

PENERAPAN ALGORITMA C4.5 DALAM *DATA MINING* UNTUK IDENTIFIKASI FAKTOR RISIKO *STROKE* PADA *DATASET* MEDIS

Immanuel Clement Onggo Putra¹, Hartana Wijaya²

Fakultas Sains dan Teknologi, Universitas Buddhi Dharma
Jalan Imam Bonjol No. 41, Tangerang, Indonesia
Email: ¹immanuelcop@gmail.com, ²hartana.wijaya@ubd.ac.id

Abstrak

Stroke adalah masalah kesehatan di seluruh dunia yang terjadi secara signifikan. Berdasarkan data Organisasi Kesehatan Dunia (WHO), insiden *stroke* baru mencapai 13,7 juta kasus per tahun dengan angka kematian sebesar 5,5 juta jiwa. *Data mining* adalah proses dari pengumpulan, pengolahan, dan analisis data untuk memperoleh informasi yang penting. Algoritma C4.5 merupakan algoritma yang paling umum digunakan dalam *data mining* untuk membangun pohon keputusan berdasarkan data yang telah diberi label. Masalah yang ingin diselesaikan dari penelitian ini adalah mengolah data kesehatan untuk mengetahui risiko *stroke* itu sulit dan lama dan masyarakat sulit mendapat informasi tentang *stroke* yang benar dan terpercaya. Untuk teknik pengumpulan data dari *dataset* yang diambil dari situs *Kaggle* dengan *dataset Stroke Prediction* dengan 5110 *record* dan 12 atribut, serta studi pustaka yang mencari informasi-informasi yang terkait dengan penelitian ini untuk membantu penerapan algoritma *data mining* C4.5 dalam mendeteksi penyebab *stroke*. Pustaka yang digunakan dari berbagai media seperti internet, buku, jurnal, dan media lainnya. Algoritma pohon keputusan di *data mining* dapat diterapkan dalam mendeteksi penyebab seseorang terkena *stroke*. Setelah dilakukan perhitungan manual dan menggunakan perangkat lunak *RapidMiner*, akurasi hasil prediksi menjadi tolok ukur seberapa efektif algoritma ini dalam mengidentifikasi penyebab *stroke* pada seseorang, dengan menghasilkan model klasifikasi dengan tingkat akurasi 94,89% dan nilai AUC sebesar 0.709. Ada 11 responden yang mengisi kuesioner dengan hasilnya adalah 45,44% memilih sangat setuju, 44,56% memilih setuju, dan 10% yang netral.

Kata Kunci

C4.5, Database, Data Mining, Stroke, Teknik Informatika

Latar Belakang

Stroke adalah masalah kesehatan global yang signifikan, menduduki peringkat kedua sebagai penyebab utama kematian dan kecacatan di seluruh dunia [1]. Di Indonesia sendiri, *stroke* juga menjadi masalah kesehatan serius yang ditandai dengan kerusakan otak akibat gangguan aliran darah non-traumatik [2]. Kondisi ini dapat mengakibatkan berbagai dampak klinis seperti defisit sensorik dan motorik, gangguan kognitif, serta masalah psikologis seperti stres, kecemasan, dan depresi [3]. Oleh karena itu, manajemen stres yang efektif sangat penting dalam upaya pencegahan dan pengelolaan *stroke*, mengingat dampaknya yang dapat mengancam jiwa.

Data Organisasi Kesehatan Dunia (WHO) menunjukkan bahwa insiden *stroke* baru mencapai 13,7 juta kasus per tahun dengan angka kematian sebesar 5,5 juta jiwa. Secara global, diperkirakan terdapat 50 juta penderita *stroke*, dengan 9 juta di antaranya mengalami kecacatan berat. Tingkat mortalitas kasus *stroke* mencapai 10%, dan sebagian besar kasus *stroke* serta kematian dan kecacatan yang diakibatkannya terjadi di negara-negara berpenghasilan rendah dan menengah [4]. Di Indonesia, pada tahun 2019, *stroke* tercatat sebagai penyebab kematian utama dengan tingkat mortalitas mencapai 15,4%. Prevalensi

stroke pada penduduk berusia lebih dari 15 tahun diperkirakan berkisar antara 10,9% hingga 11,3% [2].

Perkembangan teknologi telah membawa perubahan signifikan dalam bidang kesehatan, termasuk dalam deteksi penyakit *stroke*. Salah satu pendekatan yang menggabungkan teknologi dengan kesehatan adalah penggunaan *data mining*. *Data mining* adalah proses pengumpulan, pengolahan, dan analisis data untuk mendapatkan informasi penting. Teknik ini menggunakan analisis data dan algoritma untuk memproses data dalam jumlah besar, memungkinkan identifikasi pola dan informasi berharga dari *database* yang luas [5].

Dalam proses *data mining*, algoritma memainkan peran penting. Algoritma adalah serangkaian langkah logis dan terstruktur yang dirancang untuk memecahkan masalah dan menghasilkan solusi tertentu [6]. Salah satu algoritma yang umum digunakan dalam *data mining*, terutama untuk membangun pohon keputusan, adalah algoritma C4.5 atau *Decision Tree*. Algoritma ini bekerja dengan mengidentifikasi atribut yang memberikan informasi paling signifikan, kemudian memilihnya sebagai simpul utama dalam pohon keputusan [7]. Proses ini diulang secara rekursif hingga tidak ada lagi atribut yang dapat menghasilkan simpul baru. Pohon keputusan yang dihasilkan dapat diubah menjadi aturan klasifikasi untuk memprediksi kelas atau kategori dari data baru, membantu dalam deteksi dan prediksi *stroke*. Berdasarkan penjelasan sebelumnya, penelitian ini bertujuan untuk mengevaluasi keakuratan algoritma C4.5 dalam mendeteksi penyakit *stroke*. Dengan itu, penelitian ini berjudul "PENERAPAN ALGORITMA C4.5 DALAM *DATA MINING* UNTUK IDENTIFIKASI FAKTOR RISIKO *STROKE* PADA *DATASET* MEDIS".

Metode Penelitian

Data Sekunder

Data sekunder merupakan informasi yang didapatkan oleh peneliti yang bukan secara langsung, melainkan melalui sumber yang sudah tersedia atau dari pihak lain yang sudah mengumpulkan data tersebut sebelumnya. Peneliti hanya perlu menyalin, mengakses, atau meminta data yang sudah dikumpulkan oleh pihak lain di lapangan [8].

Klasifikasi

Klasifikasi merupakan teknik pengelompokan data ke dalam kategori atau kelas yang sudah ditentukan. Metode ini bertujuan untuk meningkatkan keakuratan prediksi dan analisis, terutama saat menghadapi volume data yang besar [9]. Tujuan klasifikasi dalam penambahan data adalah untuk mengelompokkan data ke dalam kategori yang sudah ada. Proses ini dimulai dengan menggunakan data yang sudah diberi label kategori sebagai acuan [5].

Data Mining

Data mining merupakan bidang ilmu yang memanfaatkan metodologi dari statistik, pembelajaran mesin, basis data, pengenalan pola, dan visualisasi data untuk menghasilkan pengetahuan berharga dan pola tersembunyi dari kumpulan data berskala besar [5]. Dalam pengertian lain, *data mining* adalah proses pengumpulan informasi penting dari data dalam jumlah besar. Informasi ini diubah menjadi pengetahuan baru yang berguna untuk pengambilan keputusan yang lebih baik. Proses ini menggunakan berbagai teknik, seperti klasifikasi, klustering, asosiasi, dan regresi, untuk menemukan pola tersembunyi, hubungan, keanehan, atau tren yang sulit ditemukan secara manual. Hasil dari *data mining* bisa berupa model prediksi, aturan klasifikasi, segmentasi pelanggan, atau rekomendasi produk, yang semuanya dapat digunakan untuk membuat keputusan bisnis yang lebih strategis dan berdasarkan data [10].

Algoritma C4.5

Algoritma C4.5, yang dikembangkan oleh Ross Quinlan, merupakan metode pembentukan pohon keputusan yang berlandaskan pada pemilihan atribut dengan nilai gain tertinggi berdasarkan entropi sebagai kriteria pemisahan optimal pada setiap simpul pohon [11]. Algoritma C4.5 bekerja dengan dua cara utama: pertama, merancang dan membangun struktur pohon keputusan berdasarkan data yang diberikan. Kedua, dari pohon keputusan tersebut, algoritma ini menghasilkan aturan-aturan (*rule model*) berupa pernyataan "jika-maka" (*if then*) yang dapat digunakan untuk mengklasifikasikan atau memprediksi hasil berdasarkan kondisi tertentu. Algoritma C4.5, juga dikenal sebagai *decision tree*, bekerja dengan mengevaluasi setiap titik keputusan dan memilih opsi terbaik hingga tidak ada lagi pilihan yang bisa diambil. Para peneliti telah memanfaatkan algoritma ini untuk membantu pengambilan keputusan yang lebih cepat, akurat, dan efisien. Dalam penelitian ini, algoritma C4.5 digunakan untuk mengidentifikasi penyebab *stroke* dengan menganalisis gejala-gejala yang dialami pasien melalui pohon keputusan yang dibangun dari data yang ada.

Java

Java diluncurkan oleh Sun Microsystems pada tahun 1995, yang merupakan bahasa pemrograman berorientasi objek yang bersifat lintas platform, memungkinkan eksekusi kode pada beragam arsitektur perangkat keras, termasuk komputer desktop dan perangkat mobile. *Java* dapat dijalankan di Windows, Linux, Unix, dan DOS dan banyak digunakan dalam pembuatan dan perancangan aplikasi baik untuk desktop, situs web, seluler, dll [12]. *Java*, bahasa pemrograman populer yang diciptakan oleh tim Sun Microsystems yang dipimpin oleh Patrick Naughton dan James Gosling, dirancang untuk fleksibilitas dan kesederhanaan. Awalnya bernama "*Oak*", namanya kemudian diubah menjadi "*Java*" karena konflik merek dagang. *Java* memungkinkan pengembangan aplikasi lintas platform, termasuk komputer dan perangkat seluler, dan bahkan memiliki peramban sendiri yang disebut "*HotJava*" [13].

NetBeans

NetBeans merupakan aplikasi *Integrated Development Environment* (IDE) yang populer di kalangan pengembang perangkat lunak, digunakan untuk menulis, mengkompilasi, men-debug, dan mendistribusikan program. Mirip dengan *Microsoft Visual Studio*, *NetBeans* memiliki cakupan yang lebih luas dalam pengembangan aplikasi dan merupakan perangkat lunak *open-source* [14]. *NetBeans* adalah perangkat lunak pengembangan yang dirancang khusus untuk bahasa pemrograman *Java*. Sebagai lingkungan pengembangan terintegrasi (IDE) yang bersifat *open-source*, *NetBeans* tersedia secara gratis dan mendukung berbagai sistem operasi seperti Windows, Linux, Mac, dan Solaris.

RapidMiner

RapidMiner merupakan perangkat lunak komprehensif yang dirancang untuk melakukan analisis *data mining* yang kompleks. Kemampuannya didukung oleh beragam mode operasi yang memfasilitasi berbagai teknik dan metodologi dalam proses *data mining*. Perangkat lunak ini memiliki sifat bersumber terbuka sehingga dapat dipakai oleh semua orang. *RapidMiner* pada saat kali pertama dirilis, awalnya menggunakan nama *Yale* yang berarti *Yet another learning environment*. Perangkat lunak tersebut dirilis oleh Ralf Klinkenbert pada tahun 2001 [15]. *RapidMiner* adalah perangkat lunak lengkap yang dirancang untuk memfasilitasi analisis *data mining*, *text mining*, serta prediksi. Dengan mengintegrasikan berbagai teknik deskriptif dan prediktif, perangkat lunak ini mampu menghasilkan wawasan berharga yang dapat dimanfaatkan dalam pengambilan keputusan strategis. Dilengkapi dengan lebih dari 500 operator *data mining* yang mencakup fungsi *input*, *output*, *preprocessing*, dan visualisasi, *RapidMiner* menawarkan kemudahan untuk memenuhi kebutuhan analisis data yang kompleks [16].

Basis Data

Basis data atau *database* merupakan himpunan terstruktur dari informasi atau data yang disimpan dalam sistem komputer, yang kemudian dapat diproses dan diolah menggunakan perangkat lunak guna memperoleh informasi yang relevan [17]. *Database* adalah kumpulan data terstruktur yang saling terhubung, dirancang untuk memungkinkan akses dan penggunaan ulang informasi secara efisien [18]. Basis data merupakan suatu struktur data yang terorganisasi secara logis dan sistematis, dirancang untuk disimpan dan dikelola dalam lingkungan komputasi. Struktur ini memfasilitasi penyimpanan, pengambilan, pembaruan, dan pengelolaan informasi secara efisien. *Database* berfungsi sebagai repositori informasi yang dapat diakses, diperbarui, dan dianalisis menggunakan perangkat lunak manajemen basis data untuk dapat mendukung pengambilan keputusan dan proses bisnis [19].

XAMPP

XAMPP adalah perangkat lunak bebas lintas platform yang menggabungkan beberapa komponen perangkat lunak penting dalam satu paket instalasi. Perangkat lunak ini mengintegrasikan *Apache HTTP Server*, *MySQL database server*, serta dukungan untuk bahasa pemrograman PHP (versi 4 dan 5). Dengan instalasi yang mudah dan gratis, XAMPP banyak digunakan untuk pengembangan dan pengujian aplikasi web berbasis PHP di lingkungan lokal, baik pada sistem operasi Linux maupun Windows [20].

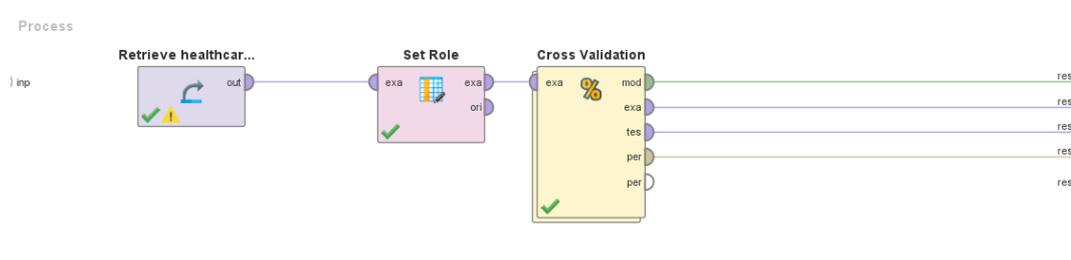
Black Box

Black box testing merupakan metode pengujian perangkat lunak yang berfokus pada hasil keluaran (*output*) berdasarkan masukan (*input*) yang diberikan, tanpa perlu memahami detail internal perangkat lunak tersebut, termasuk kode program yang digunakan. Proses pengujian dilakukan dengan memasukkan berbagai data pada formulir yang tersedia untuk memastikan perangkat lunak berfungsi sesuai dengan kebutuhan pengguna [21].

Hasil

Pembahasan Metode dan Algoritma

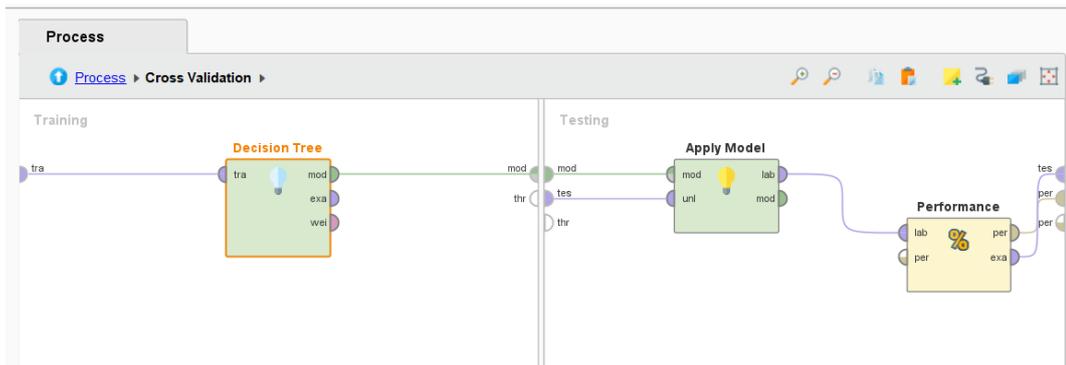
Berikut ini adalah operator yang digunakan dalam aplikasi *RapidMiner*. Pengujian dengan *RapidMiner* tersebut menggunakan 5110 data pasien yang didapatkan dari situs web *Kaggle* dengan 248 pasien positif dan 4.862 pasien positif.



Gambar 1 Perancangan Operator di *RapidMiner*

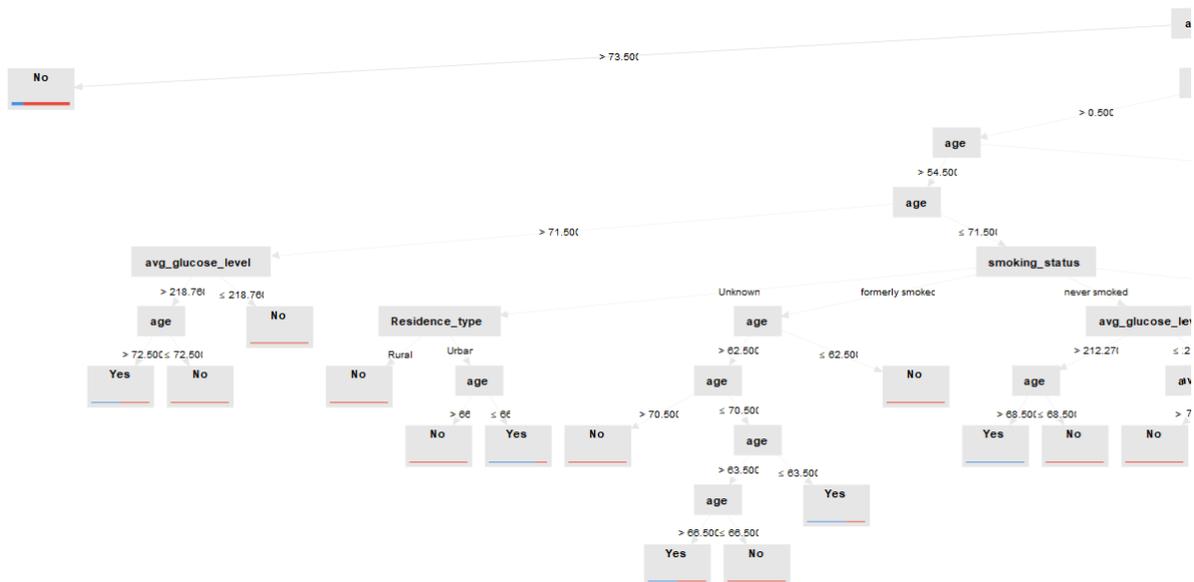
Gambar diatas merupakan tahap awal dalam menguji data menggunakan aplikasi *RapidMiner*. Pada tahap awal masukan 3 operator yaitu, operator *retrieve dataset*, *set role* dan *cross validation*. *Retrieve dataset* berguna untuk melakukan impor dataset yang sudah dimasukkan ke repositori lokal. *Set role* berfungsi untuk membedakan baris yang berisi nama atribut koordinat dengan baris yang berisi prediksi posisi. Baris-baris ini akan dimasukkan ke dalam kategori 'label'. Dengan demikian, saat data dikategorikan, baris-baris dalam kategori 'label' tidak akan ikut terhitung dan mempengaruhi hasil akhir. Sementara itu, *cross validation* merupakan teknik yang membagi *dataset* menjadi beberapa bagian, dimana setiap bagian digunakan secara bergantian sebagai data pelatihan untuk membangun model dan data

pengujian untuk mengevaluasi performanya. Tujuannya adalah untuk mendapatkan estimasi performa model yang lebih akurat dan menghindari *overfitting*.

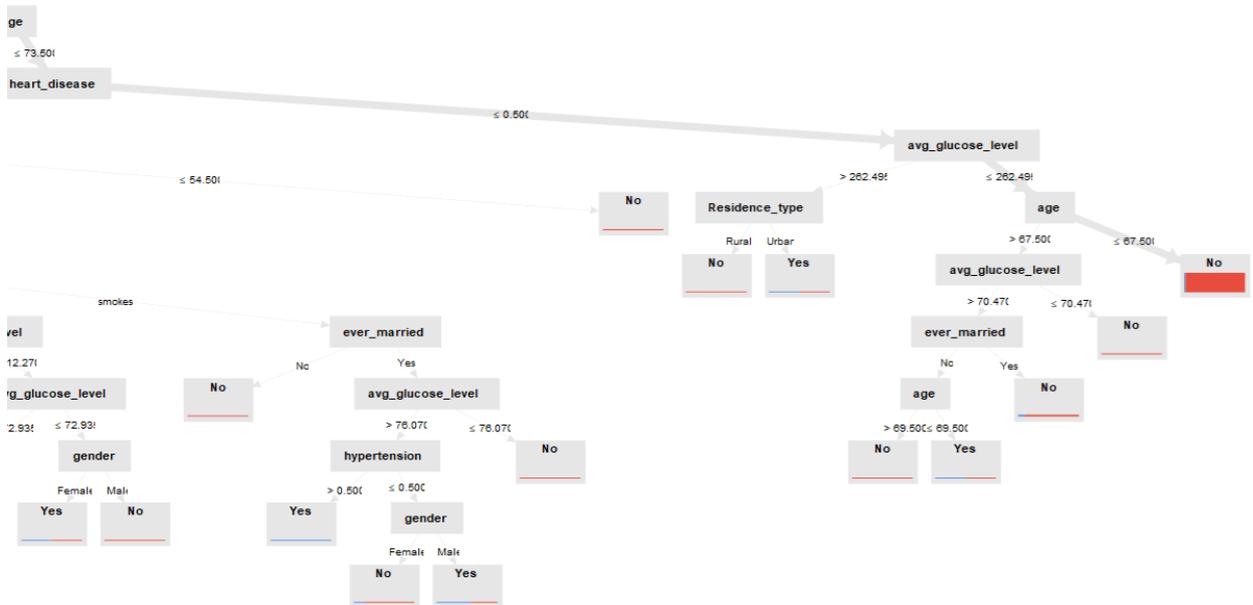


Gambar 2 Perancangan Operator di *RapidMiner* (Cross Validation)

Pada tahap berikutnya adalah dimasukkannya beberapa operator di dalam operator *cross validation* dengan cara mengeklik ganda pada operator *cross validation*, di dalam operator *cross validation* terdapat halaman kosong dan akan dimasukkan 3 buah operator, yaitu *decision tree*, *apply model*, dan *performance*. Algoritma *decision tree* tersebut merupakan algoritma yang digunakan untuk mengklasifikasikan seseorang yang menderita penyakit stroke. Dalam proses pengambilan keputusan menggunakan *decision tree*, *gain_ratio* pada kolom kriteria menjadi pertimbangan utama untuk mencapai hasil yang diinginkan. Syarat minimal *confidence* adalah 0.1, syarat *minimal gain* adalah 0.01, dan jumlah *minimum data* dalam setiap *leaf* adalah 2. *Apply model* berfungsi untuk menampilkan representasi awal dari data yang sedang kita olah, sedangkan *performance* digunakan untuk mengevaluasi seberapa baik suatu algoritma bekerja ketika diterapkan pada data uji. Pohon keputusan yang dihasilkan menggunakan *RapidMiner* dapat dilihat di bawah ini.



Gambar 3 Pohon Keputusan Bagian 1



Gambar 4 Pohon Keputusan Bagian 2

Pengujian Confusion Matrix

Confusion Matrix (matriks kebingungan) berfungsi untuk mengukur seberapa akurat model klasifikasi dalam menebak kelas-kelas data. Di dalamnya ada 4 istilah: *True Positive* (TP) dan *True Negative* (TN) menunjukkan tebakan yang benar, masing-masing untuk kelas positif dan negatif, sedangkan *False Positive* (FP) dan *False Negative* (FN) menunjukkan tebakan yang salah. Berikut adalah contoh matriks kebingungan untuk algoritma C4.5.

accuracy: 94.89% +/- 0.44% (micro average: 94.89%)

	true Positive Stroke	true Negative Stroke	class precision
pred. Positive Stroke	16	28	36.36%
pred. Negative Stroke	233	4833	95.40%
class recall	6.43%	99.42%	

Gambar 5 Hasil Confusion Matrix

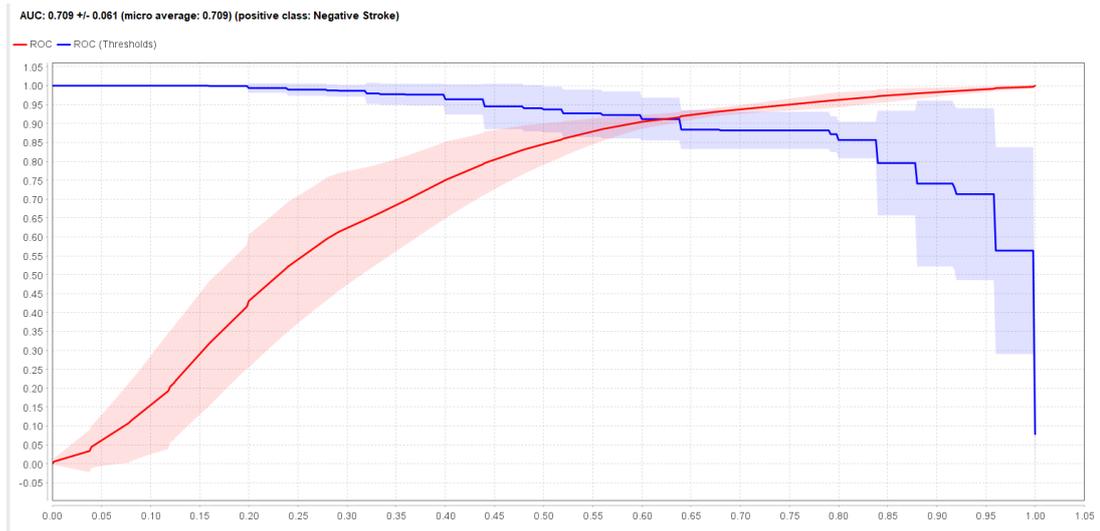
Tingkat keakuratan dari algoritma C4.5 menghasilkan tingkat akurasi dengan nilai 94.89% yang didapat dengan melakukan perhitungan berikut, dimana nilai *True Positive* adalah 16 data, *True Negative* adalah 4833 data, *False Positive* adalah 233 data, dan *False Negative* adalah 28 data:

$$\left(\frac{16 + 4833}{16 + 28 + 233 + 4833} \right) \times 100\% = 94.89\%$$

Pengujian AUC

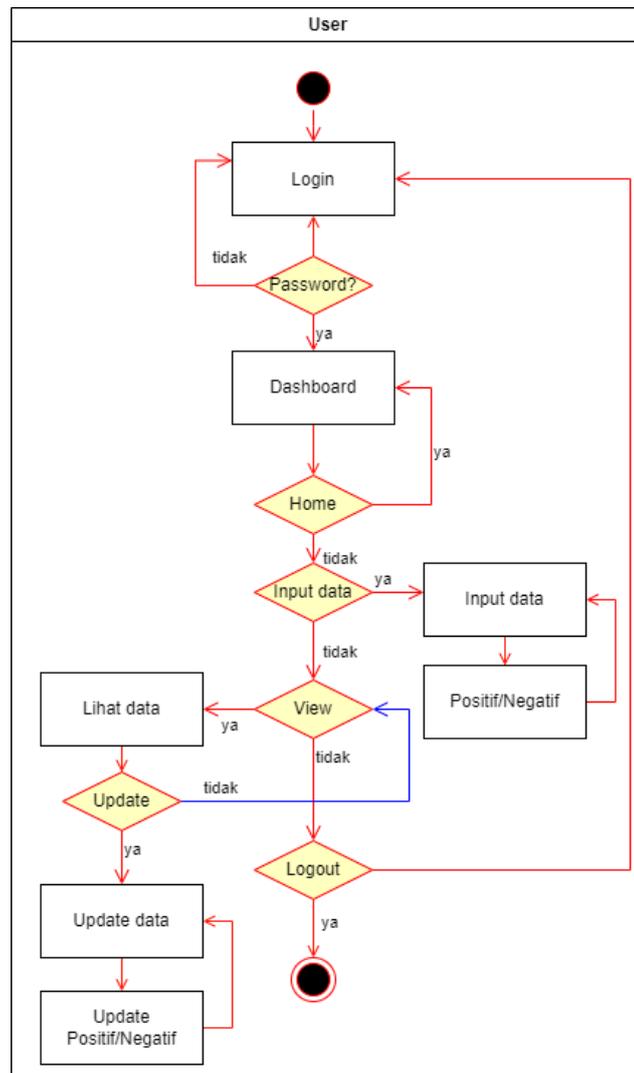
AUC merupakan ukuran performa model klasifikasi biner, yaitu seberapa baik model tersebut dapat membedakan kelas positif dan negatif. Semakin tinggi nilai AUC, semakin baik pula kemampuan model dalam melakukan klasifikasi.

Berikut adalah grafik dari hasil dari pengujian AUC dari 5110 data, terdiri dari 248 pasien positif dan 4.862 pasien negatif menggunakan algoritma C4.5 dengan hasil AUC dengan nilai 0.709 dari 1, hasil ini termasuk dalam *fair classification*. Garis merah merupakan ROC (*receiver operating characteristic*), garis biru merupakan *threshold* dari ROC, sedangkan titik potongnya merupakan titik pertemuan antara garis ROC dan garis *threshold* ROC.



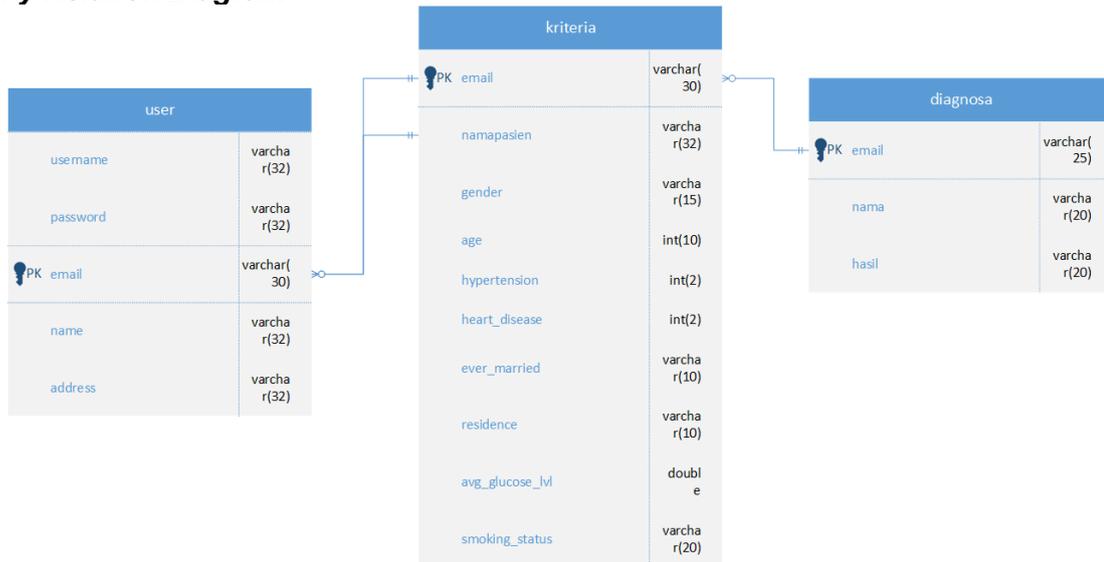
Gambar 6 Grafik AUC

Activity Diagram



Gambar 7 Activity Diagram

Entity Relation Diagram



Gambar 8 Perancangan ERD Database

Tampilan Program

1. Halaman Login

Username

Password

Login Register

Gambar 9 Tampilan Program Login

Halaman ini berisi formulir *login* di mana pengguna harus memasukkan nama pengguna dan kata sandi untuk mengakses halaman utama.

2. Halaman Register

Username

Password

Email

Name

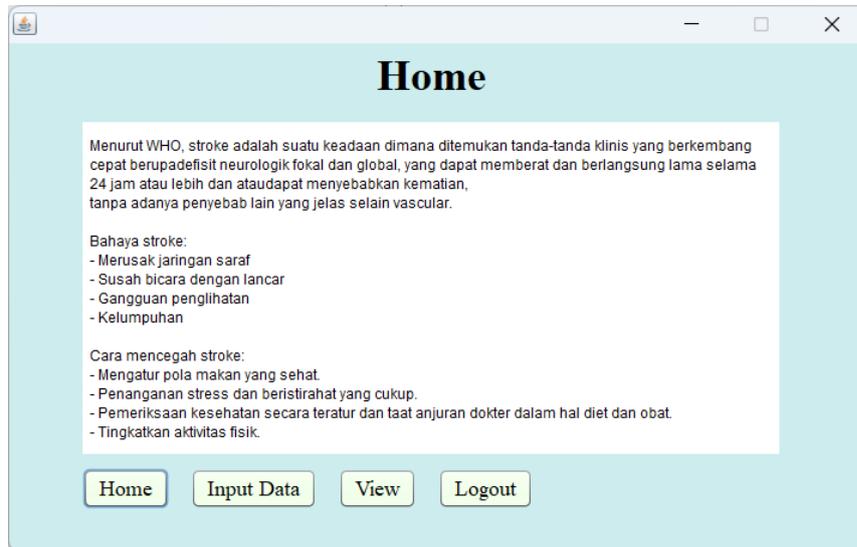
Address

Register

Gambar 10 Tampilan Program Register

Di dalam halaman ini berfungsi untuk menampilkan *form* untuk melakukan pendaftaran. Pengguna yang belum memiliki *username* dan *password* dapat melakukan pendaftaran di halaman ini.

3. Halaman *Home*



Gambar 11 Tampilan Program *Home*

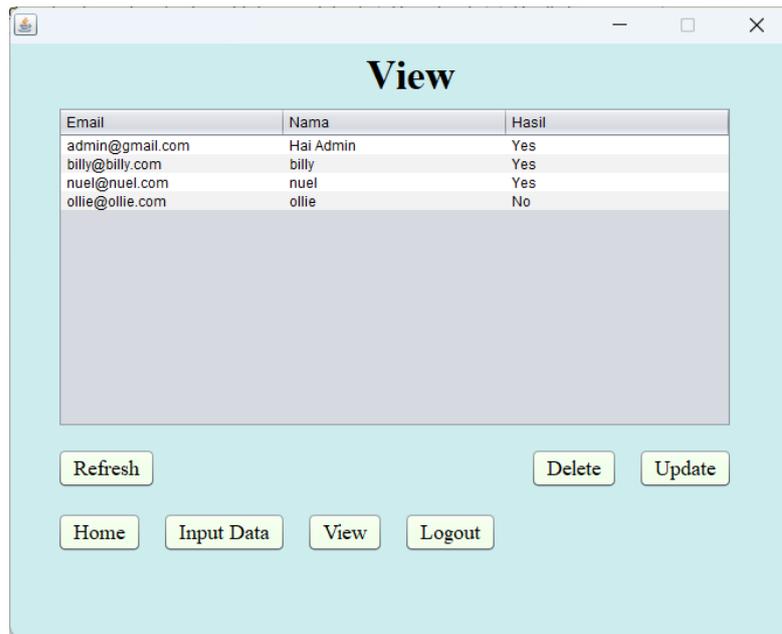
Untuk melihat Halaman Utama, pengguna wajib melakukan *login*. Halaman beranda ini memiliki 4 tombol yang terdiri dari tombol *Home*, *Input Data*, *View*, dan *Logout*. Terdapat pula informasi ringkas mengenai *stroke*, penyebab, dan cara mencegah *stroke*.

4. Halaman *Input Data*

Gambar 12 Tampilan Program *Input Data*

Pada halaman *Input Data*, pengguna dapat memasukkan data-data yang terkait dengan *stroke* yang akan menghasilkan jawaban berupa pengguna tersebut positif/negatif *stroke*.

5. Halaman View



Gambar 13 Tampilan Program *View*

Salah satu tombol yang sudah ada di halaman utama adalah tombol *View*, di halaman ini pengguna dapat melihat tabel yang berisi surel, nama, dan hasil diagnosis *stroke*.

6. Halaman *Update*

The screenshot shows a web application window titled "Update". At the top center, the word "Update" is displayed in a large, bold font. Below it is a form with several fields and buttons. The fields include: "ID" (text input with "lie@ollie.com" and a "Search" button), "Gender" (dropdown menu with "Male" selected), "Age" (text input with "61"), "Ever Married" (dropdown menu with "Yes" selected), "Residence Type" (dropdown menu with "Urban" selected), "Average Glucose Level" (text input with "87"), "Hypertension" (radio buttons for "Yes" and "No", with "No" selected), "Heart Desiase" (radio buttons for "Yes" and "No", with "No" selected), and "Smoking Status" (dropdown menu with "Never Smoked" selected). At the bottom right of the form is an "Update" button, and at the bottom left is a "Back" button.

Gambar 14 Tampilan Program *Update*

Halaman ini berfungsi untuk memperbarui data yang sudah ada, yang dapat diakses melalui halaman *View*.

Pengujian *Black Box*

Pengujian *black box* bertujuan untuk memeriksa apakah aplikasi atau program yang dikembangkan telah berfungsi dengan baik sesuai harapan. Pengujian ini berfokus pada fungsionalitas aplikasi tanpa melihat kode internalnya, dan hasilnya akan menunjukkan apakah aplikasi sudah sesuai atau masih mengandung kesalahan. Hasil pengujian *black box* dapat dilihat di bawah ini.

Tabel 1 Hasil Pengujian *Black Box*

Scene Pengujian	Skenario	Hasil yang Diharapkan	Hasil Pengujian
Halaman Login	Memasukkan nama pengguna dan kata sandi yang terdaftar di <i>database</i> , lalu mengklik tombol <i>Login</i> .	Mengarahkan pengguna ke halaman <i>Home</i> dan menampilkan pesan "Login berhasil!".	Sesuai
Halaman Login	Mencoba masuk dengan kombinasi nama pengguna dan kata sandi yang tidak terdaftar atau tidak cocok dengan data yang tersimpan di <i>database</i> , lalu menekan tombol <i>Login</i> .	Menampilkan pesan <i>error</i> bahwa nama pengguna atau kata sandi yang dimasukkan tidak sesuai.	Sesuai
Halaman Register	Memasukkan data diri dari pengguna yang berisi nama pengguna, kata sandi, surel, nama, dan alamat, lalu menekan tombol <i>Register</i> .	Menampilkan pesan "Registrasi berhasil. Silakan login dengan username dan password Anda.", dan mengarahkan pengguna ke halaman <i>Login</i> .	Sesuai
Halaman Register	Membiarkan semua atau beberapa kolom di formulir pendaftaran kosong, lalu menekan tombol <i>Register</i> .	Menampilkan pesan <i>error</i> bahwa pengguna harus mengisi semua kolom terlebih dahulu.	Sesuai
Halaman Home	Mengklik tombol <i>Home</i> .	Diarahkan kembali ke halaman <i>Home</i> .	Sesuai
Halaman Home	Mengklik tombol <i>Input Data</i> .	Mengarahkan pengguna ke halaman <i>Input Data</i> .	Sesuai
Halaman Home	Mengklik tombol <i>View</i> .	Mengarahkan pengguna ke halaman <i>View</i> .	Sesuai
Halaman Home	Mengklik tombol <i>Logout</i> .	Pengguna akan <i>logout</i> dan diarahkan ke halaman <i>Login</i> .	Sesuai
Halaman Input Data	Memasukkan data yang berisi ID yang berupa surel, umur, riwayat penyakit jantung, tingkat glukosa rata-rata, status perokok, jenis tempat tinggal, jenis kelamin, status pernikahan, dan riwayat hipertensi, lalu menekan tombol <i>Input</i> .	Menampilkan 3 pesan. Pesan pertama "Data kriteria berhasil dimasukkan", lalu pesan kedua "Data diagnosa berhasil dimasukkan", lalu pesan terakhir yang menampilkan hasil diagnosa apakah positif/negatif <i>stroke</i> .	Sesuai
Halaman Input Data	Memasukkan surel yang sesuai di <i>database</i> di kolom ID, lalu menekan tombol <i>Search</i> .	Menampilkan pemberitahuan nama pengguna ditemukan dengan nama pengguna dari surel tersebut.	Sesuai

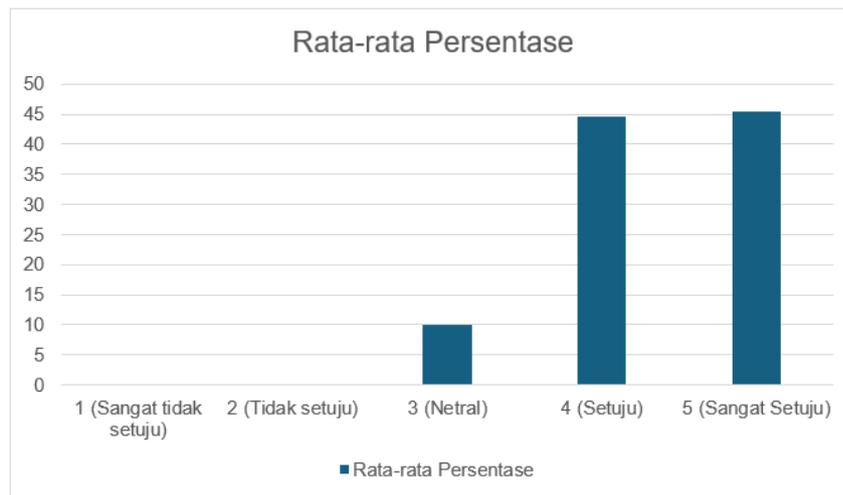
Halaman Input Data	Membiarkan kolom ID kosong atau memasukkan surel yang tidak ada di <i>database</i> , lalu menekan tombol <i>Search</i> .	Menampilkan pesan <i>error</i> "Email tidak ditemukan dalam database".	Sesuai
Halaman Input Data	Membiarkan semua atau beberapa kolom di formulir input data kosong, lalu menekan tombol <i>Input</i> .	Menampilkan pesan <i>error</i> .	Sesuai
Halaman View	Mengklik tombol <i>Refresh</i> .	Menampilkan ulang halaman <i>View</i> dengan data tabel yang sudah diperbarui.	Sesuai
Halaman View	Memilih salah satu data pada tabel, lalu menekan tombol <i>Delete</i> .	Menampilkan pesan konfirmasi "Apakah kamu yakin ingin menghapus data ini?", jika menekan <i>Yes</i> maka akan menampilkan pesan "Data berhasil dihapus" dan melakukan <i>refresh</i> data di tabel halaman <i>View</i> , sedangkan jika menekan <i>No</i> maka tidak akan terjadi apa-apa dan kembali ke halaman <i>View</i> .	Sesuai
Halaman View	Memilih salah satu data pada tabel, lalu menekan tombol <i>Update</i> .	Mengarahkan pengguna ke halaman <i>Update</i> untuk menyunting data yang dipilih.	Sesuai
Halaman View	Tidak memilih data yang ingin dihapus pada tabel, lalu menekan tombol <i>Delete</i> .	Menampilkan pesan <i>error</i> bahwa pengguna harus memilih baris yang ingin dihapus.	Sesuai
Halaman View	Tidak memilih data yang ingin diupdate pada tabel, lalu menekan tombol <i>Update</i> .	Menampilkan pesan <i>error</i> bahwa pengguna harus memilih baris yang ingin diupdate.	Sesuai
Halaman Update	Mengubah dan memperbaharui data yang sudah dipilih di halaman <i>View</i> dengan mengubah data di kolom-kolom yang sudah terisi sebelumnya dengan data-data baru, lalu menekan tombol <i>Update</i> .	Menampilkan 3 pesan. Pesan pertama "Data berhasil diupdate di dalam tabel Diagnosa", lalu pesan kedua "Data kriteria berhasil disimpan di dalam tabel Kriteria", lalu pesan terakhir yang menampilkan hasil diagnosa apakah positif/negatif <i>stroke</i> yang sudah diupdate.	Sesuai
Halaman Update	Membiarkan semua atau beberapa kolom di formulir <i>update</i> data kosong, lalu menekan tombol <i>Update</i> .	Menampilkan pesan <i>error</i> .	Sesuai
Halaman Update	Mengklik tombol <i>Back</i> .	Mengarahkan pengguna kembali ke halaman <i>View</i> .	Sesuai

Kuesioner

Dari 11 responden yang menjawab kuesioner di atas dengan 10 pertanyaan, rata-rata persentase yang menjawab 1 (sangat tidak setuju) adalah 0%, 2 (tidak setuju) adalah 0%, 3 (netral) adalah 10%, 4 (setuju) adalah 44,56%, dan 5 (sangat setuju) adalah 45,44%.

Tabel 2 Rata-rata Persentase Keseluruhan Kuesioner

Jawaban	Rata-rata Persentase
1 (Sangat tidak setuju)	0%
2 (Tidak setuju)	0%
3 (Netral)	10%
4 (Setuju)	44,56%
5 (Sangat setuju)	45,44%



Gambar 15 Grafik Rata-Rata Persentase Keseluruhan Kuesioner

Kesimpulan

Pengujian dan evaluasi yang sudah dilakukan menghasilkan simpulan sebagai berikut:

1. Algoritma pohon keputusan di *data mining* dapat diterapkan dalam mendeteksi penyebab seseorang terkena *stroke*. Setelah dilakukan perhitungan manual dan menggunakan perangkat lunak *RapidMiner*, akurasi hasil prediksi menjadi tolok ukur seberapa efektif algoritma ini dalam mengidentifikasi penyebab *stroke* pada seseorang.
2. Implementasi algoritma C4.5 dengan perangkat lunak *RapidMiner* pada dataset *Kaggle* terkait kasus *stroke* menghasilkan model klasifikasi dengan tingkat akurasi 94,89%. Akurasi tersebut dihitung berdasarkan *confusion matrix* ($True\ Positive + True\ Negative$) / ($True\ Positive + False\ Positive + True\ Negative + False\ Negative$). Nilai AUC sebesar 0.709 mengindikasikan bahwa model memiliki performa klasifikasi yang baik dalam mengidentifikasi faktor-faktor penyebab *stroke*.
3. Dari 11 responden yang menjawab kuesioner di atas dengan 10 pertanyaan, rata-rata persentase yang menjawab 1 (sangat tidak setuju) adalah 0%, 2 (tidak setuju) adalah 0%, 3 (netral) adalah 10%, 4 (setuju) adalah 44,56%, dan 5 (sangat setuju) adalah 45,44%.

Referensi :

- [1] World Health Organization, *Noncommunicable diseases*. 2018.
- [2] Kemenkes RI, *Laporan Riskesdas 2018 Nasional*. Jakarta: Kemenkes RI, 2019.
- [3] S. E. P. Munce, I. D. Graham, N. M. Salbach, S. B. Jaglal, C. L. Richards, J. J. Eng, J. Desrosiers, M. MacKay-Lyons, S. Wood-Dauphinee, N. Korner-Bitensky, N. E. Mayo, R. W. Teasell, M. Zwarenstein, J. Mokry, S. Black, and M. T. Bayley. "Perspectives of health care professionals on the facilitators and barriers to the implementation of a stroke rehabilitation guidelines cluster randomized controlled trial," *BMC Health Services Research*, vol. 17, no. 1, 2017.
- [4] Y. Turana, J. Teng kawan, Y. C. Chia, M. Nathaniel, J. G. Wang, A. Sukonthasarn, C. H. Chen, H. Van Minh, P. Buranakitjaroen, J. Shin, S. Siddique, J. M. Naites, S. Park, B. W. Teo, J. Sison, A. Ann Soenarta, S. Hoshide, J. C. Tay, G. Prasad Sogunuru, ... K. Kario. "Hypertension and stroke in Asia: A comprehensive review from HOPE Asia," *Journal of Clinical Hypertension*, vol. 23, no. 3, p. 513–521, 2021.
- [5] I. Werdiningsih, B. Nuqoba, and Muhammadun, *Data Mining Menggunakan Android, Weka, dan SPSS*. Surabaya: Airlangga University Press, 2020.
- [6] Kani, *Algoritma dan Pemrograman*. Tangerang Selatan: Universitas Terbuka, 2020.
- [7] A. Azahari and N. Nursobah. "Rekomendasi Penerimaan Beasiswa Yayasan Untuk Siswa Baru SMK TI Airlangga dengan Algoritma C4.5". *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 5, no. 2, p. 609–614, 2021.
- [8] Sugiyono. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta, 2015.
- [9] A. S. Osman. "Data Mining Techniques: Review," *International Journal of Data Science Research*, vol. 2, no. 1, p. 1–5, 2019.
- [10] J. Suntoro, *Data Mining: Algoritma dan Implementasi dengan Pemrograman PHP*. Jakarta: Elex Media Komputindo, 2019.
- [11] A. R. Sukma, R. Halfis, and A. Hermawan. "Klasifikasi Channel Youtube Indonesia Menggunakan Algoritma C4.5," *Jurnal Teknik Komputer*, vol. 5, no. 1, p. 21–28, 2019.
- [12] E. Mardiani, N. Rahmansyah, H. Kurniawan, A. Muliawati, and D. S. Permana, *Membuat Aplikasi Penjualan Menggunakan Java Netbeans, MySQL, dan iReport*. Jakarta: Elex Media Komputindo, 2017.
- [13] Nofriadi, *Java Fundamental dengan Netbeans 8.0.2*. Sleman: Deepublish, 2015.
- [14] Maya, *Membangun Sistem Informasi dengan Java NetBeans dan MySQL*. Yogyakarta: Penerbit Andi, 2015.
- [15] A. C. Putri, F. E. Hariyanto, N. L. E. Andini, and Z. C. S. Zulkarnaen. "Klasifikasi Rumah Tangga Miskin di Provinsi Papua Tahun 2017 Menggunakan Metode Naive Bayes," *Jurnal Sains Matematika Dan Statistika*, vol. 7, no. 1, p. 89–95, 2021.
- [16] T. Wahyudi, M. Kusmira, and A. B. Hikmah. "Penerapan Data Mining Pengajuan Pembiayaan Perumahan (Consumen Financing) Individual Menggunakan Algoritma C4.5," *Jurnal Khatulistiwa Informatika*, vol. 7, no. 1, p. 66–71, 2019.
- [17] R. Abdulloh, *7 in 1 Pemrograman Web untuk Pemula*. Jakarta: Elex Media Komputindo, 2018.
- [18] J. K. Kawistara and P. Hidayatullah, *Pemrograman Web (Cetakan Pertama)*. Bandung: Informatika, 2017.
- [19] R. Kurniawan and S. Marhamelda. "SISTEM PENGOLAHAN DATA PESERTA DIDIK PADA LKP PRIMA TAMA KOMPUTER DUMAI DENGAN MENGGUNAKAN BAHASA PEMROGRAMAN PHP," *INFORMATIKA*, vol. 11, no. 1, p. 37–45, 2019.
- [20] R. Roza, M. N. Fauzan, and W. I. Rahayu, *Tutorial Sistem Informasi Prediksi Jumlah Pelanggan Menggunakan Metode Regresi Linier Berganda Berbasis Web Menggunakan Framework Codeigniter*. Bandung: Kreatif Industri Nusantara, 2020.
- [21] B. A. Priyaangga, D. B. Aji, M. Syahroni, N. T. S. Aji, and A. Saifudin. "Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions," *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, vol. 3, no. 3, p. 150–157, 2020.