

# Rancang Bangun Aplikasi Android “Kuliah Apa?” Berbasis Flutter dan TensorFlow Lite

Arif Taufiq M Pratama  
Program Studi Informatika – Program Sarjana  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
arif.pratama@students.uii.ac.id

Ahmad R. Pratama  
Jurusan Informatika  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
ahmad.raffie@uii.ac.id

**Abstract**—Pemilihan program studi di bangku kuliah adalah salah satu hal penting bagi calon mahasiswa baru, terutama yang masih duduk di bangku SMA. Terlebih lagi dikarenakan adanya pandemi COVID-19 yang melanda seluruh dunia berakibat pada terdisrupsi proses penerimaan mahasiswa baru (PMB), tak terkecuali di Universitas Islam Indonesia (UII). Penelitian ini dilakukan sebagai bagian dari inovasi dalam proses PMB di UII dengan memanfaatkan teknologi *machine learning* untuk merancang dan membangun sebuah aplikasi Android yang dapat memberikan rekomendasi program studi kepada calon mahasiswa baru berdasarkan nilai mata pelajaran, minat, dan data diri lainnya. Aplikasi yang dikembangkan dengan framework Flutter dan TensorFlow Lite ini telah dinyatakan lulus pengujian fungsionalitas dan juga telah melalui proses pengujian usability oleh pengguna dengan nilai di atas rata-rata. Dengan dikembangkannya aplikasi ini, diharapkan proses PMB di UII dapat berjalan dengan lebih baik dan calon mahasiswa baru dapat mendapatkan kemudahan dalam menentukan pilihan program studi yang akan ditempuhnya.

**Keywords**—*aplikasi Android, rekomendasi, program studi kuliah, Flutter, TensorFlow Lite*

## I. PENDAHULUAN

Bagi siswa-siswi menengah atas, proses pemilihan program studi jenjang sarjana merupakan salah satu momen pengambilan keputusan paling penting. Pada tahap ini lah untuk pertama kalinya siswa-siswi dihadapkan pada pilihan program studi yang akan menentukan masa depan mereka kedepannya. Apalagi masih banyak ditemukan calon mahasiswa bingung menentukan pilihan program studi sarjana mereka, dan pada akhirnya mereka memilih program studi yang tidak sesuai dengan potensinya atau hanya mengikuti teman semata.

Dibandingkan dengan proses perpindahan dari jenjang pendidikan lainnya, perpindahan dari jenjang pendidikan menengah atas menuju jenjang pendidikan tinggi merupakan proses yang berbeda. Tidak seperti perpindahan pada jenjang sebelumnya, pada proses perpindahan menuju pendidikan tinggi inilah untuk pertama kalinya siswa dihadapkan pada pilihan program studi bahkan sejak sebelum memulai proses pendidikan di jenjang pendidikan tinggi tersebut.

Terlebih dengan adanya pandemi COVID-19 yang sedang terjadi di seluruh dunia yang menyebabkan banyaknya sekolah maupun instansi pendidikan tidak dibuka, khususnya di Indonesia dan dilaksanakannya pembelajaran secara daring. Ini tentu akan menyebabkan meningkatnya siswa-siswi yang kebingungan dalam menentukan masa depan pendidikannya dikarenakan keterbatasan dalam bimbingan dengan guru baik itu guru akademik maupun konseling. Hal ini diperkuat oleh pengakuan ibu Wanti

Yulianti, M.Pd. yang merupakan salah satu guru bimbingan konseling di SMA Negeri 1 Tasikmalaya tentang meningkat secara drastisnya jumlah orang tua murid yang meminta bimbingan terkait program studi yang paling cocok bagi anak-anak mereka.

Ini tentu akan menjadi masalah bagi mereka ke depannya jika siswa-siswi tersebut tidak mampu mengikuti pembelajaran di kampus tujuan mereka nanti yang pada akhirnya mereka hanya akan merasa bahwa mereka “salah jurusan”. Berdasar penelitian Indonesia Career Center Network (ICCN) tahun 2017, diketahui sebanyak 87 persen mahasiswa Indonesia mengakui bahwa program studi yang diambil tidak sesuai dengan minat dan potensinya sehingga mempengaruhi performa akademik para mahasiswa dan terdapat beberapa yang lebih memilih untuk keluar pada tingkat satu.

Mengenali potensi diri sendiri agar dapat menemukan pilihan yang tepat dalam pemilihan jalur program studi kuliah merupakan hal yang penting agar hal-hal yang telah disebutkan tadi tidak terjadi pada calon mahasiswa baru. Maka dari itu, aplikasi “Kuliah Apa?” menyediakan fitur rekomendasi untuk membantu pemilihan jalur kuliah sarjana bagi para calon mahasiswa baru menggunakan pembelajaran mesin untuk memproses input data berupa nilai beberapa mata pelajaran, jenis kelamin, jenis dan jurusan sekolah, dan minat dari calon mahasiswa baru.

Aplikasi ini sendiri dikembangkan dengan tujuan agar para calon mahasiswa baru memiliki gambaran akan potensi yang mereka miliki dan kecocokannya dengan program studi yang telah disediakan. Sehingga, hal-hal seperti bingung menentukan pilihan kuliah, merasa “salah jurusan”, tidak bisa mengikuti pelajaran kuliah, dan hal-hal lain yang dapat membuat potensi akademik para calon mahasiswa baru menurun atau tidak bagus dapat dihindari.

Pada penelitian ini, rekomendasi yang akan dihasilkan aplikasi akan dikembangkan menggunakan *machine learning* yang dikembangkan secara terpisah di penelitian induk [1] untuk mendapat rekomendasi program studi yang cocok bagi para calon mahasiswa baru yang nantinya akan digunakan dalam proses PMB di UII.

## II. BATASAN MASALAH

Pada penelitian ini, terdapat beberapa model *machine learning* yang telah dikembangkan oleh anggota tim kolaborasi. Namun, model yang berhasil dikembangkan memiliki tingkat akurasi dengan rentang diantara 13%-70%.

Model dari *machine learning* yang akan digunakan merupakan model dengan tingkat akurasi yang rendah yaitu menggunakan *deep learning* yang memiliki akurasi diantara

13%-14%. Model dengan tingkat akurasi yang jauh lebih baik masih belum bisa digunakan dalam aplikasi karena anggota dari tim kolaborasi menemukan kendala teknis yang tidak dimungkinkannya untuk dilakukannya *deployment* model tersebut kedalam aplikasi.

### III. KAJIAN PUSTAKA

Terdapat beberapa penelitian terdahulu, khususnya di Indonesia, yang telah membahas topik tentang rekomendasi program studi ini. Namun demikian, penelitian-penelitian tersebut lebih banyak berfokus pada metode *rule based*, baik yang berupa aturan asosiasi (*association rule*), inferensi fuzzy, atau metode *Preference ranking organization method for enrichment evaluation* (Promethee) untuk mendapat rekomendasi program studi [1, 3-4]. Masih jarang penelitian sejenis di Indonesia yang menggunakan pendekatan sains data berupa *machine learning* dalam proses pemberian rekomendasi tersebut.

Kajian literatur menunjukkan bahwa terdapat beberapa faktor yang dapat mempengaruhi dan berperan besar dalam pemilihan program studi sarjana yang cocok bagi para calon mahasiswa baru. Faktor seperti faktor kepribadian, jenis kelamin, minat atas bidang studi merupakan beberapa faktor dalam proses pemilihan program studi kuliah [2], maka dari itu dalam proses pengembangan faktor-faktor tersebut akan menjadi sebuah data masukan guna menentukan rekomendasi yang tepat bagi para calon mahasiswa baru selain dari faktor nilai akademik maupun nilai ujian (NEM) [5-7].

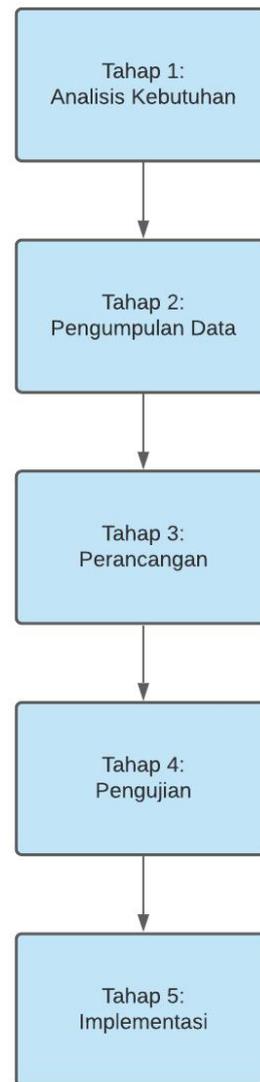
Aplikasi akan dikembangkan dalam bentuk sebuah aplikasi bergerak. Selain agar aplikasi mudah dipakai oleh para calon mahasiswa baru, pemilihan platform aplikasi bergerak sendiri didasari tingkat akses dari para calon mahasiswa baru untuk dapat memakai aplikasi. Kita semua tahu bahwa mayoritas dari pada siswa-siswi sekolah menengah atas memiliki *smartphone* yang dimana pengunduhan dan penggunaan aplikasi dianggap lebih mudah, oleh karena itu pemilihan platform aplikasi bergerak dianggap memiliki tingkat aksesibilitas yang lebih tinggi dari pada *platform* lainnya seperti desktop maupun lainnya.

### IV. METODE PENELITIAN

Seperti yang telah ditunjukkan pada Gambar 1, penelitian ini akan dijalankan dengan beberapa tahapan.

#### A. Analisis Kebutuhan

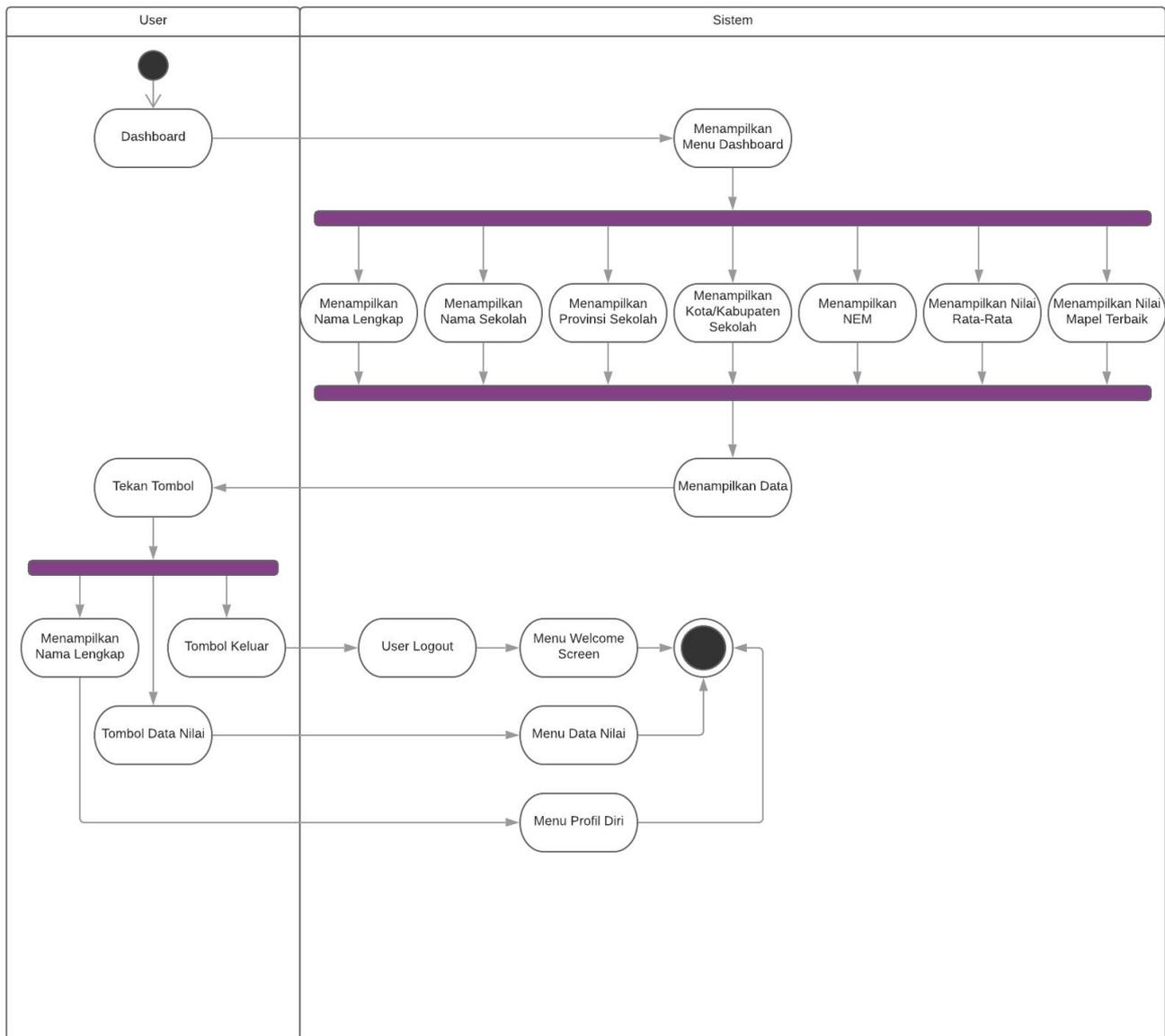
Tahapan pertama dari penelitian ini adalah analisis kebutuhan. Dari hasil kajian pustaka dan analisis kebutuhan, didapatkan bahwa aplikasi akan memiliki data masukan berupa data internal seperti data diri, minat, dan data capaian akademik sekolah [1]. Selain itu, aplikasi juga memiliki kebutuhan akan model dari *machine learning* yang akan dipakai yang dikembangkan oleh anggota tim kolaborasi.



Gambar 1. Tahapan penelitian

Perancangan aplikasi dilakukan guna memberikan pandangan mengenai sistem yang akan dibangun. Sistem akan diberi nama “Kuliah Apa?”. Pada akhir dari tahap perancangan, aplikasi harus dapat terintegrasi dengan baik dengan database, aplikasi harus dapat menyimpan data user ke dalam database, aplikasi harus dapat menampilkan data user, aplikasi harus dapat terintegrasi dengan model *machine learning*, aplikasi harus dapat menampilkan tiga besar rekomendasi pilihan program studi yang dihasilkan dari model *machine learning*.

Sebagai gambaran akan bagaimana sistem bekerja terutama pada bagian fitur utama yaitu fitur dashboard untuk mengetahui rekomendasi dari program studi, dapat dilihat pada Gambar 2 tentang *activity diagram* dari fitur dashboard.



Gambar 2. Activity diagram untuk fitur dashboard

### B. Pengumpulan Data

Data berupa data capaian akademik seluruh mahasiswa program sarjana di UII yang didapatkan dari didapat dari Badan Sistem Informasi (BSI) UII, baik mahasiswa yang masih aktif, yang telah lulus, maupun yang mengalami kegagalan studi (misal: *drop out*) juga diperlukan guna mengembangkan model *machine learning* yang akan dikerjakan oleh anggota tim kolaborasi nantinya. Data yang diperlukan dalam penelitian ini adalah berupa data kategorik dan numerik. Data kategorik berisikan data jenis kelamin, jenis sekolah, jurusan sekolah, dan minat/hobi. Sementara data numerik berisikan data nilai NEM dan nilai beberapa mata pelajaran seperti agama, matematika, bahasa Indonesia, bahasa Inggris, biologi, fisika, kimia, geografi, sejarah, dan ekonomi.

Data-data tersebut digunakan oleh anggota tim kolaborasi untuk keperluan pengembangan pemodelan dari model *machine learning* yang digunakan dan dikembangkan secara terpisah pada penelitian induk [1].

### C. Perancangan

Perancangan dari aplikasi sendiri akan menggunakan Android Studio yang dikembangkan Google dan JetBrains. Android Studio adalah *environment* pengembangan terintegrasi resmi untuk sistem operasi Android Google, yang dibangun di atas perangkat lunak IntelliJ IDEA JetBrains.

Perancangan aplikasi akan dilakukan menggunakan *framework* Flutter yang tersedia di Android Studio. Penggunaan dari Flutter sendiri dipilih dikarenakan Flutter merupakan suatu *framework* lintas platform untuk perancangan aplikasi bergerak yang dikembangkan oleh Google yang memungkinkan para *developer* untuk tidak menulis ulang kode program jika ingin meluncurkan aplikasinya di *platform* yang berbeda.

Selain itu, penggunaan Flutter lebih dipilih dibandingkan dengan salah satu *framework* aplikasi bergerak lainnya seperti React Native adalah Flutter memiliki tingkat “kerapian” yang baik, baik dalam penulisan sintaks kode

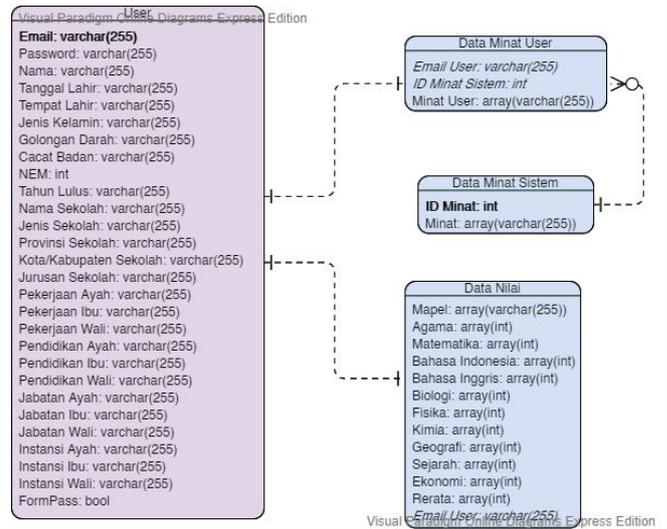
program maupun pada tingkat SDK [8]. Dan Flutter didukung oleh banyaknya *plugins* yang dikembangkan oleh Google sendiri maupun yang dikembangkan oleh pihak lain.

Untuk basis data, aplikasi akan dibangun terintegrasi dengan Firestore sehingga data pengguna dapat tersimpan di *cloud database* milik Google Firebase. Firestore merupakan basis data berbasis cloud NoSQL dan menggunakan JSON tree sebagai skema dari struktur basis datanya [1]. Firestore memungkinkan untuk penyimpanan data menjadi lebih fleksibel dan mudah dipakai dalam perancangan aplikasi [9]. Selain itu, Firebase menyediakan dukungan yang cukup lengkap seperti dukungan analitik dan dukungan apabila perangkat sedang *offline* yang artinya Firestore memungkinkan data untuk disimpan terlebih dahulu dalam perangkat apabila tidak adanya koneksi internet lalu kemudian akan diupload ke dalam cloud database setelah adanya koneksi internet. Dengan banyaknya dukungan yang diberikan oleh Firestore maka Firestore dipilih untuk menjadi teknologi basis data yang digunakan pada aplikasi. Gambaran basis data dalam sistem akan terlihat pada *Entity Relationship Diagram* (ERD) seperti Gambar 3.

Dan untuk mendukung keperluan dari menjalankan model *machine learning* yang telah dikembangkan anggota tim kolaborasi maka, dibutuhkan *plugins* dari Flutter yang bernama TensorFlow Lite. TensorFlow Lite sendiri merupakan *framework deep learning open source* untuk inferensi *on-device*. TensorFlow Lite dikembangkan oleh TensorFlow guna mendukung sistem pembelajaran mesin ke perangkat.

TensorFlow Lite menyediakan framework untuk model TensorFlow yang telah dilatih untuk dikompresi dan diterapkan ke dalam aplikasi *mobile* atau sejenisnya. Dengan bantuan TensorFlow Lite Interpreter, aplikasi dapat memanfaatkan potensi pembuatan inferensi dari model terlatih untuk mendapatkan hasil. Data masukan yang dimasukan user nantinya akan dimasukan oleh aplikasi kedalam model *machine learning* yang telah dikembangkan untuk mendapat hasil akan tetapi, model tersebut haruslah diinterpretasikan terlebih dahulu supaya dapat terbaca dan bekerja dengan baik pada perangkat dan TensorFlow Lite menyediakan proses interpretasi tersebut dengan TensorFlow Lite Interpreter.

Google sendiri pada dasarnya memberi dukungan pembelajaran mesin kepada Flutter melalui *plugins* Firebase ML Kit. Namun ada beberapa alasan mengapa TensorFlow Lite akhirnya dipilih. Pertama, TensorFlow Lite dapat berjalan tanpa adanya koneksi internet tidak seperti ML Kit yang pada beberapa kesempatan dan kondisi membutuhkan perangkat untuk terkoneksi pada internet. Kedua ada pada *pricing* atau harga, memang keduanya tidak mematok harga/gratis untuk digunakan namun bagi ML Kit jika jumlah pengguna atau user telah melebihi jumlah batas yang ditetapkan maka penggunaan dari ML Kit sendiri akan dikenai *pricing* dari Google. Lalu terakhir ada pada masalah fleksibilitas yang dimana penulis merasa penggunaan TensorFlow Lite lebih fleksibel dibanding ML Kit terutama dalam penggunaan model *custom* untuk model pembelajaran mesinnya.



Gambar 3. Entity Relationship Diagram aplikasi

Penggunaan dari TensorFlow Lite sendiri sebelum diinstall pada Flutter harus ditambahkan file *install.bat* untuk Android dan file *install.sh* bagi iOS kedalam *root folder* dari aplikasi. Kedua file tersebut dapat diperoleh pada halaman web “[https://pub.dev/packages/tflite\\_flutter](https://pub.dev/packages/tflite_flutter)”. Setelah proses penginstalan berhasil, model dari *machine learning* yang berformat “.tflite” di-*import* ke dalam folder *assets*, lalu dideklarasikan pada file “*pubspec.yaml*”. Setelah itu, model dapat dipanggil dan diinterpretasikan melalui method “*Interpreter*”. Guna mengetahui jumlah data masukan ataupun keluaran dan tipenya maka untuk memastikannya dapat dipakai method “*getInputTensors()*” atau “*getOutputTensors()*”. Sebelumnya perlu dideklarasikan juga untuk mengalokasikan data tensor pada model yang telah diinterpretasi menggunakan “.*allocateTensors()*”. Untuk memasukan data sendiri terdapat dua cara yaitu dengan menggunakan method “.*run(inputs, outputs)*” jika data masukan hanya satu atau menggunakan method “.*runForMultipleInputs(inputs, outputs)*”. Lalu setelah data masukan berhasil dimasukan dan diproses maka seperti yang telah disebutkan bahwa data keluaran akan berbentuk list data tensor yang bertipe float sehingga data keluaran agar dapat dibaca haruslah dikonversikan ke dalam *string*. Terakhir, proses dari interpretasi model haruslah ditutup agar tidak membebani memori pengguna. Untuk lebih lengkapnya dalam penggunaan TensorFlow Lite untuk aplikasi ini dapat dilihat pada Gambar 4.

TensorFlow Lite sendiri memiliki beberapa kekurangan beberapa diantaranya yaitu untuk versi perangkat Android yang digunakan haruslah Android versi 6.0 (SDK 23.0) dan tensorflow Lite masih belum mendukung data masukan dan keluaran berupa *string*. Data masukan dan keluaran dari TensorFlow Lite sendiri haruslah berupa data tensor dan harus salah satu dari jenis primitif yaitu *float, int, long, dan byte*. Jadi jenis data masukan atau keluaran dari *string* haruslah dikonversi terlebih dahulu menjadi data-data tersebut. Konversi dapat dilakukan dengan berbagai cara salah satunya yaitu dengan *tokenize* ataupun mengubah *string* menjadi jenis data yang mendukung.

```

253
254 Stream<String> loadModel(List<double> input) async* {
255     var outputs = List<double>(37).reshape([1, 37]);
256     print('Loading Model');
257     final interpreter = await Interpreter.fromAsset('new_rf.tflite');
258     interpreter.allocateTensors();
259     print(interpreter.getInputTensors());
260     print(interpreter.getOutputTensors());
261     print('Model Loaded');
262     interpreter.run(input, outputs);
263     print(outputs);
264     interpreter.close();
265     jurusan(outputs);
266 }
267

```

Gambar 4. Kode program TensorFlow Lite pada aplikasi

#### D. Pengujian

Pengujian dari aplikasi akan menggunakan dua metode pengujian yaitu menggunakan metode pengujian *black box* untuk menguji fungsionalitas aplikasi dan pengujian menggunakan metode UMUX (Usability Metric for User Experience) untuk menguji usabilitas dari aplikasi. Sementara itu, untuk menguji keluaran rekomendasi dari aplikasi akan melihat apakah hasil keluaran aplikasi sudah sesuai dengan keluaran dari hasil pengujian yang dilakukan oleh anggota tim kolaborasi terhadap model *machine learning* yang digunakan.

##### a) Black Box Testing

Black Box testing atau lebih dikenal dengan testing fungsionalitas adalah suatu metode teknik pengujian yang merancang kasus uji berdasarkan informasi dari spesifikasi [10]. Pengujian ini dapat dianalogikan menjadi sebuah kotak hitam yang dimana tester tidak dapat melihat apa isi dari kotak hitam tersebut.

##### b) UMUX (Usability Metric for User Experience)

Metode testing UMUX merupakan sebuah metode instrumen tingkatan pendek/jenang rating untuk mengumpulkan data pengguna kuantitatif tentang usabilitas

dari suatu perangkat lunak [11]. Pengujian dilakukan dengan memberikan responden 4 pertanyaan yang dimana setiap pertanyaan akan diberi tingkatan atau rating dari 1 sampai dengan 7, 1 artinya “sangat tidak setuju” sedangkan 7 artinya “sangat setuju”.

##### c) Pengujian Rekomendasi

Pengujian dari keluaran rekomendasi yang dihasilkan oleh aplikasi akan melihat kesesuaian dengan pengujian yang dilakukan oleh anggota tim kolaborasi. Anggota tim kolaborasi sendiri melakukan pengujian terhadap model yang dipakai dengan memakai beberapa evaluator seperti *confusion matrix* (untuk mengambil nilai *precision*, *recall*, dan *F-1 score*), kurva ROC, log loss, dan koefisien Gini.

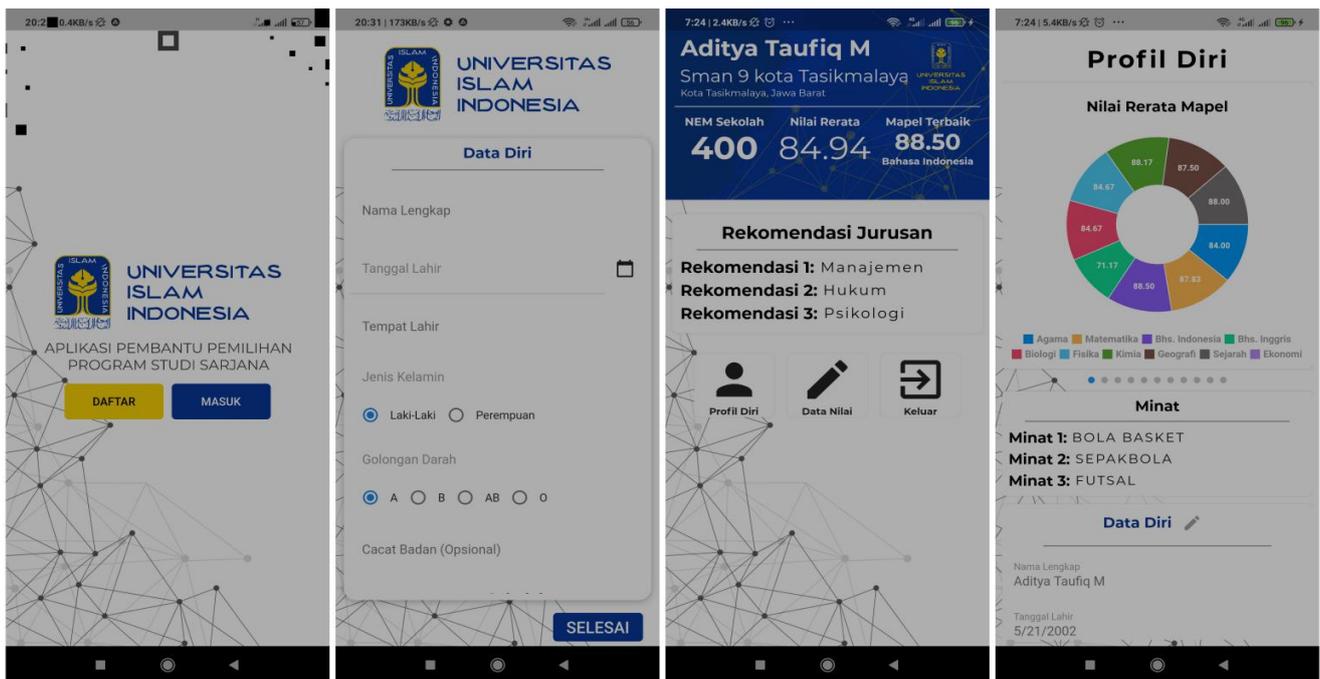
##### E. Implementasi

Penerapan dari aplikasi ini ditujukan pada siswa-siswi sekolah menengah atas dan masyarakat umum yang belum dan ingin melanjutkan studinya dari tingkat SMA menuju tingkat sarjana.

Aplikasi ini diharapkan dapat membantu mereka untuk dapat menentukan jurusan atau program studi mana yang cocok untuk mereka sesuai dengan potensinya sehingga tidak terjadi kebingungan, mengikuti teman, ataupun memilih program studi yang dipaksakan yang dapat berakibat ketidak nyamanan para calon mahasiswa dalam mengikuti pelajaran yang dapat berakibat menurunnya prestasi akademik mereka.

## V. HASIL DAN PEMBAHASAN

Penelitian ini telah berhasil melakukan perancangan dan pengembangan aplikasi Android “Kuliah Apa?” untuk memberikan tiga rekomendasi bagi para calon mahasiswa baru untuk lebih bisa memilih program studi sesuai dengan potensi yang mereka miliki. Gambar 5 menyajikan tampilan antarmuka dari aplikasi tersebut.



Gambar 5. Tangkapan layar antarmuka aplikasi “Kuliah Apa?”

Meskipun aplikasi yang berhasil dikembangkan difokuskan pada platform Android, namun dikarenakan penggunaan framework Flutter dalam perancangannya, secara teoritis aplikasi ini juga dapat diimplementasikan di platform iOS tanpa perlu melakukan proses rancang bangun ulang dari nol. Pun demikian, untuk saat ini perancangan aplikasi difokuskan pada platform Android terlebih dahulu dikarenakan keterbatasan waktu dan sumber daya dan mengingat pangsa pasar Android di Indonesia yang jauh lebih besar jika dibandingkan dengan platform iOS.

Aplikasi “Kuliah Apa?” ini juga telah dapat menyesuaikan tampilan dengan ukuran layar perangkat pengguna. Namun, dengan dinamisnya tampilan layar ini maka jika resolusi layar perangkat pengguna lebih kecil dibanding dengan resolusi pada perangkat ponsel pintar pada umumnya di masa sekarang, maka terdapat kemungkinan bahwa tampilan aplikasi akan mengalami sedikit gangguan.

Integrasi modul *machine learning* menggunakan plugins TensorFlow Lite juga telah berhasil dilakukan. Pun demikian, perlu menjadi catatan bahwa model yang digunakan pada aplikasi ini masih memiliki tingkat akurasi yang tidak terlalu tinggi sehingga masih terbuka lebar peluang untuk peningkatan ke depannya [1]. Kabar baiknya, perancangan aplikasi ini telah dirancang secara khusus agar proses pembaharuan model *machine learning* tersebut dapat dilakukan dengan mudah.

Hasil dari pengujian fungsionalitas aplikasi melalui metode *Black Box testing* seperti yang tersaji pada Tabel 1 memberikan hasil yang memuaskan di mana fitur-fitur yang terdapat dalam aplikasi telah bekerja sesuai dengan apa yang telah diharapkan.

Tabel 1. Hasil pengujian fungsionalitas

No	Fitur	Keterangan
1	Welcome Screen	Menu daftar dan menu masuk.
2	Menu Daftar	Registrasi dan verifikasi akun baru
3	Menu Masuk	Login dengan username dan password yang benar. Gagal login dengan kombinasi yang salah.
4	Lupa Password	Penggantian password melalui verifikasi email.
5	Profil Diri	Input nama, tempat/tanggal lahir, jenis kelamin, golongan darah, dll.
6	Data Nilai dan Minat	Input nilai mata pelajaran dan minat sesuai basis data yang tersimpan.
7	Dashboard	Menampilkan tiga besar rekomendasi prodi berdasarkan perhitungan model <i>machine learning</i> .

Selanjutnya dari tiga hari pengumpulan data yang ditentukan untuk pengujian usability, berhasil dikumpulkan 30 responden dari kalangan siswa-siswi sekolah menengah atas tingkat akhir atau kelas 12. Tabel 2 menampilkan

ringkasan hasil pengisian kuesioner UMUX oleh ketiga puluh orang responden tersebut.

Tabel 2. Hasil pengujian usability dengan UMUX

No	Pertanyaan	Skor	
		Mean	SD
1	Aplikasi ini sesuai dengan kebutuhan saya	4,50	1,36
2	Saya memiliki pengalaman buruk dalam menggunakan aplikasi ini	4,23	1,48
3	Aplikasi ini mudah digunakan	4,30	1,32
4	Saya harus menghabiskan banyak waktu untuk menggunakan aplikasi ini dengan benar	3,53	1,31

Dari hasil pengujian bahwa semua pertanyaan sudah mendapatkan nilai rata-rata di atas nilai tengah (3 dari 7), yang dimana nilai tersebut artinya secara umum memang sudah positif tanggapannya. Untuk *scoring* atau penghitungan metode UMUX tiap pertanyaan akan dianggap sebagai *item* dan perhitungan dapat dilakukan dengan menggunakan rumus setiap *item* ganjil diberi score [score user - 1] dan setiap *item* genap diberi score [7 - score user] lalu jumlahkan score tersebut dan dibagi oleh 24 dan dikalikan 100. Rumus dari penghitungan dari *score* pengujian menggunakan metode UMUX dapat dilihat pada Gambar 6.

$$\frac{[Q1 - 1] + [7 - Q2] + [Q3 - 1] + [7 - Q4]}{24} \cdot 100$$

Gambar 6. Rumus *scoring* metode UMUX

Berdasarkan nilai data dari setiap pertanyaan, maka didapatkan bahwa *score* dari pengujian ini mendapat nilai didapatkan nilai skor rata-rata 69 dari nilai maksimal 100. Berdasarkan nilai tersebut, tingkat usability dari aplikasi ini sedikit berada di atas rata-rata dan berada di kategori baik [12-13]. Meski demikian, nilai yang didapatkan dari perhitungan skor UMUX ini juga menunjukkan pengalaman yang belum terlalu konsisten antar beberapa pengguna yang berbeda. Hal ini menunjukkan masih terbuka lebar peluang untuk perbaikan, baik dari sisi antarmuka (UI) maupun dari sisi pengalaman pengguna itu sendiri (UX).

Lalu, pengujian terhadap keluaran rekomendasi akan dilakukan dengan menyesuaikan hasil keluaran yang dikeluarkan aplikasi dengan hasil keluaran yang dihasilkan dari pengujian model yang dilakukan anggota tim kolaborasi seperti yang ditampilkan pada Tabel 3.

Data masukan dalam pengujian haruslah data bertipe *double* karenanya data yang bertipe *string* harus lah diubah kedalam tipe *double* menggunakan pelabelan yang telah disediakan.

Urutan data masukan berupa jenis kelamin, jenis sekolah, jurusan sekolah, minat/hobi, NEM, rata-rata nilai agama,

rata-rata nilai matematika, rata-rata nilai bahasa Indonesia, rata-rata nilai bahasa Inggris, rata-rata nilai biologi, rata-rata nilai fisika, rata-rata nilai kimia, rata-rata nilai geografi, rata-rata nilai sejarah, dan rata-rata nilai ekonomi.

Tabel 3. Hasil pengujian rekomendasi

No	Input	Output	
		Aplikasi	Model
1	0.0, 3.0, 27.0, 8.0, 400.0, 84.0, 87.8, 88.5, 71.1, 84.6, 84.6, 88.1, 87.5, 88.0, 0.0	21, 14, 28	21, 14, 28
2	1.0, 1.0, 5.0, 42.0, 360.0, 84.0, 87.8, 88.5, 71.1, 0.0, 0.0, 0.0, 87.5, 88.0, 90.0	21, 14, 28	21, 14, 28
3	0.0, 1.0, 5.0, 42.0, 200.0, 70.0, 60.8, 50.5, 71.1, 0.0, 0.0, 0.0, 80.5, 72.0, 83.0	21, 14, 16	21, 14, 16
4	1.0, 2.0, 42.0, 5.0, 350.0, 75.0, 80.8, 83.5, 71.1, 80.0, 90.0, 82.0, 0.0, 0.0, 0.0	14, 21, 28	14, 21, 28

Sesuai yang ditampilkan oleh Tabel 3, dari pengujian rekomendasi yang dihasilkan aplikasi, hasil keluaran telah sesuai dengan hasil yang dihasilkan dari pengujian model *machine learning* yang dipakai yang dilakukan oleh anggota tim kolaborasi.

## VI. KESIMPULAN

Dalam penelitian ini, penulis telah berhasil mengembangkan aplikasi Android “Kuliah Apa?” untuk membantu calon mahasiswa baru dalam memilih program studi di jenjang sarjana dengan bantuan *machine learning*. Aplikasi “Kuliah Apa?” ini sendiri dikembangkan dengan menggunakan framework Flutter yang memungkinkan untuk aplikasi berjalan lintas platform, menggunakan TensorFlow Lite sebagai perantara antara aplikasi dan model machine learning untuk menghasilkan rekomendasi, dan Google Firebase sebagai database untuk menyimpan data pengguna.

Dari hasil pengujian didapatkan bahwa semua fungsionalitas dasar dari aplikasi ini telah berjalan dengan baik. Sementara itu, pengujian kepada pengguna juga menunjukkan tingkat usability aplikasi ini sudah cukup baik mengingat skor UMUX yang didapatkan berada di atas rata-rata berdasarkan peringkat persentilnya. Dan untuk pengujian hasil keluaran rekomendasi aplikasi sudah sesuai dengan yang dilakukan oleh anggota tim kolaborasi.

Dengan dikembangkannya aplikasi “Kuliah Apa?” diharapkan para calon mahasiswa dapat memilih program studi sarjana mana yang sesuai dengan potensi yang mereka miliki sehingga potensi akan ketidakcocokan program studi dan bakat para calon mahasiswa dapat sesuai dan dapat terhindar dari penurunan nilai akademik mereka di kampus tujuan mereka nantinya.

## ACKNOWLEDGMENT

Penulis ingin mengucapkan terimakasih pada Badan Sistem Informasi Universitas Islam Indonesia yang telah menyediakan data awal, Rio Rizky Aryanto, S.Si yang

mengembangkan model machine learning, dan ibu Wanti Yulianti, M.Pd yang telah membantu dalam proses pengumpulan data, dan kepada para responden yang telah melakukan pengujian terhadap aplikasi.

## DAFTAR PUSTAKA

- [1] A. R. Pratama, R. R. Aryanto, and A. T. M. Pratama, *Purwarupa Awal Sistem Rekomendasi Pemilihan Program Studi Sarjana Berbasis Pembelajaran Mesin*, 2020.
- [2] K. M. Galotti, E. Ciner, H. E. Altenbaumer, H. J. Geerts, A. Rupp, and J. Woulfe, “Decision-making styles in a real-life decision: Choosing a college major,” *Personality and Individual Differences*, vol. 41, no. 4, 2006.
- [3] B. B. Zwibelman and R. T. Plant, “Choosing a college major: a prototype decision support system,” *Computers in Human Behavior*, vol. 10, no. 3, 1994.
- [4] B. Albakri, S. Abuhamdeiyeh, and M. Alrefayah, *Rule-Based Expert System To Lead Freshmen Students In Choosing A Suitable College Major In Palestine Polytechnic University*, 2017.
- [5] J. Noble and R. Sawyer, “ACT Research Report Series: Predicting Different Levels of Academic Success in College Using High School GPA and ACT Composite Score,” *PsycEXTRA Dataset*, 2002.
- [6] M. M. Sulphay, N. S. Alkahtani, and A. M. Syed, “Relationship between admission grades and academic achievement,” *Entrepreneurship and Sustainability Issues*, vol. 5, no. 3, 2018.
- [7] E. M. Allensworth and K. Clark, “Are GPAs an Inconsistent Measure across High Schools?,” *Are GPAs an Inconsistent Measure of College Readiness across High Schools? Examining Assumptions about Grades versus Standardized Test Scores*, 2019.
- [8] W. Wenhao, *React Native vs Flutter, cross-platform mobile application frameworks*, 2018.
- [9] C. Khawas and P. Shah, “Application of Firebase in Android App Development-A Study,” *International Journal of Computer Applications*, vol. 179, no. 46, 2018.
- [10] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, 2012.
- [11] J. J. Bosley, “Creating a Short Usability Metric for User Experience (UMUX) Scale,” *Interacting with Computers*, vol. 25, no. 4, 2013
- [12] Finstad, K., “The usability metric for user experience”. *Interacting with Computers*, vol. 22, no. 5, pp.323-327, 2010.
- [13] J. Sauro, “5 Ways to Interpret a SUS Score,” *MeasuringU*, 19-Sep-2018. [Online]. Available: <https://measuringu.com/interpret-sus-score/>.