

# Klasifikasi Data Penerimaan Calon Mahasiswa Baru Menggunakan Algoritma *Random Forest*

<http://dx.doi.org/10.28932/jutisi.vXiX.X>

Riwayat Artikel

Received: xx Bulan 20xx | Final Revision: xx Bulan 20xx | Accepted: xx Bulan 20xx

Creative Commons License 4.0 (CC BY – NC)



Kafka Febianto Agiharta<sup>□#1</sup>, Avinash<sup>□#2</sup>, Sherly Santiadi<sup>□#3</sup>, Devion Tanrico<sup>#4</sup>,

<sup>#</sup>Program Studi Magister Ilmu Komputer, Universitas Kristen Maranatha

Jl. Surya Sumantri No. 65, Bandung, 40164, Indonesia

<sup>1</sup>[mi2279806@student.it.maranatha.edu](mailto:mi2279806@student.it.maranatha.edu)

<sup>2</sup>[mi2279005@student.it.maranatha.edu](mailto:mi2279005@student.it.maranatha.edu)

<sup>3</sup>[mi2279801@student.it.maranatha.edu](mailto:mi2279801@student.it.maranatha.edu)

<sup>3</sup>[mi2279802@student.it.maranatha.edu](mailto:mi2279802@student.it.maranatha.edu)

**Abstrak** — Administrasi penerimaan mahasiswa baru selalu meningkat setiap tahun. Oleh karena itu, Universitas Kristen Maranatha berusaha untuk membangun sistem yang dapat membantu mengklasifikasikan berkas administrasi ini. Sangat penting untuk membuat model yang dapat menunjukkan kecenderungan mahasiswa calon ini diterima di departemen pilihan sebagai bahan pertimbangan dalam proses seleksi berikutnya. Dalam penelitian ini, algoritma *Random Forest* diterapkan untuk mengatasi masalah ini. Metrik evaluasi model peramalan yang digunakan meliputi Laporan Klasifikasi, Kurva Karakteristik Operasi Penerima, dan Matriks Konfusi. Selain itu, metodologi agile diterapkan dalam mengembangkan dua output: sebuah situs web dan aplikasi yang dapat diintegrasikan dengan model prediksi untuk klasifikasi penerimaan mahasiswa baru menggunakan algoritma *Random Forest*.

**Kata kunci** — aplikasi; klasifikasi; random forest

## Data Classification of New Student Admissions Using the Random Forest Algorithm

**Abstract** — The administration of new student admissions is always increasing each year. Therefore, Maranatha Christian University is trying to build a system that can help classify these administrative files. It is very important to create a model that can show the tendency of these prospective students to be accepted in the chosen department as material for consideration in the next selection process. In this study, the *Random Forest* algorithm was applied to address this issue. The evaluation metrics of forecasting models used include Classification Report, Receiver Operation Characteristic Curve, and Confusion Matrix. In addition, an agile methodology was applied in developing two outputs: a website and an application that can be integrated with the prediction model for the classification of new student admissions using the *Random Forest* algorithm.

**Keywords** — application; classification; random forest

## I. PENDAHULUAN

Setiap tahun, universitas-universitas di berbagai negara menerima ribuan pendaftaran dari calon mahasiswa yang ingin bergabung. Proses seleksi ini memiliki peran yang sangat penting karena mempengaruhi komposisi mahasiswa serta berdampak pada pencapaian akademik universitas tersebut. Di Indonesia, persaingan untuk diterima di perguruan tinggi sangatlah sengit karena kesadaran masyarakat terhadap pentingnya pendidikan semakin meningkat. Meskipun demikian, proses seleksi ini kerap menjadi rumit dan berpotensi menghasilkan keputusan yang subjektif, sehingga dapat menimbulkan ketidakpuasan.

Random Forest sebagai algoritma digunakan untuk mencari solusi terkait permasalahan ini. *Random Forest* merupakan sebuah pengembangan dari metode *Decision Tree* yang menggunakan beberapa *Decision Tree*. *Random Forest* memiliki proses seleksi fitur yang mampu mengambil fitur terbaik, sehingga dapat meningkatkan performa model klasifikasi. Dengan adanya seleksi fitur, *Random Forest* dapat efektif bekerja pada *big data* dengan parameter yang kompleks.

Untuk mengimplementasi sistem penerimaan mahasiswa baru, digunakan *framework React Native* untuk menangani bagian calon mahasiswa baru. *React Native* adalah sebuah *framework JavaScript* yang populer di kalangan pengembang. *Framework* ini digunakan secara luas karena kemudahannya dalam pembelajarannya dan struktur kode yang sederhana. Kelebihan lain dari *React Native* adalah kemampuannya untuk mengembangkan aplikasi berbasis *Android* dan *iOS* dengan mudah dalam satu proses pengembangan. Hal ini mempermudah pengguna dalam menggunakannya sesuai dengan sistem operasi dari masing-masing *smartphone*.

Pengiriman data dari *platform mobile* menggunakan arsitektur REST API yang memungkinkan berbagai sistem dalam hal ini *platform mobile* ke *website* dan sebaliknya untuk berkomunikasi dan mengirim / menerima data dengan cara yang sangat sederhana. setiap panggilan API REST memiliki hubungan antara kata kerja HTTP dan URL. Sumber daya dalam *database* dalam suatu aplikasi dapat dipetakan dengan titik akhir API di REST, atau lebih dikenal dengan *Endpoint*. Data yang telah dikirim akan disimpan kedalam *database* menggunakan MySQL.

MySQL merupakan aplikasi atau sistem untuk mengelola *database* atau manajemen data dan mampu mengelola sistem dengan jumlah mencapai 40 buah *database*, dimana bisa berisi lebih dari 10.000 tabel dan dengan 500 di antaranya memiliki 7 juta baris [2]. Data pendaftaran dan nilai yang masuk dapat dilihat dan dimanipulasi secara langsung pada MySQL. Data tersebut akan ditindaklanjuti oleh seorang administrator aplikasi penerimaan mahasiswa baru.

Perancangan aplikasi tersebut menggunakan *framework Laravel* berbasis *web*. *Laravel* itu sendiri adalah sebuah *framework* berbasis PHP untuk membuat aplikasi *web* dengan sintaks yang ekspresif dan elegan. Sudah banyak proyek-proyek web yang sudah dihasilkan menggunakan *framework Laravel* karena kemudahannya karena menawarkan konsep MVC (*Model, View, Controller*) seperti penelitian perancangan aplikasi yang dibuat oleh Sopian Aji, Dany Pratmanto, Angga Ardiansyah, dan Saifudin [3].

Tentu saja penelitian ini membutuhkan kerja sama dan komunikasi tim yang ahli pada bidang masing-masing. Maka digunakanlah sebuah metodologi *Agile* untuk mengimplementasikan teknik manajemen proyek yang menggunakan siklus pengembangan singkat, yang juga dikenal sebagai *sprint*. *sprint* berfokus pada perbaikan terus-menerus dalam pengembangan produk atau layanan. Teknik *Agile* terfokus terhadap pengembangan dan perbaikan secara terus-menerus dalam pengembangan produk. Dalam pengembangan yang tangkas, perangkat lunak yang dirilis secara bertahap, mengurangi upaya proses, menghasilkan kode berkualitas tinggi, dan melibatkan pelanggan secara langsung dalam proses.

Adapun tujuan pembangunan sistem penerimaan calon mahasiswa baru dengan mengimplementasikan algoritma *Random Forest* dalam prediksi penerimaan calon mahasiswa baru. diharapkan dapat memberikan dampak bagi calon mahasiswa untuk mendapatkan visualisasi atau gambaran jelas penerimaan dengan memakai aplikasi *mobile* untuk sistem penerimaan calon mahasiswa baru dilengkapi dengan probabilitas mahasiswa tersebut diterima dalam jurusan yang dipilih. dan dengan adanya *website* admin, proses pengelolaan sistem penerimaan calon mahasiswa baru akan menjadi lebih efisien dan terorganisir. melalui *website* admin, staf administrasi dapat mengelola data pendaftaran, mengelola hasil prediksi penerimaan, melihat laporan, dan melakukan tindakan administratif yang diperlukan. Dengan adanya *website* admin, proses pengelolaan sistem penerimaan calon mahasiswa baru akan menjadi lebih efisien dan terorganisir.

Tujuan lainnya adalah meningkatkan efisiensi proses penerimaan. Dengan menggunakan *algoritma Random Forest*, sistem dapat melakukan prediksi penerimaan calon mahasiswa baru dengan cepat dan akurat. Hal ini akan mengoptimalkan waktu dan upaya yang diperlukan oleh staf administrasi dalam proses seleksi.

## II. KAJIAN LITERATUR DAN ULASAN REFERENSI

### A. Random Forest

Algoritma *Random Forest* merupakan sebuah algoritma yang cukup populer dalam memprediksi baik secara regresi maupun klasifikasi. *Random Forest* sendiri merupakan gabungan dari beberapa pohon klasifikasi maupun regresi yang menggunakan model sederhana untuk melakukan pemisahan variabel untuk menentukan hasil prediksi. Pohon tersebut

biasanya disebut dengan *Decision Tree*. Pendekatan yang dilakukan di dalam *Random Forest* menggunakan pendekatan *bagging (bootstrap aggregating)*. Metode ini digunakan untuk mengurangi variansi di dalam sebuah model dengan cara menggabungkan beberapa model yang sudah dilatih pada *subset* yang berbeda pada data *training*. Hal ini dilakukan untuk mengurangi *overfitting* dan meningkatkan akurasi serta *robustness* di dalam sebuah model.

#### B. REST API

*REST API (Representational State Transfer Application Programming Interface)* adalah arsitektur komunikasi yang digunakan untuk menghubungkan sistem-sistem yang berbeda secara terdistribusi melalui jaringan. Fungsinya adalah sebagai perantara antara aplikasi klien dan server, memungkinkan aplikasi untuk mengakses dan memanipulasi data yang ada di server. *REST API* menggunakan protokol *HTTP* dan metode seperti *GET*, *POST*, *PUT*, dan *DELETE* untuk mengambil, membuat, memperbarui, dan menghapus data.

#### C. MySQL

*MySQL* adalah sistem manajemen basis data relasional (RDBMS) yang populer dan open-source. Fungsinya adalah untuk menyimpan, mengelola, dan mengakses data secara efisien. *MySQL* menyediakan platform yang kuat untuk mengelola basis data yang dapat digunakan oleh berbagai jenis aplikasi, baik itu aplikasi web, mobile, maupun enterprise.

#### D. React Native

*React Native*, sebuah *framework* yang diciptakan oleh *Facebook*, telah menjadi alat penting dalam pengembangan aplikasi *mobile*. Dengan memanfaatkan *JavaScript* dan *React*, *React Native* memungkinkan pembuatan aplikasi *mobile* dengan performa hampir setara dengan aplikasi *native*, namun dengan efisiensi waktu dan manajemen kode yang lebih baik. Fungsi utamanya adalah untuk memfasilitasi pembuatan aplikasi *mobile* lintas *platform*, memungkinkan pengembang untuk menggunakan basis kode yang sama untuk *Android* dan *iOS*. Selain itu, fitur *JavaScript* dan *React* seperti *state* dan *props*, *hooks*, dan komponen dapat digunakan, membuat proses pengembangan lebih efisien dan terstruktur.

#### E. Metodologi Agile

Metodologi Agile adalah pendekatan pengembangan perangkat lunak yang menekankan pada iterasi, kolaborasi, dan adaptasi. Dalam Agile, proyek dibagi menjadi siklus kecil yang disebut "*sprint*" dengan durasi dua hingga empat minggu. Setiap *sprint* berakhir dengan pengiriman fitur perangkat lunak yang dapat diuji dan berfungsi. Tim Agile bekerja secara kolaboratif, terdiri dari anggota dengan keahlian beragam yang melakukan pertemuan harian untuk berbagi pembaruan dan mengatasi hambatan. Metodologi Agile juga mendorong tim untuk beradaptasi dengan perubahan kebutuhan dan merespons secara fleksibel daripada tetap terikat pada rencana awal.

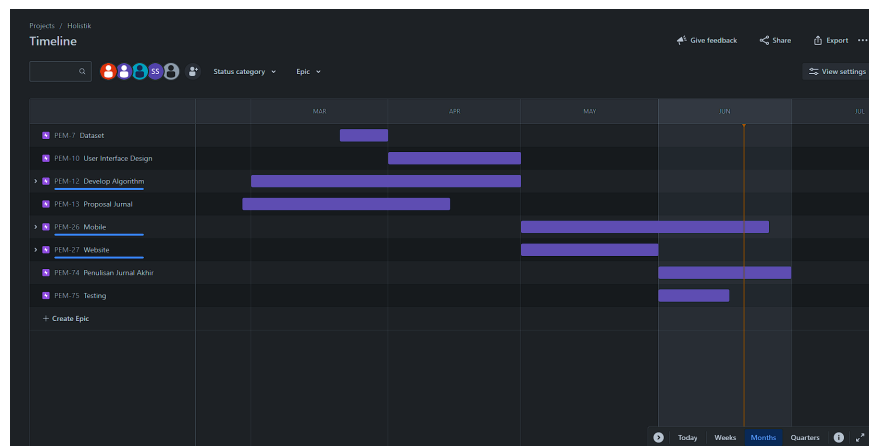
### III. METODOLOGI PENELITIAN

Data yang digunakan dalam penelitian ini berasal dari *dataset* Penerimaan Mahasiswa Baru (PMB) tahun 2021 sampai dengan 2023 di Universitas Kristen Maranatha. *Dataset* ini mencakup nilai rapor siswa, pilihan jurusan saat pendaftaran, dan status akhir apakah diterima atau ditolak. *Dataset* ini diperoleh dengan persetujuan resmi. Data tersebut kemudian melalui proses *feature engineering* untuk memastikan validitasnya sebelum digunakan untuk klasifikasi calon mahasiswa baru di universitas tersebut. Pengembangan aplikasi ini terbagi menjadi dua segmen: satu berbasis *web* untuk *admin* dan yang lainnya berbasis *mobile* untuk calon mahasiswa. Bagian *web* dirancang untuk *admin*, yang bertugas menerima data pendaftaran dari calon mahasiswa. Bagian *mobile* dirancang untuk calon mahasiswa, yang bertujuan untuk mengajukan pendaftaran program studi yang diminati dan menerima hasil penerimaan.

Bagian berbasis *web* untuk *admin* bertujuan untuk mengumpulkan dan mengelola data pendaftaran dari calon mahasiswa. Ini mencakup fitur untuk mengumpulkan informasi pribadi, riwayat pendidikan, dan dokumen pendukung lainnya. Selain itu, fitur pengelolaan data juga disediakan untuk memudahkan *admin* dalam mengelola data calon mahasiswa, termasuk pengeditan, penghapusan, dan pencarian data. Keamanan dan otorisasi juga menjadi prioritas dalam bagian ini, dengan mekanisme keamanan yang memadai untuk melindungi data pendaftaran dari akses yang tidak sah.

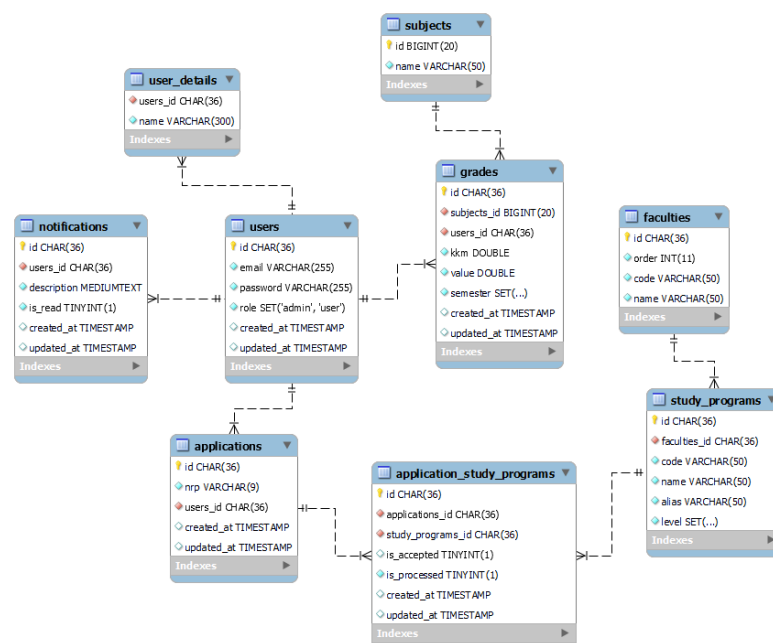
Bagian berbasis *mobile* untuk calon mahasiswa dirancang untuk memudahkan mereka dalam mendaftarkan program studi dan menerima hasil penerimaan. Aplikasi ini harus memiliki antarmuka yang responsif dan mudah digunakan, memungkinkan calon mahasiswa untuk mengisi formulir pendaftaran dan melihat hasil penerimaan mereka dengan mudah. Aplikasi ini juga harus dapat diakses melalui berbagai perangkat *mobile* dan sistem operasi yang umum digunakan oleh calon mahasiswa.

Untuk mengatur dan mengkoordinasikan kedua bagian pengembangan ini, digunakan metodologi *Scrum*. Metodologi ini memungkinkan tim untuk bekerja dalam *sprint* atau iterasi, dengan tujuan tertentu untuk setiap *sprint*. Digunakan aplikasi untuk mengimplementasi proses dalam *Scrum* ini seperti pada Gambar 3.1. Dengan pendekatan ini, pengembangan aplikasi dapat menjadi lebih terstruktur, transparan, dan dapat beradaptasi dengan perubahan yang mungkin terjadi selama proses pengembangan.



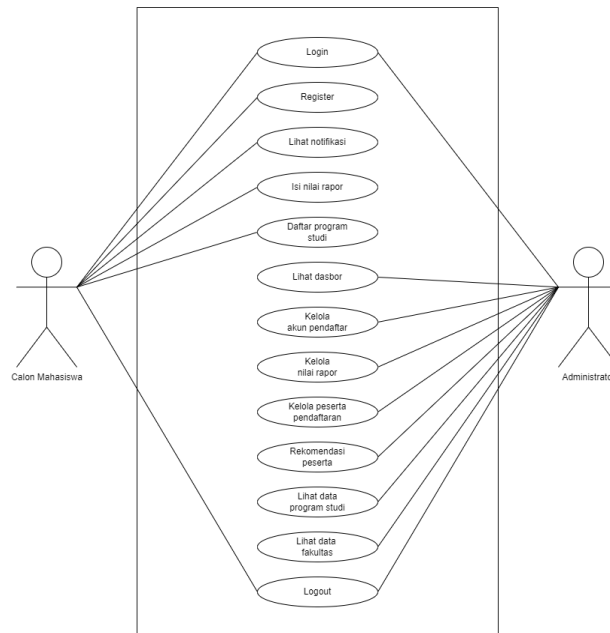
Gambar 3. 1 Tampilan Halaman Aplikasi Scrum

Dalam penelitian ini, aplikasi *MySQL* digunakan untuk merancang *database* yang terdiri dari sembilan tabel, yaitu "*users*", "*user\_details*", "*applications*", "*subjects*", "*grades*", "*notifications*", "*application\_study\_programs*", "*faculties*", dan "*study\_programs*", seperti yang ditunjukkan pada gambar 3.2. Tabel "*users*" berisi informasi akun para calon mahasiswa yang mendaftar. Tabel "*user\_details*" menyimpan detail identitas dari calon mahasiswa tersebut. Informasi pendaftaran peserta disimpan dalam tabel "*applications*", sementara data mata pelajaran disimpan dalam tabel "*subjects*". Tabel "*grades*" berisi nilai rapor siswa, dan tabel "*notifications*" menyimpan informasi notifikasi status yang akan ditampilkan kepada calon mahasiswa. Tabel "*applications\_study\_programs*" berisi data pendaftaran calon mahasiswa ke program studi tertentu. Tabel "*faculties*" menyimpan informasi tentang fakultas, dan tabel "*study\_programs*" berisi data tentang program studi yang ditawarkan.



Gambar 3. 2 Diagram Tabel Database MySQL

Untuk merancang aplikasi ini, diperlukan diagram *use case* yang melibatkan dua aktor, yaitu Calon Mahasiswa dan Administrator, serta tiga belas *use case*. Calon Mahasiswa memiliki kemampuan untuk "Login", "Register", "Lihat Notifikasi", "Isi Nilai Rapor", "Daftar Program Studi", dan "Logout". Di sisi lain, Administrator memiliki kemampuan untuk "Login", "Lihat Dasbor", "Kelola Akun Pendaftar", "Kelola Nilai Rapor", "Kelola Peserta Pendaftaran", "Rekomendasi Peserta", "Lihat Data Program Studi", "Lihat Data Fakultas", dan "Logout". Diagram *use case* ini dapat dilihat pada Gambar 3.3.



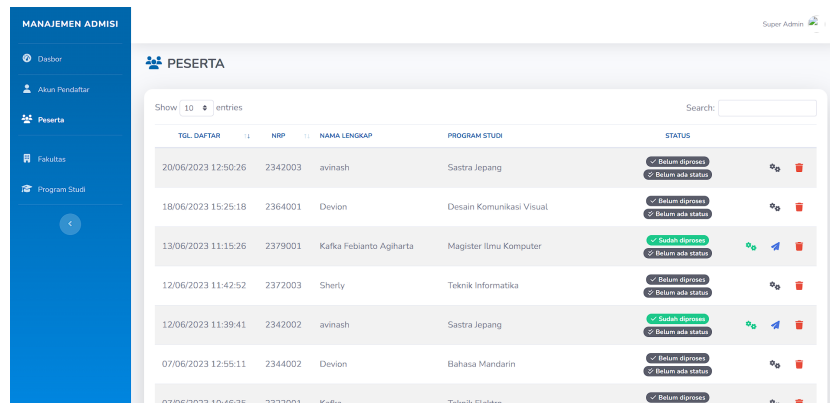
Gambar 3. 3 Use Case Diagram

Gambar 3.4 merupakan hasil implementasi sistem menampilkan halaman tampilan *web administrator*. Di dalam halaman ini administrator dapat melihat nilai rapor dari calon mahasiswa yang mendaftar ke Universitas Kristen Maranatha.

MANAJEMEN ADMISI				
★ NILAI RAPOR				
Matematika	Nilai (Ganjil) 94	KKM (Ganjil) 36	Nilai (Genap) 64	KKM (Genap) 78
Bahasa Inggris	Nilai (Ganjil) 78	KKM (Ganjil) 69	Nilai (Genap) 19	KKM (Genap) 45
Bahasa Indonesia	Nilai (Ganjil) 97	KKM (Ganjil) 17	Nilai (Genap) 92	KKM (Genap) 54
Fisika	Nilai (Ganjil) 39	KKM (Ganjil) 9	Nilai (Genap) 18	KKM (Genap) 68
Kimia	Nilai (Ganjil) 58	KKM (Ganjil) 38	Nilai (Genap) 18	KKM (Genap) 94
Biologi	Nilai (Ganjil) 19	KKM (Ganjil) 63	Nilai (Genap) 2	KKM (Genap) 26

Gambar 3. 4 Tampilan Halaman Nilai Rapor

Administrator juga dapat melihat keseluruhan pendaftar yang telah berhasil melakukan pendaftaran seperti pada Gambar 3.5.



TGL DAFTAR	NRP	NAMA LENGKAP	PROGRAM STUDI	STATUS
20/06/2023 12:50:26	2342003	avinash	Sastra Jepang	Belum diproses Belum ada status
18/06/2023 15:25:18	2364001	Devion	Desain Komunikasi Visual	Belum diproses Belum ada status
13/06/2023 11:15:26	2379001	Kafka Febianto Agiharta	Magister Ilmu Komputer	Sudah diproses Belum ada status
12/06/2023 11:42:52	2372003	Sherly	Teknik Informatika	Belum diproses Belum ada status
12/06/2023 11:39:41	2342002	avinash	Sastra Jepang	Sudah diproses Belum ada status
07/06/2023 12:55:11	2344002	Devion	Bahasa Mandarin	Belum diproses Belum ada status
07/06/2023 10:46:36	2333001	Kafka	Teknik Elektro	Belum diproses

Gambar 3. 5 Tampilan Halaman Peserta

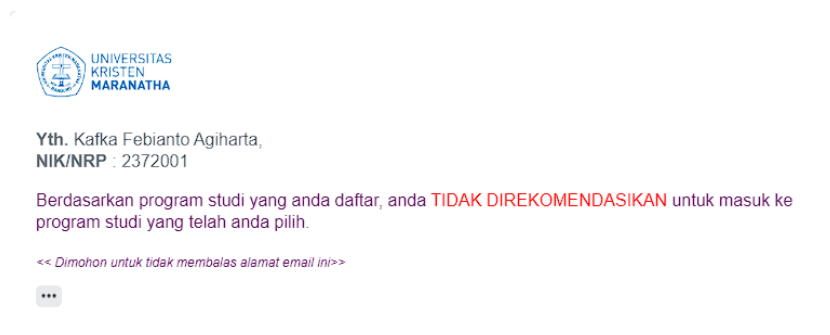
Setelah model *machine learning* berhasil memberikan rekomendasi kepada admin, admin dapat memproses data nilai peserta untuk dianalisis kedalam model yang sudah dilatih untuk menerima atau menolak hasil rekomendasi yang diberikan seperti pada Gambar 3.6.




  
**Proses Rekomendasi Peserta**  
 Ingin memproses rekomendasi peserta ini?

Gambar 3. 6 Proses Rekomendasi Peserta

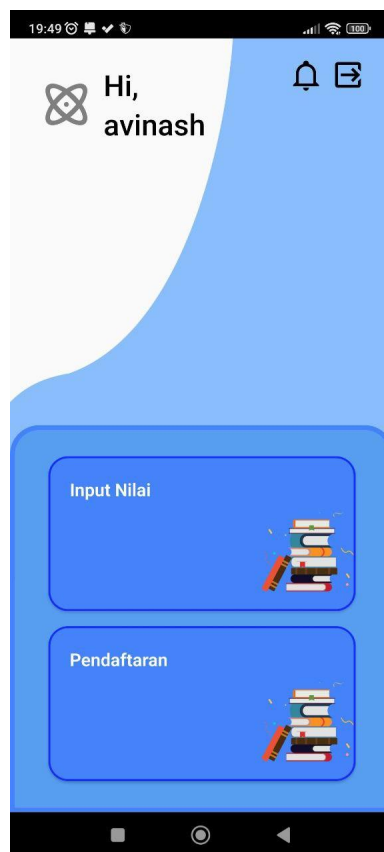
Setelah admin menentukan untuk menerima atau menolak hasil rekomendasi, pendaftar akan memperoleh pesan hasil rekomendasi program studi yang dipilih melalui *email* yang dikirim oleh admin seperti terlihat pada gambar 3.7.



  
**UNIVERSITAS KRISTEN MARANATHA**  
 Yth. Kafka Febianto Agiharta,  
 NIK/NRP : 2372001  
 Berdasarkan program studi yang anda daftar, anda **TIDAK DIREKOMENDASIKAN** untuk masuk ke program studi yang telah anda pilih.  
 << Dimohon untuk tidak membalas alamat email ini >>  
 \*\*\*

Gambar 3. 7 Pesan Email Terkirim ke Peserta

Gambar-gambar berikut dibawah merupakan hasil implementasi sistem menampilkan halaman tampilan *mobile* calon mahasiswa. Pada Gambar 3.8 merupakan tampilan halaman dasbor dari aplikasi *mobile*, di halaman dasbor ini menampilkan dua menu utama yaitu input nilai dan pendaftaran, serta di pojok kanan atas menampilkan dua *icon* yaitu *icon* lonceng untuk menuju ke halaman notifikasi dan *icon* logout untuk menjalankan proses keluar dari akun yang digunakan pada aplikasi.



*Gambar 3. 8 Tampilan Halaman Dasbor*

Pada gambar 3.9 merupakan tampilan halaman *view* nilai dari aplikasi *mobile*, di halaman *view* nilai ini menampilkan daftar nilai yang telah di input oleh pengguna aplikasi, dan di samping kanan judul terdapat *icon* yang dapat ditekan untuk melakukan proses perubahan nilai mata kuliah.

Mata Pelajaran	KKM Ganjil	Nilai Ganjil	KKM Genap	Nilai Genap
Matematika	100	50	70	90
Bahasa Indonesia	70	35	70	90
Bahasa Inggris	70	80	70	65
Fisika	70	80	70	60
Kimia	70	95	70	55
Biologi	70	65	70	80
Ekonomi	70	85	70	90
Geografi	70	55	70	85
Sosiologi	70	70	70	75
Bahasa	70	80	70	60

Gambar 3. 9 Halaman View Nilai

Pada Gambar 3.10 merupakan tampilan halaman input nilai dari aplikasi *mobile*, di halaman input nilai ini menampilkan daftar nilai yang dapat di input oleh pengguna aplikasi, dan terdapat tombol untuk menyimpan nilai yang telah kita *edit*.



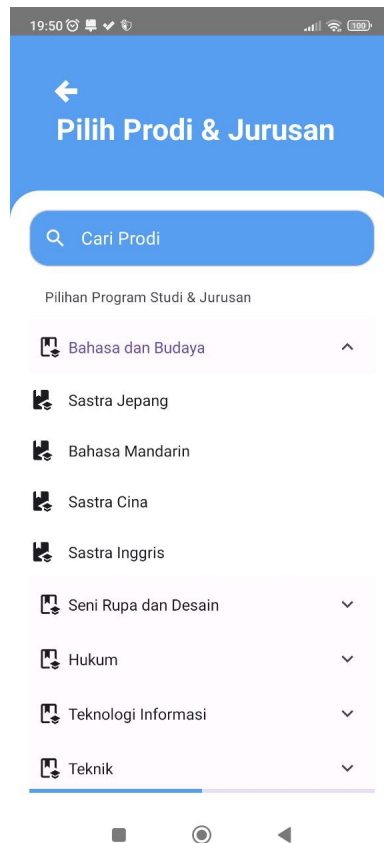
The screenshot shows a mobile application interface with a blue header bar containing a back arrow, the title 'Input Nilai Anda', and a red icon. Below the header is a list of subjects, each with four input fields for scores. The subjects and their scores are:

Subject	Score 1	Score 2	Score 3	Score 4
Inggris	70	80	70	60
Fisika	70	80	70	60
Kimia	70	95	70	55
Biologi	70	65	70	80
Ekonomi	70	85	70	90
Geografi	70	55	70	85
Sosiologi	70	70	70	75
Bahasa Mandarin	70	80	70	60
Bahasa Jepang	70	80	70	65
Bahasa Korea	70	90	70	70

At the bottom of the form is a black 'Submit' button. The mobile status bar at the top shows the time 19:49, signal strength, and battery level at 100%.

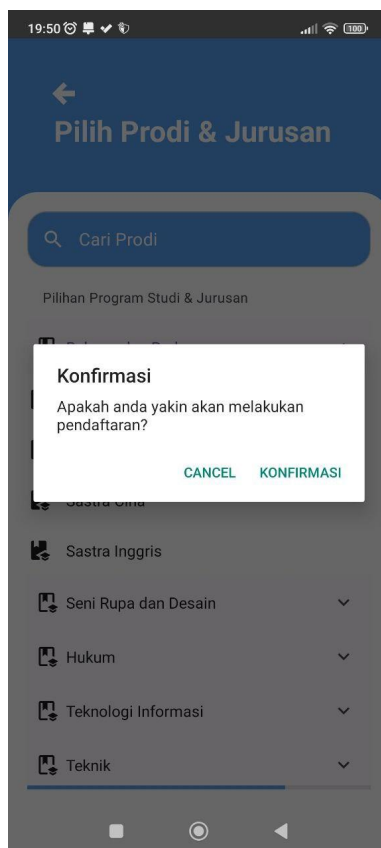
Gambar 3. 10 Tampilan Halaman Input Nilai

Pada Gambar 3.11 merupakan tampilan halaman pilih fakultas dan program studi dari aplikasi *mobile*, di halaman ini merupakan halaman proses pendaftaran program studi. Halaman ini menampilkan semua pilihan fakultas yang terwakilkan dengan *icon* berwarna putih serta program studi dengan *icon* warna hitam yang dapat dipilih.



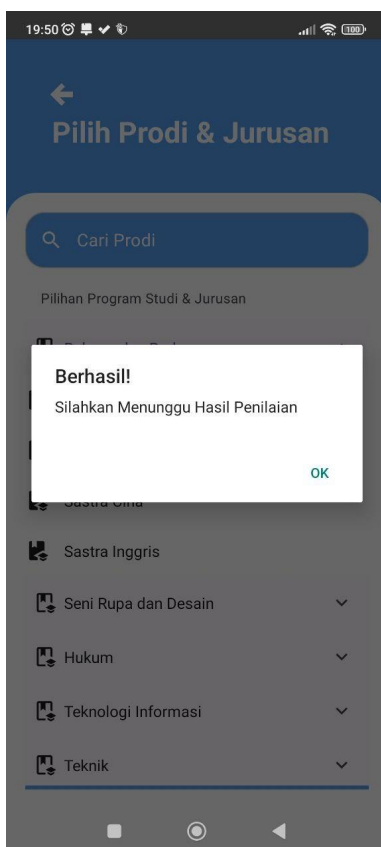
Gambar 3. 11 Tampilan Halaman Fakultas dan Program Studi

Pada gambar 3.12 merupakan tampilan halaman *pop up* konfirmasi pemilihan program studi dari aplikasi *mobile*. *Pop up* konfirmasi ini akan muncul saat pengguna telah memilih pilihan program studi.



*Gambar 3. 12 Tampilan pop up Konfirmasi Pendaftaran*

Pada gambar 3.13 merupakan tampilan halaman *pop up* berhasil melakukan pendaftaran dari aplikasi *mobile*. *Pop up* konfirmasi ini akan muncul saat pengguna telah melakukan konfirmasi pada *pop up* konfirmasi pada Gambar 12. Selanjutnya pengguna dapat menunggu hasil yang akan diberikan lewat *email* maupun notifikasi ada aplikasi *mobile*.



*Gambar 3. 13 Tampilan pop up Berhasil Melakukan Pendaftaran*

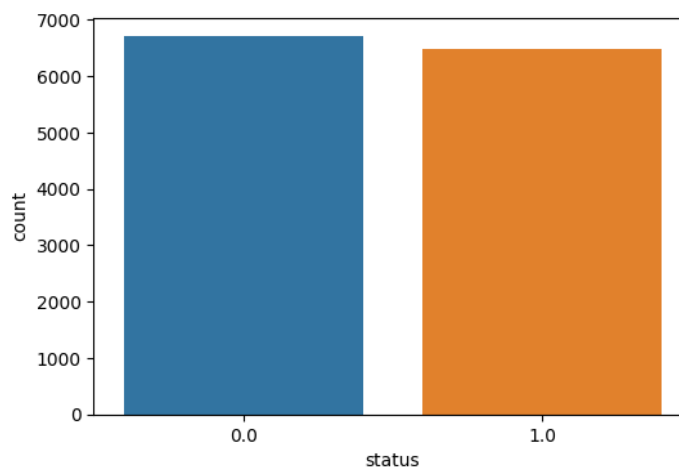
#### IV. HASIL DAN PEMBAHASAN

*Dataset* yang digunakan merupakan dataset nilai rapor calon mahasiswa angkatan 2021-2023. *Dataset* ini terdiri dari 13,188 baris dengan 54 kolom seperti pada tabel 4.1:

Nilai Biologi Ganjil	KKM Biologi Ganjil	....	Program Studi	Status
80	75	...	Pendidikan Dokter	1
76	75	...	Psikologi	0
75	75	...	Manajemen	1

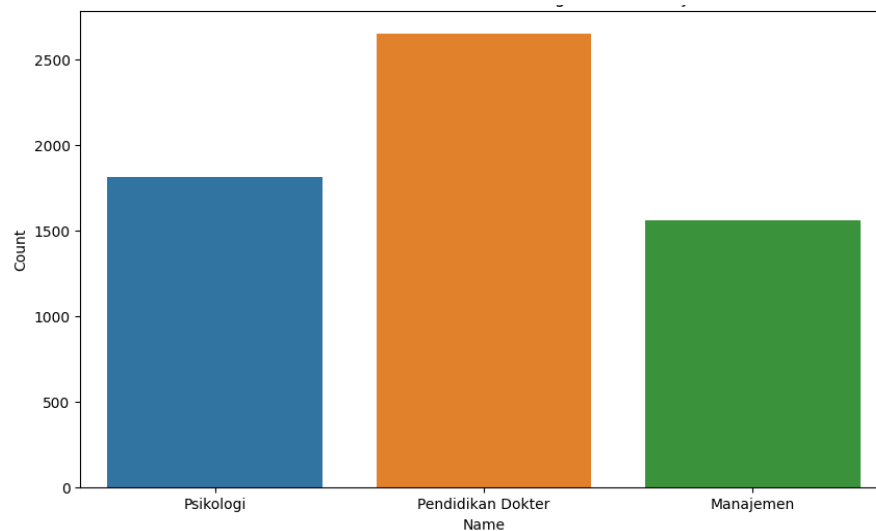
Tabel 4. 1 Tabel Dataset

*Dataset* yang mentah pastinya memiliki data yang *error*, artinya sistem tidak dapat menghitung, memproses, memanipulasi data tersebut. Maka dari itu perlu dilakukan pembersihan *dataset*. *Dataset* dibersihkan dengan cara menghapus nilai-nilai rapor dan KKM mata pelajaran yang bernilai '*NULL*'. Visualisasi dataset yang akan digunakan sebagai label pada model pertama yaitu model untuk memprediksi apakah calon siswa akan direkomendasikan lebih lanjut atau tidak seperti pada Gambar 4.1.



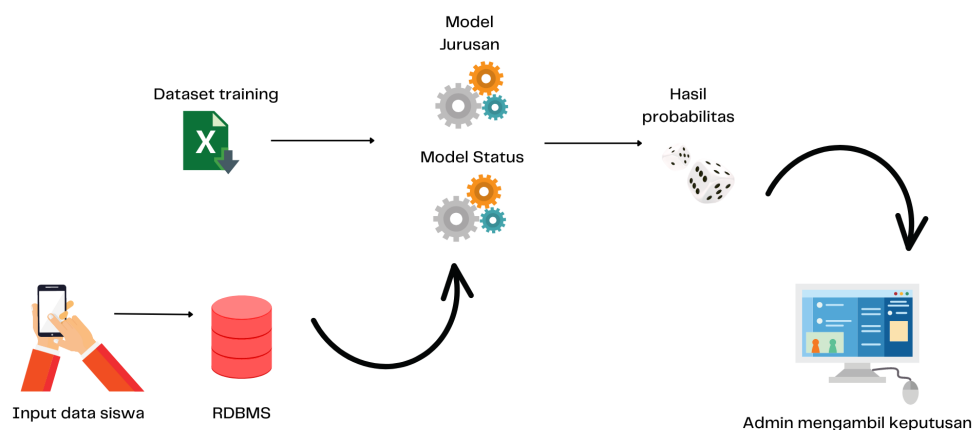
Gambar 4. 1 Rasio Perbandingan Data yang Diterima dan Ditolak

Sedangkan visualisasi yang akan digunakan sebagai label pada model kedua yaitu model untuk memprediksi berapa probabilitas calon siswa akan diterima ke dalam program studi “pendidikan dokter, psikologi, manajemen” seperti pada Gambar 4.2.



Gambar 4. 2 Program Studi yang Digunakan Sebagai Label

Berikut merupakan arsitektur tingkat tinggi dari implementasi pemodelan *machine learning* ke dalam aplikasi mobile seperti pada Gambar 4.3.



Gambar 4. 3 Arsitektur Implementasi Pemodelan Dalam Aplikasi

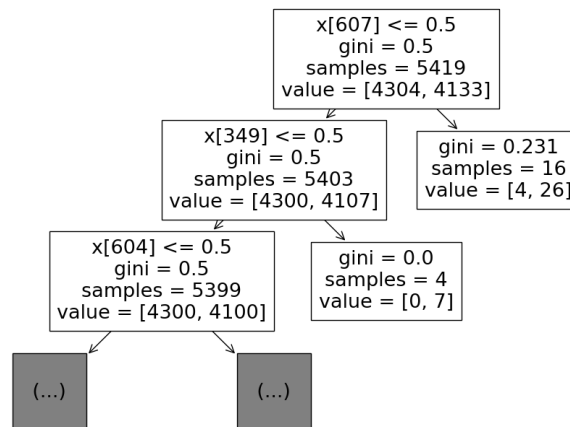
Alur skema dimulai dari input data siswa berupa nilai, KKM, dan informasi yang dibutuhkan. Data-data tersebut kemudian akan disimpan ke dalam RDBMS. Selanjutnya, model *machine learning* akan mencoba memprediksi berdasarkan informasi-informasi yang ada untuk menentukan apakah calon siswa tersebut akan direkomendasikan untuk masuk ke Universitas Kristen Maranatha. Model selanjutnya akan mencoba memprediksi berapa besar probabilitas calon siswa tersebut untuk masuk ke dalam jurusan “pendidikan dokter, psikologi, manajemen”. Selanjutnya, hasil probabilitas tersebut akan ditampilkan ke dalam *website* sehingga admin dapat memutuskan untuk menerima atau menolak calon siswa tersebut berdasarkan hasil rekomendasi *machine learning* dan juga faktor eksternal lainnya.

	Nilai Sosiologi	KKM Sosiologi	Nilai Sosiologi	...	Program Studi	Status
--	--------------------	------------------	--------------------	-----	------------------	--------

	Ganjil		Genap			
1	75	78	77	...	Manajemen	1

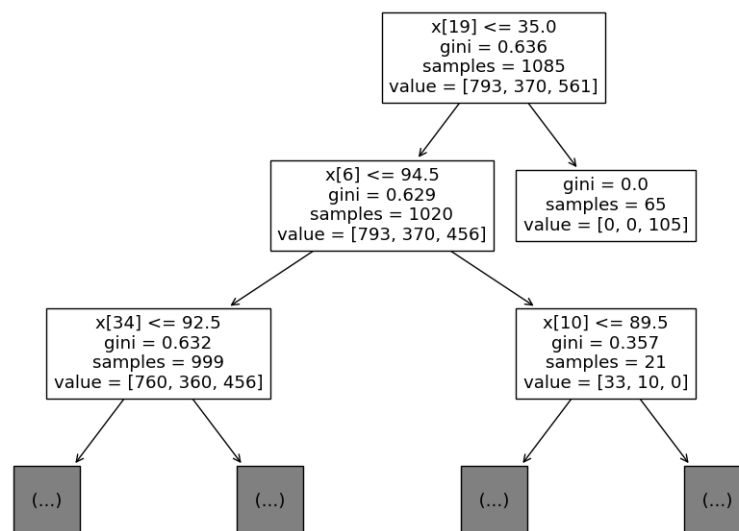
Tabel 4. 2 Data Testing

Tabel 4.2 merupakan salah satu data testing yang digunakan untuk melakukan uji coba pada model *machine learning*. Hasil probabilitas yang diberikan oleh model 1 sebesar 97% direkomendasikan untuk masuk ke Universitas Kristen Maranatha dan probabilitas Program Studi Manajemen sebesar 55%, Program Studi Psikologi sebesar 21% dan Program Studi Pendidikan Dokter sebesar 23%.



Gambar 4. 4 Plot Tree Model 1

Untuk mengetahui fitur yang digunakan sebagai penentu *split* awal model 1 dapat dilihat ada gambar 4.4. Pemisahan dataset pertama kali dilakukan pada fitur ke-607 yang berupa hasil *one hot encoding* pada fitur jurusan dan hal ini masih terus berlanjut sampai dengan *max\_depth* yang dihasilkan ketika menggunakan *randomize search* yaitu *None*.



Gambar 4. 5 Plot Tree Model 2

Penentu *split* awal model 2 dapat dilihat ada gambar 4.5. Pemisahan dataset pertama kali dilakukan pada fitur ke-19 yang berupa nilai Geografi. Dilanjutkan dengan fitur ke-6 yaitu nilai Fisika, pemisahan ini langsung dapat memisahkan dataset secara sempurna untuk kelas Manajemen, sedangkan untuk kelas Pendidikan Dokter dan Psikologi masih terus berlanjut sampai *max\_depth* yang dihasilkan ketika menggunakan *randomize search* yaitu 5.

	F1-Score	ROC
Model 1	0.97	0.98
Model 2	0.55	[0.61, 0.53, 0.68]

Tabel 4. 3 Performa Machine Learning

Performa *machine learning* ditunjukkan pada tabel 4.3 pada model 1 untuk memberikan rekomendasi lebih lanjut kepada admin atau tidak sebesar 97%. Sedangkan untuk hasil akurasi pada model 2 untuk menentukan probabilitas calon siswa akan diterima ke dalam jurusan “pendidikan dokter, psikologi, manajemen” sebesar 55%. Nilai *area under curve* pada model 1 sebesar 98% dan nilai *area under curve* pada model 1 sebesar 61% pada *class* 0, 53% pada *class* 1 dan 68% pada *class* 2.

Untuk pengujian menggunakan metode *blackbox testing*, Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian black box bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi akan. Pengujian Black Box bertumpu pada memastikan tiap proses sudah berfungsi sesuai dengan kebutuhan yang diharapkan. Penguji dapat mengartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari sistem. Sehingga pengujian merupakan suatu cara pelaksanaan program yang bertujuan menemukan kesalahan atau error kemudian memperbaikinya sehingga sistem dapat dikatakan layak untuk digunakan.



Gambar 4. 6 Prinsip Kerja BlackBox Testing

Pengujian form login website ditunjukkan pada tabel 4.4 ,menampilkan semua kemungkinan yang dapat terjadi saat admin *website* melakukan proses masuk ke dalam website admin.

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
----	--------------------	-----------------------	-----------------	------------



1	Login menggunakan email password yang salah	Sistem akan menampilkan pesan gagal login	Sesuai harapan	Valid
2	Login menggunakan password di bawah 6 karakter	Sistem akan menampilkan pesan password minimal 6 karakter	Tidak sesuai harapan	Tidak valid
3	Login tanpa menggunakan format email	Sistem akan menampilkan pesan email tidak sesuai	Sesuai harapan	Valid

Tabel 4. 4 Pengujian Form Login Website

Pengujian form pembuatan akun pendaftar ditunjukan pada tabel 4.5 ,menampilkan semua kemungkinan yang dapat terjadi saat admin *website* melakukan proses pembuatan akun pendaftar

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Membuat akun pendaftar tanpa mengisi apa – apa	Sistem akan menampilkan pesan field kosong harus di isi	Sesuai harapan	Valid
2	Mengklik tombol buak akun pendaftar baru tanpa mengisi apa - apa	Sistem akan menampilkan pesan gagal membuat akun karena field nya belum di isi	Sesuai harapan	Valid
3	Mengisi Email tanpa menggunakan format email saat mendaftar	Sistem akan menampilkan pesan email tidak sesuai	Sesuai harapan	Valid

4	Mengisi password lalu menggunakan button unhide untuk melihat apakah password sesuai dengan apa yang kita ketikan atau tidak	Sistem akan menampilkan password yang kita ketikan bila di klik unhide/hide	Tidak sesuai harapan	Tidak valid
5	Mengisi kolom search sesuai dengan nama yang di cari	Sistem menampilkan hasil sesuai dengan pencarian pada search	Sesuai harapan	Valid
6	Mengisi kolom search dengan tanggal yang ingin di cari	Sistem menampilkan hasil tanggal yang di cari	Sesuai harapan	Valid
7	Mengisi kolom search dengan lambang yang tidak ada pada data	Sistem menampilkan tulisan no data dari apa yang di cari	Sesuai harapan	valid

*Tabel 4. 5 Pengujian Akun Pendaftar*

Pengujian ditunjukan pada tabel 4.6 ,menampilkan semua kemungkinan yang dapat terjadi saat admin *website* melakukan proses pencarian data fakultas

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengisi kolom search dengan huruf yang tidak ada pada data fakultas	Sistem akan menampilkan no data pada hasil pencarian	Tidak sesuai harapan	Tidak valid
2	Mengisi kolom search dengan data yang sesuai pada data fakultas	Sistem akan menampilkan data hasil yang di cari	Tidak sesuai harapan	Tidak valid

*Tabel 4. 6 Pengujian Fakultas*

Pengujian yang ditunjukkan pada tabel 4.7 ,menampilkan semua kemungkinan yang dapat terjadi saat admin *website* melakukan proses pencarian data program studi

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengisi kolom search dengan huruf yang tidak ada pada data program studi	Sistem akan menampilkan no data pada hasil pencarian	sesuai harapan	valid
2	Mengisi kolom search dengan data yang sesuai pada program studi	Sistem akan menampilkan data hasil yang di cari	sesuai harapan	valid

*Tabel 4. 7 Pengujian Program Studi*

Pengujian form registrasi calon mahasiswa ditunjukkan pada tabel 4.8, menampilkan semua kemungkinan yang dapat terjadi saat pengguna melakukan proses pendaftaran akun

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengisi alamat email yang salah saat registrasi	Sistem akan memunculkan pesan email tidak sesuai	Sesuai harapan	Valid
2	Mengosongkan data email nama dan password	Sistem akan memunculkan pesan harap mengisi data yang kosong	Sesuai Harapan	Valid
3	Mengisi data registrasi secara asal dan tidak sesuai ketentuan	Sistem akan memunculkan pesan error pada form	Sesuai Harapan	Valid

		registrasi		
--	--	------------	--	--

Tabel 4. 8 Pengujian Halaman Registrasi Pengujian Halaman Registrasi

Pengujian form login calon mahasiswa ditunjukan pada tabel 4.9 ,menampilkan semua kemungkinan yang dapat terjadi saat pengguna melakukan proses login ke dalam aplikasi *mobile*

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Login menggunakan email password yang salah	Sistem akan menampilkan pesan gagal login	Sesuai harapan	Valid
2	Login menggunakan password di bawah 6 karakter	Sistem Akan memunculkan pop up error pada aplikasi	Sesuai harapan	valid
3	Login tanpa menggunakan format email	Sistem Akan memunculkan pop up error pada aplikasi	Sesuai harapan	Valid

Tabel 4. 9 Pengujian Halaman Login

Pengujian input nilai calon mahasiswa ditunjukan pada tabel 4.10 ,menampilkan semua kemungkinan yang dapat terjadi saat pengguna melakukan memasukan nilai rapor ke dalam aplikasi *mobile*

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menginput nilai dengan Simbol yang tidak sesuai	Sistem Akan memunculkan pop up error pada aplikasi	Sesuai harapan	Valid
2	Menginput nilai dengan angka lebih dari 100 pada sistem	Sistem akan memberikan limit nilai hanya bs di isi sampai 100	Sesuai harapan	valid

Tabel 4. 10 Pengujian Halaman Input Nilai

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Menginput nilai dengan Simbol yang tidak sesuai	Sistem Akan memunculkan pop up error pada aplikasi	Sesuai harapan	Valid
2	Menginput nilai dengan angka lebih dari 100 pada sistem	Sistem akan memberikan limit nilai hanya bs di isi sampai 100	Sesuai harapan	valid

Tabel 4. 10 Pengujian Halaman Input Nilai

Pengujian pencarian jurusan dan program studi yang ingin dipilih oleh calon mahasiswa ditunjukkan pada tabel 4.11, menampilkan semua kemungkinan yang dapat terjadi saat pengguna memasukan pencarian jurusan dan program studi

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1	Mengisi kolom search dengan huruf yang tidak ada pada data prodi jurusan	Sistem akan tidak menampilkan data sama sekali	Sesuai harapan	Valid
2	Mengisi kolom search dengan data yang sesuai pada data fakultas	Sistem akan menampilkan data hasil yang di cari	Sesuai harapan	valid

Tabel 4. 11 Pengujian Pilih Prodi dan Jurusan

## V. DISKUSI DAN EVALUASI

Dalam penelitian tersebut, model *Random Forest* menghadapi kendala dalam mengatasi masalah data tidak seimbang. Meskipun pendekatan SMOTE telah diterapkan, masih terdapat ruang untuk perbaikan. Beberapa metode yang dapat dicoba untuk meningkatkan prediksi adalah mengumpulkan lebih banyak data untuk mencapai keseimbangan *dataset*, menerapkan metode undersampling dengan folding, dan menggunakan model klasifikasi alternatif seperti *Gradient Boosting*, *SVM*, atau *neural network*. Dengan menerapkan metode-metode ini, diharapkan model *Random Forest* dapat mengatasi masalah data tidak seimbang dengan lebih baik dan menghasilkan prediksi yang lebih akurat.

## VI. KESIMPULAN DAN SARAN

Dari hasil penelitian yang dilakukan, masih terdapat *room of improvement* yang perlu dilakukan oleh model *Random Forest* terkait permasalahan data tidak seimbang. Walaupun sudah diterapkan pendekatan SMOTE namun masih belum

sepenuhnya berhasil menangani permasalahan data tidak seimbang. Dalam rangka meningkatkan hasil prediksi, terdapat beberapa metode alternatif yang layak untuk dicoba. Pertama, dapat dilakukan upaya pengumpulan data untuk mencapai keseimbangan yang lebih baik dalam *dataset* yang digunakan. Dengan mengumpulkan lebih banyak data, terutama dari kelas minoritas, dapat membantu meningkatkan performa prediksi. Metode kedua yang dapat diimplementasikan adalah *undersampling* dengan *folding*, di mana jumlah sampel dari kelas mayoritas dikurangi dengan mempertahankan integritas data. Terakhir, sebagai alternatif lain, penggunaan model klasifikasi lain seperti *Gradient Boosting*, *SVM*, dan *neural network* juga dapat dieksplorasi guna meningkatkan hasil prediksi secara signifikan. Dengan menerapkan metode-metode ini, diharapkan dapat meningkatkan performa model dan mengatasi permasalahan data tidak seimbang dalam konteks klasifikasi.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada seluruh dosen, staf, dan rekan-rekan di Universitas Kristen Maranatha yang telah memberikan dukungan penuh dalam penulisan artikel ini. Dukungan dan bantuan dari dosen sangat berharga dalam menyelesaikan tugas ini. Tanpa kerjasama dan kontribusi mereka, artikel ini tidak akan terwujud dengan baik. Semoga artikel ini dapat memberikan manfaat dan kontribusi dalam bidang penelitian yang relevan serta menjadi bagian dari peningkatan pengetahuan dan pemahaman dalam hal tugas holistik.

#### DAFTAR PUSTAKA

- [1] R. Supriyadi, W. Gata, N. Maulidah, and A. Fauzi, Penerapan Algoritma Random Forest Untuk Menentukan Kualitas Anggur Merah, vol. 13, no. 2, pp. 67–75, Dec. 2020.
- [2] R. H. Fajar Mu'Alim, "Implementasi Metode Random Forest Untuk Penjurusan Siswa Di Madrasah Aliyah Negeri Sintang," Implementasi Metode Random Forest Untuk Penjurusan Siswa Di Madrasah Aliyah Negeri Sintang, vol. 14, pp. 116-125, 2022.
- [3] Rani, D.A. and Jakaria, Implementasi Forum Alumni Pondok Pesantren As-Shiddiqiyah Berbasis Web, vol. 2, no. 1, pp. 81-90, 2018.
- [4] S. Sabdho Cathur Kuncoro, "Rancang Bangun Aplikasi Chatting Berbasis Mobile," Rancang Bangun Aplikasi Chatting Berbasis Mobile, p. 491, 2022.
- [5] I. A. T. Adlan Husein Malahella, "Pemanfaatan Framework React Native dalam Pengembangan Aplikasi," Pemanfaatan Framework React Native dalam Pengembangan Aplikasi, vol. 4, pp. 3179-3182, 2020.
- [6] H. M. Mahir, "Pembangunan Aplikasi Pemantauan Kegiatan Siswa," Pembangunan Aplikasi Pemantauan Kegiatan Siswa, vol. 1, pp. 21-23, 2019.
- [7] A. F. F. M. B. Suhari, "Sistem Informasi Kepegawaian Menggunakan Sistem Informasi Kepegawaian Menggunakan," Sistem Informasi Kepegawaian Menggunakan Sistem Informasi Kepegawaian Menggunakan, vol. 12, pp. 1-5, 2022.
- [8] A. F. P. Kevin Oloan Simatupang, "Metode Agile Dalam Perancangan Sistem Informasi Reservasi Metode Agile Dalam Perancangan Sistem Informasi Reservasi," Metode Agile Dalam Perancangan Sistem Informasi Reservasi Metode Agile Dalam Perancangan Sistem Informasi Reservasi, vol. 3, pp. 608-609, 2022.
- [9] A. A. Romi Choirudin, "Implementasi Rest Api Web Service Dalam Membangun Aplikasi Multiplatform Untuk Usaha Jasa," Implementasi Rest Api Web Service Dalam Membangun Aplikasi Multiplatform Untuk Usaha Jasa, vol. 2, pp. 284-286, 2019.
- [10] D. E. M. K. Amin Rulloh, "Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android," Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android, vol. 1, pp. 1-5, 2017.
- [11] S. Aji, D. Pratmanto, A. Ardiansyah and Saifudin, "Implementasi Framework Laravel Dalam Perancangan Sistem Informasi Desa," Indonesian Journal on Software Engineering (IJSE), vol. 7, no. 2, pp. 237-246, 2021.
- [12] M. W. A. Yahya Dwi Wijaya, "Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions," Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions, vol. 4, p. 23, 2021.