IMPLEMENTASI DECISION TREE CLASSIFIER PADA SISTEM REKOMENDASI PENERIMAAN MAHASISWA BARU FAKULTAS TEKNOLOGI INFORMASI UKDW

Skripsi

oleh:

JOSHUA PUTRA SETYADI
71170173

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2023

IMPLEMENTASI DECISION TREE CLASSIFIER PADA SISTEM REKOMENDASI PENERIMAAN MAHASISWA BARU FAKULTAS TEKNOLOGI INFORMASI UKDW

Skripsi

Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

JOSHUA PUTRA SETYADI
71170173

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
2023

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

IMPLEMENTASI DECISION TREE CLASSIFIER PADA SISTEM REKOMENDASI PENERIMAAN MAHASISWA BARU FAKULTAS TEKNOLOGI INFORMASI UKDW

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 25 Januari 2023

JOSHUA PUTRA SETYADI
71170173

HALAMAN PERSETUJUAN

Judul Skripsi	:	IMPLEMENTASI DECISION TREE CLASSIFIER PADA SISTEM REKOMENDASI PENERIMAAN MAHASISWA BARU FAKULTAS TEKNOLOGI INFORMASI UKDW
Nama Mahasiswa	:	JOSHUA PUTRA SETYADI
N I M	:	71170173
Matakuliah	:	Skripsi (Tugas Akhir)
Kode	:	TI0366
Semester	:	Gasal
Tahun Akademik	:	2022/2023

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 25 Januari 2023

Dosen Pembimbing I	Dosen Pembimbing II
Lucia Dwi Krisnawati, Dr. Phil.	Yuan Lukito, S.Kom., M.Cs.

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA

Saya yang bertanda tangan di bawah ini:

NIM	:	71170173
Nama	:	Joshua Putra Setyadi
Prodi / Fakultas	:	Teknologi Informasi / Informatika
Judul Tugas Akhir	:	Implementasi Decision Tree Classifier Pada Sistem Rekomendasi Penerimaan Mahasiswa Baru Fakultas Teknologi Informasi UKDW

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan **Hak Bebas Royalti Non Ekslusif** (*Non-exclusive Royalty-free Right*) serta bersedia Tugas Akhirnya dipublikasikan secara online dan dapat diakses secara lengkap (*full access*).

Dengan Hak Bebas Royalti Noneksklusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk *database*, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 27 Januari 2023

Yang menyatakan,

71170173 – Joshua Putra Setyadi

HALAMAN PENGESAHAN

IMPLEMENTASI DECISION TREE CLASSIFIER PADA SISTEM REKOMENDASI PENERIMAAN MAHASISWA BARU FAKULTAS TEKNOLOGI INFORMASI UKDW

Oleh: JOSHUA PUTRA SETYADI / 71170173

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 4 Januari 2023

Yogyakarta, 25 Januari 2023
Mengesahkan,

Dewan Penguji:

1. *Lucia Dwi Krisnawati, Dr. Phil.*
2. Yuan Lukito, S.Kom., M.Cs.
3. Laurentius Kuncoro Probo Saputra, S.T., M.Eng.
4. R. Gunawan Santosa, Drs. M.Si.

Dekan	Ketua Program Studi
(Restyandito,S.Kom.,MSIS.,Ph.D.)	(Gloria Virginia,Ph.D.)

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS SECARA ONLINE UNIVERSITAS KRISTEN DUTA WACANA YOGYAKARTA

Saya yang bertanda tangan di bawah ini:

NIM	:	71170173
Nama	:	Joshua Putra Setyadi
Prodi / Fakultas	:	Teknologi Informasi / Informatika
Judul Tugas Akhir	:	Implementasi Decision Tree Classifier Pada Sistem Rekomendasi Penerimaan Mahasiswa Baru Fakultas Teknologi Informasi UKDW

bersedia menyerahkan Tugas Akhir kepada Universitas melalui Perpustakaan untuk keperluan akademis dan memberikan **Hak Bebas Royalti Non Ekslusif** (*Non-exclusive Royalty-free Right*) serta bersedia Tugas Akhirnya dipublikasikan secara online dan dapat diakses secara lengkap (*full access*).

Dengan Hak Bebas Royalti Noneksklusif ini Perpustakaan Universitas Kristen Duta Wacana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk *database*, merawat, dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenar-benarnya.

Yogyakarta, 27 Januari 2023

Yang menyatakan,

71170173 – Joshua Putra Setyadi

Karya sederhana ini dipersembahkan kepada Kedua Orang Tua,

Kedua Dosen Pembimbing, dan Adikku Ezra Putra Setyadi,

Tanpanya, karya ini mungkin selesai setahun lebih dulu

If fighting is sure to result in victory, then you must fight

Sun Tzu

KATA PENGANTAR

Segala puji dan syukur kepada Tuhan yang maha kasih, karena atas segala rahmat, bimbingan, dan bantuan-Nya maka akhirnya Skripsi dengan judul "Implementasi Decision Tree Classifier Pada Sistem Rekomendasi Penerimaan Mahasiswa Baru Fakultas Teknologi Informasi UKDW" ini telah selesai disusun.

Penulis memperoleh banyak bantuan dari kerja sama baik secara moral maupun spiritual dalam penulisan Skripsi ini, untuk itu tak lupa penulis ucapkan terima kasih yang sebesar-besarnya kepada:

1. Tuhan yang maha kasih,
2. Orang tua yang selama ini telah sabar membimbing, memberi dana dan mendoakan penulis tanpa kesal untuk selama-lamanya,
3. Restyandito, S.Kom, MSIS., Ph.D selaku Dekan FTI, yang menandatangani pengesahan seminar,
4. Gloria Virginia, S.Kom., MAI., Ph.D selaku Kaprodi Informatika, yang menjadi narasumber dosen seleksi FTI dan yang memberikan topik penelitian kepada penulis,
5. Dr. Phil. Lucia Dwi Krisnawati selaku Dosen Pembimbing II, yang telah memberikan ilmunya dan dengan penuh kesabaran membimbing penulis,
6. Yuan Lukito, S.Kom., M.Cs, selaku Dosen Pembimbing II, yang telah memberikan ilmu dan kesabaran dalam membimbing penulis,
7. Lain-lain yang telah mendukung moral, spiritual untuk belajar selama ini.

Laporan skripsi ini tidak lepas dari segala kekurangan dan kelemahan, untuk itu segala kritikan dan saran yang bersifat membangun guna kesempurnaan skripsi ini sangat diharapkan. Semoga proposal/skripsi ini dapat bermanfaat bagi pembaca semua dan lebih khusus lagi bagi pengembangan ilmu komputer dan teknologi informasi.

Yogyakarta, 7 Desember 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

IMPLEMENTASI DECISION TREE CLASSIFIER PADA SISTEM REKOMENDASI PENERIMAAN MAHASISWA BARU FAKULTAS TEKNOLOGI INFORMASI UKDW

Oleh

JOSHUA PUTRA SETYADI

71170173

Seleksi penerimaan mahasiswa baru Universitas Kristen Duta Wacana mempunyai dua jalur pendaftaran, seleksi jalur reguler, dan jalur prestasi. Untuk membantu Fakultas Teknologi Informasi dalam melakukan seleksi, dikembangkanlah sebuah purwarupa sistem yang mampu memberikan rekomendasi penerimaan berdasarkan ciri-ciri mahasiswa yang diterima sebelumnya.

Penelitian ini menggunakan metode pembelajaran mesin *decision tree* dan *random forest* untuk mempelajari pola-pola data penerimaan mahasiswa tahun 2015 sampai 2021 yang disimpan gudang data Fakultas Teknologi Informasi. Data tersebut berupa sekolah asal, pilihan program studi, dan nilai-nilai tes potensial akademik calon mahasiswa pada seleksi jalur reguler, dan rata-rata nilai UAN dan rapor pada seleksi jalur prestasi. Performa luaran-luaran metode tersebut, berupa rata-rata akurasi dan nilai presisi, sensitivitas, *f-score*, dan *feature importance* kemudian diukur dan dibandingkan.

Berdasarkan hasil yang diperoleh dalam melakukan seleksi jalur reguler, algoritma *random forest* mempunyai *f-score* yang lebih tinggi daripada *decision tree* sebesar 81% berbanding 73%. Pada seleksi jalur prestasi, algoritma *decision tree* lebih baik dengan nilai *f-score* yang lebih tinggi daripada *random forest* sebesar 93% berbanding 91%. *Feature importance* luaran-luaran metode menyimpulkan bahwa variabel yang paling menentukan penerimaan mahasiswa jalur reguler

adalah pilihan program studi pertama dengan nilai *gini* model *decision tree* sebesar 0,338 dan model *random forest* sebesar 0,197, dan rata-rata nilai rapor paling menentukan penerimaan mahasiswa apabila seleksi melalui jalur prestasi dengan nilai *gini* model *decision tree* sebesar 0,736 dan model *random forest* sebesar 0,543.

Kata-kata kunci : penerimaan mahasiswa baru, klasifikasi, prediksi, *decision tree*

ABSTRACT

DECISION TREE CLASSIFIER IMPLEMENTATION FOR DUTA WACANA CHRISTIAN UNIVERSITY NEW STUDENT ENROLLMENT RECOMMENDATION SYSTEM

By
JOSHUA PUTRA SETYADI
71170173

The enrollment of Duta Wacana Christian University is categorized to regular and scholarship. To assist student enrollment of Information Technology Faculty, a web-based application prototype capable of classifying and recommending enrollees based on the characteristics of previously accepted students is developed.

This research applies decision tree and random forest learning method to recognize patterns in the student registration data of 2015 to 2021 which is stored in IT Faculty's Data Warehouse. These data include school of origin, study program selections, academic potential test scores for the regular enrollment, and high school report card average score for the scholarship enrollment. Learning performance outputted from these methods is measured and compared to one another based on their average accuracy, precision, sensitivity, f-score, and feature importance.

These comparisons conclude that random forest model offers better learning performance with higher f-score (81% > 73%) than decision tree model on classifying regular enrollment entries. Random forest model also offers smaller range of feature importance than decision tree model. However, decision tree model offers a higher accuracy (93% > 92%) than random forest model when classifying scholarship enrollments. Feature importance of these methods concludes the first choice of study program as its main feature of classifying regular enrollees, scoring 0,197 *gini* index value from the *random forest* model and 0,338 *gini* index value from the *decision tree* model. The average report card scores is the main

determinant of scholarship enrollment with 0,736 *gini* index value from *decision tree* model and 0,543 *gini* index value from the *random forest* model.

Keywords : student enrollment, classification, prediction decision tree,

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Universitas Kristen Duta Wacana (UKDW) selalu berusaha untuk menerima lebih banyak mahasiswa baru, Hal tersebut berlaku untuk Fakultas Teknologi Informasi (FTI). Salah satu kendala yang dialami oleh Penerimaan Mahasiswa Baru Fakultas Teknologi Informasi UKDW adalah mengetahui mahasiswa yang cocok untuk masuk fakultas.

FTI UKDW sendiri mempunyai tim *Data Warehouse* (DW) yang bertugas untuk mengorganisasikan data mahasiswa UKDW maupun calon mahasiswa UKDW. Data calon mahasiswa tersebut antara lain adalah nilai Tes Potensi Akademik (TPA) verbal, spasial, analogi, numerik, rata-rata nilai Ujian Akhir Nasional (UAN) dan rapor Sekolah Menengah Atas (SMA) mereka. Sementara data mahasiswa tersebut berupa data nilai studi mahasiswa antara lain seperti transkripsi nilai dan Indeks Prestasi Kumulatif (IPK).

Apabila pola-pola pada data tersebut bisa digunakan untuk memprediksi keberhasilan studi calon mahasiswa tersebut berdasarkan penelitian sebelumnya maka sebuah model yang memberikan rekomendasi penerimaan mahasiswa baru bisa dikembangkan, dan tim promotor juga mampu menyesuaikan strategi pendekatan mereka dengan sekolah-sekolah menengah atas atau kejuruan secara lebih efisien karena dasar kompetensi siswa yang direkomendasi tidak berasal dari asumsi, melainkan dari analisis data.

Oleh karena itu, penelitian ini mengusulkan implementasi model algoritma *decision tree* untuk mengembangkan sebuah sistem rekomendasi terima masuk FTI UKDW. Model akan mengolah data nilai-nilai calon mahasiswa dari Tim DW FTI UKDW untuk memprediksikan penerimaan calon-calon mahasiswa

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas, adapun rumusan masalah penelitian sebagai berikut.

1. Pengumpulan data dari Data Warehouse FTI & Puspindika UKDW
2. Pengembangan algoritma *decision tree* pada model untuk analisis data yang ditambang
3. Implementasi model ke dalam sebuah sistem untuk melakukan seleksi penerimaan calon-calon mahasiswa masuk FTI UKDW

1.3. Batasan Masalah

Adapun batasan masalah penelitian ini yaitu ketersediaan data dari DW FTI. Sehingga, data latih dan data uji yang digunakan adalah data PMB FTI UKDW Angkatan 2015-2021. Keluaran dari purwarupa sistem akan memberikan rekomendasi penerimaan calon mahasiswa hanya untuk FTI. Data yang akan dipelajari luaran model bertipe angka.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah menerapkan algoritma *decision tree* pada model *machine learning* untuk mempelajari data seleksi penerimaan mahasiswa baru. Dengan luaran model, penulis mengembangkan sistem rekomendasi masuk FTI bagi calon-calon mahasiswa yang mempunyai potensial sukses kuliah pada FTI UKDW.

1.5. Manfaat Penelitian

Manfaat dari sistem yang dihasilkan dari penelitian ini bagi dekanat FTI UKDW adalah menjadi salah satu pertimbangan dalam seleksi penerimaan mahasiswa baru. Bagi admisi dan promosi, penelitian ini menjawab faktor yang paling menentukan penerimaan mahasiswa FTI UKDW.

1.6. Metodologi Penelitian

Adapun beberapa tahap yang dilakukan untuk mencapai hasil akhir penelitian, antara lain:

1. Studi Pustaka

   Penulis melakukan studi pustaka untuk menemukan topik-topik penelitian yang berkaitan dengan implementasi metode pembelajaran mesin pada data-data mahasiswa. Informasi yang didapat dari studi akan memperkuat pendapat yang ada.

2. Wawancara Narasumber

   Penulis melakukan wawancara dengan dosen-dosen yang melakukan seleksi penerimaan mahasiswa baru untuk FTI. Hal ini dilakukan sebagai analisis kebutuhan purwarupa sistem dan mendapatkan informasi tentang apa yang bisa dilakukan pada penelitian.

3. Pengolahan Data

   Penulis mengambil data pendaftaran mahasiswa baru FTI tahun angkatan 2015-2021 dari DW FTI dan mengembangkan *dataset* yang menjadi data latih dan uji luaran model pembelajaran.

4. Pengembangan Luaran Model

   Penulis mengembangkan luaran-luaran dari metode-metode model yang dipelajari saat melakukan studi pustaka dalam rangka mempelajari pola-pola klasifikasi pada *dataset* yang sudah dikembangkan. Luaran-luaran tersebut akan disimpan untuk dimuat pada purwarupa sistem.

5. Rancangan

   Penulis merancang antarmuka dan fitur-fitur dari purwarupa yang diperlukan untuk memenuhi *requirement* yang didapat dari wawancara narasumber.

6. Implementasi

   Penulis mengimplementasikan sistem yang sudah dirancang. Pengembangan menggunakan alur kerja *agile*. Purwarupa sistem akan memuat luaran model untuk melakukan klasifikasi dan penghitungan

prediksi pada masukan data. Purwarupa sistem diimplementasikan pada platform *streamlit cloud* agar bisa diakses oleh pengguna.

7. Analisis Sistem

   Penulis melakukan analisis purwarupa sistem dengan mencatat apa saja yang bisa dilakukan pada implementasi rancangan, kemudian membandingkan hasil uji performa luaran-luaran model dan *feature importance* untuk menentukan variabel yang paling digunakan oleh luaran model dalam melakukan klasifikasi pada *dataset*.

8. Kesimpulan dan Saran

   Penulis mengetahui batasan implementasi sistem, kelebihan dan kekurangan model-model metode yang digunakan. Hasil dari tahap analisis akan dirangkum lebih lanjut untuk menjadi kesimpulan, dan kekurangan yang dianalisis akan dirangkum sebagai saran penulis.

1.7. Sistematika Penulisan

Laporan/Proposal skripsi ini disusun dengan sistematika bagian pertama, terdiri dari empat bab: Bab 1 berisi pendahuluan yang membahas latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan pernyataan keaslian disertasi. Bab 2 yaitu Tinjauan Pustaka dan Landasan Teori yang berisi tinjauan pustaka tentang penelitian-penelitian terkait, dan berbagai tinjauan pustaka spesifik, yaitu tentang *support vector machine*, metrik evaluasi, dan metode normalisasi data.

Bab 3 yaitu Metodologi Penelitian, yang berisi rinci perancangan penelitian, pengembangan sistem beserta dengan kebutuhannya. Bab 4 yaitu Implementasi dan Pembahasan, berisi penerangan implementasi sistem rekomendasi PMB, hasil evaluasi performa luaran model, dan pembahasan penerapan pengujian perangkat lunak beserta analisis hasil uji model. Bab 5 akan membahas kesimpulan dan saran dari hasil penelitian.

BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

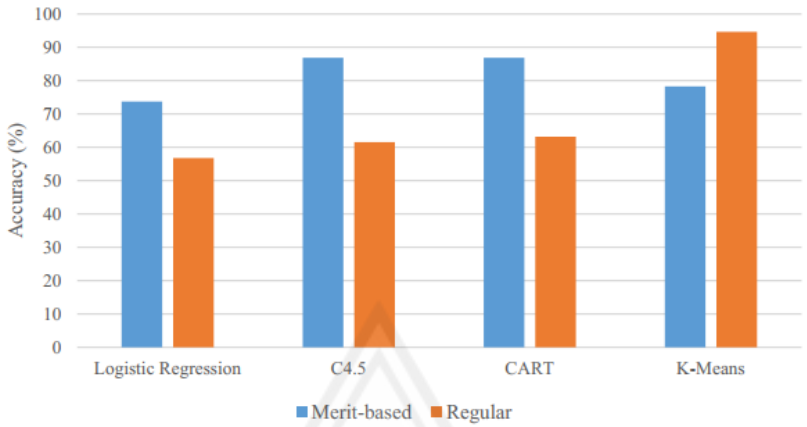
Untuk meninjau proses pengembangan purwarupa perangkat lunak, penulis meninjau Manoe, R (2012) yang mengimplementasikan algoritma *decision tree* dalam analisis pengambilan keputusan PMB jalur reguler UKDW untuk angkatan tahun 2007 sampai dengan 2011. Program hasil implementasi mampu mengambil dan mengolah masukan data penerimaan mahasiswa baru menjadi sebuah *dataset* yang bisa dianalisa oleh *decision tree*. Program juga mampu mengkategorikan pada tingkat nilai TPA *dataset* menggunakan metode statistika kuartil.

Hasil penelitian menyimpulkan bahwa pada pertimbangan keputusan PMB UKDW untuk angkatan tahun 2007-2010, nilai TPA bidang numerik menjadi pertimbangan utama. Sementara nilai TPA verbal menjadi penentu utama pada penerimaan mahasiswa angkatan tahun 2011. Kesimpulan lain yang didapat adalah bahwa faktor lain di luar hasil tes TPA yang tidak terklasifikasikan oleh *decision tree* dapat mempengaruhi pertimbangan keputusan, seperti gelombang atau tanggal tes, penyertaan berkas rapor, ijazah, atau SKHUN.

Alverina, Chrismanto, & Santosa (2018) membandingkan performa akurasi implementasi algoritma C4.5 dengan algoritma CART dalam melakukan prediksi kategori nilai IP semester pertama mahasiswa. Kedua algoritma dilatih pada data pendaftaran mahasiswa baru tahun 2008 sampai dengan 2014 yang berisi jenis (tipe), status, lokasi sekolah menengah. Kemudian luaran algoritma diuji pada data pendaftaran mahasiswa baru angkatan 2015. Berdasarkan jalur penerimaan mahasiswa, Data tersebut kemudian dibagi menjadi dua yang mana data jalur prestasi ditambahkan nilai level kompetensi bahasa inggris mahasiswa, sementara data jalur non-prestasi ditambahkan nilai TPA numerik, verbal, spasial dan analogi.

Akurasi dari algoritma C4.5 dan CART diujikan pada 24 skenario kumpulan data yang mempunyai variasi seperti *binning* variabel-variabel numerik, seimbangnya data, dan jumlah kategori prediksi. Hasil penelitian menunjukkan bahwa dalam memprediksi data jalur prestasi, kedua algoritma memberikan akurasi yang sama, sebesar 86,86% tetapi pada data jalur non-prestasi, algoritma CART memberikan akurasi yang lebih baik daripada C4.5. (63,16% berbanding 61,54%)

Santosa, Lukito, & Chrismanto (2021) melakukan klasifikasi dan prediksi nilai IPK mahasiswa Universitas Kristen Duta Wacana menggunakan algoritma *k-means* untuk meningkatkan akurasi prediksi. Pelatihan dilakukan pada data PMB 2008-2017 dan pengujian pada PMB tahun 2018. Variabel dataset antara lain status, lokasi, tipe SMA calon mahasiswa, dengan data nilai tes potensial akademik analogi, verbal, spasial, numerik calon mahasiswa dan nilai ICE yang mewakili kemampuan berbahasa inggris calon mahasiswa. Perincian lebih lanjut variabel ditulis pada Tabel 2.1. Nilai *k* yang diberikan untuk algoritma bernilai $k = 2$ berdasarkan jumlah kategori yang diinginkan, yakni prediksi nilai IPK tinggi dan rendah. Metrik akurasi, presisi, *recall* dan *f*1-measure algoritma kemudian dibandingkan dengan algoritma yang dilakukan pada penelitian sebelumnya yang menggunakan algoritma *logistic regression*, C4.5, dan CART. Hasil penelitian diilustrasikan pada Gambar 2.1 bahwa metode K-means memberikan performa yang lebih tinggi pada data jalur reguler dengan rata-rata akurasi sebesar 94,627%, sementara metode C4,5 dan CART memberikan akurasi yang lebih tinggi pada data jalur prestasi, yakni sebesar 86,86% merujuk pada Alverina, Chrismanto, & Santosa (2018).



Gambar 2.1 Perbandingan akurasi algoritma pada penelitian sebelumnya (Santosa, Lukito, & Chrismanto, 2021)

Variable Name	Variable Description	Possible Value
X1	High school status	Public = 1, Private = 2
X2	High school location	Java = 1, Outside Java = 2
X3	High school type	SMU = 1, SMK = 2
X4	English language capability	Level 1, 2, 3 dan 4
X5	First Semester GPA	0 – 4.0
X6	Numeric score	0 – 200
X7	Verbal score	0 – 200
X8	Spatial score	0 – 200
X9	Analogy/Logic	0 – 200

Tabel 2.1 Tabel variabel data penelitian (Santosa, Lukito, & Chrismanto, 2021)

Pustaka-pustaka yang ditinjau mendukung yang akan dilakukan Penulis bahwa algoritma *decision tree* dan *random forest* secara umum memberikan performa prediksi yang lebih tinggi, dan durasi pembelajaran yang relatif lebih singkat dibandingkan algoritma-algoritma lainnya yang termodifikasi sekalipun.

2.2 Landasan Teori

2.2.1 Penerimaan Mahasiswa Baru TI UKDW

Berdasarkan situs web pendaftaran mahasiswa baru UKDW, seleksi penerimaan mahasiswa baru fakultas TI dibagi menjadi dua jalur, yaitu jalur reguler dan jalur seleksi.

Pada jalur reguler, seleksi penerimaan dilakukan dua gelombang, yang mana pada setiap gelombang seleksi calon mahasiswa mengerjakan tes potensi

akademik sesuai dengan tanggal yang ditentukan. Hasil dari tes tersebut menentukan penerimaan calon mahasiswa. Apabila calon mahasiswa gagal diterima pada gelombang pertama, maka calon bisa mendaftar pada gelombang kedua.

Pada jalur prestasi, seleksi penerimaan dilakukan berdasarkan nilai rapor. Nilai rapor yang digunakan adalah nilai kelas 10 dan 11 SMA, SMK, ataupun Setara, dengan ketentuan nilai mata pelajaran yang diambil berupa Bahasa Inggris dan Matematika untuk calon mahasiswa berjurusan IPA, IPS, Bahasa, atau Kejuruan. Adapun berdasarkan wawancara dosen-dosen program studi Informatika sebagai narasumber, seleksi yang dilakukan diluar skema jalur prestasi dilakukan berdasarkan wawancara untuk mengukur nilai-nilai kualitas calon mahasiswa, seperti minat dan motivasi.

2.2.2 Machine Learning

Machine Learning dapat secara luas didefinisikan sebagai metode komputasi menggunakan pengalaman untuk membantu pekerjaan atau membuat prediksi yang akurat (Mohri, Rostamizadeh, & Talwalkar, 2018). Definisi formal *machine learning* diberikan oleh Mitchell (1997) yaitu, "Suatu program komputer yang belajar dari pengalaman *E* sehubungan dengan beberapa kelas tugas *T* dan kinerja ukuran *P*, jika kinerjanya pada tugas-tugas di *T*, sebagaimana diukur dengan *P*, meningkat dengan pengalaman *E*."

Jika diberikan *dataset* yang diobservasi *X*, set parameter θ, dan model pembelajaran $f(\theta)$, *machine learning* digunakan untuk meminimalisir kesalahan pembelajaran $E(f(\theta), X)$ antara model pembelajaran dan kebenaran dasar (*ground truth*). Tanpa hilangnya generalisasi, kita mendapatkan kesalahan pembelajaran menggunakan perbedaan antara hasil prediksi $f(\theta)$ dengan sampel dari *X* (Dua, 2013).

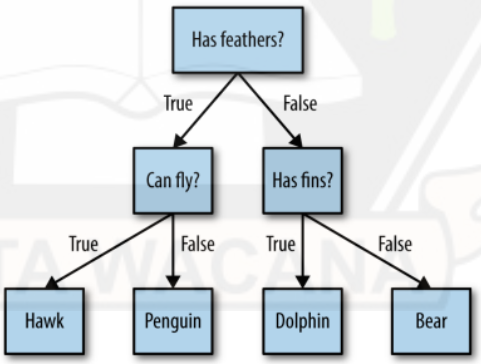
2.2.3 Supervised Learning

Supervised Learning lebih sering diasosiasikan untuk mesin klasifikasi, regresi, dan pemeringkat, yang mana sebuah mesin akan tidak hanya menempatkan sebuah data pada kelas atau peringkat tertentu, tetapi memberi prediksi numerik pada data tergantung jenis parameter apa yang diberikan pada data. Pada keseharian, mesin ini bisa digunakan untuk deteksi fitur-fitur dalam sebuah entitas, misalnya foto (Mohri, 2012).

Pada kategori ini, Mesin akan menerima sebuah himpunan data yang terlabel sebagai data latih dan memberikan label tersebut pada data yang tidak pernah dipakai oleh mesin, yakni data uji (Dua & Du, 2016).

2.2.3.1 Classification and Regression Tree (CART)

CART, diasosiasikan dengan *decision tree* pada basisnya adalah sebuah hirarki pernyataan-pernyataan "jika-maka" yang menghasilkan sebuah klasifikasi data. Seperti yang diilustrasikan pada Gambar 2.2, setiap antara node pohon mewakili sebuah pertanyaan, atau jawaban (disebut juga node ujung). Garis grafik pohon menghubungkan jawaban dengan pertanyaan yang ditentukan (Müller & Guido, 2016).



Gambar 2.2 Contoh sebuah decision tree klasifikasi binatang (Müller & Guido, 2016)

Model mulai dari seluruh *dataset S*, dan mencari setiap perbedaan nilai dari fitur-fitur *dataset* untuk mencari faktor-faktor pemisah yang paling optimal. *S* kemudian dipisah menjadi dua kelompok data, S_1 dan S_2 sehingga jumlah kuadrat

kesalahan *SSE* (*sums of square error*) diminimalkan sedemikian rupa sehingga $\overline{y_1}$ dan $\overline{y_2}$ masing-masing adalah rata-rata hasil data latih dari S_1 dan S_2. Kemudian dalam kelompok S_1 dan S_2 masing-masing metode dilakukan secara rekursif untuk mencari faktor pemisah dan nilai pisah yang mengurangi *SSE* secara efisien. Nilai *SSE* model CART dihitung sesuai dengan Persamaan 2.1 yakni sebagai berikut,

$$SSE = \sum_{i \epsilon S_1} (y_1 - \overline{y_1})^2 + \sum_{i \epsilon S_2} (y_2 - \overline{y_2})^2 \qquad [2.1]$$

Setiap node diberi metrik untuk mengukur efisiensi pemisahan, Akurasi apabila digunakan sebagai metrik mempunyai bias yang menyimpangkan fokus optimasi pemisahan pada mengurangi kesalahan klasifikasi, daripada mengklasifikasikan sebanyak mungkin data pada sebuah kelas. Sehingga, metrik yang digunakan sebagai alternatif akurasi adalah *Gini index*.

Proses penghitungan *gini index* mulai dengan pembagian sampel data oleh faktor pemisah yang dilakukan seperti Persamaan 2.1. Untuk klasifikasi dua-kelas, keluaran dari proses ini menghasilkan tabel kontingensi berukuran 2x2 pada setiap titik/node pemisahan, yang direpresentasikan pada Tabel 2.2 sebagai berikut,

Tabel 2.2 Tabel Representasi Nilai Probabilitas

	Class 1	Class 2	
$>$ split	n_{11}	n_{12}	n_{+1}
$\leq$ split	n_{21}	n_{22}	n_{+2}
	n_{1+}	n_{2+}	n

Gini index sebelum pemisahan (*split*) dihitung menurut Persamaan 2.2, di mana dengan *n* sebagai jumlah data yang diberi indeks kelas. Adapun Persamaan 2.2 sebagai berikut,

$$gini\ (sebelum\ split) = 2\left(\frac{n_{1+}}{n}\right)\left(\frac{n_{2+}}{n}\right) \qquad [2.2]$$

Gini index kemudian dihitung setelah pemisahan pada setiap node baru menurut Persamaan 2.3, yakni sebagai berikut,

$$gini\ (setelah\ split) = 2\left[\left(\frac{n_{11}}{n}\right)\left(\frac{n_{12}}{n_{+1}}\right) + \left(\frac{n_{21}}{n}\right)\left(\frac{n_{22}}{n_{+2}}\right)\right] \quad [2.3]$$

Sehingga nilai indeks gini sebuah node menjadi minimal bila kemungkinan klasifikasi kedua kelas tidak seimbang dengan kemungkinan salah satu kelas mendekati nol, dan lainnya mendekati satu. Sementara nilai indeks gini menjadi tinggi apabila kesempatan data diklasifikasikan ke dalam dua kelas sama besar (Kuhn & Johnson, 2018).

2.2.3.2 Random Forest Classifier

Random Forest menggabungkan kumpulan *decision tree*, teknik klasifikasi varians tinggi bias rendah untuk meningkatkan performa akurasinya (Awad & Khanna, 2015). Dengan membuat sampel *bootstrap* pada *dataset*, himpunan *decision tree* terdistribusi pada setiap sampel secara acak, sehingga hasil akhir klasifikasi mempunyai nilai varians rendah sehingga menghasilkan hasil klasifikasi yang berbeda setiap sampel dan *decision tree* (Kuhn & Johnson, 2018).

Algoritma *random forest* bisa disimpulkan sebagai berikut (Awad & Khanna, 2015),

- Untuk membuat *decision tree* sejumlah *B*, pilih sampel bootstrap *n* dari *dataset S*
- Untuk setiap sampel bootstrap, buatlah *decision tree*
- Pada setiap node pohon,
    - Faktor klasifikasi *m* dipilih secara acak dari semua faktor klasifikasi yang ada
    - Faktor klasifikasi yang memberikan pemisahan terbaik akan memisah node secara biner
    - Node selanjutnya dipilih acak dari himpunan *m* dari semua faktor klasifikasi yang ada dan melakukan langkah sebelumnya

- Jika ada data baru untuk klasifikasi, dilakukan voting dari himpunan *decision tree B*, dengan hasil mayoritas sebagai keluaran klasifikasi

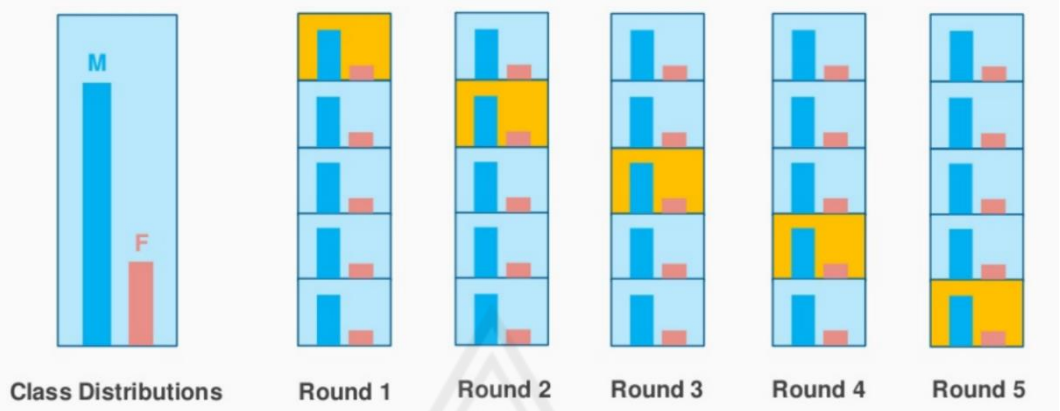
2.2.3.3 *Feature Importance*

Gini Importance, disebut juga *Feature Importance* adalah salah satu pendekatan umum untuk mengetahui fitur variabel *dataset* dengan pengaruh lemah pada model *decision tree* dan *random forest* yang didasarkan pada *Gini Index* (Doyen, et al., 2021) yang sudah dijelaskan pada sub-bab sebelumnya. Menghapus atau mengurangi fitur dengan nilai *feature importance* bisa mengurangi pelatihan berlebih dan mengatasi masalah *overfitting* pada model (Wang, et al., 2022).

Nilai ini tidak merepresentasikan kualitas prediktif secara intrinsik, melainkan seberapa pentingnya sebuah fitur *dataset* pada sebuah model tertentu (Pedregosa, et al., 2011). Fitur *dataset* yang mempunyai nilai *Feature Importance* rendah pada sebuah model, bisa menjadi lebih tinggi pada model lainnya. Sehingga untuk mengevaluasi performa sebuah model, lebih disarankan untuk menggunakan *cross validation* sebelum penghitungan *feature importance* (Buitinck, et al., 2013).

2.2.4 Stratified k-fold Cross Validation

Dataset akan dipartisi secara acak menjadi himpunan-himpunan data latih sejumlah *k*. Pada setiap fold, data diubah sedemikian rupa hingga jumlah data setiap kelas sama. Model kemudian menjalani proses pelatihan (*training*) menggunakan semua himpunan (*fold*) kecuali yang pertama, karena *fold* tersebut menjadi data uji luaran model untuk mengukur perfomanya. *Fold* tersebut kemudian digabungkan pada data latih, sementara *fold* kedua akan dijadikan sebagai data uji, dan proses ini diulang sejumlah *k* kali (Kuhn & Johnson, 2018). Untuk k = 5, proses ini diilustrasikan pada Gambar 2.3 di bawah.



Gambar 2.3 skema stratified *5-fold cross validation*. Sampel direpresentasikan setiap simbol dan dibagi menjadi 5 himpunan dengan jumlah data tiap sampel seimbang (Kuhn & Johnson, 2018).

2.2.5 Metrik Evaluasi

Untuk mengevaluasi performa klasifikasi *decision tree*, metrik yang digunakan adalah akurasi, presisi, dan *f1-score* (Soto-Murillo, et al., 2021). Akurasi adalah nilai persentase klasifikasi data positif dan negatif yang benar secara riil dari semua sampel yang diklasifikasi, diukur menurut Persamaan 2.3 sebagai berikut,

$$Accuracy = \frac{TruePositives + TrueNegatives}{Total\ Example} \qquad [2.4]$$

Sensitifitas atau *recall*, mengukur persentase *true positives* sedemikian rupa menurut menurut Persamaan 2.4, hingga dari semua sampel pasien yang sakit, berapa yang terdeteksi sebagai sakit. Adapun Persamaan 2.4 adalah sebagai berikut,

$$Sensitivity = \frac{TruePositives}{FalseNegatives + TruePositives} \qquad [2.5]$$

Spesifisitas mengukur persentase *true negatives* sedemikian rupa, hingga dari semua sampel pasien yang tidak sakit, berapa yang terdeteksi sebagai tidak sakit. Spesifitas dihitung menurut Persamaan 2.5 sebagai berikut,

$$Specificity = \frac{TrueNegatives}{FalsePositives + TrueNegatives} \quad [2.6]$$

Presisi mengukur konsistensi hasil klasifikasi ketika pengukuran diulang. Presisi dihitung menurut Persamaan 2.6 sebagai berikut,

$$Precision = \frac{TruePositives}{FalsePositives + TruePositives} \quad [2.7]$$

F1-score adalah rata-rata harmonik dari presisi dan sensitivitas, dan dihitung menurut Persamaan 2.7 sebagai berikut,

$$F1\ score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad [2.8]$$

BAB III
METODOLOGI PENELITIAN

3.1 Spesifikasi Perangkat Keras & Perangkat Lunak

Perangkat keras yang digunakan selama pengembangan aplikasi adalah laptop ASUS Vivobook A442U dengan prosessor Intel core i5-8250U, GPU Nvidia GeForce 930MX, dan SSD Samsung EVO 870 1TB, dan 8 GB RAM.

Perangkat lunak yang digunakan selama pengembangan aplikasi adalah bahasa pemrograman Python dengan platform distribusi Anaconda, lingkungan pengembangan kode Jupyter Lab, kerangka antarmuka Streamlit, dan API Scikit-Learn sebagai sumber kode algoritma *decision tree*, *random forest*, dan metrik evaluasi dan pengujian luaran model. Sumber kode purwarupa aplikasi serta implementasi model pembelajaran mesin disimpan dengan Git yang diakses dari Github.

3.2 Spesifikasi Kemampuan Sistem

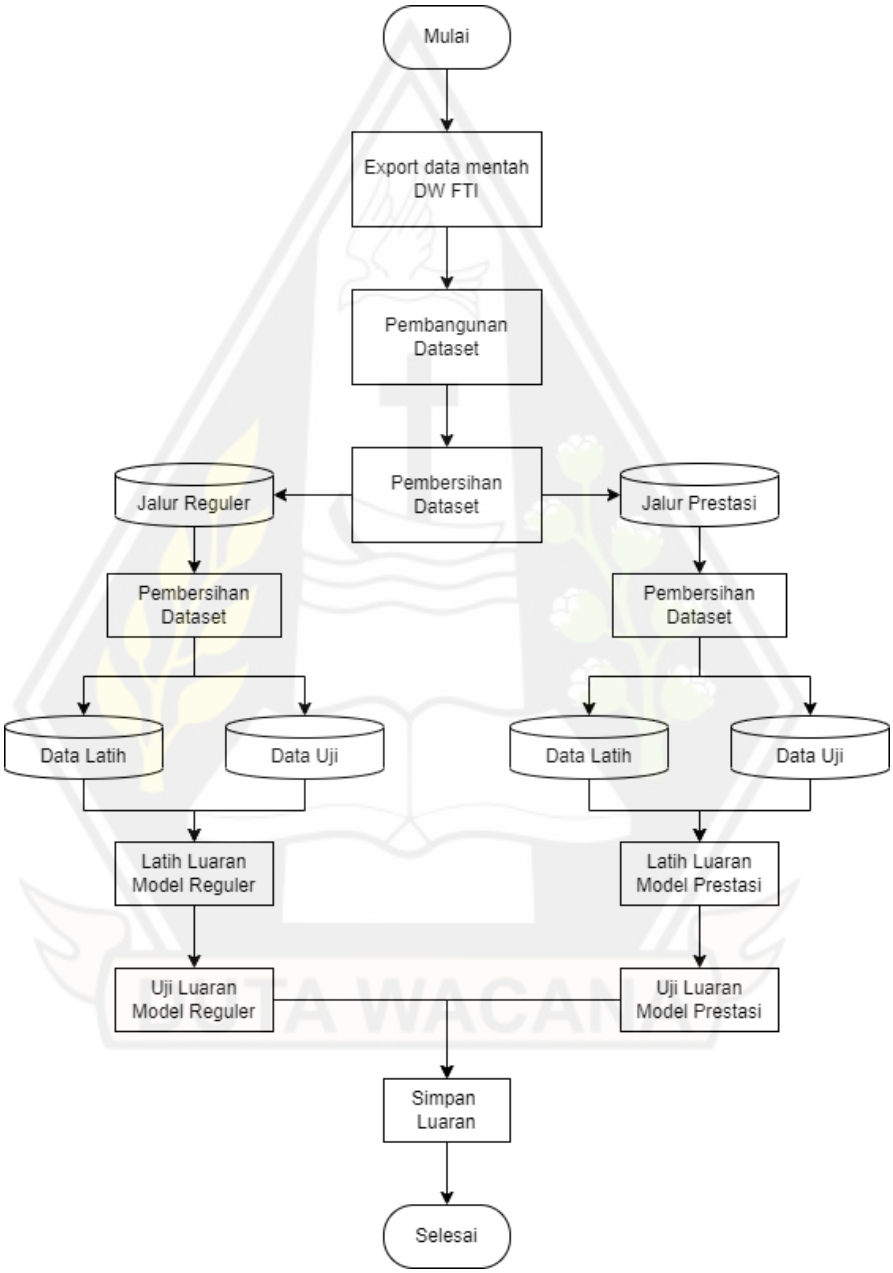
Aplikasi yang dikembangkan menghasilkan penghitungan serta hasil klasifikasi penerimaan calon mahasiswa dari masukan data lokasi, tipe, status, dan asal sekolah, pilihan program studi pertama dan kedua, serta nilai TPA calon mahasiswa untuk jalur reguler, atau rata-rata nilai ujian akhir nasional dan rapor sekolah calon mahasiswa untuk jalur prestasi.

Selama program berjalan, sekolah yang tidak terdaftar sebelumnya akan tersimpan pada daftar sekolah yang bisa diakses pada masukkan formulir sampai program dimuat ulang. Aplikasi juga mampu menerima unggahan data baru untuk digabungkan dengan data latih dan melakukan pelatihan ulang luaran model *random forest*.

3.3 Flowchart Sistem

3.3.1 Flowchart Pengembangan *Dataset* & Pembelajaran Mesin

Gambar 3.1 adalah diagram flowchart pengembangan pembelajaran mesin, yakni sebagai berikut,



Gambar 3.1 Diagram Flowchart Pengembangan Luaran Model ML

Untuk mengembangkan pembelajaran mesin, penulis melakukan studi pustaka dan dokumentasi algoritma *decision tree* dan *random forest* untuk melakukan klasifikasi. Penulis juga melakukan wawancara dosen-dosen yang melakukan seleksi penerimaan sebagai para narasumber untuk mempelajari proses penerimaan mahasiswa baru kedua jalur.

Penulis kemudian mengambil data yang diperlukan untuk pembuatan *dataset*. Praproses data meliputi pengembangan *dataset* dengan melakukan ekstraksi dan pengolahan fitur-fitur dari data mentah. Fitur-fitur yang digunakan sesuai dengan uraian sebelumnya pada Tinjauan Pustaka. Fitur-fitur yang redundan lalu dihapus untuk mengurangi dimensi *dataset* dan membantu menghemat daya mesin dalam mempelajari *dataset*. Dilakukan juga penanganan nilai-nilai pada fitur yang tidak memenuhi format data, seperti data bernilai nil, atau berbeda tipe.

Setelah pengolahan dan praproses data, *dataset* yang direncanakan untuk melatih dan menguji luaran model *random forest* mempunyai fitur-fitur seperti yang tertulis pada Tabel 3.1 yakni sebagai berikut,

Tabel 3.1 Tabel rancangan field *dataset* PMB

Variabel/Fitur *Dataset*	**I/O**	**Deskripsi Singkat**
lokasi	Input	Nilai biner mewakili bahwa asal calon mahasiswa dari jawa atau luar jawa
status	Input	Nilai biner mewakili status sekolah swasta atau negeri
tipe_sekolah	Input	Nilai biner mewakili tipe sekolah menengah atas atau sekolah menengah lainnya
id_daftar_sekolah	Input	Nomor ID sekolah menengah asal mahasiswa
id_prodi_pilihan_1	Input	Nomor ID program studi pilihan pertama calon mahasiswa
id_prodi_pilihan_2	Input	Nomor ID program studi pilihan kedua calon mahasiswa
nilai_tpa_verbal	Input	Nilai Tes Potensi Akademik verbal calon mahasiswa
nilai_tpa_spasial	Input	Nilai Tes Potensi Akademik spasial calon mahasiswa

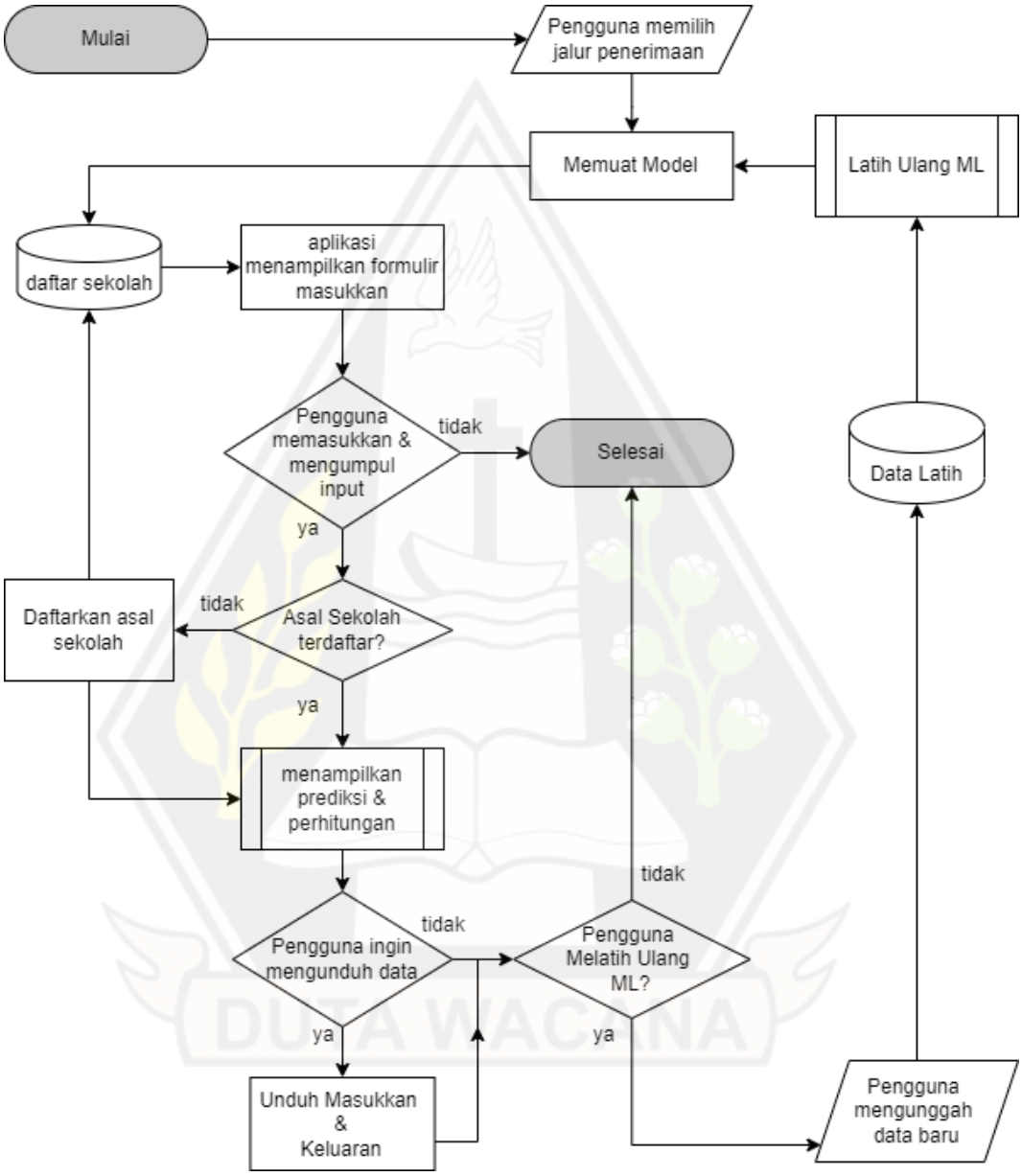
nilai_tpa_analogi	Input	Nilai Tes Potensi Akademik analogi calon mahasiswa
nilai_tpa_numerik	Input	Nilai Tes Potensi Akademik numerik calon mahasiswa
avg_nilai_uan	Input	Rata-rata nilai UAN calon mahasiswa
avg_nilai_rapor	Input	Rata-rata nilai rapor SMA calon mahasiswa
is_diterima	Output	Nilai biner penerimaan calon mahasiswa masuk Fakultas TI

Penulis kemudian melakukan *fitting* atau pelatihan dengan model algoritma *decision tree* dan *random forest*. Luaran model *random forest* disingkat menjadi RF, sementara luaran model *decision tree* disingkat menjadi DT. Sesuai dengan jumlah fitur data masing-masing jalur seleksi, parameter RF untuk klasifikasi data jalur reguler menggunakan 10 *tree*, dan 8 *tree* untuk klasifikasi data jalur prestasi. Sementara parameter DT sebagai perbandingan tidak diubah.

Luaran model-model ini kemudian disimpan untuk dimuat ulang pada purwarupa aplikasi rekomendasi menggunakan library *pickle* ke dalam dua file berformat *.sav* untuk setiap jalur seleksi. Data kedua jalur seleksi disimpan ke dalam dua file berformat *.csv*.

3.3.2 Flowchart Aplikasi

Flowchart sistem secara keseluruhan dengan seluruh fiturnya .dapat dilihat pada Gambar 3.2 yakni sebagai berikut,



Gambar 3.2 Diagram Flowchart Aplikasi

Tahap pertama pada penggunaan purwarupa aplikasi adalah pemilihan jalur PMB yang menentukan model dan formulir masukan yang dimuat. Model yang dimuat adalah model yang sudah dikembangkan sebelumnya dan disimpan dalam sebuah file berformat .sav. Pengguna akan membuka halaman baru setelah memilih jalur.

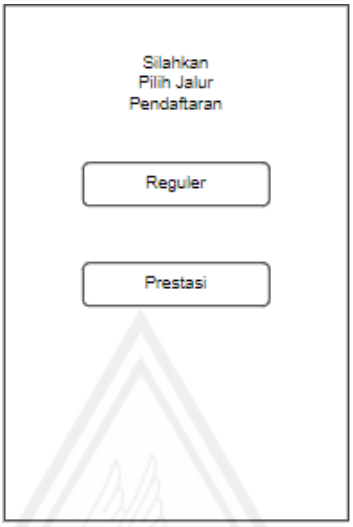
Pada tahap kedua, pengguna memasukkan dan mengumpulkan data-data seperti antara lain asal sekolah (atau nama sekolah bila asal sekolah tidak terdaftar pada opsi formulir), jurusan dan status sekolah mahasiswa, pilihan pertama dan kedua prodi calon mahasiswa, nilai tpa calon mahasiswa bila sebelumnya pengguna memilih halaman jalur reguler, atau rata-rata UAN dan rapor bila halaman jalur prestasi yang terpilih sebelumnya. Pengguna kemudian menekan tombol “submit”, dan aplikasi akan melakukan seleksi penerimaan calon mahasiswa dari data yang diberikan. Aplikasi juga akan mengeluarkan proses perhitungan seleksi dan persentase proses yang menerima atau menolak calon mahasiswa.

Pengguna diberikan tombol untuk menampilkan dan mengunduh data masukan formulir serta untuk menampilkan hasil seleksi. Apabila pengguna ingin melatih ulang model yang dimuat aplikasi, maka pengguna menekan tombol yang menerima unggahan data baru untuk ditambahkan pada data latih. Aplikasi kemudian akan melakukan *fitting* atau pelatihan ulang model untuk digunakan sebagai seleksi data masukan formulir.

3.4 Rancangan Antarmuka Sistem

3.4.1 Tampilan Awal

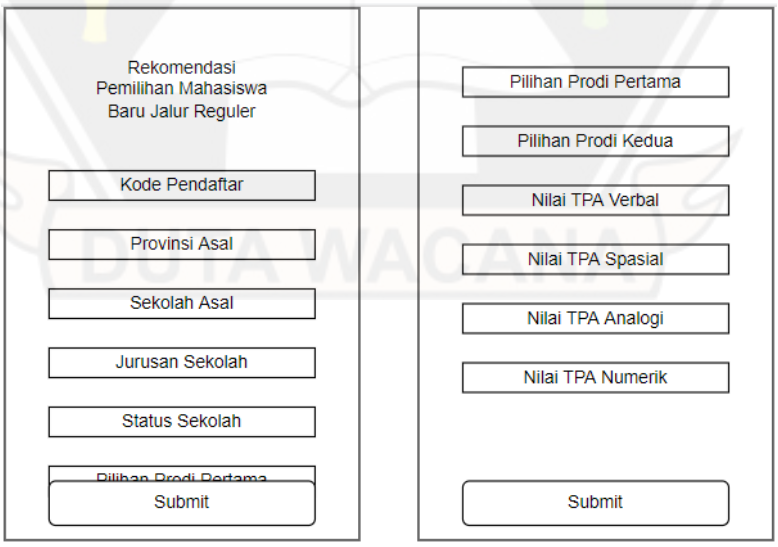
Halaman beranda aplikasi dua tombol untuk memilih jalur penerimaan mahasiswa. Apabila salah satu tombol ditekan maka aplikasi membuka halaman formulir masukan untuk jalur yang dipilih. Rancangan awal halaman bisa dilihat pada Gambar 3.3 yakni sebagai berikut,



Gambar 3.3 Halaman Beranda Aplikasi

3.4.2 Tampilan Halaman Formulir

Halaman formulir sebagai media masukan data, berupa sederet textbox, dan selectbox masukan. Pengguna memasukkan data sesuai dengan jalur penerimaan yang dipilih pengguna sebelumnya. Pengguna kemudian menekan tombol *submit* setelah selesai dan aplikasi akan menampilkan halaman keluaran seleksi aplikasi. Rancangan halaman formulir bisa dilihat pada Gambar 3.4 dan Gambar 3.5 yakni sebagai berikut,



Gambar 3.4 Rancangan Halaman Formulir Jalur Reguler

Gambar 3.5 Rancangan Halaman Formulir Jalur Prestasi

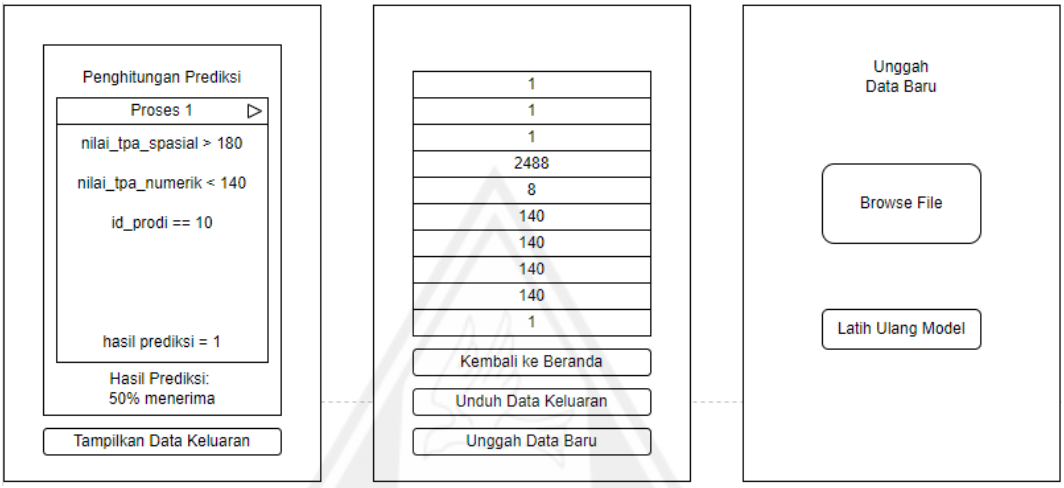
3.4.3 Tampilan Halaman Keluaran Aplikasi

Halaman keluaran aplikasi berupa *pane window* yang menunjukkan hasil seleksi, di mana pengguna juga bisa melihat penghitungan setiap keputusan yang diambil setiap *decision tree* pada model *random* forest.

Menggunakan tombol "Tampilkan Data Keluaran", pengguna bisa melihat data masukan dan hasil klasifikasi sebagai list dataframe yang diproses oleh model. Pengguna bisa mengunduh list tersebut sebagai data berformat .csv untuk disunting secara manual. Tombol "Kembali ke Beranda" apabila ditekan, maka aplikasi akan menampilkan halaman beranda di mana penggunaan aplikasi akan berulang.

Apabila pengguna mau melatih ulang model, maka pengguna menekan tombol "Unggah Data Baru" mengakses halaman unggahan data, dan menekan tombol "Browse Files" untuk mengunggah data baru yang ditambahkan pada data latih. Aplikasi kemudian menampilkan data baru tersebut dan pengguna bisa menekan tombol "Latih Ulang Model" untuk melakukan *fitting* atau latih ulang model. Aplikasi kemudian akan kembali ke halaman awal beranda setelah pelatihan

model selesai. Adapun rancangan antarmuka halaman keluaran aplikasi diilustrasikan oleh Gambar 3.6 yakni sebagai berikut,



Gambar 3.6 Rancangan halaman keluaran aplikasi

3.5 Evaluasi dan Pengujian

Pada tahap ini dilakukan perbandingan performa klasifikasi, dan kesesuaian pelatihan DT dan RF, sesuai dengan pustaka yang ditinjau. Pelatihan DT dan RF menggunakan data reguler dan data prestasi yang sudah dikembangkan sebelumnya.

Data reguler berisi 1020 data dengan fitur-fitur yakni lokasi, status dan tipe sekolah asal, pilihan pertama dan kedua program studi, nilai TPA verbal, spasial, analogi dan numerik, dengan nilai biner penerimaan calon mahasiswa sebagai variabel kelas data. 553 data lolos seleksi penerimaan, sementara 467 data ditolak.

Data prestasi berisi 797 data dengan fitur yakni lokasi, status dan tipe sekolah asal, pilihan pertama dan kedua program studi, rata-rata nilai UAN, dan rata-rata nilai rapor, dengan nilai biner penerimaan calon mahasiswa sebagai variabel kelas data. 295 data lolos seleksi penerimaan, sementara 502 data ditolak.

Pengujian performa luaran model menggunakan metrik akurasi pada Persamaan 2.3, sensitivitas pada Persamaan 2.4, presisi pada Persamaan 2.6, dan f1-score pada Persamaan 2.7.

Evaluasi pada performa DT dan RF menggunakan metode *stratified 10-fold cross validation* metrik akurasi dan *feature importance* untuk menentukan variabel yang paling digunakan oleh DT dan RF dalam melakukan klasifikasi.

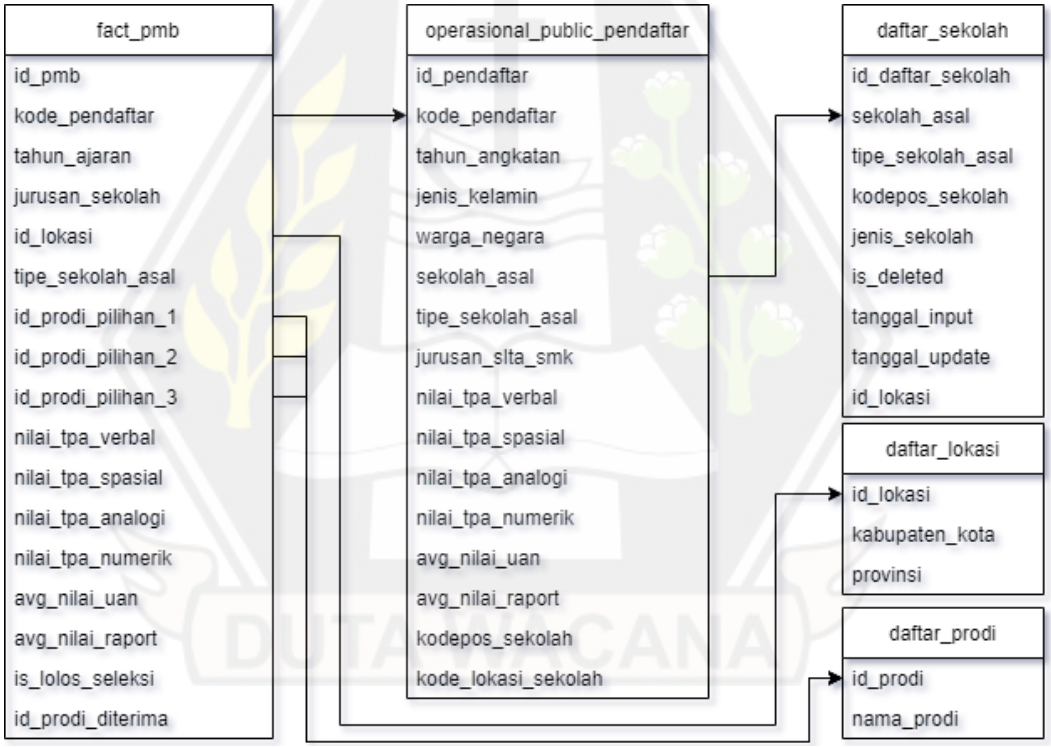
Adapun evaluasi lebih lanjut pada luaran model berupa perbandingan rata-rata nilai akurasi evaluasi DT dan RF dengan perubahan parameter max_depth benilai 10, 20, 30, 40, 50, dan nilai bawaan algoritma untuk melihat pengaruh parameter *max_depth* luaran model pada performa DT dan RF.

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

4.1 Praproses Dataset

Untuk mengembangkan *dataset* yang sesuai dengan rancangan pada metodologi penelitian, data mentah diekspor dari kueri basis data yang dimiliki oleh tim DW FTI UKDW, berisi tabel-tabel data PMB mahasiswa angkatan 2015 sampai 2021.

Data PMB diekspor dari DW FTI UKDW sesuai dengan detail yang tertulis pada metodologi penelitian. Adapun data-data tersebut terlampir pada Gambar 4.1 dengan hubungan antar data sebagai berikut,



Gambar 4.1 Diagram Data PMB FTI UKDW

Data *fact_pmb* berupa kueri langsung dari admin DW FTI UKDW dan digunakan oleh penulis sebagai acuan utama daftar field data yang diolah menjadi fitur-fitur *dataset*. Adapun data tersebut berasal dari sekumpulan data yakni,

operasional_public_pendaftar, *daftar_sekolah*, *daftar_lokasi*, dan *daftar_prodi*. Data mentah tersebut menjadi sumber kueri lanjut untuk mendapatkan informasi yang tidak ada pada kueri sebelumnya.

4.1.1 Fitur Asal Sekolah Calon Mahasiswa

Merujuk pada Gambar 4.1, untuk menghasilkan fitur asal sekolah pada dataset, penulis melakukan kueri data menggunakan field *kode_pendaftar* sebagai kunci asing kepada data *operasional_public_pendaftar*, dan kemudian menggunakan field *sekolah_asal* sebagai kunci asing kepada data *daftar_sekolah* untuk menghubungkan nomor identifikasi setiap nama sekolah pada *sekolah_asal*.

4.1.2 Fitur Lokasi Asal Calon Mahasiswa dengan Field *id_lokasi*

Field *id_lokasi* berisi nomor induk kabupaten dan kota asal calon mahasiswa. Adapun nomor induk 1 mewakili calon mahasiswa asal yang tidak terdaftar, 2 mewakili calon mahasiswa asing yang berasal dari luar negara. Rentang nomor induk 154 sampai dengan 254 merupakan kota dan kabupaten yang berada di Pulau Jawa.

Penulis mengolah field *id_lokasi* menjadi fitur nilai biner lokasi asal calon mahasiswa dengan 1 mewakili asal calon mahasiswa dari Jawa dan 0 mewakili asal calon mahasiswa dari luar Jawa.

4.1.3 Fitur Status dan Tipe Asal Sekolah Calon Mahasiswa

Penulis mengolah field *tipe_sekolah_asal* menjadi fitur status sekolah asal calon mahasiswa dengan 1 mewakili sekolah negeri dan 0 mewakili yang bukan sekolah swasta. Field *jurusan_sekolah* kemudian diolah penulis menjadi fitur tipe sekolah asal calon mahasiswa dengan nilai satu mewakili sekolah menengah atas (SMA) dan tipe sekolah lainnya seperti sekolah menengah kejuruan (SMK), *homeschooling*, dan sebagainya.

4.1.4 Fitur Pilihan Prodi dan Nilai Biner Penerimaan Calon Mahasiswa

Adapun variabel nilai biner *is_lolos_seleksi* bernilai 1 mewakili calon mahasiswa diterima dan 0 mewakili calon mahasiswa ditolak, serta variabel-variabel *id_prodi_pilihan_1*, *id_prodi_pilihan_2*, dan *id_prodi_diterima* yang berisi nomor identifikasi program-program studi yang dipilih calon mahasiswa. Di mana nilai 9 mewakili program studi Informatika dan nilai 10 mewakili program studi Sistem Informasi. Menggunakan variabel-variabel tersebut, penulis mengembangkan fitur *dataset* biner *is_diterima* yang bernilai 1 hanya apabila variabel is_lolos_seleksi bernilai 1 dan *id_prodi_diterima* bernilai 9 atau 10.

4.1.5 Fitur Nilai TPA, Rata-rata Nilai UAN dan Rapor

Variabel-variabel nilai akademik calon mahasiswa didapatkan secara langsung melalui kueri pertama dari Admin DW FTI UKDW. Variabel tersebut berupa nilai-nilai TPA calon mahasiswa untuk jalur seleksi reguler dan rata-rata nilai UAN dan rapor untuk jalur seleksi prestasi. Nilai-nilai TPA mempunyai rentang 0 sampai dengan 200. Rata-rata nilai UAN dan rapor berentang 0 sampai dengan 100.

4.1.6 Pembersihan Data

Dataset kemudian dipisah berdasarkan jalur seleksi menjadi dua, yakni data jalur reguler dan data jalur prestasi. Fitur-fitur seperti lokasi, status, tipe dan sekolah asal calon mahasiswa, pilihan program studi pertama dan kedua, dan nilai biner penerimaan digunakan oleh kedua data. Tetapi fitur nilai TPA hanya digunakan oleh data jalur reguler, sementara fitur rata-rata nilai UAN dan rapor hanya digunakan oleh data jalur prestasi.

Fitur kelas klasifikasi data adalah nilai biner penerimaan calon mahasiswa baru, *is_diterima*. Data jalur reguler dan jalur prestasi masing-masing dipisah lebih lanjut dengan rasio 60% data latih dan 40% data uji dan *random state* bernilai 42.

Adapun pembersihan data lebih lanjut dilakukan pada data jalur prestasi seperti pembagian rata-rata nilai UAN di atas 100, dan pembuangan data-data calon

mahasiswa dengan rata-rata nilai rapor yang kosong, kecuali apabila calon mahasiswa tersebut ditolak. Tabel 4.1 dan 4.2 secara urut adalah contoh dataset PMB seleksi jalur reguler dan prestasi yang digunakan, yakni sebagai berikut.

Tabel 4.1 Contoh Data PMB Jalur Reguler

lokasi	status	tipe	id_daftar_sekolah	id_prodi_pilihan_1	id_prodi_pilihan_2	nilai_tpa_verbal	nilai_tpa_spasial	nilai_tpa_analogi	nilai_tpa_numerik	is_diterima
1	0	1	164	6	9	100	90	70	90	1
1	0	1	45	9	10	10	10	10	10	1
1	0	1	2227	9	3	150	90	120	80	1
0	1	1	239	9	1	100	120	120	80	1
1	0	1	2192	9	1	10	10	10	20	1

Tabel 4.2 Contoh Data PMB Jalur Prestasi

lokasi	status	tipe	id_daftar_sekolah	id_prodi_pilihan_1	id_prodi_pilihan_2	avg_nilai_uan	avg_nilai_raport	is_diterima
0	0	1	10	3	9	0.0	0	0
0	0	0	10	4	9	0.0	0	0
0	0	0	10	9	10	0.0	0	0
1	0	1	25	8	9	0.0	0	0
1	0	1	25	3	9	0.0	0	0

4.2 Implementasi Sistem

Model algoritma *random forest classifier* dan *decision tree classifier* digunakan untuk mempelajari *dataset* yang sudah disiapkan. Implementasi pelatihan, pengujian beserta parameter-parameter algoritma-algoritma tersebut dilakukan menggunakan fungsi-fungsi dari pustaka Scikit-Learn. Luaran-luaran model tersebut dinamakan RF untuk *Random Forest classifier*, dan DT untuk *Decision Tree classifier*. Luaran model yang memberikan performa yang paling efisien dari hasil pengujian setiap jalur seleksi kemudian disimpan menjadi file berformat .*sav* untuk dimuat ulang pada purwarupa sistem.

4.2.1 Keluaran Penghitungan Luaran Model

Pustaka scikit-learn mengimplementasikan landasan teori node pohon algoritma *decision tree* menjadi sebuah struktur dengan atribut-atribut antara lain seperti *node_count*, yang berarti jumlah total node pohon, dan *max_depth*, yang berarti kedalaman terbesar pohon. Adapun node-node pohon tersebut tersimpan sebagai deretan-deretan (*arrays*) paralel, dimana masing-masing isi deretan-deretan

tersebut yang mempunyai indeks *i* memegang informasi tentang node ke-*i*, mulai dari *i = 0* yang berarti node adalah puncak pohon. Adapun isi deretan-deretan tersebut antara lain sebagai berikut,

- *feature*[i]: fitur dataset yang digunakan untuk memisah node *i*.
- *threshold*[i]: nilai ambang fitur yang digunakan node *i*.

Fungsi *decision_path()* memasukkan data uji lalu mengeluarkan sebuah matriks indikator bernama *node_indicator* yang apabila diiterasikan dengan indeks data uji (*sample_id*) berisi deret *i* indeks node-node yang dilalui data uji tersebut, dinamakan *node_index*. Kemudian fungsi *apply()* mengeluarkan nilai *i* node ujung pohon yang menjadi akhir penghitungan, dinamakan *leave_id*. Kode Python penggunaan fungsi-fungsi tersebut ada pada Gambar 4.2 baris kode 8 dan 10 sebagai penggunaan fungsi *decision_path()* dan baris kode 9 sebagai penggunaan fungsi *apply()*.

```
8     node_indicator = rfc.decision_path(test)
9     leave_id = rfc.apply(test)
10    node_index = node_indicator.indices[node_indicator.indptr[sample_id]:
11                                        node_indicator.indptr[sample_id + 1]]
```

Gambar 4.2 Kode Pengambilan Nomor Identifikasi Node-node Hitungan Luaran Model

Selanjutnya isi *node_index* diiterasikan dengan menggunakan variabel *node_id* agar nilai fitur-fitur data uji berindeks *i = sample_id* bisa dibandingkan dengan *threshold* dari fitur yang digunakan node pohon yang berindeks *i = node_id*. Perbandingan tersebut akan menampilkan informasi yang sepadan dengan simbol lebih besar “>” atau tidak lebih besar “<=”. Kode python implementasi proses ini diilustrasikan pada Gambar 4.3 yakni sebagai berikut,

```
13    for node_id in node_index:
14
15        tabulation = ""
16        if leave_id[sample_id] == node_id:
17            break
18        if (test.iloc[sample_id, feature[node_id]] <= threshold[node_id]):
19            threshold_sign = "<="
20        else:
21            threshold_sign = ">"
22
23        st.write("%s (= %s) %s %s"
24              % (X.columns.values[feature[node_id]],
25                 test.iloc[sample_id, feature[node_id]],
26                 threshold_sign,
27                 threshold[node_id]))
28    st.write("\n%sPrediction for submitted data: %s"%(tabulation,
29                                                      rfc.predict(test)[sample_id]))
```

Gambar 4.3 Kode Pengulangan Perbandingan Data Uji dengan isi Node-Node Pohon

Seperti yang bisa dilihat pada baris kode 16 dan 17, pengulangan hanya berhenti apabila nilai *node_id* sama dengan *leave_id* menandakan bahwa penghitungan sudah sampai pada node ujung pohon akhir. Setelah pengulangan selesai maka *decision tree* mengeluarkan hasil akhir klasifikasi data uji tersebut seperti pada baris kode 28 dan 29.

4.2.2 Kerangka Antarmuka Streamlit

Streamlit menggunakan satu file kode berbahasa Python dengan format *.py* untuk menayangkan antarmuka sebuah halaman aplikasi secara langsung pada web browser pengguna. Halaman aplikasi mampu memuat berbagai widget aplikasi untuk menerima masukkan dari pengguna dan memperbarui tampilan antarmuka.

Fitur kerangka Streamlit ini tetap menggunakan bahasa python sehingga kode-kode fungsi dari pustaka scikit-learn seperti pengeluaran penghitungan klasifikasi diletakkan pada file yang sama dengan kode-kode fungsi antarmuka Streamlit, dan menyederhanakan pengembangan implementasi rancangan sistem. Contoh kode ditunjukkan pada Gambar 4.4 dimana pada pada skrip, terdapat fungsi klasifikasi luaran model yang mana keluaran hasil klasifikasi langsung ditampilkan sebagai perubahan widget antarmuka pada aplikasi.

```
149     # show predicted result
150     result = regular_model.predict(test)[0]
151     st.write("%s"% (positive_counter*10)+"% Tree dalam random forest memberikan rekomendasi" )
152     st.write(negative_tab_list)
153     text_result = "DITERIMA" if result else "DITOLAK"
154     st.write("HASIL AKHIR ML: "+ text_result)
```

Gambar 4.4 Kode Fungsi Keluaran Hasil Klasifikasi Luaran Model

Untuk mengimplementasikan aplikasi halaman majemuk, file-file skrip sumber kode halaman-halaman aplikasi lainnya diletakkan dalam sebuah folder bernama *pages*. Streamlit juga mempunyai platform *cloud* yang mampu menampung kompilasi skrip-skrip dan eksekusi program apabila sumber kode sistem tersebut tersimpan pada repositori Git.

4.3 Implementasi Antarmuka Sistem

Tampilan antarmuka yang diimplementasikan penulis tidak sepenuhnya sesuai dengan rancangan yang diberikan pada sub bahasan bab 3 dikarenakan antarmuka aplikasi dirancang untuk perangkat seluler, sementara kerangka Streamlit mampu menghasilkan antarmuka yang mampu menyesuaikan semua perangkat.

4.3.1 Halaman Beranda

Tampilan halaman beranda diilustrasikan pada Gambar 4.5 yakni sebagai berikut,

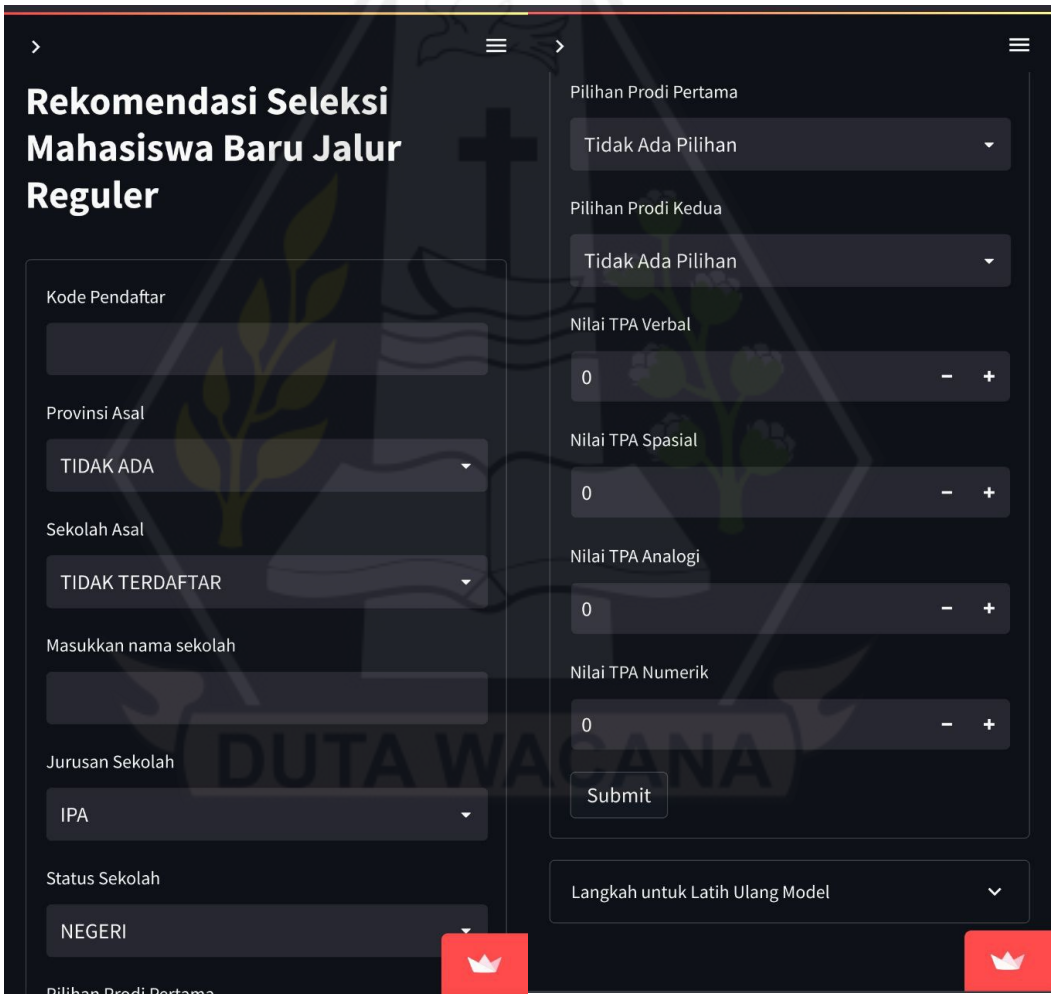
Gambar 4.5 Tampilan halaman beranda

Navigasi antar halaman aplikasi diimplementasikan sebagai sidebar yang bisa diakses pada semua halaman. Pada rancangan navigasi halaman terpisah hanya

pada halaman beranda dan keluaran. Aplikasi kemudian menampilkan halaman formulir sesuai pilihan jalur apabila pengguna sudah menekan pilihan pada sidebar.

4.3.2 Halaman Formulir Jalur Reguler

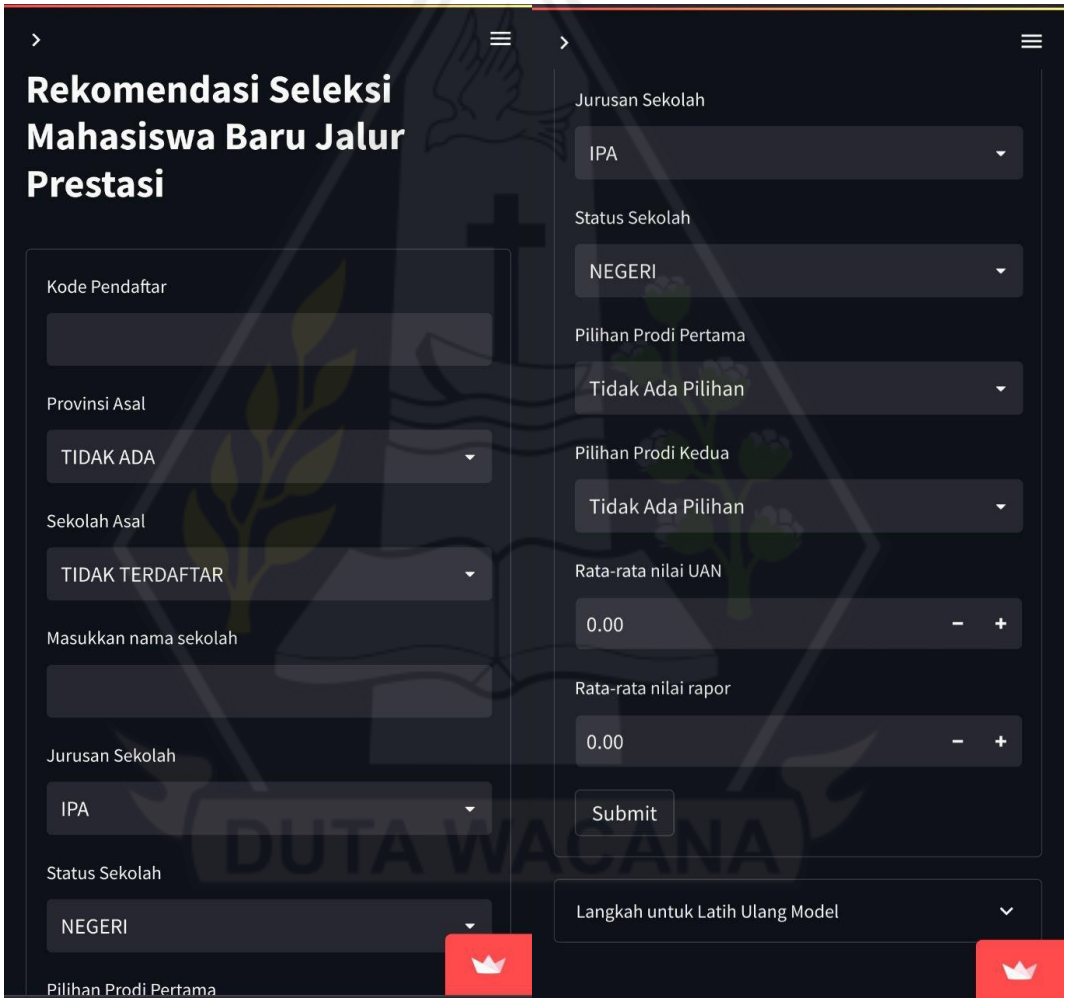
Halaman ini memuat formulir dan model yang digunakan untuk melakukan seleksi penerimaan calon mahasiswa jalur reguler. Pengguna akan mengisi formulir dengan data masukan dan menekan tombol submit setelah selesai. Tampilan halaman formulir seleksi jalur reguler diilustrasikan pada Gambar 4.6 sebagai berikut,



Gambar 4.6 Tampilan awal antarmuka halaman formulir jalur reguler

4.3.3 Halaman Formulir Jalur Prestasi

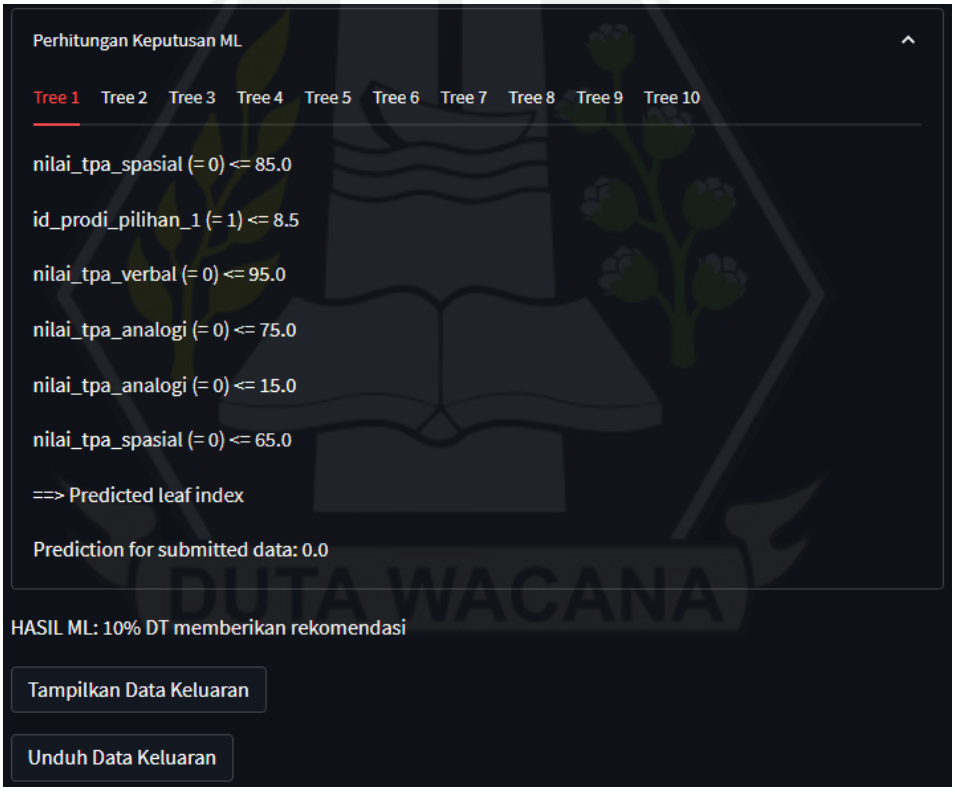
Mirip dengan halaman formulir jalur reguler tetapi berbeda pada data-data yang dimasukkan pada formulir, halaman ini memuat formulir dan model yang digunakan untuk melakukan seleksi penerimaan calon mahasiswa jalur prestasi. Pengguna akan mengisi formulir dengan data masukan dan menekan tombol submit setelah selesai. Tampilan implementasi halaman formulir seleksi jalur prestasi diilustrasikan pada Gambar 4.7 sebagai berikut,



Gambar 4.7 Tampilan awal antarmuka halaman formulir jalur prestasi

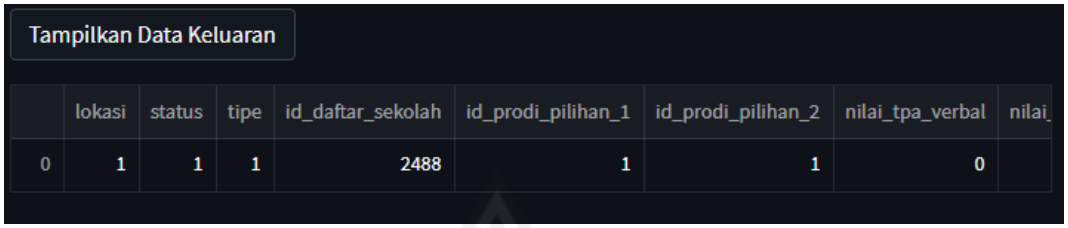
4.3.4 Keluaran Aplikasi

Pada bagian bawah kedua halaman formulir, apabila tombol “submit” ditekan maka aplikasi akan menampilkan perhitungan klasifikasi setiap *tree* pada model, beserta dengan persentasi klasifikasi *tree* pada model *Random Forest*. Adapun aplikasi akan memunculkan tombol “Tampilkan Data Keluaran” dan “Unduh Data Keluaran” yang masing-masing bila ditekan akan menampilkan data masukan formulir dan keluaran hasil klasifikasi dalam templat data, serta mengunduh data tersebut ke dalam file berformat .csv. Implementasi keluaran aplikasi tersebut diilustrasikan pada Gambar 4.8 sebagai berikut,



Gambar 4.8 Tampilan keluaran aplikasi

Sedangkan implementasi fitur tampilan data keluaran digambarkan pada Gambar 4.9, sebagai berikut,



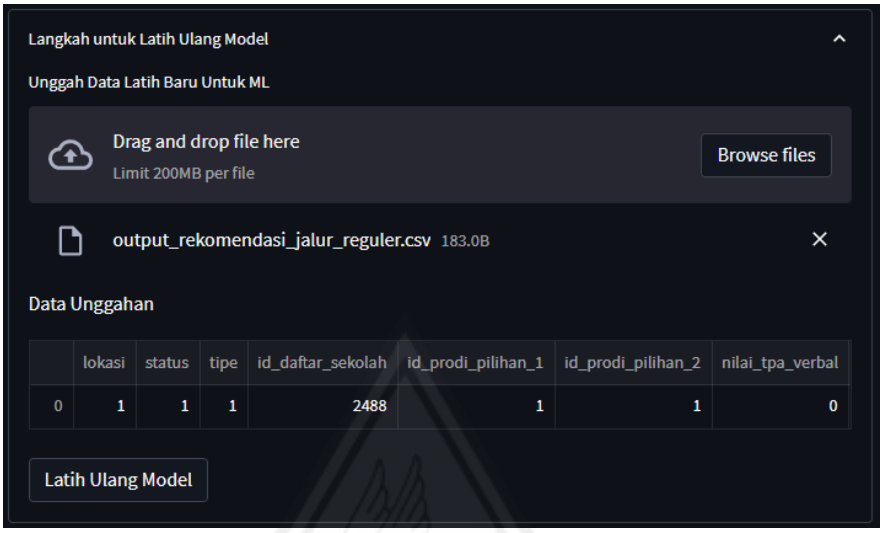
Gambar 4.9 Tampilan Data Keluaran

4.3.5 Fitur Latih Ulang Model

Untuk melatih ulang masing-masing model pada kedua halaman formulir, sebuah komponen antarmuka *expand* diletakkan di bawah formulir yang mana bila ditekan, akan memunculkan tombol opsi untuk mengunggah data baru untuk dilatih pada model. Tampilan awal implementasi fitur ini digambarkan pada Gambar 4.10 sebagai berikut,

Gambar 4.10 Tampilan awal antarmuka fungsi latih ulang model.

Pengguna kemudian mengunggah data .csv yang sebelumnya diunduh dan disunting. Setelah itu maka aplikasi akan menampilkan data baru tersebut dengan templat data, dan tombol "Latih Ulang" yang melakukan fungsi pelatihan pada model mengunakan data latih yang ditambahkan dengan data unggahan. Implementasi antarmuka setelah mengunggah data diilustrasikan pada Gambar 4.11 sebagai berikut,



Gambar 4.11 Tampilan antarmuka keseluruhan fitur latih ulang model

4.4 Analisis Sistem

Skenario pengujian dilakukan menurut sub bahasan Bab 3, setelah setiap pelatihan luaran model. Sehingga pengujian dibagi berdasarkan data seleksi jalur reguler, dan data seleksi jalur prestasi.

4.4.1 Klasifikasi Data Jalur Reguler

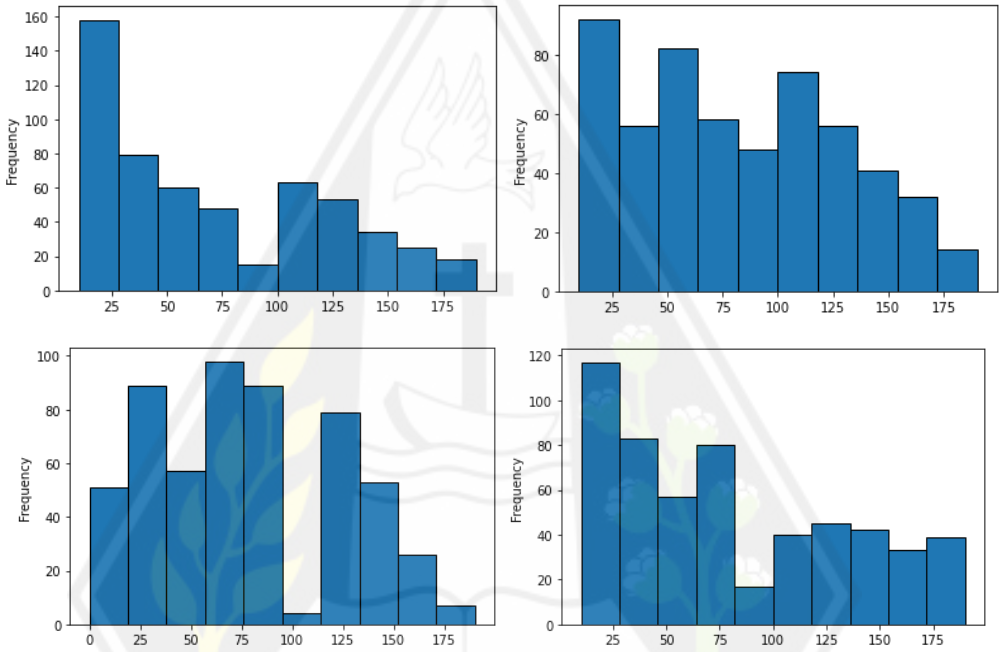
RF dan DT dilatih pada data jalur reguler yang berisikan 1020 data dengan fitur yakni lokasi, status dan tipe sekolah asal, pilihan pertama dan kedua program studi, nilai TPA verbal, spasial, analogi dan numerik, dengan nilai biner penerimaan calon mahasiswa sebagai variabel kelas data. 553 data lolos seleksi penerimaan, sementara 467 data ditolak.

Pelatihan RF dan DT pada *dataset* terpengaruh secara signifikan dengan implementasi pengembangan data. Data tidak dinormalisasikan calon-calon mahasiswa agar aplikasi mampu mengeluarkan penghitungan seleksi yang dapat dipahami oleh pengguna.

Selain itu, pada data latih terdapat banyak data calon-calon mahasiswa dengan nilai TPA yang rendah, namun lolos seleksi penerimaan. Hal ini disebabkan

karena seleksi pada kedua jalur dilakukan per gelombang. Sehingga muncul bias pada data karena terdapat kemungkinan untuk calon mahasiswa dengan nilai-nilai TPA yang lebih rendah untuk diterima karena nilai-nilai tersebut relatif sedang pada tempo hari seleksi tersebut.

Persebaran nilai-nilai TPA pada data calon-calon mahasiswa yang diterima diilustrasikan pada Gambar 4.12 yakni sebagai berikut,



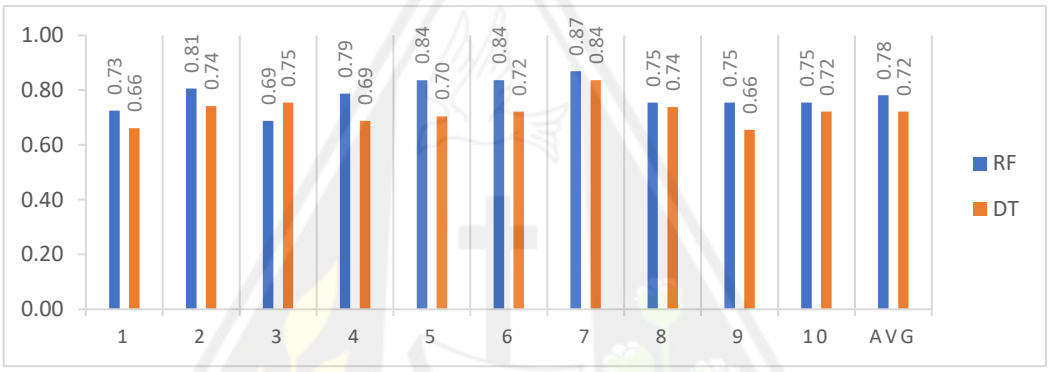
Gambar 4.12 Grafik persebaran nilai TPA sesuai baris dan kolom: verbal, spasial, analogi, dan numerik calon mahasiswa yang diterima.

Tabel 4.3 Tabel Metrik Performa RF dan DT pada Data Reguler

RF				DT			
Class	Precision	Sensitivity	F1-score	Class	Precision	Sensitivity	F1-score
Ditolak	0.78	0.82	0.80	Ditolak	0.72	0.68	0.70
Diterima	0.84	0.80	0.82	Diterima	0.74	0.78	0.76
Metrik							
Macro Avg	0.81	0.81	0.81	Macro Avg	0.73	0.73	0.73
Weighted Avg	0.81	0.81	0.81	Weighted Avg	0.73	0.73	0.73
Accuracy	0.81			Accuracy	0.73		

Berdasarkan metrik performa klasifikasi RF dan DT pada Tabel 4.3, RF mempunyai akurasi presisi, sensitivitas, dan *f-score* sebesar 81%, yang mana lebih tinggi daripada DT.

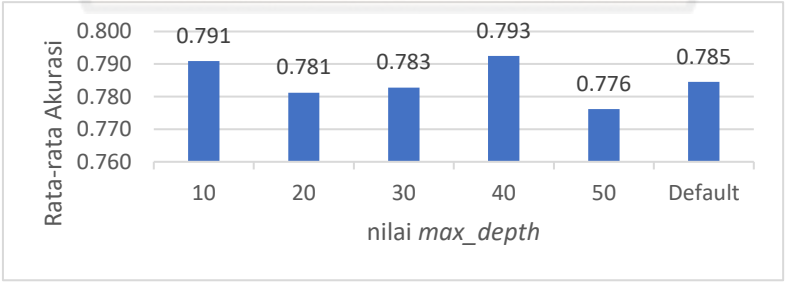
Hasil *stratified 10-fold cross validation* pada metrik akurasi yang diilustrasikan pada Gambar 4.13 menunjukkan bahwa RF memberikan performa yang lebih tinggi dengan rata-rata nilai akurasi sebesar 78%, dibandingkan dengan DT yang sebesar 72%.



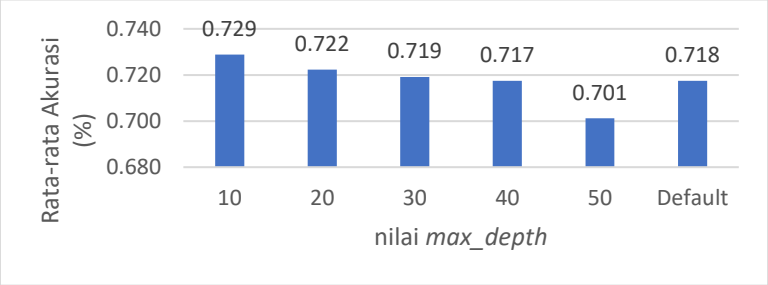
Gambar 4.13 Grafik skor akurasi RF dan DT pada data reguler

Bisa dilihat pada Gambar 4.14 untuk luaran model RF parameter *max_depth* bernilai 40 memberikan rata-rata nilai akurasi tertinggi bernilai senilai 79,3%. Gambar 4.15 menunjukkan bahwa rata-rata nilai akurasi tertinggi pada DT bernilai 72,9% dengan parameter *max_depth* bernilai 10.

Berdasarkan hasil evaluasi tersebut disimpulkan bahwa perubahan parameter *max_depth* pada rata-rata akurasi luaran model DT dan RF tidak terlalu signifikan dalam mengklasifikasikan data jalur reguler.

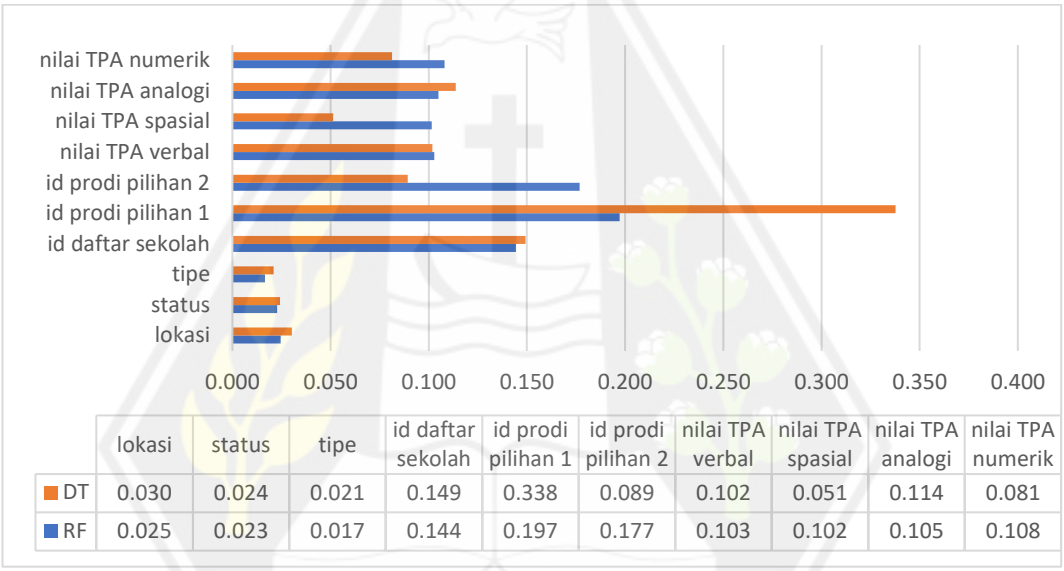


Gambar 4.14 Grafik rata-rata akurasi klasifikasi data reguler RF dengan variasi *max_depth*



Gambar 4.15 Grafik rata-rata akurasi klasifikasi data reguler DT dengan variasi *max_depth*

Evaluasi *feature importance* RF dan DT yang diilustrasikan pada Gambar 4.16, DT lebih mengutamakan variabel tersebut dengan *gini importance* sebesar 0.341 sebagai penentuan utama daripada RF, dengan *gini importance* sebesar 0.188.



	lokasi	status	tipe	id daftar sekolah	id prodi pilihan 1	id prodi pilihan 2	nilai TPA verbal	nilai TPA spasial	nilai TPA analogi	nilai TPA numerik
DT	0.030	0.024	0.021	0.149	0.338	0.089	0.102	0.051	0.114	0.081
RF	0.025	0.023	0.017	0.144	0.197	0.177	0.103	0.102	0.105	0.108

Gambar 4.16 *Feature importance* RF dan DT pada data reguler

Tabel 4.4 menunjukkan bahwa variabel data yang paling berpengaruh untuk pada DT dan RF dalam menentukan seleksi penerimaan adalah pilihan program studi pertama, yakni sebagai berikut,

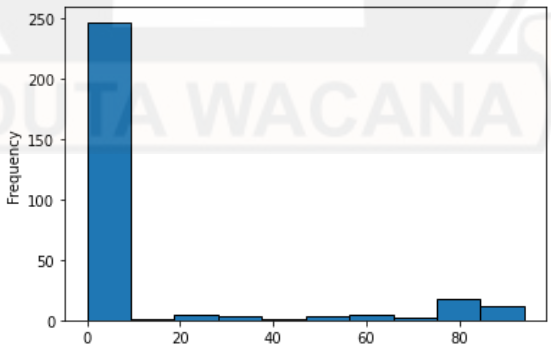
Tabel 4.4 Urutan Feature Importance Luaran Model Pada Data Jalur Reguler

Fitur RF	***Gini***	**Fitur DT**	***Gini***
id prodi pilihan 1	0.197	id prodi pilihan 1	0.338
id prodi pilihan 2	0.177	id daftar sekolah	0.149
id daftar sekolah	0.144	nilai TPA analogi	0.114
nilai TPA numerik	0.108	nilai TPA verbal	0.102
nilai TPA analogi	0.105	id prodi pilihan 2	0.089
nilai TPA verbal	0.103	nilai TPA numerik	0.081
nilai TPA spasial	0.102	nilai TPA spasial	0.051
lokasi	0.025	lokasi	0.030
status	0.023	status	0.024
tipe	0.017	tipe	0.021

Oleh karena metrik performa, rata-rata nilai akurasi yang lebih baik daripada DT, dengan rentang *feature importance* luaran model *random forest* yang lebih stabil dan tidak hanya terpaku pada satu fitur seperti DT, penulis menggunakan RF sebagai luaran model yang dimuatkan pada aplikasi.

4.4.2 Klasifikasi Data Jalur Prestasi

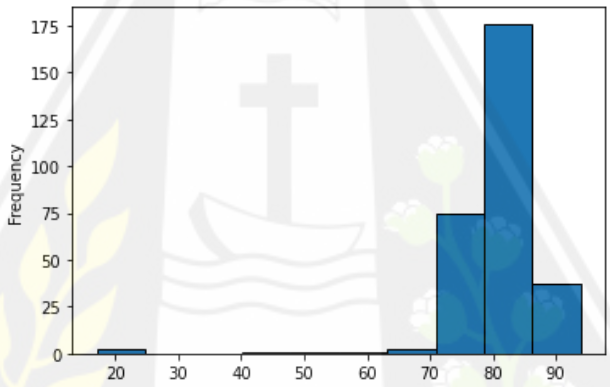
RF dan DT kemudian dilatih pada data jalur prestasi yang berisikan 797 data dengan fitur yakni lokasi, status dan tipe sekolah asal, pilihan pertama dan kedua program studi, rata-rata nilai UAN, dan rata-rata nilai rapor, dengan nilai biner penerimaan calon mahasiswa sebagai variabel kelas data. 295 data lolos seleksi penerimaan, sementara 502 data ditolak.



Gambar 4.17 Grafik persebaran nilai-rata-rata UAN

Adapun keterbatasan data di mana beberapa nilai UAN calon mahasiswa pada angkatan tahun tertentu yang tidak tersimpan pada DW FTI, sehingga kebanyakan data mempunyai rata-rata nilai UAN bernilai 0. Persebaran rata-rata nilai UAN pada data diilustrasikan pada Gambar 4.17.

Berdasarkan hasil wawancara dengan dosen-dosen yang melakukan seleksi untuk FTI, nilai rapor yang menjadi penentuan seleksi tidak ditentukan berdasarkan rata-rata nilai keseluruhan, melainkan dilihat secara bertahap dari semester pertama sampai semester keempat. Sehingga pelatihan RF dan DT pada data jalur prestasi terpengaruh secara signifikan dengan kurangnya fitur-fitur yang lebih spesifik. Persebaran rata-rata nilai rapor diilustrasikan pada Gambar 4.18 sebagai berikut,



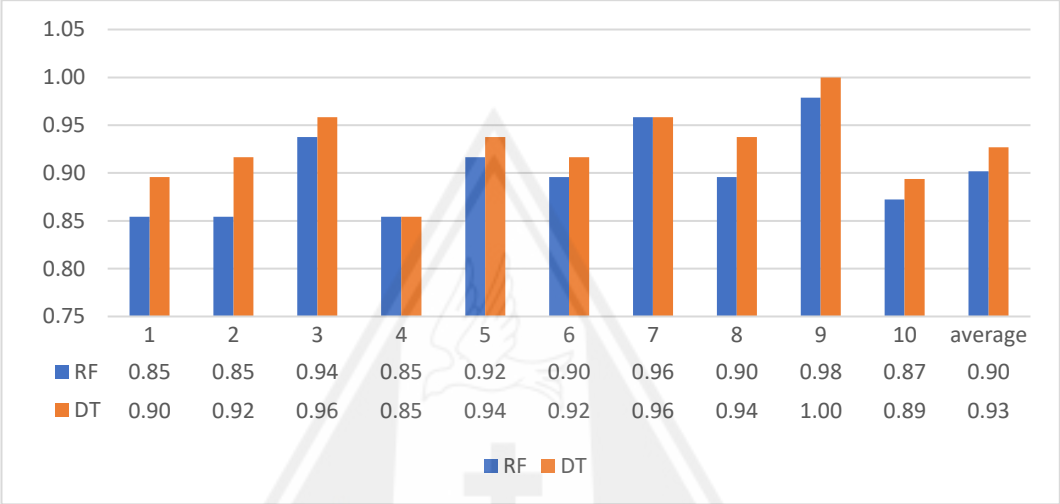
Gambar 4.18 Grafik persebaran nilai-rata-rata rapor

Berdasarkan metrik performa klasifikasi RF dan DT pada Tabel 4.5, DT mempunyai akurasi presisi, sensitivitas, dan *f-score* sebesar 93%, lebih tinggi 1% daripada RF.

Tabel 4.5 Tabel Metrik Performa RF dan DT pada Data Jalur Prestasi

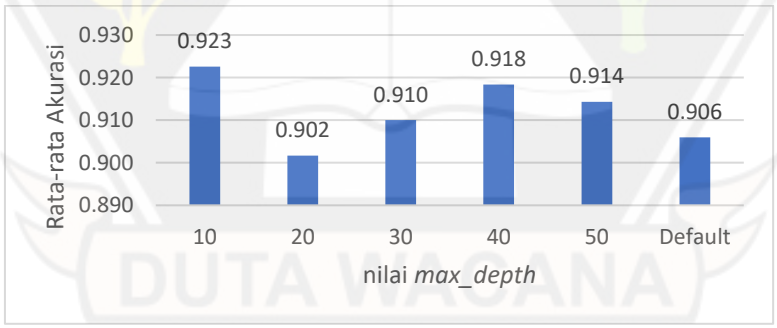
RF				DT			
Class	Precision	Sensitivity	F1-score	Class	Precision	Sensitivity	F1-score
Ditolak	0.91	0.95	0.93	Ditolak	0.95	0.94	0.95
Diterima	0.92	0.86	0.89	Diterima	0.91	0.92	0.92
Metrik							
Macro Avg	0.92	0.90	0.91	Macro Avg	0.93	0.93	0.93
Weighted Avg	0.92	0.92	0.91	Weighted Avg	0.93	0.93	0.93
Accuracy	0.92			Accuracy	0.93		

Hasil uji *stratified 10-fold cross validation* nilai akurasi RF dan DT diilustrasikan pada Gambar 4.19, DT mempunyai rata-rata nilai akurasi tertinggi sebesar 93%.



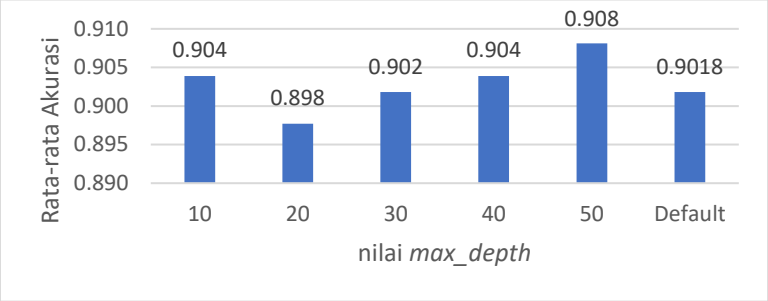
Gambar 4.19 Grafik skor akurasi RF dan DT pada data jalur prestasi

Bisa dilihat pada Gambar 4.20 untuk luaran model RF parameter *max_depth* bernilai 10 memberikan rata-rata nilai akurasi tertinggi bernilai senilai 92,3%.



Gambar 4.20 Grafik rata-rata akurasi klasifikasi data prestasi RF dengan variasi *max_depth*

Gambar 4.21 menunjukkan bahwa rata-rata nilai akurasi tertinggi pada DT bernilai 90,8% dengan parameter *max_depth* bernilai 50, yakni sebagai berikut.

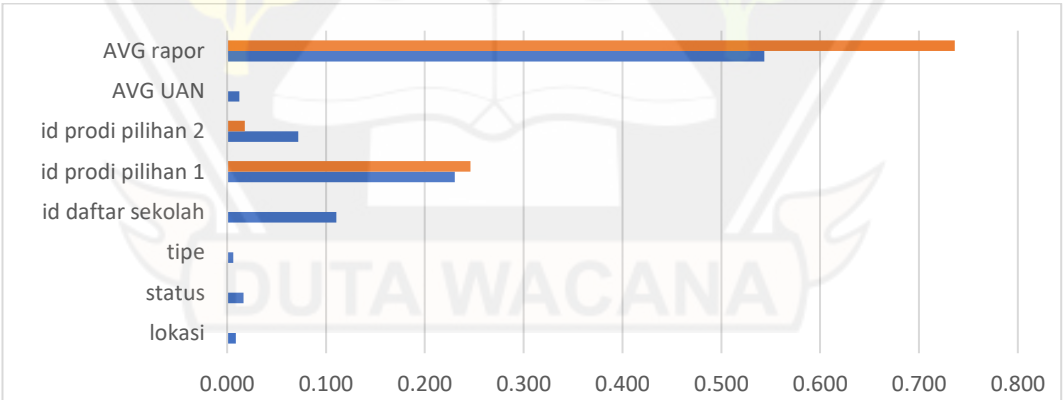


Gambar 4.21 Grafik rata-rata akurasi klasifikasi data prestasi DT dengan variasi *max_depth*

Berdasarkan hasil evaluasi tersebut disimpulkan bahwa perubahan parameter *max_depth* pada rata-rata akurasi luaran model DT dan RF tidak terlalu signifikan dalam mengklasifikasikan data jalur prestasi.

Berdasarkan evaluasi *feature importance* RF dan DT yang diilustrasikan pada Gambar 4.19, DT lebih mengutamakan rata-rata nilai rapor dengan *gini importance* sebesar 0.736 sebagai penentu pemisahan data daripada RF. DT juga tidak menggunakan fitur rata-rata nilai UAN, variabel lokasi, status dan tipe sekolah, maupun asal sekolah mahasiswa.

Tabel 4.6 menunjukkan bahwa variabel data yang digunakan oleh RF dan DT untuk seleksi penerimaan prestasi adalah rata-rata nilai rapor.



	lokasi	status	tipe	id daftar sekolah	id prodi pilihan 1	id prodi pilihan 2	AVG UAN	AVG rapor
DT	0.000	0.000	0.000	0.000	0.246	0.018	0.000	0.736
RF	0.009	0.017	0.006	0.110	0.230	0.072	0.012	0.543

Gambar 4.22 *Feature importance* RF dan DT pada data prestasi

Tabel 4.6 Urutan Feature Importance Luaran Model Pada Data Jalur Prestasi

Fitur RF	***Gini***	**Fitur DT**	***Gini***
Rata-rata rapor	0.543	Rata-rata rapor	0.736
id prodi pilihan 1	0.230	id prodi pilihan 1	0.246
id daftar sekolah	0.110	id prodi pilihan 2	0.018
id prodi pilihan 2	0.072	lokasi	0.000
status	0.017	status	0.000
Rata-rata UAN	0.012	tipe	0.000
lokasi	0.009	id daftar sekolah	0.000
tipe	0.006	Rata-rata UAN	0.000

Penulis kemudian menyimpan RF sebagai luaran model untuk dimuat pada aplikasi karena rentang nilai *feature importance* RF yang lebih sempit yang menandakan RF menggunakan lebih banyak fitur dalam penghitungan klasifikasi.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan implementasi dan analisis penelitian, maka dapat disimpulkan bahwa,

1. Pelatihan ML dipengaruhi oleh persebaran nilai pada beberapa fitur *dataset*. Sebagai contohnya banyak dijumpai pada data jalur reguler, mahasiswa lolos seleksi dengan nilai TPA yang rendah, dan banyak juga di data jalur prestasi dengan rata-rata nilai UAN yang *null*, atau tidak disimpan oleh PMB.
2. Pada klasifikasi data jalur reguler, model *random forest* mencapai *f-score* tertinggi sebesar 81% dan rata-rata akurasi sebesar 78% serta rentang *feature importance* yang lebih sempit daripada *decision tree*. Sementara pada klasifikasi data jalur prestasi, model *decision tree* memberikan *f-score* tertinggi dan rata-rata akurasi yang lebih tinggi daripada *random forest* dengan keduanya sebesar 93%, namun dengan rentang *feature importance* yang lebih lebar.
3. Tidak ada pengaruh signifikan dari perubahan nilai parameter *max_depth* DT pada rata-rata akurasi performa model *random forest* dan *decision tree* dalam melakukan klasifikasi pada data jalur reguler dan prestasi.
4. Berdasarkan *feature importance* luaran kedua model, variabel yang paling berpengaruh pada penerimaan seleksi jalur reguler adalah pilihan prodi pertama calon. Sementara variabel yang paling berpengaruh pada penerimaan seleksi jalur prestasi adalah rata-rata nilai rapor.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan penulis, adapun saran yang diberikan kepada pengembang selanjutnya untuk meningkatkan efisiensi pembelajaran sebagai berikut,

1. Pengembang dapat menambah fitur nilai bobot asal sekolah, ataupun nilai rata-rata rapor semester pertama sampai semester empat pada Data Jalur Prestasi.
2. Pengembang dapat melakukan transformasi dan penyuntingan lebih lanjut pada data latih.
3. Pengembang dapat menggunakan algoritma atau model yang berbeda untuk melakukan klasifikasi apabila model mampu memberikan performa lebih baik, seperti *Ridge Classifier*, atau *Nearest Neighbour*.

Kepada Data Warehouse FTI, pengembangan dapat dilakukan sehingga ada relasi queri data antara asal sekolah dengan data pendaftaran mahasiswa UKDW.

DAFTAR PUSTAKA

Alverina, D., Chrismanto, A. R., & Santosa, R. G. (2018). Perbandingan Algoritma C4.5 dan CART dalam Memprediksi Kategori Indeks Prestasi Mahasiswa. *Jurnal Teknologi dan Sistem Komputer, vol. 6, no. 2,*, 76-83.

Awad, M., & Khanna, R. (2015). *Efficient learning machines: theories, concepts, and applications for engineers and system designers.* Springer nature.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., . . . Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.

Doyen, S., Taylor, H., Nicholas, P., Crawford, L., Young, I., & Sughrue, M. E. (2021). Hollow-tree super: A directional and scalable approach for feature importance in boosted tree models. *PloS one,* 16(10).

Dua, S., & Du, X. (2016). *Data mining and machine learning in cybersecurity.* CRC press.

Kuhn, M., & Johnson, K. (2018). *Applied Predictive Modeling.* New York: Springer.

Manoe, R. V. (2012). *Program Bantu Pembuatan Pohon Keputusan Penerimaan Mahasiswa Baru di UKDW.* Yogyakarta: Universitas Kristen Duta Wacana.

Mitchell, T. M. (1997). *Machine learning.* New York: McGraw-Hill .

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning (2nd Edition).* MIT Press.

Müller , A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists.* O'Reilly Media, Inc.

Santosa, R. G., Lukito, Y., & Chrismanto, A. R. (2021). Classification and prediction of students' GPA using K-means clustering algorithm to assist student admission process. *Journal of Information Systems Engineering and Business Intelligence, 7(1),* 1-10.

Soto-Murillo, M. A., Galvan-Tejada, J. I., Galvan-Tejada, C. E., Celaya-Padilla, J. M., Luna-Garcia, H., Magallanes-Quintanar, R., . . . Gamboa-Rosales, H. (2021). Automatic Evaluation of Heart Condition According to the Sounds Emitted and Implementing Six Classification Methods. *Healthcare*, 317.

Wang, S., Lu, H., Khan, A., Hajati, F., Khushi, M., & Uddin, S. (2022). A machine learning software tool for multiclass classification. *Software Impacts, 13*, 100383.