tahun 2018 hingga 2021. 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 kegiatan Mathematics English Club (MEC) dari tahun 2021 hingga 2024, yang memberikan pengalaman berharga dalam pengembangan keterampilan bahasa Inggris. Selain itu, pada tahun 2024, penulis juga mendapatkan kesempatan untuk berkontribusi sebagai Asisten Laboratorium (Aslab) dalam mata kuliah Analisis Numerik, di mana penulis menggunakan keahlian dalam bahasa pemrograman Python untuk mendukung pembelajaran dalam bidang tersebut. Pada tahun 2025, penulis memperdalam pengetahuannya di bidang *machine learning* dengan mengikuti *bootcamp* Dicoding Laskar AI - Machine Learning.

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 : Muhammad Kristover Armand
NIM : 210601110051
Fakultas / Program Studi : Sains dan Teknologi / Matematika
Judul Skripsi : Klasifikasi Sekuens DNA Diabetes Melitus dengan Algoritma K-Nearest Neighbor
Pembimbing I : Ari Kusumastuti, M.Pd., M.Si
Pembimbing II : Ach. Nasichuddin, M.A

No	Tanggal	Hal	Tanda Tangan
1.	2 September 2024	Konsultasi Topik	1.
2.	9 September 2024	Konsultasi Bab I, II, dan III	2.
3.	16 September 2024	Konsultasi Bab I, II, dan III	3.
4.	23 September 2024	Konsultasi Bab I, II, dan III	4.
5.	30 September 2024	Konsultasi Bab I, II, dan III	5.
6.	7 Oktober 2024	Konsultasi Kajian Integrasi Bab I dan II	6.
7.	14 Oktober 2024	Konsultasi Kajian Integrasi Bab I dan II	7.
8.	21 Oktober 2024	Persetujuan untuk Seminar Proposal oleh Pembimbing I	8.
9.	28 Oktober 2024	Persetujuan untuk Seminar Proposal oleh Pembimbing II	9.
10.	4 November 2024	Konsultasi Bab IV dan V	10.
11.	11 November 2024	Konsultasi Bab IV dan V	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.	16 Desember 2024	Konsultasi Bab IV dan V	12.
13.	23 Desember 2024	Konsultasi Bab IV dan V	13.
14.	30 Desember 2024	Konsultasi Kajian Integrasi Bab IV	14.
15.	6 Januari 2024	Konsultasi Kajian Integrasi Bab IV	15.
16.	13 Januari 2024	Persetujuan untuk Seminar Hasil oleh Pembimbing I	16.
17.	20 Januari 2024	Persetujuan untuk Seminar Hasil oleh Pembimbing II	17.
18.	24 Februari 2025	Persetujuan untuk Sidang Skripsi oleh Pembimbing I	18.
19.	3 Maret 2025	Persetujuan untuk Sidang Skripsi oleh Pembimbing II	19.

Malang, 14 Maret 2025
Mengetahui,
Ketua Program Studi Matematika

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