**SKRIPSI**

**PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* DAN *RANDOM FOREST* UNTUK**

**PREDIKSI PENYAKIT DIABETES**

**Oleh:**

**Dinnar Nurhuda Hermawan**

**065119213**



**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PAKUAN**

**BOGOR**

**2024**

# **HALAMAN PENGESAHAN**

Judul : Perbandingan Algoritma *Support Vector Machine* dan *Random Forest*

Untuk Prediksi Penyakit Diabetes

Nama : Dinnar Nurhuda Hermawan

NPM : 065119213

Mengesahkan,

|  |  |
| --- | --- |
| Pembimbing Pendamping  Program Studi Ilmu Komputer  FMIPA - UNPAK | Pembimbing Utama  Program Studi Ilmu Komputer  FMIPA - UNPAK |
|  |  |
| **Halimah Tussa’diah, M.Kom.** | **Dr. Tjut Awaliyah Zuraiyah, M.Kom.** |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Program Studi Ilmu Komputer FMIPA - UNPAK | Dekan  FMIPA - UNPAK |
|  |  |
| **Arie Qur’ania, S.Kom., M.Kom.** | **Asep Denih, S.Kom., M.Sc., Ph.D** |

# **PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI**

Dengan ini saya menyatakan bahwa:

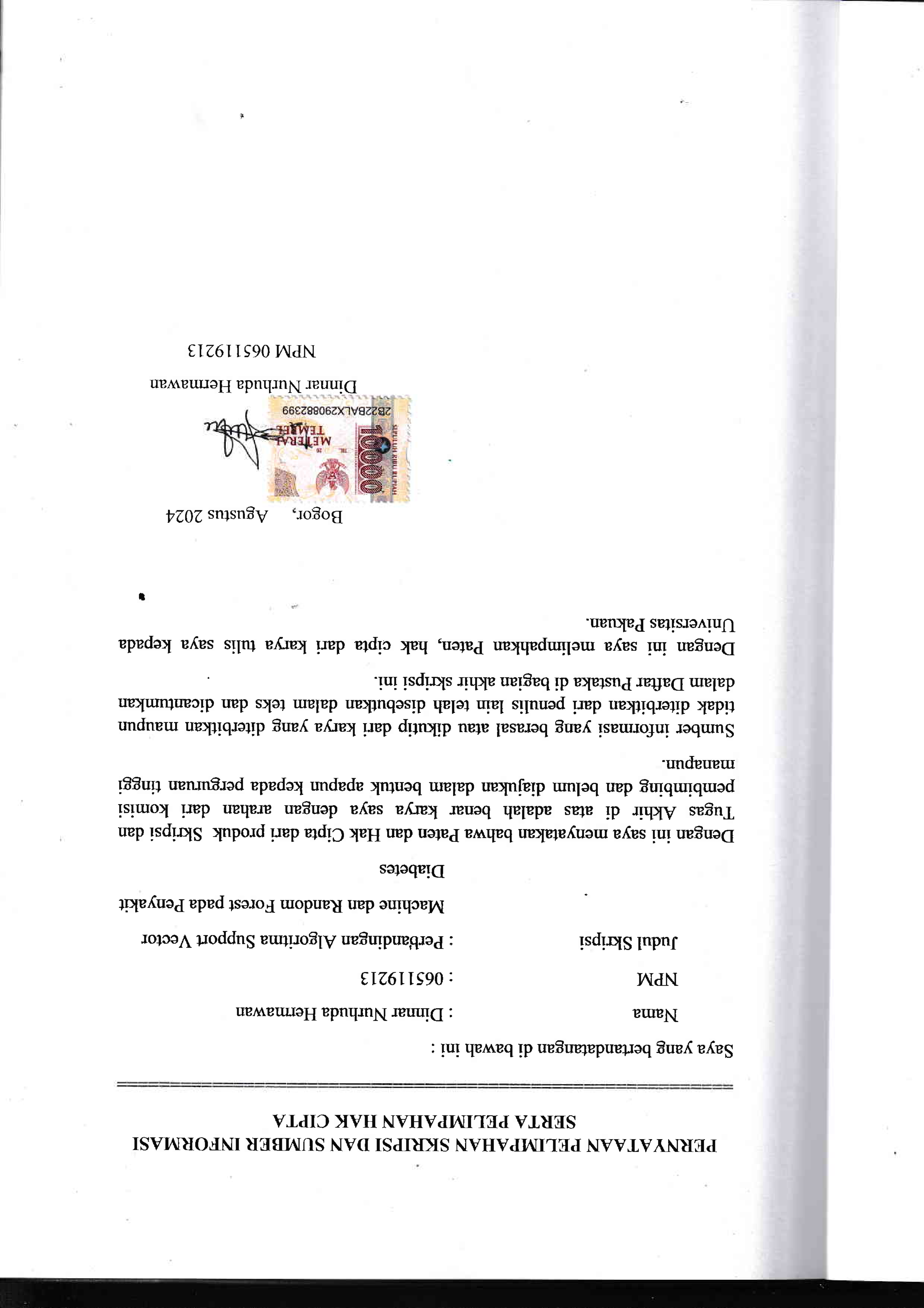
Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian dimana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyaan ini saya buat dengan sebenar-benarnya. Apabila kelas dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan berlaku.

Bogor, Agustus 2024

Dinnar Nurhuda Hermawan

NPM 065119213

**PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

**==========================================================**

Saya yang bertandatangan di bawah ini :

Nama : Dinnar Nurhuda Hermawan

NPM : 065119213

Judul Skripsi : Perbandingan Algoritma Support Vector

Machine dan Random Forest pada Penyakit

Diabetes

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, Agustus 2024

Dinnar Nurhuda Hermawan

NPM 065119213

# **RIWAYAT HIDUP**

A person in a purple suit

Description automatically generatedDinnar Nurhuda Hermawan, lahir pada tanggal 02 Maret 2000 di Bogor. Anak ke 6 dari 6 bersaudara dari pasangan Bapak Dedi Hermawan dan Ibu Tresnawati. Peneliti mengawali pendidikan di bangku sekolah dasar di SD Negeri Cikereteg 01 pada tahun 2006 dan tamat 2012. Pada tahun yang sama penulis melanjutkan pendidikan ke SMP Humanika Bogor dan tamat pada tahun 2015. Lalu melanjutkan ke tahap Sekolah Menengah Kejuruan Amaliah 1 dan tamat pada tahun 2018. Pada tahun 2019 peniliti melanjutkan pendidikan perguruan tinggi di Universitas Pakuan Fakultas Matematika dan Ilmu Pengetahuan Alam pada Program Studi Ilmu Komputer (ILKOM). Peneliti menyelesaikan kuliah strata satu (S1) pada tahun 2024.

# **KATA PENGANTAR**

Segala puji dan syukur kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya karena dengan segala curahan rahmat dan hidayahnya segala sesuatu dan aktivitas dapat berjalan dengan baik, efektif dan efisien sehingga penulis dapat menyelesaikan skripsi yang berjudul: “Perbandingan Algoritma *Support Vector Machine* dan *Random Forest* Untuk Prediksi Penyakit Diabetes “. Penulisan tugas akhir ini merupakan salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Ilmu Komputer FMIPA UNPAK Bogor.

Dalam penulisan tugas akhir ini, penulis dengan senang hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Dr. Tjut Awaliyah Zuraiyah, M.Kom., selaku Pembimbing Utama yang telah memberikan dorongan moril dan motivasi kepada penulis.
2. Halimah Tussa’diah, M.Kom., selaku Pembimbing Pendamping yang telah memberikan bimbingan, semangat dan motivasi.
3. Arie Qur’ania S.Kom., M.Kom. selaku Ketua Program Studi Ilmu Komputer FMIPA Universitas pakuan Bogor.
4. Orang tua tercinta yang selalu memberikan doa dan dukungan baik moril maupun materil.
5. Teman-teman kelas GH 2019 yang telah menjadi penyemangat kuliah dari awal masuk hingga akhir.
6. Garnis Dwi Septiani yang selalu memberikan semangat serta dukungan dalam proses penyusunan tugas akhir ini.

Saran dan kritik yang membangun dalam penulisan tugas akhir ini akan diterima dengan senang hati. Mudah-mudahan Allah SWT akan membalas semua kebaikan kepada semua pihak yang membantu. Akhir kata, semoga laporan ini dapat bermanfaat bagi kita semua. Aamiin.

Bogor, Agustus 2024

Dinnar Nurhuda Hermawan

065119213

# **DAFTAR ISI**

**Halaman**

[KATA PENGANTAR i](#_Toc167459217)

[DAFTAR ISI ii](#_Toc167459218)

[DAFTAR GAMBAR iv](#_Toc167459219)

[DAFTAR TABEL v](#_Toc167459220)

[DAFTAR LAMPIRAN vi](#_Toc167459221)

[BAB I PENDAHULUAN 1](#_Toc167459222)

[1.1. Latar Belakang 1](#_Toc167459223)

[1.2. Ruang lingkup 2](#_Toc167459224)

[1.3. Tujuan Penelitian 2](#_Toc167459225)

[1.4. Manfaat Penelitian 2](#_Toc167459226)

[BAB II TINJAUAN PUSTAKA 3](#_Toc167459227)

[2.1. Landasan Teori 3](#_Toc167459228)

[2.1.1. *Support Vector Machine* (SVM) 3](#_Toc167459229)

[2.1.2. *Random Forest* 4](#_Toc167459230)

[2.1.3. *Confusion Matrix* 5](#_Toc167459231)

[2.1.4. Normalisasi *Z-Score* 6](#_Toc167459232)

[2.1.5. *Interquartile Range* (IQR) 7](#_Toc167459233)

[2.1.6. *Receiver Operations Curve – Area Under Curve* (ROC-AUC) 7](#_Toc167459234)

[2.1.7. NumPy 7](#_Toc167459235)

[2.1.8. Pandas 8](#_Toc167459236)

[2.1.9. Matplotlib 8](#_Toc167459237)

[2.1.11. *Scikit-Learn* 8](#_Toc167459238)

[2.1.12. Diabetes Melitus 8](#_Toc167459239)

[2.2. Penelitian Terdahulu 8](#_Toc167459240)

[2.3. Tabel Perbandingan Penelitian 11](#_Toc167459241)

[BAB III METODE PENELITIAN 13](#_Toc167459242)

[3.1. Metode 13](#_Toc167459243)

[3.1.1. *Business Understanding* 13](#_Toc167459244)

[3.1.2. *Data Understanding* 13](#_Toc167459245)

[3.1.3. *Data* *Preparation* 13](#_Toc167459246)

[3.1.4. *Modeling* 14](#_Toc167459247)

[3.1.5. *Evaluation* 14](#_Toc167459248)

[3.1.6. *Deployment* 14](#_Toc167459249)

[3.2. Alat dan Bahan 14](#_Toc167459250)

[3.2.1. Alat 14](#_Toc167459251)

[3.2.2. Bahan 14](#_Toc167459252)

[BAB IV RANCANGAN DAN IMPLEMENTASI 15](#_Toc167459253)

[4.1. *Business Understanding* 15](#_Toc167459254)

[4.2. *Data Understanding* 15](#_Toc167459255)

[4.3. *Data Preparation* 15](#_Toc167459256)

[4.3.1. *Data Cleaning* 15](#_Toc167459257)

[4.3.2. *Data Transformation* 17](#_Toc167459258)

[4.3.3. *Data Splitting* 18](#_Toc167459259)

[4.4. *Modeling* 18](#_Toc167459260)

[4.4.1. Model Klasifikasi SVM 18](#_Toc167459261)

[4.4.2. Model Klasifikasi *Random Forest* 19](#_Toc167459262)

[4.5. *Evaluation* 20](#_Toc167459263)

[4.5.1. *Confusion Matrix* 20](#_Toc167459264)

[4.5.2. ROC-AUC 21](#_Toc167459265)

[4.6. *Deployment* 21](#_Toc167459266)

[BAB V HASIL DAN PEMBAHASAN 22](#_Toc167459267)

[5.1. Hasil 22](#_Toc167459268)

[5.1.1. Pemanggilan *Library* dan *Package* 22](#_Toc167459269)

[5.1.2. Deskripsi Data 22](#_Toc167459270)

[5.1.3. *Data* *Preprocessing* 25](#_Toc167459271)

[5.1.4. *Modeling* 27](#_Toc167459272)

[5.1.5. *Evaluation* 28](#_Toc167459273)

[5.1.6. *Deployment* 30](#_Toc167459274)

[5.2. Uji Coba 32](#_Toc167459275)

[BAB VI KESIMPULAN DAN SARAN 33](#_Toc167459276)

[6.1. Kesimpulan 33](#_Toc167459277)

[6.2. Saran 33](#_Toc167459278)

[DAFTAR PUSTAKA 34](#_Toc167459279)

[LAMPIRAN 36](#_Toc167459280)

# **DAFTAR GAMBAR**

**Halaman**

[**Gambar 1.** Contoh Hyperplane Pada Klasifikasi SVM 3](#_Toc167288101)

[**Gambar 2.** Contoh Ilustrasi Random Forest 5](#_Toc167288102)

[**Gambar 3.** Tabel Confusion Matrix 6](#_Toc167288103)

[**Gambar 4.** Alur Metode CRISP-DM (Alhamdani, 2022) 13](#_Toc167288104)

[**Gambar 5.** Implementasi Pada Visual Studio Code 21](#_Toc167288105)

[**Gambar 6.** Pemanggilan Library 22](#_Toc167288106)

[**Gambar 7.** Deskripsi Data 22](#_Toc167288107)

[**Gambar 8.** Barchart Jenis Kelamin 23](#_Toc167288108)

[**Gambar 9.** Visualisasi KDE Plot 23](#_Toc167288109)

[**Gambar 10.** Feature Corellations 24](#_Toc167288110)

[**Gambar 11.** Visualisasi categorical plot 25](#_Toc167288111)

[**Gambar 12.** Sebelum dan sesudah proses pengisian data null 26](#_Toc167288112)

[**Gambar 13.** Boxplot sebelum dan setelah Preprocessing 26](#_Toc167288113)

[**Gambar 14.** Normalisasi Z-Score 27](#_Toc167288114)

[**Gambar 15.** Proses Data Splitting 27](#_Toc167288115)

[**Gambar 16.** Modeling SVM dan Random Forest 28](#_Toc167288116)

[**Gambar 17.** Confusion Matrix SVM dan Random Forest 28](#_Toc167288117)

[**Gambar 18.** ROC-AUC 29](#_Toc167288118)

[**Gambar 19.** Halaman Beranda 30](#_Toc167288119)

[**Gambar 20.** Halaman Informasi 30](#_Toc167288120)

[**Gambar 21.** Halaman Prediksi 31](#_Toc167288121)

[**Gambar 22.** Halaman Visualisasi 31](#_Toc167288122)

# **DAFTAR TABEL**

**Halaman**

[**Tabel 1.** Tabel Perbandingan Penelitian 11](#_Toc167288123)

[**Tabel 2.** Data Kosong 15](#_Toc167288124)

[**Tabel 3.** Frekuensi 16](#_Toc167288125)

[**Tabel 4.** Variabel Tekanan Darah 16](#_Toc167288126)

[**Tabel 5.** Mengurutkan Data 16](#_Toc167288127)

[**Tabel 6.** Setelah Transformation Data 18](#_Toc167288128)

[**Tabel 7.** Contoh Pohon Pertama 19](#_Toc167288129)

[**Tabel 8.** Kontingensi Pertama 20](#_Toc167288130)

[**Tabel 9.** Confusion Matrix 20](#_Toc167288131)

[**Tabel 10.** Perbandingan Model 29](#_Toc167288132)

[**Tabel 11.** Hasil uji coba pada website 32](#_Toc167288133)

# **DAFTAR LAMPIRAN**

**Halaman**

[**Lampiran 1.** Definisi Variabel Penelitian 37](#_Toc167288340)

[**Lampiran 2.** Sebelum Transformation Dataset 37](#_Toc167288341)

[**Lampiran 3.** Contoh studi kasus dataset SVM 37](#_Toc167288342)

[**Lampiran 4.** Perhitungan Persamaan SVM 38](#_Toc167288343)

[**Lampiran 5.** Perhitungan Pohon dan Kontingensi Random Forest 40](#_Toc167288344)

[**Lampiran 6.** Dataset CSV Puskesmas Citapen 47](#_Toc167288345)

[**Lampiran 7.** Uji coba model pada website 49](#_Toc167288346)

# **BAB I**

# **PENDAHULUAN**

## **1.1. Latar Belakang**

Diabetes melitus merupakan salah satu penyakit yang paling luas dan bertahan lama yang menjangkiti manusia di seluruh dunia. Sekitar 425 juta orang telah menderita secara global saat ini dan diperkirakan hingga 700 juta orang akan terpengaruh pada tahun 2045 (Patil et al., 2022). Penyakit ini merupakan kondisi dimana metabolisme yang disebabkan oleh peningkatan kadar glukosa darah yang mengakibatkan masalah kesehatan seperti infark miokard, stroke, gagal ginjal, kebutaan, neuropati, gangren, dan peningkatan kecenderungan infeksi. Menurut para dokter, diabetes diduga disebabkan oleh gaya hidup dan pola makan yang berperan penting dalam penyakit ini. Penderita diabetes berisiko mengalami sejumlah masalah kesehatan sekunder, termasuk kerusakan saraf dan penyakit jantung (Hovi et al., 2022). Untuk mengontrol peningkatan terkena diabetes tipe 2 (kelenjar pankreas yang tidak dapat mencukupi kebutuhan insulin pada tubuh), perlu dilakukan diagnosis penyakit sejak dini untuk mencegah komplikasi serta mengurangi resiko masalah kesehatan yang serius. Sebelum dilakukan diagnosis, dokter harus menganalisis beberapa faktor, sehingga menjadikan pekerjaan dokter tidak efisien. Namun, teknologi dapat digunakan untuk membuat prediksi atau mendeteksi diabetes, sesuai dengan kemajuan terkini. Kemajuan teknologi ini dapat membantu mempermudah kerja dokter dalam mengantisipasi penyakit. Dibidang medis, misalnya, metode klasifikasi dapat digunakan untuk mengklasifikasikan tingkat keparahan penyakit pasien, sehingga memudahkan dokter dalam memberikan pengobatan yang paling efektif. Ada sejumlah pendekatan *machine learning* yang dapat digunakan untuk memecahkan masalah klasifikasi.

Oleh karena itu diperlukan teknik yang dapat mendukung dalam penyelesaian permasalahan diatas dengan menerapkan *data mining* untuk klasifikasi dengan memanfaatkan algoritma yaitu Algoritma *Support Vector Machine* dan *Random Forest* (Apriyani & Kumiati, 2020). Penelitian terkait metode klasifikasi salah satunya ialah perbandingan metode SVM, MLP DAN XGBoost pada data Ekspresi Gen (Fathur Rahman, 2020), hasil penelitian menunjukkan bahwa metode SVM memiliki akurasi yang lebih tinggi dibandingkan dengan metode MLP dan XGBoost. Kemudian pada penelitian Perbandingan Metode *Random Forest* dan naïve bayes Dalam Prediksi Keberhasilan Klien Telemarketing (Leonardo et al., 2020) menunjukan bahwa algoritma *Random Forest* lebih unggul.

Penelitian ini menggunakan metode SVM dikarenakan metode SVM merupakan salah satu algoritma *machine learning* yang populer untuk melakukan klasifikasi dan regresi. Metode SVM memiliki sejumlah keunggulan, antara lain kesesuaiannya untuk ruang berdimensi tinggi, efektivitas dalam kasus dimana jumlah dimensi lebih besar daripada jumlah sampel, penghematan memori karena penggunaan titik pelatihan dari fungsi keputusan (*support vector*), dan kinerja yang relatif baik bila terdapat pemisahan kelas-ke-kelas yang jelas (Trivusi, 2022). Selain itu metode yang digunakan dalam prediksi klasifikasi ini ialah *Random Forest,* metode ini efektif untuk menyimpan data dan dapat meningkatkan hasil akurasi jika ada data yang hilang serta tahan terhadap *outlier*. Selain itu, *random forest* juga memiliki proses penentuan komponen yang siap mengambil elemen-elemen terbaik

sehingga dapat berfungsi pada penyajian model penataannya. *random forest* mampu menangani data besar dengan parameter kompleks secara efektif berkat pemilihan fitur (Devella et al., 2020).

Peneliti membandingkan 2 metode SVM dan *Random* *Forest* dikarenakan pada penelitian sebelumnya oleh Rahman (2020) dan Leonardo et al., (2020) dimana dapat disimpulkan bahwa kedua metode tersebut memiliki kinerja yang sama-sama baik dalam melakukan klasifikasi, termasuk akurasi yang sangat baik dibandingkan dengan metode klasifikasi lainnya, dapat menangani data dengan jumlah yang besar, menangani *noise* dan *missing* data, kesalahan yang dibuat relatif rendah dan implementasi yang sederhana. Berdasarkan uraian tersebut maka akan dilakukan penelitian mengenai “Perbandingan Algoritma *Support Vector Machine* dan *Random Forest* Untuk Prediksi Penyakit Diabetes” untuk menentukan metode mana yang lebih efektif dalam memprediksi diabetes.

## **1.2. Ruang lingkup**

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan untuk meneliti penyakit Diabetes melitus yaitu, data dari rekam medik Puskesmas Citapen meliputi tekanan darah, glukosa, *body mass index*, usia, merokok, lingkar perut, dan hasil sebanyak 1124 data.
2. Membagi data menjadi data latih (*train*) dan data uji (*test*) dengan 4 skenario, skenario pertama data latih 70% dan data uji 30%, skenario kedua data latih 75% dan data uji 25%, skenario ketiga data latih 80% dan data uji 20%, dan skenario keempat data latih 90% dan 10% data uji dari total keseluruhan data.
3. Metode yang digunakan dalam klasifikasi penderita penyakit Diabetes melitus yaitu *Support Vector Machine* kernel RBF dan *Random Forest*.
4. Hasil evaluasi dari kinerja metode *Support Vector Machine* dan *Random Forest* menggunakan *confusion matrix* untuk mengetahui parameter *accuracy* dan *area under curve* (AUC).
5. Pada tahap deployment metode dengan hasil evaluasi terbaik dari kedua metode tersebut dibuatkan aplikasi website.

## **1.3. Tujuan Penelitian**

Membuat perbandingan metode *Support Vector Machine* dan *Random Forest* untuk mengetahui ketepatan klasifikasi pada dataset penderita penyakit diabetes melitus, lalu dibuatkan model prediksi dengan aplikasi web berdasarkan metode terbaik yang telah dikomparasi sebelumnya.

## **1.4. Manfaat Penelitian**

Manfaat dari penelitian ini antara lain, yaitu:

1. Memberikan model yang bisa digunakan oleh tenaga kesehatan dalam melakukan prediksi penyakit Diabetes melitus.
2. Masyarakat dapat mengidentifikasi prediksi penyakit Diabetes melitus berdasarkan penginputan gula darah puasa.
3. Memberikan gambaran tentang pasien di Puskesmas Citapen kepada instansi kesehatan.

# **BAB II**

# **TINJAUAN PUSTAKA**

## **2.1. Landasan Teori**

### **2.1.1. *Support Vector Machine* (SVM)**

Algoritma SVM bekerja dengan mencari petak pembatas yang dinamakan *hyperplane* yang memisahkan data kedalam 2 kelas unik. Pada saat ini, SVM dapat dikembangkan untuk mengklasifikasikan berbagai kelas. SVM akan memilih *hyperplane* yang paling ideal, dimana *hyperplane* tersebut seharusnya ideal dengan asumsi *hyperplane* tersebut berada tepat di tengah-tengah kedua kelas sehingga memiliki jarak terjauh ke data-data terluar di kedua kelas tersebut. Contoh menentukan *hyperplane* dapat dilihat pada gambar 1.

A screenshot of a computer

Description automatically generated with medium confidence

**Gambar 1.** Contoh Hyperplane Pada Klasifikasi SVM

Berdasarkan Gambar 1, terdapat banyak kemungkinan *hyperplane* yang dapat terbentuk. Dengan memanfaatkan SVM, *hyperplane* terbaik akan dilacak menggunakan margin dan *support vector*. *Hyperplane* yang terbaik adalah *hyperplane* yang mempunyai margin paling luas dan nilai data yang paling dekat dengan *hyperplane* atau *support vector* tersebut akan dimanfaatkan untuk membentuk model yang digunakan untuk pengelompokan. Persamaan 1 merupakan notasi *hyperplane* pada SVM.

(1)

w merupakan bobot suatu vektor dan b merupakan angka bias. kerangka kerja SVM adalah mencari *hyperplane* terbaik untuk mengelompokan dua kelas, kelas +1 (positif) dan kelas - 1 (negatif). Persamaan 2 merupakan fungsi klasifikasi pada SVM.

(2)

Apabila nilai > 0 maka data tersebut akan dikelompokan ke dalam kelas +1 (positif) dan apabila nilai w.x+b < 0 maka data tersebut akan dikelompokkan menjadi kelas -1 (negatif). Seperti penjelasan sebelumnya, *hyperplane* terbaik adalah *hyperplane* yang memiliki margin paling besar. margin terbesar dapat diperoleh dengan memperluas jarak antara *hyperplane* dan titik terdekatnya, khususnya . Permasalahan seperti ini biasa disebut dengan Quadratic Programming (QP), yaitu dengan menelusuri titik minimal dari persamaan 3:

(3)

Dengan batasan constraint persamaan 4:

(4)

Dari hasil tersebut akan diperoleh ≥ 0 yang disebut dengan *support vector* dan sisanya bernilai . Nilai *support vector* mempengaruhi fungsi keputusan yang dihasilkan. Secara umum, sangat jarang ditemukan data terpisah secara linear pada permasalahan di dunia nyata, jadi ketika menangani masalah-masalah nonlinier, SVM menggunakan fungsi kernel. Kernel dapat menyelesaikan permasalahan dengan atribut dan hubungan antara dua kelas yang tidak linear, dengan memetakan sampel data kedalam ruang dimensi yang lebih tinggi. SVM memiliki 4 kernel yang dapat dimanfaatkan untuk menangani persoalan linear dan nonlinier, antara lain:

1. Kernel Linear

(5)

Dengan merupakan data latih (*training*), adalah data uji

1. Kernel Polynomial

(6)

Dengan merupakan data latih (*training*), adalah data uji, adalah derajat polinomial.

1. Kernel Gaussian Radial Basis Function (RBF)

(7)

1. Sigmoid

(8)

Dengan merupakan data latih (*training*), adalah data uji, adalah koefisien.

Nilai ketepatan model yang dihasilkan menggunakan SVM sangat bergantung pada nilai batas yang digunakan dan fungsi kernel. Batasan yang digunakan dalam perhitungan SVM adalah batas *Cost* (C) dan Gamma (γ). Semakin tinggi nilai C, semakin tinggi pula penalty atas perintah tersebut (Kardono et al., 2023).

### **2.1.2. *Random Forest***

*Random forest* terdiri dari sekumpulan pohon pilihan yang telah disiapkan dengan menggunakan berbagai contoh, dan setiap atribut dipisahkan menjadi pohon yang dipilih dari subset atribut secara acak(Fauzi et al., 2020). Semakin banyak pohon yang digunakan maka keakuratan hasil akan semakin meningkat. *Random forest* terbentuk dari hasil pemilihan pohon yang digunakan untuk klasifikasi data. Pohon yang dipilih adalah pohon yang terbaik. Pohon-pohon pada *random forest* dibuat hingga mencapai ukuran terbesar dari pohon data (Nugraha et al., 2022). Ilustrasi *random forest* seperti pada gambar 2.

A diagram of a tree

Description automatically generated

**Gambar 2.** Contoh Ilustrasi Random Forest

Pembuatan model klasifikasi menggunakan *Random Forest* dengan tahapan analisis sebagai berikut:

1. *Bootstrapping* data dengan pengambilan sampel dari dataset data latih secara acak.
2. Pada tahap ini pohon dikerjakan hingga mencapai ukuran terbesarnya(tanpa dilakukan pruning).
3. Kemudian dari seluruh *features* dilakukan perhitungan untuk menghitung nilai gini indeks yang digunakan untuk membangun pohon dengan persamaan 9.

(9)

1. Selanjutnya dilakukan perhitungan gini split dari nilai gini indeks yang didapatkan dari masing-masing *features*, dengan persamaan 10.

(10)

*Features* dengan nilai gini *split* terbesar akan dijadikan *root node* pada pohon yang dibangun.

1. Kemudian dilakukan *split* hingga semua node berbentuk *leaf node* atau tidak ada *split* yang dapat dibuat kembali.
2. Apabila belum selesai membangun sampai k pohon, ulangi tahap 1-5, jika pohon yang dibangun sudah sampai pada k pohon, maka akan dilakukan *majority voting* untuk prediksi terakhir (Jackins et al., 2021).

### **2.1.3. *Confusion Matrix***

*Confussion matrix* adalah instrumen yang digunakan untuk memvisualisasikan performa model klasifikasi pada data *test* yang hasil aslinya diketahui sebelumnya. Ilustrasi tabel *confusion matrix* seperti pada gambar 3.

A diagram of values

Description automatically generated

**Gambar 3.** Tabel Confusion Matrix

*Confusion matrix* dapat digunakan untuk menghitung berbagai performa matriks dalam mengukur kinerja model yang telah dibuat seperti akurasi, sehingga dapat menggambarkan keakuratan model dalam mengklasifikasikan dengan benar (Rina, 2023). Persamaan dapat dilihat pada persamaan 11 dan 12.

(11)

(12)

### **2.1.4. Normalisasi *Z-Score***

Normalisasi z-*score* adalah suatu metode untuk mengubah ukuran data yang memiliki berbagai aspek menjadi skala yang seragam, yaitu antara 0 - 1. Standardisasi diperlukan mengingat suatu indeks data dapat memuat variasi skala estimasi yang berbeda-beda, sehingga tanpa standarisasi hasil pemeriksaan dapat dipengaruhi oleh beberapa faktor yang mempunyai skala estimasi terbesar. Perhitungan *z-score* diawali dengan perhitungan nilai rata-rata dengan persamaan 13:

(13)

Setelah itu dicari nilai standar deviasi dengan persamaan 14:

(14)

Hitung z-score dengan rumus persamaan 15:

(15)

Keterangan:

standar deviasi

nilai tengah

rata- rata

data yang akan dicari rata-ratanya

jumlah data

### **2.1.5. *Interquartile Range* (IQR)**

IQR adalah pengukuran yang digunakan untuk mengukur penyebaran data, seperti perlu mengidentifikasi adanya data *outlier* atau nilai-nilai ekstrem pada dataset. Semakin tinggi IQR maka semakin tinggi penyebaran data antara kuartil dan keragaman data tersebut (Info, 2023). Sebaliknya, jika IQR kecil, maka data cenderung lebih banyak muncul pada nilai yang sangat menengah. Langkah yang perlu dilakukan untuk menghitung IQR:

1. Urutkan data dalam urutan dari yang terkecil hingga yang terbesar.
2. Hitunglah kuartil pertama (Q1). Q1 adalah median dari data-data yang berada di bawah median utama. Jika jumlah data genap, Q1 adalah rata-rata dari dua nilai tengah di separuh bawah.
3. Lalu hitung juga kuartil ketiga (Q3). Q3 adalah median dari data-data yang berada di atas median utama. Jika jumlah data genap, Q3 adalah rata-rata dari dua nilai tengah di separuh atas.
4. Hitung IQR. Ingat, IQR adalah adalah selisih antara Q3 dan Q1. Seperti pada persamaan 16:

(16)

1. Selanjutnya mencari nilai *outlier* pada data dengan mencari nilai dari batas bawah dan batas atas dengan persamaan 17 dan 18.

(17)

(18)

### **2.1.6. *Receiver Operations Curve – Area Under Curve* (ROC-AUC)**

ROC-AUC merupakan ukuran matematis untuk membedakan performa model, memperlihatkan seberapa efektif dan akurat posisi model dengan memisahkan persepsi positif dan negatif. Lima kategori dinyatakan pada klasifikasi akurasi menggunakan *area under the curve* (AUC), yaitu:

1. Akurasi bernilai 0.90 – 1.00 = *Excellent classification*
2. Akurasi bernilai 0.80 – 0.90 = *Good classification*
3. Akurasi bernilai 0.70 – 0.80 = *Fair classification*
4. Akurasi bernilai 0.60 – 0.70 = *Poor classification*
5. Akurasi bernilai 0.50 – 0.60 = *Failure excellent* (Lestandy et al., 2020).

Nilai ROC dapat dihitung menggunakan persamaan *true positive rate* pada persamaan 19 dan *false positive rate* pada persamaan 20.

(19)

(20)

### **2.1.7. NumPy**

NumPy merupakan kependekan dari *Numerical Python*. NumPy adalah perpustakaan *Python* yang mampu melakukan proses komputasi numerik. NumPy dapat membuat objek N-dimensi array (Uswatun, 2021).

### **2.1.8. Pandas**

Pandas adalah perpustakaan *open source* dan resmi BSD dengan *Python* yang menawarkan desain dan menelaah data yang tidak sulit untuk digunakan. Pandas sering dimanfaatkan untuk membuat tabel, mengubah dimensi data, melihat data, dan lain-lain. Struktur data dasar di Pandas dikenal sebagai DataFrame, yang memudahkan kita membaca dokumen di berbagai jenis file, misalnya, .txt, .csv, dan .tsv (Miftah, 2021).

### **2.1.9. Matplotlib**

Matplotlib merupakan perpustakaan atau bundel yang paling terkenal pada bahasa *python* untuk memvisualisasikan data, misalnya membuat plot grafik untuk satu atau lebih sumbu. Setiap sumbu memiliki sumbu horizontal (x) dan sumbu vertikal (y) (Tineges, 2021).

**2.1.10. *Seaborn***

*Seaborn* merupakan perpustakaan untuk membuat bagan dan statistik menggunakan *Python*. Perpustakaan ini dibangun berdasarkan perpustakaan Matplotlib saat ini. Kemudian dikoordinasikan dengan struktur data di Pandas (Maulid, 2022).

### **2.1.11. *Scikit-Learn***

*Scikit-learn* memberikan berbagai pilihan algoritma pembelajaran yang diawasi (*supervised*) dan tidak diawasi (*unsupervised*). Bagian terbaik adalah ini merupakan perpustakaan *machine learning* yang paling mudah dan terbersih saat ini. *Scikit learn* dibuat dengan pola pikir pemrograman komputer. desain antarmuka pemrograman aplikasi tidak sulit untuk digunakan, namun kuat, sekaligus menjaga kemampuan beradaptasi untuk upaya penelitian (Gifari, 2020).

### **2.1.12. Diabetes Melitus**

Diabetes adalah penyakit yang diderita dalam kurun waktu lama dengan ditandai kadar glukosa yang tinggi. Glukosa adalah sumber energi mendasar untuk sel-sel tubuh manusia. Tetapi pada pengidap diabetes, glukosa ini tidak mampu dimanfaatkan oleh tubuh. ([Pittara](https://www.liputan6.com/me/nisa.sari), 2024).

## **2.2. Penelitian Terdahulu**

Referensi jurnal penelitian antara lain:

|  |  |  |
| --- | --- | --- |
| 1 | Nama : | Iqbal Fathur Rahman |
|  | Judul : | Implementasi Metode SVM, MLP Dan XGBoost Pada Data Ekspresi Gen |
|  | Tahun : | 2020 |
|  | Isi : | Sampel Gangguan Toleransi Glukosa (IGT) dan kadar gula darah normal (NGT) dengan kode seri GSE18732 menjadi subjek analisis klasifikasi yang dilakukan pada data *microarray* pada penelitian ini. Dalam penelitian ini, teknik pengelompokan *Support Vector Machine* (SVM), *Arsitektur multilayer Perceptron* (MLP), dan *Xtreme Gradient Boosting* (XGBoost) digunakan untuk menganalisis data tersebut. Berdasarkan analisis, metode klasifikasi *Support Vector Machine* dengan kernel *Linear* memperoleh akurasi tertinggi yaitu sebesar 91,30%. Sedangkan metode klasifikasi *Xtreme Gradient Boosting* mampu mencapai nilai akurasi sebesar 71,39%, sedangkan *Arsitektur Multilayer Perceptron* dengan satu *hidden layer* dan 100 *hidden node* mendapatkan akurasi sebesar 78,26%. |
| 2 | Nama : | Ricky Leonardo, Janis Pratama, & Chrisnatalis |
|  | Judul : | Perbandingan Metode *Random Forest* Dan Naïve Bayes Dalam Prediksi Keberhasilan Klien Telemarketing |
|  | Tahun : | 2020 |
|  | Isi : | Data yang digunakan dalam penelitian ini berasal dari sumber sekunder, yaitu website UCI Machine Learning. Random Forest dan Naive Bayes merupakan metode penelitian yang digunakan dalam penelitian ini. Sebelum melakukan persiapan dan pengujian, dilakukan preprocessing data pada dataset tersebut. Normalisasi data, atau mengubah data kategori, adalah metode yang digunakan untuk prapemrosesan. Algoritma Random Forest lebih cocok untuk memprediksi keputusan klien setelah dilakukan pengujian, dapat disimpulkan. Hal ini terlihat dari akurasi 90%, 5% lebih tinggi dibandingkan algoritma Naive Bayes. Algoritma Random Forest memiliki nilai AUC sebesar 0,97, lebih besar 1,3 dibandingkan algoritma Naive Bayes. |
| 3 | Nama : | Dicka Y, Yuliana M & Endang Setyati |
|  | Judul : | Prediksi Kecocokan Jurusan Siswa SMK Dengan *Support Vector Machine* dan *Random Forest* |
|  | Tahun : | 2023 |
|  | Isi : | Siswa SMP yang mendaftar di SMK Antartika 1 Sidoarjo pada umumnya kurang mengetahui minatnya terhadap jurusan-jurusan di sekolah. Rekayasa Perangkat Lunak, Teknik Kendaraan Ringan, dan teknik Permesinan merupakan jurusan yang ada di SMK Antartika 1 Sidoarjo. Diperlukan sistem untuk mengantisipasi pemilihan jurusan yang cocok bagi siswa baru di SMK Antartika 1 Sidoarjo berdasarkan permasalahan tersebut. Sistem ini berpotensi meningkatkan pelayanan kepada mahasiswa baru dalam menentukan jurusan yang terbagi dalam empat tingkatan yaitu sangat sesuai, sesuai, tidak sesuai, dan sangat tidak sesuai. Memanfaatkan aplikasi perbandingan yang menggunakan metode *Support Vector Machine* (SVM) dan *Random Forest* (RF) untuk mengetahui pola prediksi dari data siswa. Terdapat 14 fitur yang dimanfaatkan yang terdiri atas: Jurusan, Pendidikan\_Ayah, Pendapatan\_Ayah, Pendidikan\_Ibu, Pendapatan\_Ibu, Transportasi\_ke\_Sekolah, NUS\_MTK\_SMP, NUS\_BIND\_SMP, NUS\_BING\_SMP, Disiplin, Kewajiban, Sikap, Surat menyurat, dan Keluaran\_Kelas. Pada tahun 2020 hingga 2022, dataset siswa di SMK Antartica 1 Sidoarjo berisi 578 catatan data siswa untuk penelitian ini. Berdasarkan hasil analisis dengan metode SVM dengan kernel sigmoid diperoleh tingkat akurasi sebesar 83%, sedangkan hasil analisa dengan dengan metode RF dengan jumlah tree 150 diperoleh tingkat akurasi sebesar 82%. |
| 4 | Nama : | Ayu Aina N & Arie Wahyu W |
|  | Judul : | Perbandingan Algoritma Klasifikasi *Support Vector Machine* dan *Random Forest* pada Prediksi Status Indeks Mitigasi dan Kesiapsiagaan Bencana (IMKB) Satuan Kerja BPS di Indonesia Tahun 2020 |
|  | Tahun : | 2022 |
|  | Isi : | Tujuan dari penelitian ini adalah menghasilkan sistem klasifikasi yang dapat menggunakan sejumlah faktor yang mempengaruhi kesiapsiagaan bencana untuk memprediksi posisi satuan kerja dalam Indeks Mitigasi dan Kesiapsiagaan Bencana (IMKB). Teknik klasifikasi *Support Vector Machine* (SVM) dan *Random Forest* digunakan untuk pengolahan data pada aplikasi *R Studio*. Beberapa pengujian menunjukkan bahwa akurasi SVM dan *Random Forest* secara umum lebih baik jika dibandingkan dengan beberapa metode klasifikasi lainnya. Selain itu, SVM dapat digunakan untuk mengklasifikasikan data nonlinier, dan *Random Forest* tidak *overfit* seiring bertambahnya jumlah pohon. Dengan akurasi 78,22%, presisi 75,54%, dan *recall* 76%, metode klasifikasi *Random Forest* mengungguli SVM di ketiga area tersebut. |
| 5 | Nama : | Revika Dwi Merianti |
|  | Judul : | Implementasi Metode *Support Vector Machine* Dan *Random Forest* Pada Data Ekspresi Gen Microarray |
|  | Tahun : | 2020 |
|  | Isi : | *Support Vector Machine* (SVM) dan *Random Forest* merupakan teknik klasifikasi yang akan digunakan untuk menyelesaikan masalah medulloblastoma. Berdasarkan penelitian, akurasi klasifikasi metode *Random Forest* adalah 90,92%, sedangkan metode SVM dengan kernel linier, polinomial, dan sigmoid masing-masing sebesar 90,92%, 77,27%, dan 86,36%. Nilai AUC dari SVM dan *Random Forest* adalah 0,9754 atau 97,54%, dan termasuk dalam kelas *excellent classification*. Para peneliti merekomendasikan penggunaan SVM dan *Random Forest* untuk mengklasifikasikan data subgrup medulloblastoma karena hasil kedua metode tersebut sama dan faktor penyebab medulloblastoma masih sangat kecil. |

## **2.3. Tabel Perbandingan Penelitian**

**Tabel 1.** Tabel Perbandingan Penelitian

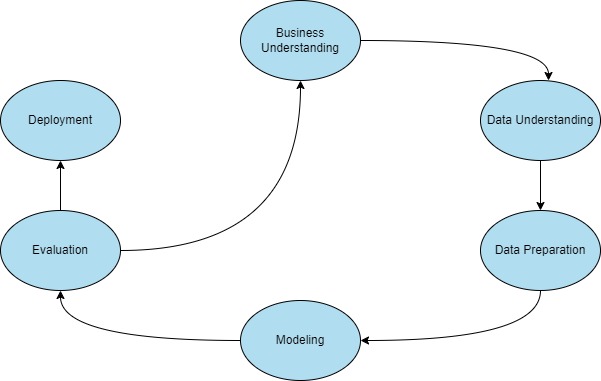
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Nama Peneliti, Tahun** | **Judul** | **Objek Penelitian** | **Jenis Perbandingan** | | | | | | | | | |
| **Metode** | | | | | **Pembagian Data** | | **Pengembangan** | | **Presentase Akurasi** |
| **SVM** | **MLP** | **XGBoost** | ***Random Forest*** | **Naïve bayes** | **Data Latih** | **Data Uji** | ***Website*** | ***Prototype*** |
| 1 | Iqbal Fathur Rahman, 2020 | Implementasi Metode SVM, MLP Dan XGBoost Pada Data Ekspresi Gen | Data Ekspresi Gen | **✓** | **✓** | **✓** | **-** | **-** | 80% dari total data 118 data | 20% dari total data 118 data | - | **✓** | 73.91%  (XGBoost) 91.30% (SVM)  78.26%  (MLP) |
| 2 | Ricky Leonardo, Janis Pratama, & Chrisnatalis, 2020 | Perbandingan Metode *Random Forest* Dan Naïve Bayes Dalam Prediksi Keberhasilan Klien Telemarketing | Keberhasilan Klien Telemarketing | - | - | **-** | **✓** | **✓** | 36.168 data | 9.043 data | - | **✓** | 90%  (*Random Forest*)  85%  (Naïve bayes) |
| 3 | Dicka Y, Yuliana M & Endang Setyati, 2023 | Prediksi Kecocokan Jurusan Siswa SMK Dengan *Support Vector Machine* dan *Random Forest* | Kecocokan Jurusan Siswa SMK | **✓** | **-** | - | **✓** | - | 80% Dari total data 578 data | 20% Dari total data 578 data | - | **✓** | 82%  (*Random Forest*)  83%  (SVM) |
| 4 | Ayu Aina & Arie Wahyu W, 2022 | Perbandingan Algoritma Klasifikasi *Support Vector Machine* dan *Random Forest* pada Prediksi Status Indeks Mitigasi dan Kesiapsiagaan Bencana (IMKB) Satuan Kerja BPS di Indonesia Tahun 2020 | Status Indeks Mitigasi dan Kesiapsiagaan Bencana (IMKB) Satuan Kerja BPS di Indonesia Tahun 2020 | **✓** | - | - | **✓** | - | 80% Dari total data 514 data | 20% Dari total data 514 data | - | **✓** | 78,22%  (*Random Forest*)  75,11%  (SVM) |
| 5 | Revika Dwi Merianti, 2020 | Implementasi Metode *Support Vector Machine* Dan *Random Forest* Pada Data Ekspresi Gen Microarray | Data Ekspresi Gen Microarray | **✓** | - | - | **✓** | **-** | 70% Dari total data 73 data | 30% Dari total data 73 data | - | **✓** | 90.92 %  (*Random Forest*) 90.92 %  (SVM) |
| 6 | Dinnar Nurhuda Hermawan, 2024 | Perbandingan Algoritma *Support Vector Machine* Dan *Random Forest* Untuk Prediksi Penyakit Diabetes | Penyakit Diabetes Pada Pasien Di Puskesmas Citapen | **✓** | - | - | **✓** | **-** | 70%, 75%,80%, 90%  Dari total data 1124 | 10%, 20%, 25%, 30%  Dari total data 1124 | **✓** | **✓** | - |

# **BAB III**

# **METODE PENELITIAN**

## **3.1. Metode**

Metode yang digunakan dalam penelitian ini adalah CRISP-DM (*Cross Industry Process for Data Mining*). CRISP-DM memiliki 6 fase untuk melakukan analisis *data mining* yaitu *business* *understanding, data understanding, data preparation, modeling, evaluation,* dan *deployment.* Sebagai sebuah rangkaian proses, alur metode CRISP-DM ditunjukan pada gambar 4.



**Gambar 4.** Alur Metode CRISP-DM (Alhamdani, 2022)

### **3.1.1. *Business Understanding***

Pada tahap ini terdapat beberapa aktivitas, misalnya memahami kebutuhan dan tujuan bisnis yang kemudian akan didefinisikan dalam *data mining*, kemudian menentukan langkah dan metodologi untuk mencapai tujuan *data mining*. Isu dalam penelitian ini adalah diabetes yang merupakan *silent killer*.

### **3.1.2. *Data Understanding***

Setelah melewati tahap pemahaman bisnis, beberapa hal yang harus dilakukan pada tahap ini antara lain memahami data yang akan digunakan, mengumpulkan data, dan mengeksplorasi data. Data yang dikumpulkan adalah data gula darah puasa dari Puskesmas Citapen pada tanggal 18 Maret 2023. Objek penelitian ini adalah data CSV yang berisi informasi data pasien yang terdiagnosis penyakit Diabetes melitus seperti tekanan darah, kadar glukosa, BMI, usia, merokok, lingkar perut, dan hasil diagnosis sebanyak 1124 data. Adapun macam-macam variabel yang digunakan pada penelitian ini dapat dilihat pada lampiran 1.

### **3.1.3. *Data* *Preparation***

*Data Preparation* dilakukan dengan 2 tahap, yaitu melalui *data cleaning* dan *data transformation*. Setelah data diperoleh, dilakukan pembersihan data dengan menghapus data yang tidak sesuai atau tidak diperlukan dalam penelitian, misalnya menghapus atau mengisi data *missing* *value* dan mengambil faktor-faktor penting yang akan diteliti secara manual. Setelah itu,

transformasi dataset dengan melakukan perubahan skala menggunakan standarisasi *z-score* sehingga memiliki rataan 0 dan simpangan baku 1.

### **3.1.4. *Modeling***

Tahap ini menggunakan *machine learning* secara langsung untuk memutuskan teknik *data mining* yang akan digunakan. Pada tahap ini, membuat skenario pemodelan terbaik dengan memilih algoritma yang akan digunakan, memisahkan data sesuai dengan ketersediaan data menjadi *data* *train* dan *data test*. Kemudian membuat model menggunakan dua model algoritma, yaitu *Support Vector Machine* dan *Random Forest*.

### **3.1.5. *Evaluation***

Tujuan dari tahap ini adalah untuk menilai kinerja pola berdasarkan algoritma yang telah ditentukan. *Receiver operating characteristic*-*area under curve* (ROC-AUC) dan *confusion matrix* adalah parameter yang digunakan untuk evaluasi komparasi algoritma dan mengetahui tingkat akurasi dan nilai *recall,* mulai dari prediksi positif yang benar, prediksi positif yang salah, prediksi negatif yang benar dan prediksi negatif yang salah. Akurasi akan ditentukan dari seluruh prediksi yang benar (baik prediksi positif maupun negatif).

### **3.1.6. *Deployment***

Tahapan terakhir yang akan dilakukan adalah pembuatan aplikasi web agar lebih mudah digunakan oleh pihak yang berkepentingan dan dapat digunakan untuk memprediksi penyakit diabetes berdasarkan analisis prediksi pola yang dihasilkan oleh algoritma terbaik dari hasil komparasi.

## **3.2. Alat dan Bahan**

### **3.2.1. Alat**

Alat yang dibutuhkan pada penelitian ini berupa perangkat lunak *(software)* dan perangkat keras *(hardware)* yaitu:

1. Perangkat Lunak (*Software*):

1. Google *Colab*

1. *Microsoft Office* 365
2. *Visual Studio Code* Ver. 1.85
3. *Streamlit*
4. OS *Windows* 10 Pro 64-bit
5. Perangkat Keras (*Hardware*):
6. Laptop Hp 14-AF118AU
7. Processor AMD A8-7410 APU with AMD Radeon R5 Graphics 2.20 GHz
8. RAM 8GB
9. ROM 500GB

### **3.2.2. Bahan**

Bahan yang dibutuhkan dalam penelitian ini adalah dataset CSV pasien Diabetes melitus dari Puskesmas Citapen.

# **BAB IV**

# **RANCANGAN DAN IMPLEMENTASI**

## **4.1.** ***Business Understanding***

Karena diabetes adalah penyebab kematian ketujuh di seluruh dunia, tujuan dari penelitian ini adalah untuk mengembangkan model *data* *mining* yang dapat digunakan sebagai indikator awal dari kondisi tersebut. Status diabetes sebagai *silent killer* harus ditangani secara efektif untuk mengurangi angka kematiannya. Mayoritas penderita diabetes tidak menyadari bahwa mereka mengidap penyakit tersebut, dan sebagian besar dari mereka mungkin hanya mencari pertolongan medis jika kondisinya sudah sangat parah sehingga sulit untuk diobati. Pengendalian dan pencegahan diabetes diantisipasi sebagai hasil dari prediksi ini.

## **4.2. *Data Understanding***

Pengumpulan data ini dilakukan dengan cara wawancara secara langsung dengan pihak pertama yaitu penanggung jawab puskesmas citapen, data yang akan digunakan pada penelitian ini yaitu sebuah data hasil pemeriksaan di Puskesmas Citapen pada periode 2022 sebanyak 1124 data dan terdiri dari 6 variabel independen dan 1 variabel dependen. Berdasarkan arahan dari tenaga medis Puskesmas Citapen, bahwa variabel Merokok, BMI, Usia, Tekanan Darah, Lingkar Perut dan Glukosa merupakan variabel yang mempengaruhi penyakit diabetes, sehingga variabel yang lain tidak digunakan pada penelitian ini.

## **4.3. *Data Preparation***

Membersihkan, mengubah, dan memisahkan data adalah bagian dari persiapan data*.* Tahap awal yang dilakukan pada penelitian ini adalah pembersihan data yang diharapkan dapat menemukan apakah terdapat data null (*missing value*) ataupun *outlier* pada dataset. Data *outlier* memiliki banyak dampak negatif karena dapat berdampak pada proses pembelajaran model dan menurunkan metrik secara signifikan. Pada setiap atribut, deteksi *outlier* dilakukan satu per satu.

### **4.3.1. *Data Cleaning***

#### **4.3.1.1. Mengisi Data Kosong**

Dalam tahapan ini, dataset terdapat data yang bernilai *null* (kosong). Maka dari itu perlu dilakukannya tahap *preprocessing* data dengan mengisi nilai data yang kosong dengan menentukan perhitungan yang cocok dengan metode mean, modus, atau median dari setiap variabel sehingga dataset ini menjadi bersih. Pada tabel 2 contoh terdapat nilai kosong pada dataset.

**Tabel 2.** Data Kosong

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  diabetes |
| 1 | 1 | 166 | 63 | 23.6 | 33 | 93 | 1 |
| 2 | 1 | 119 | 68 | 21.7 | 51 | 78 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 0 | 173 | 82 | 26.1 | 54 | 108 | 1 |
| 6 |  | 338 | 92 | 27.9 | 55 | 85 | 1 |
| 7 | 1 | 123 | 71 | 31.6 | 26 | 97 | 0 |
| 8 | 0 | 135 | 82 | 22.3 | 24 | 73 | 0 |

Pada tabel 3 kolom ke 6 terdapat data yang bernilai *null* pada variabel merokok, karena variabel ini sebenarnya termasuk varibel non-numerik, yaitu YA disimbolkan dengan angka 1 dan TIDAK disimbolkan dengan angka 0. Menurut Afifah (2020) metode yang cocok untuk mengisi nilai *null* dengan tipe non-numerik pada *data cleaning* dapat dilakukan dengan metode modus atau nilai yang paling sering muncul dalam kolom tersebut. Berdasarkan hal tersebut, maka pada tabel 3 kolom ke 6 diisi dengan nilai modus. Adapun mencari nilai modus yang terdapat pada kolom variabel Merokok dan mengelompokan data untuk menentukan frekuensi tiap data pada tabel 3.

**Tabel 3.** Frekuensi

|  |  |
| --- | --- |
| Nilai | Frekuensi |
| 0 | 4 |
| 1 | 3 |

Dari tabel 3 dapat dilihat, 0 merupakan nilai modus dari kolom tersebut karena memiliki data terbanyak yaitu 3. Selanjutnya kolom kosong tersebut dapat diisi dengan nilai modus yang telah didapatkan yaitu nilai 0.

#### **4.3.1.2. Mengganti Nilai Outliers**

Tahap selanjutnya pada data *cleaning*, mencari nilai *outliers* pada dataset dengan metode *Interquartile range* (IQR). Menurut Setiawan (2020), mendeteksi *outliers* menggunakan metode IQR cukup mudah dilakukan. Setelah *outliers* ditemukan, selanjutnya mengganti nilai *outliers* dengan metode mean. Menurut Afifah (2020) nilai standar yang biasa digunakan untuk menggantikan nilai *outliers* adalah dengan nilai rata-rata dari seluruh nilai dalam variabel atau kolom tersebut. Tabel 4 contoh data pada variabel TekananDarah untuk menghitung IQR.

**Tabel 4.** Variabel Tekanan Darah

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TekananDarah | 66 | 71 | 45 | 67 | 76 | 70 | 75 | 76 | 77 |

Pada tabel 5 hal pertama yang perlu dilakukan yaitu mengurutkan seluruh nilai dari yang terkecil hingga terbesar sehingga didapatkan nilai seperti pada tabel 5.

**Tabel 5.** Mengurutkan Data

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TekananDarah | 45 | 66 | 67 | 70 | 71 | 75 | 76 | 76 | 77 |

Dari tabel 5 kita mendapatkan nilai median (nilai tengah) yaitu 71. Selanjutnya menghitung kuartil pertama (Q1) dengan menghitung median dari data-data yang berada dibawah median utama, sehingga didapatkan nilai Q1 yaitu 66.5. Lalu menghitung nilai kuartil ketiga (Q3) dengan menghitung median dari data-data yang berada diatas median utama, sehingga didapatkan nilai Q3 yaitu 76. Setelah didapatkan nilai Q1 dan Q3, selanjutnya mencari nilai IQR dengan menghitung selisih antara Q3 dan Q1 menggunakan persamaan 16.

IQR = 76 – 66.5 = 9.5

Langkah selanjutnya mendeteksi *outliers*, menggunakan metode IQR dengan membuat batas atas dan batas bawah berdasarkan persamaan 17 dan 18.

Batas Bawah = 67 – 1.5 × 9.5 = 52,75

Batas Atas = 76 + 1.5 × 9.5 = 90.25

Setelah mendapatkan nilai dari batas atas dan batas bawah, kita dapat melihat pada tabel 5 terdapat nilai yang melewati batas bawah yaitu nilai 45. Nilai tersebut perlu diganti dengan menemukan nilai rata-rata atau mean dari kolom dengan persamaan 13:

Mean = ((45 + 66 + 67 +70 + 71 + 75 + 76 + 76 + 77) / 9) = 69.2

### **4.3.2. *Data Transformation***

Pada tahapan ini dataset dilakukan standarisasi data dengan menggunakan *z-score* agar dimensi skala data yang ada pada dataset ini menjadi lebih rendah dan memiliki skala yang sama yaitu rata-rata 0 (terpusat) dan standar deviasi 1 (Anzihory, 2021). Untuk melihat contoh dataset yang akan dilakukan standarisasi data dapat dilihat pada lampiran 2.

Berikut ini adalah ilustrasi perhitungan untuk mengetahui nilai rata-rata dan deviasi standar pada setiap variabel menggunakan persamaan 13 dan 14.

Merokok:

Mean = 0.5 std = 0.5

Gula Darah:

Mean = 166.9 std = 84.9

Tekanan Darah:

Mean = 75.5 std = 10.0

BMI:

Mean = 26.4 std = 5.1

Usia:

Mean = 40.9 std = 15.6

Lingkar Perut

Mean = 89.5 std = 11.2

Diagnosis Diabetes:

Mean = 0.6 std = 0.5

Mencari nilai *z-score* dari setiap kolom menggunakan persamaan 15.

Id 1 z- Merokok = (1 - 0.5) / 0.5 = 1

Id 1 z-Gula darah = (166 – 166.9) / 84.9 = 0.0

Berikut ini adalah hasil transformasi data menggunakan *z-score normalization* dengan microsoft excel untuk seluruh variabel, dapat dilihat pada tabel 6.

**Tabel 6.** Setelah Transformation Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  diabetes |
| 1 | 1 | 0 | -1 | 0 | -1 | 0 | 1 |
| 2 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |
| 3 | -1 | 0 | 1 | 0 | 1 | 2 | 1 |
| 4 | -1 | -1 | -1 | 0 | -1 | 0 | -1 |
| 5 | -1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 1 | 2 | 2 | 1 | 1 | 0 | 1 |
| 7 | 1 | -1 | 0 | 2 | -1 | 1 | -1 |
| 8 | -1 | 0 | 1 | -1 | -1 | -1 | -1 |
|  |  |  |  |  |  |  |  |

### **4.3.3. *Data Splitting***

Proses selanjutnya adalah membagi data menjadi data latih dan data uji dengan jumlah perbandingan 70:30, 75:25, 80:20, dan 90:10 dari total keseluruhan data dengan menggunakan *library* dari *sklearn* untuk *splitting* data. Proses *ini* dilakukan secara acak untuk menghindari terjadinya bias terhadap data latih dan data uji.

## **4.4. *Modeling***

Algoritma *Support Vector Machine* dan *Random Forest* digunakan untuk pemodelan, dan bahasa pemrograman *Python* serta perpustakaan *scikit-learn* digunakan untuk mengimplementasikan algoritma.

### **4.4.1. Model Klasifikasi SVM**

Langkah pertama dalam mengembangkan model klasifikasi dengan *Support Vector Machine* adalah menentukan nilai parameter dan jenis kernel, khususnya kernel RBF. Data model klasifikasi SVM kemudian digunakan untuk memulai pelatihan SVM. Proses klasifikasi dapat dilanjutkan jika model terbaik untuk data telah dipilih. Studi kasus perhitungan menggunakan model SVM RBF pada lampiran 3.

Karena kelas yang digunakan pada metode SVM adalah -1 dan 1, maka dari hasil standarisasi, akan ditentukan nilai *output* () dengan menjadikan variabel diagnosis diabetes menjadi nilai *output* (), untuk mempermudah perhitungan selanjutnya seluruh label yang terdapat pada tabel diubah menjadi label . untuk variabel merokok, untuk variabel Gula darah, untuk variabel tekanan darah, untuk varibel BMI, untuk variabel Usia, untuk variabel LingkarPerut, dan untuk variabel diagnosis diabetes.

Selanjutnya menghitung nilai *weight* () dan bias (), untuk menghitung nilai *weight* mengunakan persamaan 1. Sehingga didapatkan beberapa persamaan berikut:

Setelah memperoleh persamaan diatas, maka dilakukan pencarian nilai ,, , ,, dan nilai , dengan menjumlahkan persamaan 17 dan 18, 19 dan 20, 21 dan 22, 23 dan 24, 18 dan 20, 18 dan 23, 20 dan 23. Untuk melihat hasil perhitungan persamaan secara keseluruhan dapat dilihat pada lampiran 4.

Hasil penjumlahan dari beberapa persamaan adalah sebagai berikut:

= 1

Selanjutnya menentukan nilai *hyperplane*, maka digunakan persamaan 2 sehingga diperoleh *hyperplane* sebagai berikut:

### **4.4.2. Model Klasifikasi *Random Forest***

Dilanjutkan ke pembuatan model selanjunya yaitu *Random Forest*. Contoh studi kasus perhitungan menggunakan model *Random Forest* pada lampiran 5. Asumsikan bahwa ada 5 pohon yang akan dibuat. Karena dataset tidak memiliki banyak tupel, maka pohon utama dapat dibuat dengan memasukkan semua karakteristik dan menghilangkan baris nomor 5. Indeks Gini akan digunakan untuk mencari akar atau atribut terbaik dalam kasus ini. Tujuan dari tugas ini adalah mencari atribut terbaik dari berbagai pohon yang dibangun (*best split*). Contoh pohon pertama dapat dilihat pada tabel 7:

**Tabel 7.** Contoh Pohon Pertama

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  diabetes |
| 1 | 2 | 0 | -1 | 0 | -1 | 0 | 1 |
| 2 | -1 | 1 | -1 | 0 | -1 | 1 | -1 |
| 3 | -1 | 0 | 1 | 0 | 1 | 2 | 1 |
| 4 | -1 | -1 | 1 | 0 | 0 | 0 | -1 |
| 6 | -1 | -1 | -1 | 0 | 1 | 0 | -1 |
| 7 | 2 | -1 | 0 | 2 | -1 | -1 | -1 |
| 8 | -1 | 0 | 1 | -2 | 1 | -1 | -1 |

Ubah menjadi tabel kontingensi, dapat dilihat pada tabel 8:

**Tabel 8.** Kontingensi Pertama

|  |  |  |  |
| --- | --- | --- | --- |
| Merokok | Diagnosis | | |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 1 | 2 |
| < 0 | 1 | 4 | 5 |
| Total | 2 | 5 | 7 |

Selanjutnya dari masing-masing kelas atribut yang tersedia, gunakan persamaan 9 untuk menghitung indeks Gini. sehingga mendapatkan gini indeks:

Dengan menggunakan persamaan 10, carilah Gini *split* setelah mendapatkan semua indeks Gini. Sehingga didapatkan hasil sebagai berikut:

Langkah terakhir adalah memilih akar terbaik. Atribut gula darah merupakan yang terbaik berdasarkan perhitungan di atas karena memiliki indeks Gini *split* yang paling rendah. Untuk melihat hasil perhitungan pohon dan kontingensi secara keseluruhan dapat dilihat pada lampiran 4.

## **4.5. *Evaluation***

Setelah melakukan pengklasifikasian dan didapatkan hasil prediksi. Kinerja algoritma akan dievaluasi untuk membandingkan nilai prediksi terbaik dengan menggunakan *confusion matrix* dan ROC-AUC. Pada umumnya, dengan membandingkan nilai prediksi algoritma klasifikasi dengan nilai variabel target data pengujian sebagai data aktual, kinerja algoritma klasifikasi dapat diukur.

### **4.5.1. *Confusion Matrix***

Setelah melakukan analisis metode klasifikasi dan memperoleh hasil prediksi. Nilai prediksi terbaik kemudian dibandingkan menggunakan kinerja algoritma *confusion matrix*. Contoh perhitungan *confusion matrix* seperti pada tabel 9.

**Tabel 9.** Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Data prediksi SVM | Data aktual | |
| Diabetes | Tidak diabetes |
| Diabetes | 59 | 8 |
| Tidak diabetes | 12 | 146 |
| Data prediksi *Random Forest* | Data aktual | |
| Diabetes | Tidak diabetes |
| Diabetes | 64 | 3 |
| Tidak diabetes | 9 | 149 |

Berdasarkan hasil *output* diatas dapat diketahui bahwa prediksi diabetes dengan nilai sebenarnya adalah 59 untuk SVM dan 64 untuk *Random Forest* (*true positif*), prediksi diabetes dengan nilai sebenarnya tidak diabetes adalah 8 untuk SVM dan 3 untuk *Random Forest* (*false positif*), prediksi tidak diabetes dengan nilai sebenarnya diabetes adalah 12 untuk SVM dan 9 untuk *Random Forest* (*false negatif*), prediksi tidak diabetes dengan nilai sebenarnya tidak diabetes adalah 146 untuk SVM dan 149 untuk *Random Forest* (*true negatif*). Dari hasil tersebut dapat dihitung nilai akurasi dari hasil klasifikasinya dengan persamaan 11:

*Accuracy* SVM (%) =

*Accuracy Random Forest* (%) =

Berdasarkan *confusion matrix* hasil model yang di peroleh nilai akurasi yang tinggi yaitu menghasilkan nilai akurasi sebesar 91% dan 94%, dimana akurasi ini bisa dikatakan baik.

### **4.5.2. ROC-AUC**

Dengan menggunakan *area under the ROC curve* (AUC) untuk membandingkan model SVM dan *Random Forest* yang memiliki nilai AUC terbesar, nilai akurasi model sebelumnya sebesar 91% dan 94% menggunakan data uji akan ditinjau kembali.

## **4.6. *Deployment***

Tahap ini merupakan tahapan terakhir yang akan dilakukan, pada tahap ini aplikasi website akan dibuat agar lebih mudah digunakan oleh pihak yang berkepentingan dan dapat digunakan untuk memprediksi penyakit diabetes berdasarkan analisis prediksi pola yang dihasilkan oleh algoritma terbaik dari hasil komparasi. Pembuatan aplikasi website menggunakan editor teks *Visual Studio Code* dengan *framework* *Streamlit* dapat dilihat pada gambar 5.

A screenshot of a computer

Description automatically generated

**Gambar 5.** Implementasi Pada Visual Studio Code

# **BAB V**

# **HASIL DAN PEMBAHASAN**

## **5.1. Hasil**

### **5.1.1. Pemanggilan *Library* dan *Package***

Pemanggilan *library* dan *package* ini dilakukan agar *Python* selaku bahasa pemrograman dapat menjalankan perintah-perintah untuk proses pada *machine learning*. Berikut *library-library* yang digunakan pada penelitian ini 1) *numpy*, sebagai modul numerik untuk menganalisa data dalam bentuk *array* atau *matriks*. 2) *matplotlib* dan *seaborn*, yaitu untuk visualisasi data dalam bentuk plot dan diagram 3) *pandas*, untuk mengolah dan membaca data yang digunakan didalam program. 4) *sklearn*, yaitu untuk membangun model *machine learning*. Pemanggilan *library* dan *package* dapat dilihat pada gambar 6.

A screenshot of a computer

Description automatically generated

**Gambar 6.** Pemanggilan Library

### **5.1.2. Deskripsi Data**

Dataset pada penelitian ini dibagi menjadi dua, yaitu data *train* dan data *test* dengan 4 skenario data *training* dan data *testing*. Sebelum membagi data menjadi dua, dataset terlebih dahulu dilakukan *data cleaning* dan *data* *transformation*, karena terdapat beberapa data yang bernilai *null* dan agar data sesuai seperti yang diharapkan. Informasi mengenai dataset dapat dilihat pada gambar 7.

A screenshot of a computer

Description automatically generated

**Gambar 7.** Deskripsi Data

Berdasarkan Gambar 7 diketahui bahwa data terdiri dari 1124 sampel dengan jumlah variabel sebanyak 8 dan terdiri dari 2 kelas, yaitu 0 untuk terdiagnosis normal dan 1 untuk terdiagnosis diabetes. Jumlah sampel terdiagnosis

normal sebanyak 787 pasien dan jumlah sampel terdiagnosis diabetes sebanyak 337. Pada variabel Usia dapat diketahui usia terendah yaitu berusia 13 tahun dan tertua berusia 76 tahun, dapat diketahui juga rata-rata pada dataset yaitu berusia 36 tahun. Pada variabel JenisKelamin dapat diketahui jenis kelamin laki-laki dan perempuan memiliki jumlah yang hampir sama banyak, 49% untuk jenis kelamin laki-laki dan 51% untuk jenis kelamin perempuan. Pada variabel Merokok hanya sekitar 25% dari keseluruhan data yang merokok dan dapat diketahui terdapat 6 data yang bernilai *null*. Pada variabel TekananDarah diketahui terdapat 6 data yang bernilai null. Pada variabel BMI dapat diketahui rata-rata pasien memiliki *body mass index* yaitu 22.9. Pada variabel LingkarPerut diketahui rata-rata pasien memiliki lingkar perut sebesar 80 cm. Pada variabel Glukosa terdapat pasien dengan kadar gula darah yaitu 71 dan tertinggi yaitu 593. Terakhir pada variabel HasilDiagnosis sekitar 25% dari keseluruhan data terdiagnosis diabetes. Selanjutnya visualisasi *barchart* pengelompokan hasil diagnosis berdasarkan jenis kelamin seperti pada gambar 8.

A screenshot of a computer

Description automatically generated

**Gambar 8.** Barchart Jenis Kelamin

Gambar 8 menunjukkan bahwa pasien wanita sedikit lebih banyak terkena diabetes dibandingkan pasien pria. Kemudian dari dataset tersebut dapat dilihat visualisasi kepadatan probabilitas variabel dari dataset Puskesmas Citapen seperti pada gambar 9.

**A graph of a curve

Description automatically generated with medium confidence**

**Gambar 9.** Visualisasi KDE Plot

Berdasarkan gambar 9, diketahui bahwa sampel pasien dari dataset diabetes berada pada rentang usia belasan tahun sampai usia 70 tahun. Pasien rentang usia belasan hingga 40 tahun paling sedikit terserang penyakit diabetes, sedangkan pasien diantara rentang usia 40 hingga 70 tahun cenderung lebih banyak terserang penyakit diabetes, hal ini disebabkan oleh berkurangnya kemampuan organ pankreas dalam tubuh untuk memproduksi hormon insulin dan peningkatan konsentrasi glukosa karena proses penuaan.

Kekuatan fitur (variabel) pada target (kelas) ditentukan dengan menggunakan korelasi fitur. Jika variabel Merokok, Umur, Jenis Kelamin, Tekanan Darah, IMT, Lingkar Perut, dan Glukosa mempunyai nilai yang tinggi terhadap Hasil Diagnosis, maka variabel tersebut sangat berpengaruh, sedangkan jika lebih rendah maka kurang atau tidak berpengaruh. *Feature correlations* dapat dilihat pada gambar 10.

**A screenshot of a green and white grid

Description automatically generated**

**Gambar 10.** Feature Corellations

Dapat diketahui dari Gambar 10, bahwasanya jika hasil korelasi mendekati nilai 1, maka korelasi tersebut dikatakan baik, namun jika mendekati nilai -1 maka korelasi dikatakan buruk. Dapat dilihat dari hasil visualisasi di atas, yang memiliki nilai korelasi baik terdapat pada variabel “Glukosa” dan “LingkarPerut” yaitu memiliki nilai hasil korelasinya 0.58 dan 0.47, adapun nilai korelasi yang buruk terdapat pada variabel “JenisKelamin”, yaitu memiliki nilai -0.0031, jika hasil korelasi yang berbentuk diagonal menyamping tersebut itu merupakan korelasi variabel dengan dirinya sendiri.

Lalu dilakukan visualisasi *categorical plot* seperti pada gambar 11, plot dilakukan pada setiap variabel berdasarkan hasil diagnosisnya sehingga membentuk suatu pola persebaran data pada masing-masing variabel. Pola yang tampak jelas dapat dilihat pada *categorical plot* terhadap Glukosa, LingkarPerut, dan BMI, pesebaran data pasien yang terdiagnosis diabetes rata-rata memiliki nilai yang tinggi pada variabel-variabel tersebut. Hal tersebut selaras dengan *correlations features* pada gambar 10, dimana ketiga variabel tersebut memiliki nilai yang besar dalam mempengaruhi seseorang berpotensi memiliki penyakit diabetes.

A graph of blue and orange dots

Description automatically generatedA graph of different colored dots

Description automatically generatedA graph of blue and orange dots

Description automatically generatedA graph of different colored dots

Description automatically generatedA graph of a number of blue and orange dots

Description automatically generated

**Gambar 11.** Visualisasi categorical plot

### **5.1.3. *Data* *Preprocessing***

#### **5.1.3.1. Data Cleaning**

Dalam proses ini dimana data sebelumnya yang sudah didapatkan akan dilakukan tahap *data* *preprocessing* pada *Python*. Dimana untuk proses *preprocessing* data ini melakukan *data cleaning* (mengisi data yang kosong atau data bernilai *null*). Pada gambar 12 dapat dilihat terdapat 6 data dengan nilai *null* pada variabel Merokok dan TekananDarah. Untuk mengisi data yang kosong pada variabel tersebut, dilakukannya perhitungan nilai modus pada variabel merokok dan perhitungan mean pada variabel tekanan darah dan dilanjutkan dengan mengisi data bernilai *null* dengan nilai yang telah didapatkan. Untuk proses *preprocessing* tersebut bisa dilihat pada gambar 12.

A graph with blue lines

Description automatically generated A graph with text on it

Description automatically generated

**Gambar 12.** Sebelum dan sesudah proses pengisian data null

Tahap selanjutnya mendeteksi nilai *outliers* pada masing-masing variabel dengan menggunakan visualisasi *boxplot* dan metode *Interquartile Range* untuk mendapatkan data *outlier*. Pada variabel BMI, TekananDarah, LingkarPerut, dan Glukosa terdapat beberapa data *outlier* yang melewati batas atas dan batas bawah *boxplot*. Selanjutnya dicari data-data tersebut dengan metode IQR, sehingga dapat dilakukannya *data cleaning* dengan pengisian data menggunakan perhitungan mean gambar 13.

A graph of a number of people

Description automatically generated with medium confidenceA graph of a number of boxes

Description automatically generated with medium confidence

**Gambar 13.** Boxplot sebelum dan setelah Preprocessing

#### **5.1.3.2. Data Transformation**

Setelah dataset bebas dari data *outliers*, selanjutnya dilakukan transformasi data dengan menggunakan *z-score* dalam bentuk *distribution plot* seperti pada gambar 14. Dapat dilihat dengan jelas bahwa terdapat perbedaan kualitas data sebelum preprocessing dilakukan dan setelah preprocessing selesai. Perbedaan paling jelas yang terlihat adalah nilai tengah data saat prapemrosesan. Proses *data transformation* bisa dilihat pada gambar 14.

A graph of a line

Description automatically generated with medium confidenceA graph of a line graph

Description automatically generated with medium confidence

**Gambar 14.** Normalisasi Z-Score

#### **5.1.3.3. Data Splitting**

Dataset pada penelitian ini dibagi menjadi dua, yaitu data *train* dan data *test* dengan 4 skenario presentase *spliting* yaitu, skenario pertama 70% data *train* dan 30% data *test*, skenario kedua 75% data *train* dan 25% data *test,* skenario ketiga 80% dan 20%, danskenario keempat 90% data *train* dan 10% data *test*. Setelah pembagian data, pada skenario pertama didapatkan 786 data latihdan 338 data uji, pada skenario kedua didapatkan 843 data latih dan 281 data uji, pada skenario ketiga didapatkan 899 data latih dan 225 data uji, dan pada skenario keempat didapatkan 1011 data latih dan 113 data uji.Pembagian data dapat dilihat pada gambar 15.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Gambar 15.** Proses Data Splitting

### **5.1.4. *Modeling***

Dalam tahapan ini, data sebelumnya yang telah diproses dimasukan ke dalam model algoritma SVM dan *Random Forest,* lalu diproses untuk mendapatkan hasil klasifikasi terbaik diantara algoritma SVM dan *Random Forest.* Pemodelan dapat dilihat pada gambar 16.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Gambar 16.** Modeling SVM dan Random Forest

### **5.1.5. *Evaluation***

Visualisasi *confusion matrix* untuk prediksi terhadap data *testing* dengan model SVM didapatkan 59 untuk *true positive* (TP), kemudian 8 untuk data *false positive* (FP), 12 untuk data *false negative* (FN), dan 146 untuk data *true negative* (TN). Untuk model *Random Forest* didapatkan 64 untuk *true positive* (TP), kemudian 3 untuk data *false positive* (FP), 9 untuk data *false negative* (FN), dan 149 untuk data *true negative* (TN). Selanjutnya dilakukan perhitungan nilai akurasi dari model klasifikasi SVM dan *Random Forest* yang dibuat menggunakan *sklearn.metrics* dengan mengimport *accuracy,* didapatkan hasil dari perhitungan *matrix* SVM yaitu sebesar 0.90 dan *Random Forest* yaitu sebesar 0.95. Adapun hasil dari visualisasi *confusion matrix* bisa dilihat pada gambar 17.

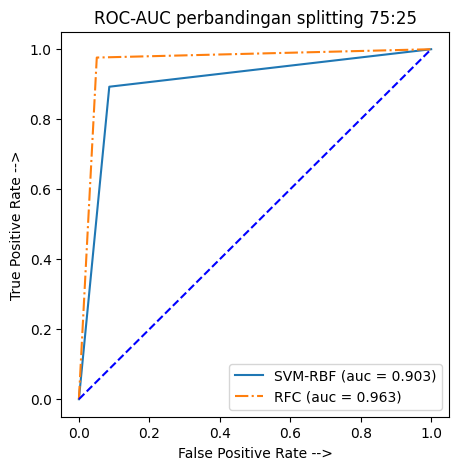
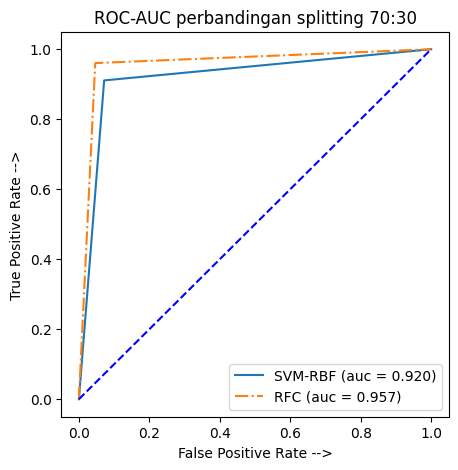
A screenshot of a computer

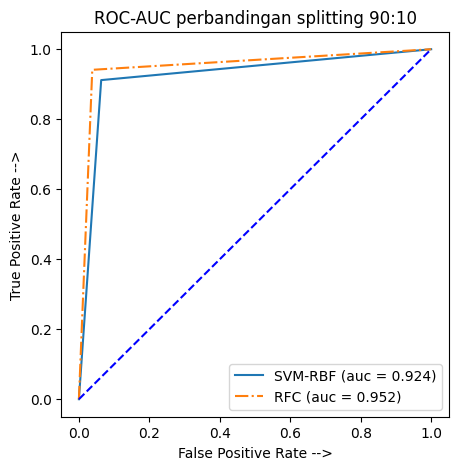
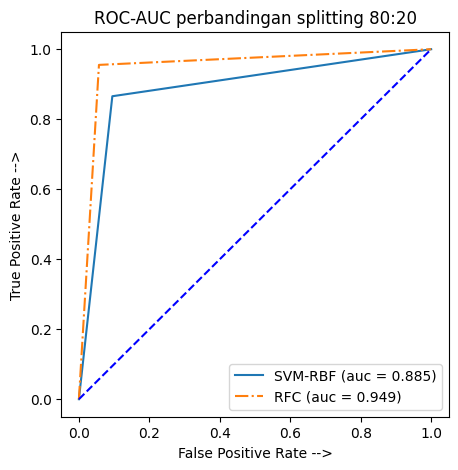
Description automatically generatedA screenshot of a computer

Description automatically generated

**Gambar 17.** Confusion Matrix SVM dan Random Forest

*Area under curve* ROC akan digunakan untuk menguji nilai akurasi pada data uji yang diperoleh dari model sebelumnya dimana kurva dibuat berdasarkan pada perhitungan dengan *confusion matrix* yaitu diantaranya *false positif rate* (FPR) berdasarkan persamaan 19 dan dengan *true positif rate* (TPR) pada persamaan 20. Hasil AUC yang didapatkan untuk model klasifikasi SVM sebesar 0.90 dan nilai untuk *Random Forest* sebesar 0.96. Gambar 18 hasil dari AUC untuk membandingkan model klasifikasi SVM dan *Random Forest.*





**Gambar 18.** ROC-AUC

Pada Gambar 18 dapat dilihat bahwa model *Random Forest* dengan garis berwarna birumenghasilkan performa terbaik pada *spliting* data 75:25 dan dengan nilai AUC 0.96 (mendekati 1) dibandingkan dengan model SVM dengan nilai AUC terbaik 0.95 pada *splitting* data 70:30. Berdasarkan pengujian dataset pasien di Puskesmas Citapen menggunakan 2 model algoritma yaitu SVM dan *Random Forest,* diperoleh hasil penelitian eksperimen yang dapat disimpulkan ke dalam tabel 10.

**Tabel 10.** Perbandingan Model

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rasio | 70:30 | | 75:25 | | 80:20 | | 90:10 | |
| Perbandingan | SVM | *Random Forest* | SVM | *Random Forest* | SVM | *Random Forest* | SVM | *Random Forest* |
| Akurasi | 0.92 | 0.95 | 0.90 | 0.95 | 0.89 | 0.94 | 0.92 | 0.95 |
| AUC | 0.92 | 0.95 | 0.90 | 0.96 | 0.88 | 0.94 | 0.92 | 0.95 |
| *Recall* | 0.92 | 0.96 | 0.90 | 0.96 | 0.92 | 0.95 | 0.89 | 0.95 |

Dari hasil *confusion matrix, accuracy,* dan kurva ROC-AUC model *Random Forest* dipilih untuk digunakan ke tahap *deployment* dikarenakan lebih unggul dibandingkan dengan model SVM pada *splitting* data 75:25 memiliki akurasi 95% dan AUC sebesar 96%.

### **5.1.6. *Deployment***

Setelah model *Random Forest* dan pembagian data dengan skenario 75% data latih dan 25% data ujiterpilih sebagai model terbaik, selanjutnya dibuatkan aplikasi *website* dengan menggunakan *framework Streamlit* dengan teks editor *Visual Studio Code* agar pemodelan prediksi penyakit diabetes dapat diakses lebih mudah. Dalam aplikasi ini terdapat halaman beranda yang merujuk pada halaman awal yang ditampilkan di *browser* web seperti pada gambar 19.

A screenshot of a computer

Description automatically generated

**Gambar 19.** Halaman Beranda

Pada halaman beranda terdapat informasi mengenai penjelasan aplikasi website yang telah dibuat dan pilihan menu yang berada pada sidebar. Pada pilihan halaman terdapat 3 halaman selain beranda, yaitu halaman Informasi, Halaman Prediksi, dan halaman Visualisasi. Tampilan halaman informasi seperti pada gambar 20.

A screenshot of a computer

Description automatically generated

**Gambar 20.** Halaman Informasi

Halaman Informasi menjelaskan penyakit diabetes melitus dan faktor-faktor apa saja yang mempengaruhi kemungkinan seseorang terdiagnosis penyakit diabetes. Tampilan halaman prediksi seperti pada gambar 21.

A screenshot of a computer

Description automatically generated

**Gambar 21.** Halaman Prediksi

Dapat dilihat pada gambar 21 halaman Prediksi berisi form-form yang dapat diisi untuk melakukan prediksi penyakit diabetes, jika semua form telah diisi sesuai data dan mengklik tombol prediksi, maka akan muncul diagnosis seseorang terkena penyakit diabetes atau tidak.

Halaman terakhir yaitu halaman Visualisasi, pada halaman ini terdapat visualisasi data dari model yang telah dibuat sebelumnya yaitu visualisasi *correlations features* dan KDE plot seperti pada gambar 22.

A screenshot of a computer

Description automatically generated

**Gambar 22.** Halaman Visualisasi

## **5.2. Uji Coba**

Uji coba dilakukan untuk mengevaluasi efektifitas model pada *website*. Pada penelitian ini uji coba dilakukan untuk memastikan bahwa model dapat menghasilkan hasil yang diinginkan pada *website.* Uji coba menggunakan 10 data teratas dari dataset. Berdasarkan hasil uji coba, model dapat memprediksi keseluruhan data dengan tepat sesuai dengan data aktual. Hasil uji coba dapat dilihat pada tabel 11 dan lampiran 7.

**Tabel 11.** Hasil uji coba pada website

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Usia** | **Merokok** | **Tekanan Darah** | **BMI** | **Lingkar Perut** | **Glukosa** | **Aktual** | **Prediksi** | **Ket.** |
| 47 | 0 | 84 | 18.7 | 70 | 86 | 0 | 0 | Sesuai |
| 22 | 1 | 71 | 26.9 | 90 | 113 | 0 | 0 | Sesuai |
| 58 | 0 | 79 | 15.8 | 60 | 85 | 0 | 0 | Sesuai |
| 35 | 1 | 59 | 22.8 | 82 | 129 | 1 | 1 | Sesuai |
| 31 | 1 | 81 | 26.1 | 93 | 128 | 1 | 1 | Sesuai |
| 60 | 0 | 66 | 26.4 | 87 | 158 | 1 | 1 | Sesuai |
| 35 | 1 | 64 | 22.8 | 84 | 108 | 0 | 0 | Sesuai |
| 34 | 1 | 84 | 21.2 | 78 | 101 | 0 | 0 | Sesuai |
| 38 | 0 | 70 | 20.1 | 74 | 108 | 0 | 0 | Sesuai |
| 16 | 0 | 65 | 19.7 | 70 | 104 | 0 | 0 | Sesuai |

# **BAB VI**

# **KESIMPULAN DAN SARAN**

## **6.1. Kesimpulan**

Dari hasil *corellation features,* dapat diketahui bahwa variabel yang sangat mempengaruhi terhadap hasil diagnosis yaitu variabel Glukosa dengan memiliki nilai korelasi terbesar yaitu 0.58, selanjutnya yaitu variabel LingkarPerut dengan nilai 0.47, variabel BMI dengan nilai 0.43, variabel Usia dengan nilai 0.34, variabel TekananDarah dengan nilai 0.13, dan variabel Merokok dengan nilai 0.1. Adapun pada variabel JenisKelamin memiliki nilai -0.0031 yang menandakan bahwa variabel tersebut sangat tidak berpengaruh terhadap hasil diagnosis. Pada visualisasi KDE plot, dapat diketahui bahwa pasien diantara rentang usia 40 hingga 70 tahun cenderung lebih banyak terserang penyakit diabetes, Hal ini disebabkan oleh berkurangnya kemampuan organ pankreas dalam tubuh untuk memproduksi hormon insulin dan peningkatan konsentrasi glukosa karena proses penuaan.

Klasifikasi dilakukan menggunakan algoritma SVM dan *Random Forest*. Sebelumnya, pada algoritma SVM dilakukan pengujian akurasi pada kernel RBF, Linear, Sigmoid, dan Polynomial, hasil pengujian tersebut didapatkan kesimpulan bahwa kernel RBF dipilih untuk dibandingkan dengan *Random Forest* dikarenakan kernel RBF memiliki akurasi yang paling stabil diantara kernel yang lain. Tahap pertama dilakukan yaitu tahap *cleaning* dan *transformation* data*,* setelah itu dilakukan pembagian data latih dan data uji dengan 4 skenario, yaitu 70:30, 75:25, 80:20, dan 90:10. Dari hasil evaluasi model, skenario 75:25 merupakan hasil terbaik untuk diterapkan pada model karena memiliki tingkat akurasi dan nilai AUC yang lebih tinggi dibandingkan dengan 3 skenario lainnya pada algoritma *Random Forest*.

Model pada penelitian ini dievaluasi dengan 2 metode, yaitu dengan *confusion matrix* dan ROC-AUC. Sehingga diketahui bahwa algoritma SVM didapatkan hasil akurasi 90% dan nilai AUC 0.90. Pada algoritma *Random Forest* didapatkan hasil akurasi 95% dan nilai AUC 0.96. Algoritma Random Forest mengungguli algoritma SVM dalam hal menghasilkan nilai klasifikasi, seperti yang ditunjukkan oleh perbandingan keduanya. Selanjutnya *Random Forest* dipilih untuk dibuatkan sistem aplikasi web menggunakan *framework* *Streamlit*.

Berdasarkan serangkaian proses yang telah dilakukan dalam penelitian ini, maka dapat dapat disimpulkan bahwa penelitian ini berhasil membangun sebuah model terbaik dan aplikasi web untuk prediksi penyakit diabetes menggunakan algoritma *Random Forest*.

## **6.2. Saran**

Berikut adalah beberapa saran agar penelitian ini dapat diperbaiki dan dikembangkan untuk penelitian selanjutnya:

1. Menambahkan data prediksi dalam bentuk file.
2. Mengimplementasikan teknik evaluasi *k-fold cross-validation* untuk mengukur kinerja model agar lebih akurat.
3. Menambahkan prediksi pra-diabetes pada hasil model.

# **DAFTAR PUSTAKA**

Afifah, L. (2020). Cara Menangani *Missing Values* di *Project Data Science*. Diakses pada 12 Januari 2024, dari <https://ilmudatapy.com/menangani-missing-values/>

Alhamdhani, M. A. Y. (2022). Hasil Penelitian Prediksi Harga *Cryptocurrency Ethereum* Menggunakan Algoritma *Deep Learning*.

Anzihory, E. (2021). Normalisasi vs Standarisasi. Diakses pada 24 Januari 2024, dari <https://anzihory.medium.com/normalisasi-vs-standarisasi-101093633e18>

Apriyani, H., & Kumiati. (2020). Perbandingan Metode *Naïve Bayes* Dan *Support Vector Machine* Dalam Klasifikasi Penyakit Diabetes Melitus. In *Journal of Information Technology Ampera* (Vol. 1, Issue 3). <https://journal-computing.org/index.php/journal-ita/index>

Devella, S., Rahmawati, F. N., & Yohannes. (2020). Implementasi *Random Forest* Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT. *Jurnal Teknik Informatika Dan Sistem Informasi*, *7*(2). http://jurnal.mdp.ac.id

Fauzi, A., Supriyadi, R., & Maulidah, N. (2020). Deteksi Penyakit Kanker Payudara dengan Seleksi Fitur berbasis *Principal Component Analysis* dan *Random Forest*. In *Jurnal* (Vol. 2, Issue 1). <http://ejournal.bsi.ac.id/ejurnal/index.php/infortech96>

Hovi, S., Hadiana, A., & Umbara, F, R. (2022). Prediksi Penyakit Diabetes Menggunakan Algoritma *Support Vector Machine* (SVM). *INFORMASI ARTIKEL ABSTRAK* (Vol. 4, Issue 1). <https://e-journal.unper.ac.id/index.php/informatics>

Info, R. (2023). Pengertian Jangkauan Interkuartil dan Cara Menghitungnya dalam Matematika. Diakses pada 23 Januari 2024, dari https://kumparan.com/ragam-info/pengertian-jangkauan-interkuartil-dan-cara-menghitungnya-dalam-matematika-21IEl4rLaGw/full

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

Kardono, D. Y., Pranoto, Y. M., & Setyati, E. (2023). Prediksi Kecocokan Jurusan Siswa SMK Dengan *Support Vector Machine* dan *Random Forest*. *Teknika*, *12*(1), 11–17. <https://doi.org/10.34148/teknika.v12i1.567>

Leonardo, R., Pratama, J., & Chrisnatalis. (2020). *Address*: Universitas Prima Indonesia, Teknik Informatika. *Jl. Sekip Sei Kambing Medan*, *123*. <https://archive.ics.uci.edu/ml/datasets/Bank+Ma>

Lestandy, M., Syafa’ah, L., & Faruq, A. (2020). *Classification of potential blood donors using machine learning algorithms approach*. *Jurnal Teknologi Dan Sistem Komputer*, *8*(3), 217–221. <https://doi.org/10.14710/jtsiskom.2020.13619>

Maulid, R. (2022). Perbandingan *Machine Learning* *Python* Antara Matplotlib vs *Seaborn*. Diakses pada 20 Mei 2024, dari https://dqlab.id/perbandingan-machine-learning-python-antara-matplotlib-vs-seaborn

Merianti, D. (2020). Implementasi Metode *Support Vector Machine* Dan *Random Forest* Pada Data Ekspresi Gen *Microarray*.

Miftah, S. (2021). Belajar *Python* Mengenal Pandas dan Series untuk Meningkatkan Kompetensi Data. Diakses pada 20 Mei 2024, dari https://dqlab.id/belajar-python-mengenal-pandas-dan-series-untuk-meningkatkan-kompetensi-data

Nugraha, R., Yuwono, E., Prasetyohadi, L., Arief, Y. B., Patria, H., & Teknologi Sepuluh Nopember Jl, I. (2022). Analisis Konsumsi Energi Listrik Pelanggan Dan Biaya Pokok Produksi Penyediaan Energi Listrik dengan *Machine Learning*. In *Jurnal Sains Komputer & Informatika (J-SAKTI* (Vol. 6, Issue 1).

Nurkhaliza A., & Wijayanto A. (2022). Mitigasi dan Kesiapsiagaan Bencana (IMKB) Satuan Kerja BPS di Indonesia Tahun 2020. *Maret*, *7*(1), 2622–4615. https://doi.org/10.32493/informatika.v7i1.16117

Patil, R., Tamane, S., Rawandale, S. A., & Patil, K. (2022). *A modified mayfly-SVM approach for early detection of type 2 diabetes mellitus*. *International Journal of Electrical and Computer Engineering*, *12*(1), 524–533. <https://doi.org/10.11591/ijece.v12i1.pp524-533>

Rahman, I. F. (2020) Implementasi Metode SVM, MLP Dan XGBoost Pada Data Ekspresi Gen.

Rina. (2023). Memahami *Confusion Matrix: Accuracy, Precision, Recall, Specificity, dan F1-Score* untuk Evaluasi Model Klasifikasi. Diakses pada 13 Oktober 2023, dari https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf

Pittara. (2024). Diabetes. Diakses pada 06 Mei 2024, dari https://www.alodokter.com/diabetes

Setiawan, S. (2020). Mendeteksi *Univariate Outliers* dengan Metode IQR (*Python*). Diakses pada 13 Januari 2023, dari <https://stevkarta.medium.com/mendeteksi-univariate-outliers-dengan-metode-iqr-python-3adfad87de82>

Tineges, R. (2021). Mengenal Matplotlib untuk Visualisasi Data dengan Python. Diakses pada 20 Mei 2024, dari https://dqlab.id/mengenal-matplotlib-untuk-visualisasi-data-dengan-python

Trivusi. (2022). Penjelasan Lengkap Algoritma *Support Vector Machine* (SVM). Diakses pada 25 Mei 2023, dari <https://www.trivusi.web.id/2022/04/algoritma-svm.html>

Uswatun, L. (2021). Python : Kenali NumPy Array dalam *Python*. Diakses pada 20 Mei 2024, dari <https://dqlab.id/kenali-numpy-array-dalam-python>

Widya, J. (2020). Belajar *Machine Learning* Dengan *Library Python* : *Scikit-Learn*. Diakses pada 20 Mei 2024, dari https://dqlab.id/belajar-machine-learning-dengan-library-python-scikit-learn

# **LAMPIRAN**

**Lampiran 1.** Definisi Variabel Penelitian

|  |  |
| --- | --- |
| Nama Variabel | Definisi Operasional Variabel |
| Tekanan Darah | Tekanan darah adalah ukuran yang dapat menentukan seberapa kuat jantung mengalirkan darah ke seluruh tubuh. |
| Kadar Glukosa | Berapa banyak gula atau glukosa dalam darah. |
| Index Massa Tubuh | Perbandingan tinggi badan dan berat badan seseorang merupakan ukuran yang digunakan untuk mengetahui status gizinya. |
| Usia | Satuan waktu yang mengukur keberadaan suatu benda atau makhluk. |
| Merokok | Seseorang yang dengan sengaja menghisap tembakau yang telah dilinting dan biasanya dibungkus dengan kertas, daun, atau kulit jagung. |
| Lingkar Perut | Pengukuran yang dilakukan di sekitar perut setinggi pusar. |
| Hasil | Sesuatu yang dihasilkan oleh usaha (diciptakan, dijadikan). |

**Lampiran 2.** Sebelum Transformation Dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  Terkena diabetes |
| 1 | 1 | 166 | 63 | 23.6 | 33 | 93 | 1 |
| 2 | 1 | 119 | 68 | 21.7 | 51 | 78 | 0 |
| 3 | 0 | 173 | 82 | 26.1 | 54 | 108 | 1 |
| 4 | 0 | 88 | 64 | 24.1 | 24 | 86 | 0 |
| 5 | 0 | 235 | 82 | 23.9 | 60 | 96 | 1 |
| 6 | 1 | 338 | 92 | 27.9 | 55 | 85 | 1 |
| 7 | 1 | 123 | 71 | 31.6 | 26 | 97 | 0 |
| 8 | 0 | 135 | 82 | 22.3 | 24 | 73 | 0 |

**Lampiran 3.** Contoh studi kasus dataset SVM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  diabetes |
| 1 | 1 | 0 | -1 | 0 | -1 | 0 | 1 |
| 2 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |
| 3 | -1 | 0 | 1 | 0 | 1 | 2 | 1 |
| 4 | -1 | -1 | -1 | 0 | -1 | 0 | -1 |
| 5 | -1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 1 | 2 | 2 | 1 | 1 | 0 | 1 |
| 7 | 1 | -1 | 0 | 2 | -1 | -1 | -1 |
| 8 | -1 | 0 | 1 | -1 | -1 | -1 | -1 |

**Lampiran 4.** Perhitungan Persamaan SVM

(23)

(24)

Menjumlahkan persamaan (1) dan (2)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

=1

Menjumlahkan persamaan (3) dan (4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

=1

Menjumlahkan persamaan (5) dan (6)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

=1

Menjumlahkan persamaan (7) dan (8)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

-2 -2 = 2

- + - = 2

= - 2 + 1 – 1 = - 2

Menjumlahkan persamaan (2) dan (5)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

+ =1

= 1 – 1 = 0

Menjumlahkan persamaan (3) dan (7)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

- + + + = 1

= 1 +1 – 1 – 0= 1

Menjumlahkan persamaan (7) dan (4)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_+

– - – =1

= - 1 – 1 +1 + - 2 = - 3

Selanjutnya menguji data uji yang akan dilakukan pada *hyperplane*, yaitu data , , , , , dan , dimana masing-masing data sebanyak 2 seperti pada tabel Data Uji SVM:

**Tabel.** Data Uji SVM

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id |  |  |  |  |  |  |  |
| 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 |
| 2 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |

Persamaan kelas dengan nilai digunakan untuk melakukan klasifikasi pada data uji pada tabel. *sign* tersebut digunakan untuk memeriksa hasil perhitungan yang dilakukan terhadap data uji. Jika nilainya lebih besar dari nol maka bernilai 1, sedangkan nilainya lebih kecil sama dengan nol akan memperoleh nilai -1. Hasil pengujian data dapat dilihat pada tabel Fungsi Klasifikasi Pada Data Uji:

**Tabel.** Fungsi Klasifikasi Pada Data Uji

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Data uji | | | | | | | Fungsi klasifikasi |
|  |  |  |  |  |  |  |  |
| 1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 |  |
| 2 | 1 | -1 | -1 | -1 | 1 | -1 | -1 |  |

Pada tabel diperlihatkan hasil klasifikasi data dari perhitungan yang telah dilakukan. Dari hasil klasifikasi data, diperoleh nilai +1 dan -1. +1 merupakan pasien yang terdiagnosis diabetes. -1 merupakan pasien yang tidak terdiagnosis diabetes. Hasil pengujian data tersebut sesuai dengan hasil diagnosis yang pada variabel y.

**Lampiran 5.** Perhitungan Pohon dan Kontingensi Random Forest

**Tabel.** Contoh Studi Kasus

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  diabetes |
| 1 | 2 | 0 | -1 | 0 | -1 | 0 | 1 |
| 2 | -1 | 1 | -1 | 0 | -1 | 1 | -1 |
| 3 | -1 | 0 | 1 | 0 | 1 | 2 | 1 |
| 4 | -1 | -1 | 1 | 0 | 0 | 0 | -1 |
| 5 | -1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 6 | -1 | -1 | -1 | 0 | 1 | 0 | -1 |
| 7 | 2 | -1 | 0 | 2 | -1 | -1 | -1 |
| 8 | -1 | 0 | 1 | -2 | 1 | -1 | -1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Gula Darah | Diagnosis | | |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 2 | 4 |
| < 0 | 0 | 3 | 3 |
| Total | 2 | 5 | 7 |
| Tekanan darah | Diagnosis | | |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 3 | 4 |
| < 0 | 1 | 2 | 3 |
| Total | 2 | 5 | 7 |
| BMI | Diagnosis | | |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 4 | 6 |
| < 0 | 0 | 1 | 1 |
| Total | 2 | 5 | 7 |
| Usia | Diagnosis | | |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 3 | 4 |
| < 0 | 1 | 2 | 3 |
| Total | 2 | 5 | 7 |
| Lingkar Perut | Diagnosis | | |
| Ya | Tidak | Total |
| >=0 | 2 | 3 | 5 |
| <0 | 0 | 2 | 2 |
| Total | 2 | 5 | 7 |

Gini indeks:

Gini split:

Selanjutnya membuat iterasi pohon kedua dengan menghilangkan atribut Tekanan darah, juga menghilangkan baris 4 dan 6 dari dataset. Maka hasilnya seperti pada tabel dibawah ini.

**Tabel.** Contoh Pohon Kedua

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | BMI | Usia | Lingkar Perut | Diagnosis  diabetes |
| 1 | 2 | 0 | 0 | -1 | 0 | 1 |
| 2 | -1 | 1 | 0 | -1 | 1 | -1 |
| 3 | -1 | 0 | 0 | 1 | 2 | 1 |
| 5 | -1 | 1 | 0 | 1 | 0 | 1 |
| 7 | 2 | -1 | 2 | -1 | -1 | -1 |
| 8 | -1 | 0 | -2 | 1 | -1 | -1 |

Lakukan perhitungan gini indeks dan gini split indeks dengan rumus seperti pada iterasi pohon pertama, dan hasilnya akan seperti pada tabel dibawah ini.

**Tabel.** Kontingensi Kedua

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Merokok | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 1 | 2 | 0.5 | 0.5 |
| < 0 | 2 | 2 | 4 | 0.5 |
| Total | 3 | 3 | 6 | 1 |
| Gula Darah | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 3 | 2 | 5 | 0.48 | 0.4 |
| < 0 | 0 | 1 | 1 | 0 |
| Total | 3 | 3 | 6 | 0.48 |
| BMI | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 3 | 2 | 5 | 0.48 | 0.4 |
| < 0 | 0 | 1 | 1 | 0 |
| Total | 3 | 3 | 6 | 0.48 |
| Usia | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 1 | 3 | 0.44 | 0.44 |
| < 0 | 1 | 2 | 3 | 0.44 |
| Total | 3 | 3 | 6 | 0.88 |
| Lingkar Perut | Diagnosis |  |  | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 3 | 1 | 4 | 0.37 | 0.24 |
| < 0 | 0 | 2 | 2 | 0 |
| Total | 3 | 3 | 6 | 0.37 |

Berdasarkan gini split indeks diatas, maka atribut yang menjadi akar adalah Gula darah, karena memiliki nilai gini split indeks yang kecil. Pada iterasi ke 3, asumsikan kita melakukan bagging pada dataset dengan menghilangkan atribut BMI, baris yang dihilangkan adalah baris 2 dan 7, hasilnya dapat dilihat pada tabel dibawah ini.

**Tabel.** Contoh Pohon Ketiga

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | Usia | Lingkar Perut | Diagnosis  Terkena diabetes |
| 1 | 2 | 0 | -1 | -1 | 0 | 1 |
| 3 | -1 | 0 | 1 | 1 | 2 | 1 |
| 4 | -1 | -1 | 1 | 0 | 0 | -1 |
| 5 | -1 | 1 | 1 | 1 | 0 | 1 |
| 6 | -1 | -1 | -1 | 1 | 0 | -1 |
| 8 | -1 | 0 | 1 | 1 | -1 | -1 |

Lakukan perhitungan gini indeks dan gini split indeks dengan rumus seperti pada iterasi pohon pertama, dan hasilnya akan seperti tabel dibawah ini.

**Tabel.** Kontingensi Ketiga

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Merokok | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 0 | 1 | 0 | 0.4 |
| < 0 | 2 | 3 | 5 | 0.48 |
| Total | 3 | 3 | 6 | 0.48 |
| Gula Darah | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 3 | 1 | 4 | 0.37 | 0.24 |
| < 0 | 0 | 2 | 2 | 0 |
| Total | 3 | 3 | 6 | 0.37 |
| Tekanan darah | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 2 | 4 | 0.5 | 0.5 |
| < 0 | 1 | 1 | 2 | 0.5 |
| Total | 3 | 3 | 6 | 1 |
| Usia | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 3 | 5 | 0.48 | 0.4 |
| < 0 | 1 | 0 | 1 | 0 |
| Total | 3 | 4 | 6 | 0.48 |
| Lingkar Perut | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 3 | 2 | 5 | 0.48 | 0.4 |
| < 0 | 0 | 1 | 1 | 0 |
| Total | 3 | 3 | 6 | 0.48 |

Berdasarkan gini split indeks diatas, maka atribut yang menjadi akar adalah Gula darah, karena memiliki nilai gini split indeks yang kecil. Pada iterasi ke 4, asumsikan kita melakukan bagging pada dataset dengan menghilangkan atribut Usia, baris yang dihilangkan adalah baris 1, hasilnya dapat dilihat pada tabel dibawah ini.

**Tabel.** Contoh Pohon Keempat

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah | BMI | Lingkar Perut | Diagnosis  Terkena diabetes |
| 2 | -1 | 1 | -1 | 0 | 1 | -1 |
| 3 | -1 | 0 | 1 | 0 | 2 | 1 |
| 4 | -1 | -1 | 1 | 0 | 0 | -1 |
| 5 | -1 | 1 | 1 | 0 | 0 | 1 |
| 6 | -1 | -1 | -1 | 0 | 0 | -1 |
| 7 | 2 | -1 | 0 | 2 | -1 | -1 |
| 8 | -1 | 0 | 1 | -2 | -1 | -1 |

Lakukan perhitungan gini indeks dan gini split indeks dengan rumus seperti pada iterasi pohon pertama, dan hasilnya akan seperti tabel dibawah ini.

**Tabel.** Kontingensi Keempat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Merokok | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 4 | 6 | 0.44 | 0.37 |
| < 0 | 0 | 1 | 1 | 0 |
| Total | 2 | 5 | 7 | 0.44 |
| Gula Darah | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 2 | 4 | 0.5 | 0.28 |
| < 0 | 0 | 3 | 3 | 0 |
| Total | 2 | 5 | 7 | 0.5 |
| Tekanan darah | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 3 | 5 | 0.48 | 0.34 |
| < 0 | 0 | 2 | 2 | 0 |
| Total | 2 | 5 | 7 | 0.48 |
| BMI | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 4 | 6 | 0.44 | 0.37 |
| < 0 | 0 | 1 | 1 | 0 |
| Total | 2 | 5 | 7 | 0.44 |
| Lingkar Perut | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 3 | 5 | 0.48 | 0.34 |
| < 0 | 0 | 2 | 2 | 0 |
| Total | 2 | 5 | 7 | 0.48 |

Berdasarkan gini split indeks diatas, maka atribut yang menjadi akar adalah Gula darah, karena memiliki nilai gini split indeks yang kecil. Pada iterasi ke 5, asumsikan kita melakukan bagging pada dataset dengan menghilangkan atribut Gula darah, baris yang dihilangkan adalah baris 1, hasilnya dapat dilihat pada tabel dibawah ini.

**Tabel.** Contoh Pohon Kelima

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Tekanan darah | BMI | Usia | Lingkar Perut | Diagnosis  Terkena diabetes |
| 1 | 2 | -1 | 0 | -1 | 0 | 1 |
| 2 | -1 | -1 | 0 | -1 | 1 | -1 |
| 4 | -1 | 1 | 0 | 0 | 0 | -1 |
| 5 | -1 | 1 | 0 | 1 | 0 | 1 |
| 6 | -1 | -1 | 0 | 1 | 0 | -1 |
| 7 | 2 | 0 | 2 | -1 | -1 | -1 |
| 8 | -1 | 1 | -2 | 1 | -1 | -1 |

Lakukan perhitungan gini indeks dan gini split indeks dengan rumus seperti pada iterasi pohon pertama, dan hasilnya akan seperti tabel dibawah ini.

**Tabel.** Kontingensi Kelima

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Merokok | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 1 | 2 | 0.5 | 0.37 |
| < 0 | 1 | 4 | 5 | 0.32 |
| Total | 2 | 5 | 7 | 0.82 |
| Tekanan darah | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 3 | 4 | 0.37 | 0.4 |
| < 0 | 1 | 2 | 3 | 0.44 |
| Total | 2 | 5 | 7 | 0.77 |
| BMI | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 4 | 6 | 0.44 | 0.37 |
| < 0 | 0 | 1 | 1 | 0 |
| Total | 2 | 5 | 7 | 0.44 |
| Usia | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 1 | 3 | 4 | 0.37 | 0.4 |
| < 0 | 1 | 2 | 3 | 0.44 |
| Total | 2 | 5 | 7 | 0.81 |
| Lingkar Perut | Diagnosis | | | Gini indeks | Gini split indeks |
| Ya | Tidak | Total |
| ≥ 0 | 2 | 3 | 5 | 0.48 | 0.34 |
| < 0 | 0 | 2 | 2 | 0 |
| Total | 2 | 5 | 7 | 0.48 |

Terlihat pada perhitungan diatas, bahwa atribut terbaik adalah lingkar perut, karena memiliki nilai gini split indeks terkecil.

Atribut Gula Darah merupakan yang terbaik dan layak dijadikan *root* karena mempunyai nilai *split* terkecil, berdasarkan iterasi pada kelima pohon dengan menggunakan dataset yang tersedia. terlepas atribut lingkar perut menjadi atribut terbaik pada siklus pohon kelima, hal ini semata-mata karena bagging yang dilakukan terhadap atribut Gula darah. Oleh karena itu, jika pohon tersebut terus-menerus diiterasi dan menyertakan atribut gula darah, maka akan dipastikan atribut tersebut menjadi akar dan atribut terbaik. Gambar dibawah ini contoh hasil dari perhitungan klasifikasi *Random Forest*.

A diagram of a company structure

Description automatically generated

**Gambar.** Hasil klasifikasi Random Forest

Selanjutnya dilakukan pengujian model klasifikasi yang telah dibuat menggunakan algoritma *Random Forest* terhadap data testing sebanyak 2 data pada tabel data uji *Random Forest*.

**Tabel.** Data Uji Random Forest

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Merokok | Gula darah | Tekanan darah |  | Usia | Lingkar Perut | Diagnosis Diabetes |
| 1 | -1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 2 | -1 | -1 | -1 | 0 | 1 | 0 | -1 |

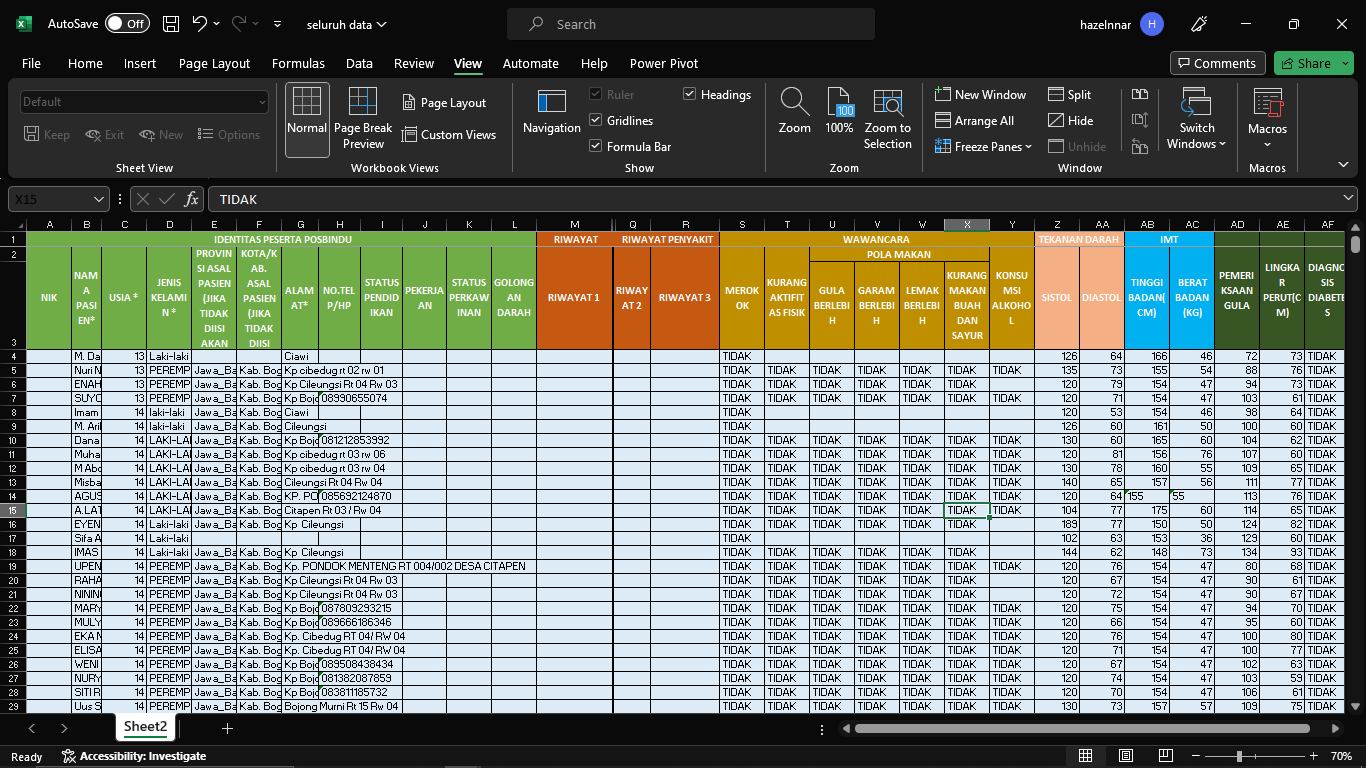
Dari setiap data uji, dilakukan pengklasifikasian dari model klasifikasi yang telah dibuat dengan memasukan data ke setiap pohon sehingga bisa dilakukan *voting* terbanyak dari setiap pohon. Hasil *voting* dapat dilihat pada tabel *Majority Voting.*

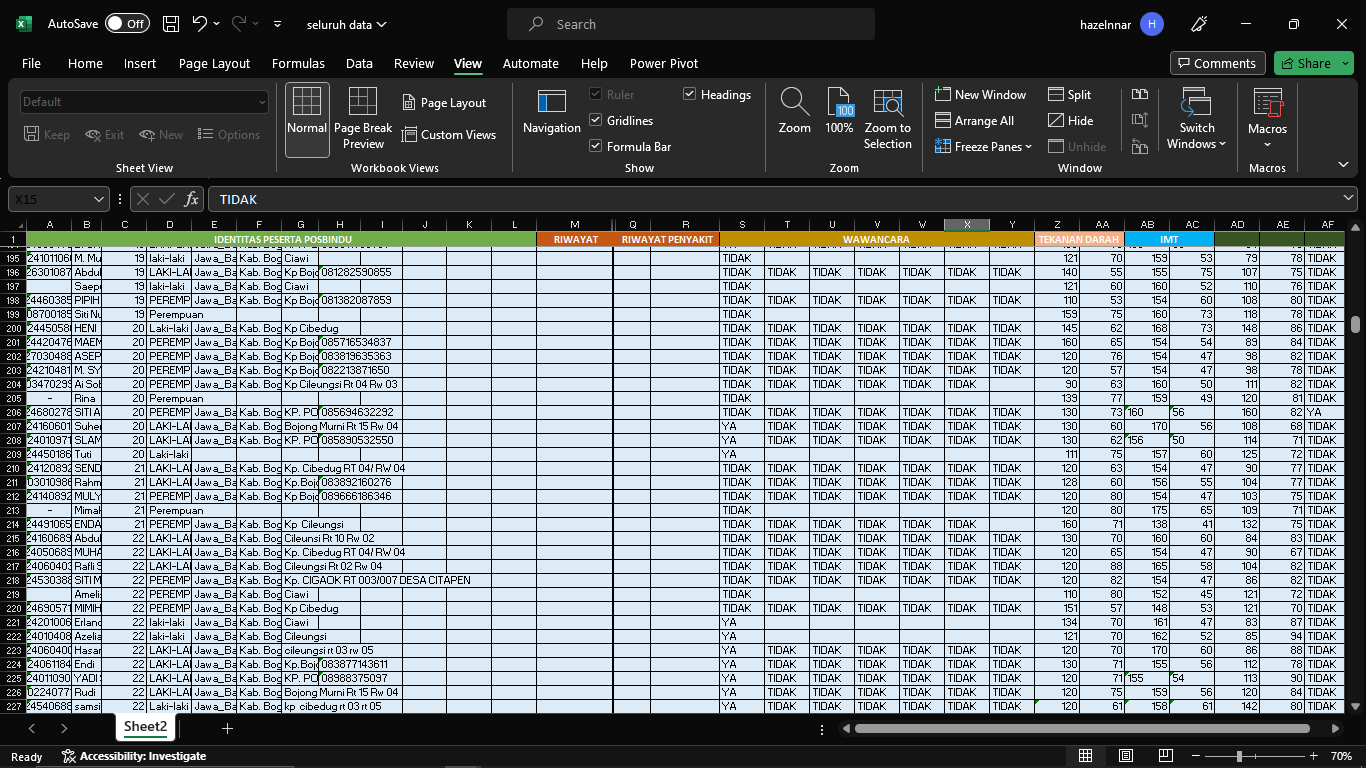
**Tabel.** Majority Voting

|  |  |  |
| --- | --- | --- |
|  | Data pertama | Data kedua |
| Pohon pertama | Ya | Tidak |
| Pohon kedua | Ya | Tidak |
| Pohon ketiga | Tidak | Tidak |
| Pohon keempat | Ya | Tidak |
| Pohon kelima | Ya | Tidak |
| ***Majority Voting*** | **Ya** | **Tidak** |

Dari hasil pengujian terhadap 2 data uji, data pertama memiliki hasil *voting* terbanyak Ya sebanyak 4 dari 5 pohon, pada data kedua memiliki hasil *voting* terbanyak Tidak dari keseluruhan pohon. Maka dari hasil *voting* terhadap masing-masing data, data pertama terdiagnosis diabetes dan data kedua tidak terdiagnosis diabetes. Hasil pengujian data tersebut sesuai dengan hasil variabel dependen pada tabel Data Uji.

**Lampiran 6.** Dataset CSV Puskesmas Citapen





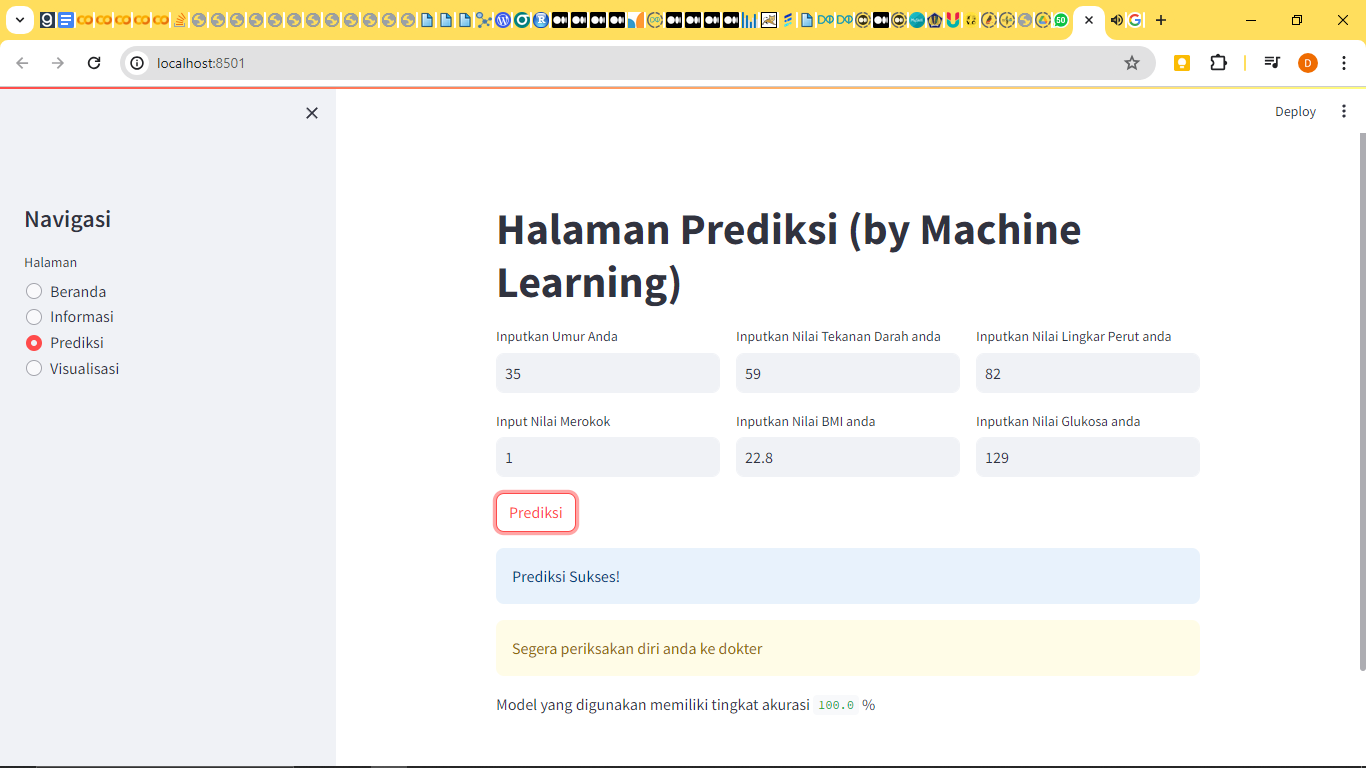
Seluruh dataset dapat dilihat pada link: https://drive.google.com/file/d/1XYbHmr0dV0EZC1oNpFby9bpY931nitk\_/view?usp=sharing

**Lampiran 7.** Uji coba model pada website

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated