

**PERANCANGAN APLIKASI COLLABOLIO SEBAGAI
PLATFORM SOSIAL KOLABORATIF BERBASIS ANDROID**

SKRIPSI

Disusun sebagai salah satu syarat untuk menempuh gelar Sarjana Teknik (S. T)



Disusun oleh:

Tio Fahdi

NIM. 3332200055

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SULTAN AGENG TIRTAYASA
2024**

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya sebagai penulis Skripsi berikut:

Judul : Perancangan Aplikasi Collabolio Sebagai Platform Sosial
Kolaboratif Berbasis Android
Nama Mahasiswa : Tio Fahdi
NIM : 3332200055
Fakultas/Jurusan : Teknik/Teknik Elektro

Menyatakan dengan sesungguhnya bahwa skripsi tersebut diatas adalah benar benar hasil karya asli saya dan tidak memuat hasil karya orang lain, kecuali dinyatakan melalui rujukan yang benar dan dapat dipertanggungjawabkan. Apabila dikemudian hari ditemukan hal-hal yang menunjukkan bahwa sebagian atau seluruh karya ini bukan karya saya, maka saya bersedia dituntut melalui hukum yang berlaku. Saya juga bersedia menanggung segala akibat hukum yang timbul dari pernyataan yang secara sadar dan sengaja saya nyatakan melalui lembar ini.

Cilegon, 10 September 2024



Tio Fahdi
NIM. 3332200055

LEMBAR PENGESAHAN

Dengan ini ditetapkan bahwa Skripsi berikut.

Judul : Perancangan Aplikasi Collabolio Sebagai Platform Sosial
Kolaboratif Berbasis Android
Nama Mahasiswa : Tio Fahdi
NIM : 3332200055
Fakultas/Jurusan : Teknik/Teknik Elektro

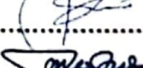
Telah diuji dan dipertahankan pada tanggal 26 November 2024 Melalui Sidang Skripsi di Fakultas Teknik Universitas Sultan Ageng Tirtayasa Cilegon dan dinyatakan LULUS / ~~TIDAK LULUS~~

Dewan Penguji,

Tanda Tangan

Pembimbing 1 : Rian Fahrizal, S.T., M.Eng.
Pembimbing 2 : Ceri Ahendyarti, S.T., M.Eng.
Penguji 1 : Prof. Dr. Ir. Alimuddin ST., MM., MT
Penguji 2 : Masjudin, S.T., M.Eng.


.....

.....

.....

.....

Mengetahui,

Ketua Jurusan Teknik Elektro



Dr. Eng/Rocky/Alfanz, S.T., M.Sc.

NIP. 198103282010121001

PRAKATA

Puji dan syukur saya panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan penelitian yang berjudul Perancangan Aplikasi Collabolio Sebagai Platform Sosial-Kolaboratif Berbasis Android ini tepat pada waktunya. Laporan penelitian ini ditulis dalam rangka memenuhi syarat untuk menyelesaikan Program Sarjana (S1) Jurusan Teknik Elektro Universitas Sultan Ageng Tirtayasa.

Saya menyadari bahwa laporan akhir ini tidak dapat terselesaikan dengan baik tanpa adanya bimbingan, motivasi, dukungan dan bantuan dari berbagai pihak. Untuk itu saya sampaikan rasa hormat dan terima kasih yang setulus-tulusnya kepada :

1. Dr. Eng. Rocky Alvanz, S.T., M.Sc. selaku Ketua Program Studi Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa.
2. Keluarga dan Orang tua yang selalu memberikan doa, dukungan, motivasi dan semangat selama menjalani proses penelitian.
3. Bapak Dr. Suhendar, S.Pd., M.T. selaku dosen wali akademik saya terdahulu.
4. Bapak Fadil Muhammad, S.T., M.T. selaku dosen wali akademik saya Saat ini.
5. Bapak Prof. Dr. Ir. Supriyanto, ST., M.Sc., IPM. Selaku dosen pembimbing studi independent sementara pada saat program Bangkit berlangsung.
6. Bapak Rian Fahrizal, S.T., M.Eng. Selaku dosen pembimbing 1 yang telah membimbing saya dalam topik penelitian skripsi.
7. Ibu Ceri Ahendyarti, S.T., M.Eng. Selaku dosen pembimbing 2 dan pembimbing MBKM Studi Independen yang telah membimbing dalam topik penelitian Perancangan Aplikasi Collabolio Sebagai Platform Sosial-Kolaboratif Berbasis Android.
8. Pihak MBKM dan Bangkit Academy 2023 yang telah memberikan saya kesempatan untuk dapat mengikuti program ini.
9. Teman-teman Bangkit Academy 2023 terutama anggota tim capstone yang telah berbagi pengetahuan, pengalaman, dan semangat yang luar biasa.

Saya menyadari, bahwa laporan akhir ini jauh dari sempurna dan masih perlu banyak perubahan serta perbaikan. Oleh karena itu, saran dan kritik yang

sifatnya dapat membangun sangat diharapkan untuk menyempurnakan laporan akhir ini. Besar harapan penulis, semoga laporan akhir ini dapat bermanfaat.

Serang, 12 Agustus 2024

A handwritten signature in black ink, consisting of a large, stylized 'X' shape with a vertical line extending downwards from the center, and a horizontal line extending to the right from the top right of the 'X'.

Tio Fahdi

ABSTRAK

Tio Fahdi

Teknik Elektro

Perancangan Aplikasi Collabolio Sebagai Platform Sosial Kolaboratif Berbasis Android

Pengangguran merupakan masalah yang sangat sentral dalam perekonomian baik negara maju maupun berkembang. Peningkatan penggunaan teknologi dalam kehidupan sehari-hari telah membawa dampak positif dan negatif, termasuk peningkatan pengangguran di kalangan lulusan perguruan tinggi yang kurang memiliki soft skill, seperti kerja sama tim. Untuk mengatasi masalah ini, proyek "Collabolio" hadir sebagai platform sosial-kolaboratif dengan fitur seperti *dating apps* untuk membantu pengguna menemukan rekan yang tepat sesuai preferensi keahlian yang dibutuhkan untuk berkolaborasi dalam proyek-proyek kehidupan nyata. Platform ini bertujuan meningkatkan kemampuan kerja sama dan kolaborasi lulusan perguruan tinggi. Salah satu fitur utama Collabolio adalah matchmaking berdasarkan keterampilan dan minat pekerjaan. Hasil akhir dari penelitian berupa aplikasi android pencarian rekan project dengan fitur seperti *dating apps*.

Kata Kunci : Pengangguran, Sosial-kolaboratif, *Dating Apps*, Aplikasi Android.

ABSTRACT

Tio Fahdi

Electrical Engineering

Designing Collabolio Application as A Collaborative Social Platform Based on Android

Unemployment remains a central issue in both developed and developing economies. The increased integration of technology into daily life has brought both positive and negative impacts, including a rise in unemployment among college graduates lacking soft skills, such as teamwork. To address this issue, the 'Collabolio' project emerges as a social collaborative platform utilizing feature like dating apps to assist users in finding suitable collaborators for real-life projects. Collabolio name derived from a combination of 'collaboration' and 'portfolio'. The platform aims to enhance cooperation and collaboration skills among college graduates. A key feature of Collabolio is its matchmaking based on skills and job interests. Output from this project is an Android application for project partner search with feature like dating apps.

Keywords: Unemployment, Social-Collaborative, Dating Apps, Android App.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERNYATAAN KEASLIAN SKRIPSI	ii
LEMBAR PENGESAHAN	iii
PRAKATA	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah.....	4
1.6 Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Perancangan Aplikasi	6
2.2 Platform Sosial-Kolaboratif.....	7
2.2 Android OS (Operating System)	8
2.3 Kotlin.....	9
2.4 Android Studio	9
2.5 Firebase.....	10
2.6 Profil Perusahaan.....	11

2.7	Kajian Pustaka	12
BAB III METODOLOGI PERCOBAAN		15
3.1	Metodologi Percobaan.....	15
3.2	Diagram Alir.....	16
3.3	Skenario Aplikasi	17
3.4	Komponen Penelitian	19
3.5	Skenario Pengujian Aplikasi	21
3.6	Tempat Penelitian.....	22
3.7	Pengambilan Data.....	23
BAB IV PEMBAHASAN.....		24
4.1	Analisis Kebutuhan dan Konsep Awal Aplikasi	24
4.2	Desain Logo Aplikasi.....	26
4.3	Desain Low Fidelity dan High Fidelity	28
4.4	Implementasi Desain ke Dalam Aplikasi	31
4.5	Perancangan Program Aplikasi	40
4.6	Pengujian dan Pengembangan Aplikasi	54
BAB V PENUTUP		57
5.1	Kesimpulan.....	57
5.2	Saran.....	57
DAFTAR PUSTAKA.....		59
LAMPIRAN A HASIL PERANCANGAN APLIKASI.....		A-1
LAMPIRAN B TAMPILAN PADA FIREBASE		B-1
LAMPIRAN C HASIL SURVEI APLIKASI.....		C-1
LAMPIRAN D DOKUMENTASI KEGIATAN		D-1

DAFTAR GAMBAR

Gambar 2.1 Logo Perusahaan	11
Gambar 2.2 Bangkit Academy 2023 Organizational Chart	12
Gambar 3.1 Diagram Alir Penelitian	17
Gambar 3.2 Use Case Diagram.....	18
Gambar 3.3 Activity Diagram.....	19
Gambar 3.4 Skala Interpretasi Hasil Skor SUS [32]......	22
Gambar 4.1 Logo Aplikasi Collabolio	26
Gambar 4.2 Palet Warna Pada Logo	28
Gambar 4.3 Desain Wireframe	29
Gambar 4.4 Desain Mockup	30
Gambar 4.5 Desain <i>Splash Screen Activity</i>	32
Gambar 4.6 Desain <i>Welcome Activity</i>	33
Gambar 4.7 Desain <i>Login Activity</i>	34
Gambar 4.8 Desain <i>Register Activity</i>	35
Gambar 4.9 Desain <i>InputBio Activity</i>	36
Gambar 4.10 Desain <i>Swipe Fragment</i>	37
Gambar 4.11 Desain <i>Favorite Fragment</i>	38
Gambar 4.12 Tampilan ke Aplikasi Lain.	39
Gambar 4.13 Desain <i>Profile Fragment</i> dan Navigation Drawer	40
Gambar 4.14 Hasil Autentikasi Pada Firebase.....	44
Gambar 4.15 Distribusi Respon Pengguna	55

DAFTAR TABEL

Tabel 4.1 Analisis Fitur Aplikasi	24
Tabel 4.3 Hasil Penilaian Responden.....	54
Tabel 4.4 Hasil Rata-rata Skor SUS.....	55

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan Teknologi Komunikasi terutama teknologi berbasis perangkat bergerak (*mobile*) yang begitu cepat telah memberikan kemudahan bagi manusia dalam memenuhi kebutuhan informasi [1]. Perkembangan dunia digital secara tidak langsung mendorong masyarakat untuk memiliki *smartphone*. Hampir semua orang di Indonesia kini memiliki *smartphone* karena kemudahan yang ditawarkan. Kecanggihan dan kemudahan tersebut sering dimanfaatkan dalam kehidupan sehari-hari, seperti untuk berbelanja, memesan tiket, atau layanan transportasi. Bahkan, seiring waktu, *smartphone* juga memungkinkan pekerjaan dilakukan dari mana saja. [2].

Pengangguran merupakan masalah yang sangat sentral dalam perekonomian baik negara maju maupun berkembang. Salah satu masalah paling menantang adalah tingkat pengangguran yang memengaruhi laju pertumbuhan ekonomi negara [3]. Organisasi Perburuhan Internasional (ILO) mendefinisikan pengangguran sebagai orang yang berusia kerja, tidak bekerja, tetapi secara aktif mencari pekerjaan [4]. Bertambahnya jumlah pengangguran disebabkan oleh berkurangnya tingkat dalam pembangunan ekonomi. Penurunan pada pembangunan ekonomi menyebabkan penurunan lapangan kerja. Jika tenaga kerja tidak terserap secara optimal, maka akan menimbulkan pengangguran [5].

Pekerjaan di Indonesia terus berkembang terutama di bidang perkotaan dan pekerjaan bergensi. Hal ini yang mendorong meningkatnya minat masyarakat untuk bisa menempuh pendidikan tinggi. Data Direktorat Jenderal Pendidikan Tinggi menunjukkan peningkatan lulusan perguruan tinggi setiap tahun. Namun, hal ini juga menyebabkan masalah pengangguran terdidik, di mana jumlah lulusan tidak seimbang dengan lapangan pekerjaan yang tersedia [6]. Pengangguran terdidik didefinisikan sebagai kondisi di mana seseorang yang memiliki kualifikasi pendidikan tinggi, setidaknya tingkat SMA atau sederajat, sedang mencari pekerjaan atau tidak bekerja [7].

Selain dari jumlah lapangan kerja yang tidak seimbang, pengangguran terdidik di Indonesia disebabkan oleh beberapa faktor. Pertama, Pekerja lebih selektif dalam mencari pekerjaan. Lulusan perguruan tinggi cenderung mencari pekerjaan yang sesuai dengan minat mereka, yang kadang-kadang sulit dipenuhi oleh pasar tenaga kerja [6]. Ditambah lagi, tuntutan kualifikasi tertentu oleh pasar kerja dapat membuat lulusan yang tidak memiliki keterampilan yang sesuai menjadi menganggur[8]. Perubahan cepat dalam kebutuhan pasar kerja, keterbatasan pengalaman kerja, serta ketidakcocokan antara pendidikan dengan kebutuhan industri juga menjadi tantangan yang harus diatasi. Terakhir, krisis ekonomi dan pandemi seperti COVID-19 juga memperburuk masalah pengangguran terdidik [9].

Sejak awal pandemi COVID-19 pada tahun 2020, tingkat pengangguran di Indonesia terus meningkat. Pengurangan aktivitas di luar ruangan mengakibatkan sejumlah perusahaan mengalami kerugian, yang pada gilirannya memaksa mereka untuk mengurangi jumlah pekerja[10]. Menurut data Badan Pusat Statistik (2021), pengangguran meningkat selama pandemi, terutama di sektor pendidikan. Pengangguran terdidik mengalami kenaikan dari 6,69% pada tahun 2020 menjadi 8,55% pada tahun 2021 di kalangan tenaga kerja dengan latar belakang pendidikan perguruan tinggi [6].

Calon pekerja yang akan memasuki dunia kerja harus siap menghadapi persaingan yang ketat dalam mencari pekerjaan. Situasi ini menjadi tantangan bagi *fresh graduate* dan mahasiswa semester akhir, yang perlu memikirkan cara untuk meningkatkan pengalaman agar dapat bersaing di dunia kerja. [11]. Tujuan Utama dari pendidikan tinggi tidak hanya bertujuan untuk memastikan bahwa lulusan-lulusan memiliki kemampuan untuk memasuki pasar kerja. Lebih dari itu, pendidikan tinggi juga bertujuan untuk membekali lulusan dengan keterampilan-keterampilan yang diperlukan untuk berhasil dalam pekerjaan [12]. Terdapat beberapa keterampilan atau *soft skill* yang wajib dimiliki di abad ke 21 ini, yaitu keterampilan untuk komunikasi, berpikir kritis, penyelesaian masalah, kolaborasi, dan kreativitas [13].

Salah satu aspek penting dalam cara kita bekerja adalah kemampuan berkolaborasi atau bekerja sama. Kemampuan berkolaborasi sangat penting bagi setiap individu dan kemampuan tersebut perlu ditingkatkan. Beberapa penelitian

menunjukkan bahwa keterampilan berkolaborasi seringkali kurang diperhatikan, terutama pada mahasiswa maupun pekerja. Keterampilan berkolaborasi mencakup kemampuan bekerja bersama dan bertanggung jawab terhadap apa yang menjadi tugasnya[14].

Untuk mengatasi masalah dari kurangnya pengalaman dan *soft skill* di kalangan lulusan pendidikan tinggi ini, sebagai hasil capstone project pada program Bangkit 2023 dibuat sebuah aplikasi Collabolio sebagai platform sosial-kolaboratif yang merupakan gabungan dari dua aplikasi yaitu dating apps seperti tinder dan aplikasi LinkedIn. Salah satu fitur utama Collabolio adalah *matchmaking* atau lebih tepatnya menggunakan algoritma rekomendasi. Algoritma rekomendasi sendiri adalah teknik komputasional yang digunakan untuk menyaring dan menganalisis data pengguna guna memberikan rekomendasi yang relevan atau personal kepada pengguna. Tujuannya adalah untuk memprediksi preferensi pengguna terhadap item tertentu berdasarkan informasi yang dikumpulkan dari aktivitas pengguna atau karakteristik item itu sendiri [15].

Parameter preferensi yang digunakan dalam aplikasi Collabolio ini yaitu *skill* dan *interest*. Hasil dari rekomendasi tersebut akan di tampilkan pada aplikasi dalam bentuk *card stack view* dengan cara slide kiri atau kanan untuk melihat list yang tersedia. Collabolio dirancang untuk membantu pengguna menemukan rekan kerja yang tepat sebagai sarana berkolaborasi dalam proyek-proyek di kehidupan nyata. Platform ini memungkinkan individu yang memiliki ide proyek untuk menemukan rekan pengembang dari berbagai jenis keahlian untuk membangun proyek mereka bersama-sama.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, didapatkan rumusan masalah yang diperoleh sebagai berikut

1. Bagaimana merancang aplikasi Collabolio berbasis android yang dapat membantu pengguna dalam mencari rekan kolaborasi?
2. Bagaimana cara mengimplementasikan fitur, elemen, dan komponen yang sesuai pada aplikasi Collabolio?

3. Bagaimana cara memastikan aplikasi Collabolio andal dan stabil, serta memberikan pengalaman yang optimal ketika digunakan oleh pengguna?

1.3 Tujuan Penelitian

Adapun tujuan pada penelitian mengenai aplikasi Collabolio ini adalah sebagai berikut.

1. Merancang Aplikasi Collabolio berbasis android untuk membantu pengguna dalam mencari rekan kolaborasi dengan *user experience* yang baik dan optimal.
2. Mengetahui implementasi fitur, elemen, dan komponen yang sesuai dengan aplikasi Collabolio.
3. Mengetahui keandalan dari aplikasi Collabolio dengan menggunakan metode pengujian *system usability test*.

1.4 Manfaat Penelitian

Berdasarkan pada penelitian mengenai aplikasi Collabolio ini, diperoleh manfaat sebagai berikut.

1. Membantu mengatasi masalah kurangnya pengalaman di kalangan lulusan perguruan tinggi dengan meningkatkan keterampilan kerja sama dan kolaborasi tim.
2. Memberikan platform bagi individu yang memiliki ide proyek untuk menemukan rekan pengembang lain dengan keterampilan yang sesuai.
3. Sebagai referensi pada penelitian selanjutnya terkait dengan perancangan aplikasi berbasis android.

1.5 Batasan Masalah

Berikut batasan masalah dari perancangan aplikasi Collabolio sebagai berikut:

1. Perancangan aplikasi menggunakan software android studio dengan bahasa pemrograman kotlin.
2. Perancangan berfokus pada desain dan program aplikasi android.

3. Aplikasi hanya berfokus untuk pengguna mahasiswa atau lulusan baru yang membutuhkan media untuk mencari rekan *project*.
4. Sistem operasi android dan minimal versi android yang digunakan adalah android 7.1 (Nougat)

1.6 Sistematika Penulisan

Dalam penulisan laporan praktikum ini terdiri dari 5 bab. Isi setiap bab adalah sebagai berikut.

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang, rumusan masalah, tujuan percobaan, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berikan dasar teori yang berkaitan dengan pembahasan kecepatan motor dan karakteristik beban.

BAB III METODOLOGI PENELITIAN

Bab ini berikan dasar teori yang mendukung penelitian perancangan aplikasi Collabolio berbasis android.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan hasil dan analisis yang diperoleh dari percobaan yang dilaksanakan.

BAB V KESIMPULAN

Bab ini berisikan kesimpulan dari percobaan yang dilaksanakan dan saran untuk percobaan atau kegiatan mendatang.

BAB II

TINJAUAN PUSTAKA

2.1 Perancangan Aplikasi

Perancangan Aplikasi merupakan suatu proses yang melibatkan definisi dan rancangan suatu aplikasi dengan menggunakan berbagai teknik yang sesuai. Dalam proses ini, diperinci dan dideskripsikan arsitektur aplikasi, detail komponen yang akan digunakan, serta keterbatasan yang mungkin dihadapi selama proses pengembangan [16]. Istilah "aplikasi" mengacu pada program yang siap digunakan dan dirancang untuk menjalankan fungsi atau tugas tertentu bagi pengguna atau aplikasi lainnya. Aplikasi ini dapat dimanfaatkan oleh target atau audiens yang dituju. [17].

Dalam konteks perancangan aplikasi, fokus utamanya adalah pada pembuatan tampilan antarmuka program yang akan digunakan oleh pengguna. Ini melibatkan aspek-aspek seperti desain grafis, navigasi, dan cara pengguna berinteraksi dengan aplikasi [18]. Perancangan aplikasi dapat mencakup segala hal mulai dari rancangan visual hingga kebutuhan fungsional aplikasi tersebut. Proses perancangan aplikasi merupakan tahap awal yang dibutuhkan dan penting dilakukan oleh pengembang untuk merinci bagaimana aplikasi akan bekerja, bagaimana pengguna akan berinteraksi dengannya, dan mengatasi batasan-batasan yang mungkin ada. Dengan perancangan yang baik, pengembang dapat menciptakan aplikasi yang efisien, mudah digunakan, dan memenuhi kebutuhan pengguna dengan baik.[17].

Pengembangan aplikasi *mobile* (*Mobile Development*) merupakan proses pembuatan atau pengembangan software untuk smartphone dan tablet. *Software* ini dapat diinstal sebelumnya pada perangkat melalui *mobile apps store*, atau diakses melalui web browser. Bahasa pemrograman yang biasa digunakan dalam proses pengembangan aplikasi yaitu Bahasa Kotlin, Java, Swift, HTML5, C#, Dart dan lain-lain[19].

Saat ini, terdapat dua platform utama di pasar smartphone modern. Salah satunya adalah iOS dari Apple Inc., yang menjadi sistem operasi untuk jajaran ponsel pintar iPhone yang terkenal. Platform lainnya adalah Android dari Google,

yang tidak hanya digunakan pada perangkat Google, tetapi juga diadopsi oleh banyak produsen perangkat lain yang fokus pada smartphone dan perangkat pintar buatan Google [20].

Pengembangan aplikasi *mobile* saat ini sedang berkembang dengan cepat. Dari industri ritel, telekomunikasi, dan *e-commerce* hingga asuransi, perawatan kesehatan, dan pemerintahan, organisasi di berbagai industri dimana mereka harus memenuhi harapan pengguna agar bisa melakukan transaksi dan mengakses informasi dengan cara yang nyaman dan real-time. Saat ini, perangkat *mobile*, beserta aplikasi *mobile* yang mulai terlihat perkembangannya, merupakan cara paling populer bagi individu dan bisnis untuk terhubung ke internet. Agar tetap relevan, responsif, dan akses yang mudah, maka organisasi perlu mengembangkan aplikasi *mobile* yang dibutuhkan oleh pelanggan, mitra, dan karyawan mereka [19].

2.2 Platform Sosial-Kolaboratif

Platform dapat diartikan sebagai wadah digital yang banyak digunakan oleh manusia untuk berbagai keperluan. Secara umum, platform adalah media yang digunakan untuk menjalankan sistem sesuai dengan program yang telah dirancang. Sebagai contoh, untuk pembelajaran daring, platform yang digunakan berbasis digital. Sementara itu, digital platform adalah kumpulan perangkat lunak yang membentuk suatu sistem tertentu, yang bisa diakses melalui PC atau sistem Android. Pada sistem Android, digital platform dapat berupa aplikasi [21].

Kolaborasi adalah proses interaksi yang kompleks dan bervariasi, melibatkan beberapa orang untuk bekerja sama dengan menggabungkan pemikiran secara berkelanjutan dalam menghadapi suatu masalah dan Setiap pihak yang terlibat saling bergantung satu sama lain [22]. Pekerjaan yang dilakukan secara kelompok memerlukan koordinasi, komunikasi, dan kerja sama dari setiap anggota agar tujuan proyek dapat tercapai dengan optimal. Salah satu dampak positif dari teknologi informasi adalah kemudahan bagi seluruh anggota kelompok kerja untuk mencapai hasil yang lebih efisien dan efektif [23]. Sementara itu, kolaborasi sosial adalah proses interaksi dan kerja sama antara individu atau kelompok dalam konteks sosial atau komunitas. Dalam kolaborasi sosial, individu atau kelompok

berkolaborasi untuk mencapai tujuan tertentu dengan berbagi pengetahuan, sumber daya, ide, atau dukungan dalam lingkungan sosial tersebut [22].

Berdasarkan penjelasan tersebut, Platform sosial kolaboratif adalah suatu sistem atau aplikasi digital yang memungkinkan pengguna untuk berinteraksi, berbagi, dan bekerja sama secara online dalam suatu lingkungan yang terbuka dan terstruktur. Ini memungkinkan individu atau kelompok untuk saling berkomunikasi, berbagi ide, file, atau sumber daya, serta bekerja sama untuk mencapai tujuan bersama. Platform ini mencakup berbagai fitur yang dibutuhkan oleh pengguna untuk saling berbagi informasi. Dengan demikian, platform sosial kolaboratif dapat memfasilitasi kolaborasi tim, proyek, atau komunitas dalam berbagai konteks, dari bisnis hingga pendidikan dan bahkan kegiatan sosial.

2.2 Android OS (Operating System)

Android adalah sebuah sistem operasi yang dirancang untuk perangkat mobile berbasis kernel Linux. Sistem operasi ini mencakup komponen seperti sistem operasi inti, perangkat lunak tengah (*middleware*), dan aplikasi. Salah satu aspek utama dari Android adalah bahwa ia menyediakan platform terbuka yang memungkinkan para pengembang untuk membuat beragam aplikasi [24]. Sejarah Android dimulai ketika Google Inc. mengakuisisi Android Inc., sebuah perusahaan startup yang fokus pada pengembangan perangkat lunak untuk ponsel dan *smartphone*. Untuk mengembangkan Android lebih lanjut, Google bersama dengan sejumlah perusahaan terkemuka dalam industri perangkat keras, perangkat lunak, dan telekomunikasi, membentuk *Open Handset Alliance*. Konsorsium ini terdiri dari 34 perusahaan, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Bersama-sama, mereka menyatakan komitmennya untuk mendukung pengembangan perangkat mobile yang bersifat *open source* [25].

Pada tanggal 5 November 2007, Android resmi diluncurkan, dan bersama *Open Handset Alliance*, mereka memberikan dukungan terhadap pendekatan *open source* dalam pengembangan perangkat *mobile*. Selain itu, Google merilis kode-kode Android di bawah lisensi Apache, yang merupakan lisensi perangkat lunak *open source*. Ini memungkinkan sistem operasi Android untuk didistribusikan secara bebas dan digunakan oleh berbagai vendor *smartphone* [26]. Sebagai hasil

dari pendekatan *open source* ini, Android telah menjadi dasar bagi sebagian besar produsen smartphone, termasuk HTC, Motorola, Samsung, LG, dan banyak lagi. Hal ini disebabkan karena Android memberikan fleksibilitas dan kebebasan bagi vendor untuk mengadaptasi sistem operasi sesuai dengan kebutuhan mereka, yang membuatnya menjadi salah satu sistem operasi paling populer di dunia untuk perangkat *mobile* [25].

2.3 Kotlin

Kotlin adalah sebuah bahasa pemrograman yang didasarkan pada *Java Virtual Machine* (JVM), yang telah meraih popularitas yang signifikan dalam dunia pengembangan aplikasi Android. Bahasa ini terkenal karena pendekatannya yang pragmatis, menggabungkan konsep *Object Oriented Programming* (OOP) dan pemrograman fungsional [27]. Salah satu keunggulan utama Kotlin adalah kemampuannya untuk berinteroperabilitas dengan Java, memungkinkan penggunaan keduanya dalam satu proyek pengembangan. Kotlin tidak hanya terbatas pada pengembangan aplikasi Android, melainkan dapat digunakan untuk beragam platform pengembangan, termasuk aplikasi desktop, web, dan pengembangan backend. Kotlin awalnya dikembangkan oleh JetBrains, perusahaan di balik IntelliJ IDEA, sebuah *Integrated Development Environment* (IDE) yang populer [28].

Google secara resmi mendukung Kotlin sebagai bahasa pemrograman alternatif untuk pengembangan aplikasi Android. Hal ini menandai tingkat penerimaan dan dukungan yang kuat terhadap Kotlin dalam ekosistem pengembangan Android. Dengan fitur-fitur modernnya dan integrasi yang baik dengan API Android, Kotlin telah menjadi pilihan populer bagi pengembang Android yang mencari alternatif yang lebih ekspresif dan efisien dibandingkan dengan Java[29].

2.4 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) yang dikembangkan oleh Google untuk mendukung pengembangan aplikasi Android. IDE ini merupakan evolusi dari Eclipse IDE dan didasarkan pada IntelliJ IDEA,

sebuah IDE Java yang populer. Android Studio direncanakan untuk menjadi IDE resmi pengembangan aplikasi Android, menggantikan peran Eclipse. Dibandingkan dengan Eclipse yang menggunakan *Android Development Tools* (ADT), Android Studio menggunakan Gradle sebagai *environment* pembangunan aplikasi [30]. Android Studio memiliki banyak fitur baru yang membuatnya menjadi pilihan yang lebih unggul untuk para pengembang Android, termasuk alat bantu pengembangan yang lebih canggih dan integrasi yang lebih kuat dengan berbagai komponen Android. Android Studio mendukung berbagai bahasa pemrograman, dengan bahasa utama yang digunakan untuk pengembangan aplikasi Android adalah Java dan Kotlin. Java adalah bahasa resmi untuk pengembangan Android dan telah digunakan secara luas dalam komunitas pengembang Android. Namun, Kotlin juga menjadi semakin populer sebagai alternatif yang lebih modern dan ekspresif [31].

2.5 Firebase

Firebase adalah platform pengembangan aplikasi seluler dan web yang dikembangkan oleh Google. Firebase menyediakan berbagai alat dan layanan yang memungkinkan pengembang untuk membangun, mengelola, dan memantau aplikasi dengan lebih mudah. Firebase sering digunakan untuk pengembangan aplikasi seluler, web, dan perangkat lunak di berbagai platform [32].

Firebase pertama kali didirikan oleh James Tamplin dan Andrew Lee pada tahun 2011 sebagai "Firebase, Inc." Saat itu, Firebase adalah platform berbagi file yang sangat sederhana. Pada tahun 2014, Firebase diluncurkan ulang dengan fokus pada pengembangan aplikasi seluler dan web dengan menyediakan berbagai layanan yang memungkinkan pengembang untuk mengintegrasikan fitur penting, seperti otentikasi, basis data real-time, penyimpanan file, dan analitik ke dalam aplikasi mereka. Firebase awalnya adalah perusahaan independen, tetapi pada tahun 2014, Google mengakuisisi Firebase, dan sejak saat itu, platform ini terus berkembang di bawah naungan Google [33].

Firebase menawarkan berbagai fitur dan layanan untuk membantu pengembang dalam pengembangan, pengelolaan, dan pertumbuhan aplikasi mereka. Beberapa fitur utama yang tersedia di Firebase meliputi Fitur Authentication, Cloud Firestore dan Realtime Database yang menyediakan solusi

basis data berbasis cloud untuk penyimpanan dan sinkronisasi data secara real-time, Cloud Storage, Cloud Functions, dan Crashlytics. Selain itu, Firebase juga memiliki fitur seperti Analytics untuk memantau performa aplikasi, In-App Messaging dan Cloud Messaging untuk interaksi pengguna, serta App Check untuk melindungi aplikasi dari penyalahgunaan atau aktivitas tidak sah [34].

2.6 Profil Perusahaan

Dicoding adalah platform *developer end-to-end* yang terdiri dari program pembelajaran *online* dan platform (Dicoding Academy), jaringan dan acara offline (Dicoding Event), serta pembuatan produk/solusi (Dicoding Challenge). Dicoding menghubungkan perusahaan teknologi dengan pengembang yang tepat dan sebaliknya untuk berkolaborasi dalam menciptakan produk berskala global [35].



Gambar 2.1 Logo Perusahaan

Visi Dicoding adalah menjadi platform edukasi teknologi terdepan yang mendorong akses literasi digital yang lebih luas untuk semua. Dicoding memiliki misi untuk mengakselerasi transisi Indonesia menuju dunia digital melalui pendidikan teknologi yang mentransformasi kehidupan. Dicoding hadir sebagai platform pendidikan teknologi yang membantu menghasilkan talenta digital berstandar global. Semua demi mengakselerasi Indonesia agar menjadi yang terdepan [36].

Bangkit Academy adalah salah satu program yang ada di dalam program Magang dan Studi Independen Bersertifikat (MSIB) yang dirancang oleh Google

bersama GoTo, Traveloka, DeepTech Foundation dan Dicoding sebagai platform pembelajaran bagi peserta. Program ini dirancang untuk membekali mahasiswa dengan keterampilan yang relevan dengan kebutuhan industri saat ini [37]. Terdapat 3 *learning path* di Bangkit Academy 2023 ini yaitu *Machine Learning*, *Cloud Computing*, dan *Android Development*.

Struktur organisasi Bangkit Academy dirancang untuk mendukung kelancaran pelaksanaan program dan memastikan koordinasi yang efektif antara berbagai pihak yang terlibat [37]. Berikut adalah struktur organisasi Bangkit Academy:



Gambar 2.2 Bangkit Academy 2023 Organizational Chart [37]

2.7 Kajian Pustaka

Kajian pustaka ini berisi tentang penelitian sebelumnya yang relevan dengan topik penelitian mengenai Perancangan Aplikasi Collabolio Sebagai Platform Sosial Kolaboratif Berbasis Android. Berikut adalah beberapa penelitian yang menjadi referensi bagi penelitian yang telah dilaksanakan.

Hasil penelitian yang dilakukan oleh [38], yang berjudul “*Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement*”. Penelitian tersebut berisi tentang perancangan aplikasi berbasis web dan PHP yang bertujuan untuk mengetahui rekomendasi karir yang cocok bagi mahasiswa Teknik Informatika

dengan cara mengklasifikasikan keahlian dan keterampilan mahasiswa berdasarkan referensi dari data *professional* TI pada Linked In. Kemudian metode *Collaborative Filtering* dengan algoritma K-NN digunakan untuk menentukan klasifikasi berdasarkan kedekatan antara keterampilan mahasiswa dengan lapangan kerja teknologi informasi. Outputnya adalah rekomendasi bidang pekerjaan teknologi informasi yang dihasilkan dari perhitungan keterampilan mahasiswa IT.

Hasil penelitian [39] yang berjudul “Rancang Bangun Aplikasi Siska (Sistem Informasi Karier) Berbasis Android” yang berisi tentang perancangan aplikasi dengan *software* pengembangan flutter yang berfungsi untuk memberikan informasi secara mendetail terkait lowongan pekerjaan yang disediakan oleh *job-agency* kepada para *job-seeker* yang dapat diakses melalui *smartphone* mereka. Dalam perancangan tersebut, terdapat tiga entitas utama yang terhubung. Pertama, Admin bertugas mengatur jalannya aplikasi, mulai dari manajemen pengguna hingga pembaruan informasi aplikasi. Kedua, *Job-seeker* sebagai pengguna utama yang dapat mencari lowongan sesuai keahlian, mulai dari pendaftaran hingga proses tes (pada versi website) dan jadwal interview. Ketiga, *Job-agency*, berperan sebagai penyedia lowongan pekerjaan.

Hasil penelitian [40] yang berjudul “Analisa Rancang Bangun Aplikasi Biro Jodoh Islami Menggunakan Algoritma *Profile Matching* Berdasarkan Platform Android”. Penelitian tersebut berisi tentang perancangan aplikasi pencarian jodoh berbasis android dengan sistem *profile matching* atau *matchmaking* yang dimana aplikasi akan menampilkan hasilnya berdasarkan preferensi dan kriteria yang dipilih oleh pengguna.

Hasil penelitian [41] yang berjudul “Aplikasi Pencarian Rekan dan Tempat Singgah saat *Touring* Berbasis Android”. Aplikasi tersebut dibuat untuk mencari rekan dan tempat singgah secara online untuk masyarakat yang ingin melakukan perjalanan wisata. Pengguna yang akan melakukan perjalanan pariwisata dapat melihat informasi pengguna lain yang ingin melakukan perjalanan pariwisata ke tujuan yang sama, Kemudian pengguna dapat membuat pencarian rekan dan merespon pencarian rekan pengguna lain. Pengguna juga dapat melakukan pencarian tempat singgah dan merespon pencarian tempat singgah user lain.

Berdasarkan penelitian [38], [39], [40], dan [41] sebagai acuan dari penelitian terdahulu, maka ditentukan topik penelitian yang dilakukan pada penyusunan laporan ini dengan menggabungkan beberapa parameter dan fitur dari penelitian terdahulu untuk menciptakan platform sosial kolaboratif berbasis android.

BAB III

METODOLOGI PERCOBAAN

3.1 Metodologi Percobaan

Dalam penelitian, diperlukan suatu metodologi penelitian yang berperan sebagai panduan dalam menjalankan penelitian. Keberadaan metodologi penelitian ini bertujuan untuk mengatur alur penelitian agar lebih sistematis sehingga proses penelitian dapat berjalan dengan baik dan terorganisir.

1. Studi Literatur

Studi literatur dilakukan bertujuan untuk menemukan informasi mengenai perancangan aplikasi, membantu pengembang untuk memahami kebutuhan pengguna, mengetahui tren teknologi terkini, serta mendapatkan inspirasi desain dari aplikasi sukses yang sudah ada. Dengan menganalisis literatur terkait, pengembang dapat memperoleh pemahaman mendalam tentang persyaratan fungsional yang diperlukan, menghindari kesalahan yang telah terjadi pada penelitian aplikasi sebelumnya, dan bahkan menggunakan informasi tersebut sebagai dasar untuk inovasi.

2. Analisis Kebutuhan dan Konsep Awal Aplikasi

Analisis kebutuhan pengguna bertujuan untuk memastikan bahwa aplikasi yang dikembangkan dapat memenuhi harapan, kebutuhan, dan preferensi pengguna dengan sebaik mungkin. Dengan melakukan analisis ini, tim pengembangan dapat memahami secara mendalam siapa pengguna potensial aplikasi, masalah apa yang terjadi pada mereka, dan bagaimana aplikasi dapat memberikan solusi yang paling efektif dan memuaskan bagi mereka. Hal ini melibatkan pengumpulan dan evaluasi informasi tentang preferensi pengguna, kebiasaan penggunaan, tantangan yang dihadapi, serta harapan dan tujuan pengguna dalam menggunakan aplikasi.

3. Perancangan Desain Aplikasi

Perancangan desain aplikasi bertujuan untuk menciptakan UI (*User Interface*) dan UX (*User Experience*) yang intuitif, efisien, dan memberikan kesan yang baik kepada pengguna. Perancangan awal berupa desain logo aplikasi, desain *low fidelity* (*wireframe*) berupa gambar manual pada kertas dan *high fidelity* (*mockup*) dari tampilan UI aplikasi yang dibuat menggunakan software Figma.

Setelah didapat rancangan awal, kemudian dilanjut dengan mengimplementasikan desain UI tersebut di *software* android studio dalam bentuk file XML.

4. Perancangan Program Aplikasi

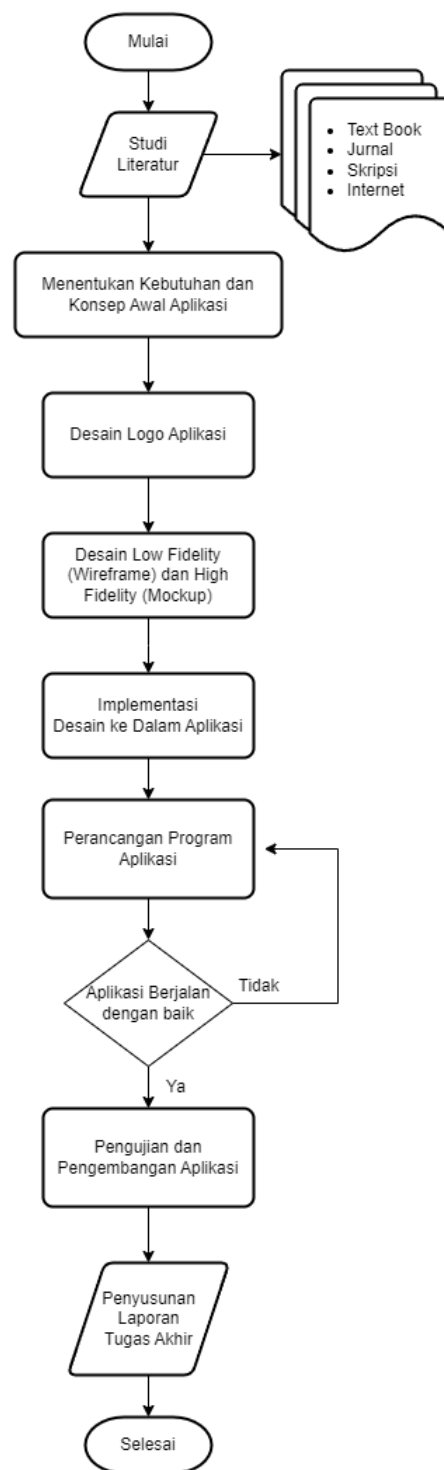
Perancangan program aplikasi bertujuan untuk membangun infrastruktur android yang baik dan terstruktur. Hal ini melibatkan proses merencanakan struktur keseluruhan aplikasi, termasuk arsitektur perangkat lunak, desain database, serta pemilihan arsitektur dan metode yang tepat untuk membangun aplikasi yang baik dan efisien. Perancangan infrastruktur aplikasi Android juga mempertimbangkan aspek keamanan, skalabilitas, dan kinerja aplikasi. Perancangan aplikasi ini menggunakan *software* android studio dengan Bahasa pemrograman Kotlin. Selain itu ditambahkan beberapa komponen untuk membangun aplikasi seperti *activity*, *fragment*, desain XML, dan komponen utama lainnya.

5. Pengujian dan Pengembangan Aplikasi

Pengujian dan pengembangan aplikasi bertujuan untuk memverifikasi dan memastikan bahwa aplikasi tersebut sesuai dengan standar kualitas serta memastikan kinerja dari aplikasi tetap optimal dan relevan. Selain itu, pada proses pengujian dan pengembangan aplikasi, pengembang akan memastikan fungsionalitas yang sesuai, dimana semua fitur beroperasi dengan baik. Selain itu, pengujian dilakukan untuk mengevaluasi performa aplikasi, memeriksa kompatibilitas dengan berbagai perangkat dan versi Android, serta memastikan kualitas antarmuka pengguna yang baik. Pengujian aplikasi dilakukan dengan metode *System Usability Scale (SUS)* untuk mengetahui keandalan aplikasi dan respon dari pengguna secara langsung.

3.2 Diagram Alir

Dalam pelaksanaan penelitian ini, penting untuk menerapkan diagram alir (flowchart) guna memfasilitasi penciptaan suatu urutan perencanaan yang lebih terstruktur dan terorganisir. Diagram alir memberikan gambaran visual tentang langkah-langkah atau proses yang terlibat dalam penelitian, mulai dari tahap awal hingga tahap akhir. Berikut merupakan diagram alir dari penelitian perancangan aplikasi Collabolio.



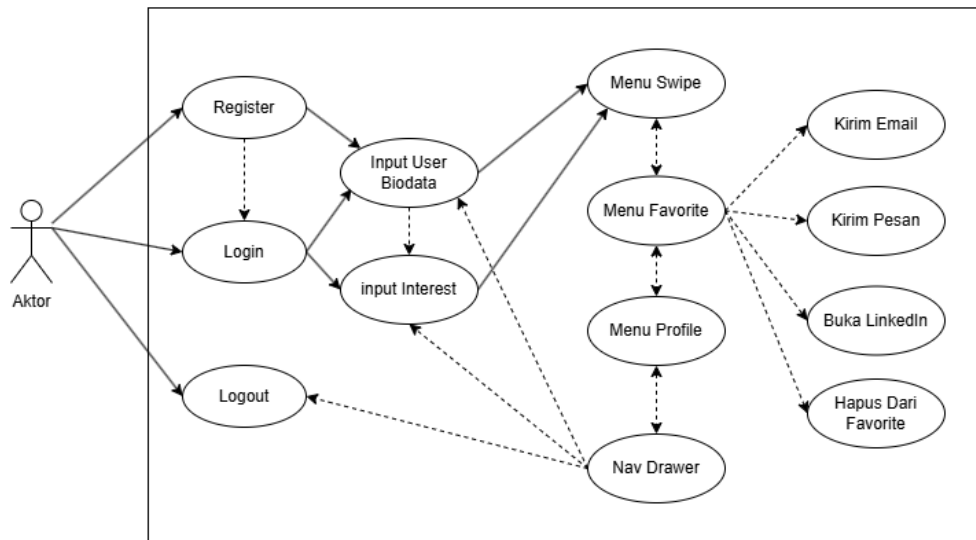
Gambar 3.1 Diagram Alir Penelitian

3.3 Skenario Aplikasi

Untuk scenario aplikasi disini menggunakan pendekatan dari *use case diagram* dan *activity diagram*.

3.3.1 Use Case Diagram

Use Case Diagram adalah salah satu jenis diagram UML (Unified Modeling Language) yang mengilustrasikan interaksi antara sistem dan aktor yang terlibat.



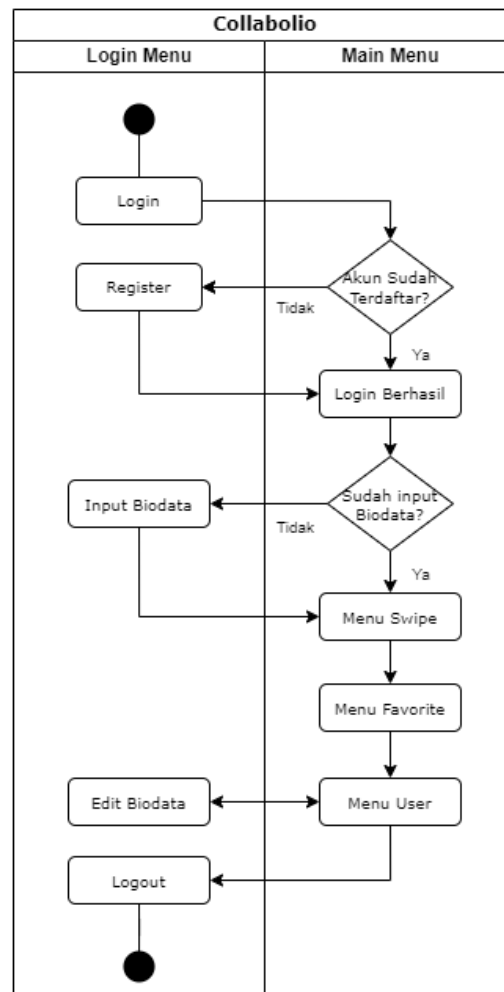
Gambar 3.2 Use Case Diagram

Berdasarkan gambar diatas, aktor disini dapat melakukan register, login dan logout. Ketika aktor berhasil melakukan register dan login, pengguna akan diminta untuk mengisi biodata, kemudian pengguna juga diminta untuk mengisi interest atau keahlian rekan yang dibutuhkan. Setelah itu, pengguna akan dibawa ke menu utama yang terdapat 3 tab tampilan yaitu menu swipe, menu *favorite*, dan menu profile. Pada menu *favorite*, pengguna dapat melakukan beberapa aksi kepada pengguna yang disukai, seperti mengirim email, mengirim pesan, membuka profile linkedIn rekan, dan menghapus rekan dari daftar *favorite*. Selain itu terdapat juga *navigation drawer* untuk melakukan pengeditan data, membuka menu *about*, dan melakukan *logout*.

3.3.2 Activity Diagram

Diagram aktivitas (*activity diagram*) adalah salah satu jenis diagram dalam UML (*Unified Modeling Language*) yang digunakan untuk menggambarkan alur kerja atau perilaku proses dalam sistem perangkat lunak. Diagram ini menunjukkan

serangkaian aktivitas, tindakan, dan keputusan yang terjadi dalam suatu proses dari awal hingga akhir. Berikut *Activity* diagram dari aplikasi Collabolio.



Gambar 3.3 *Activity Diagram*

3.4 Komponen Penelitian

Dalam penelitian ini, komponen penelitian menjadi elemen penting yang membentuk kerangka kerja Perancangan Aplikasi Collabolio Sebagai Platform Sosial-Kolaboratif Berbasis Android. Berikut komponen yang digunakan untuk penelitian.

3.4.1 Spesifikasi Alat

Untuk mendukung pengembangan aplikasi Collabolio, diperlukan alat-alat berikut dengan spesifikasi yang memadai yaitu

1. Laptop

Laptop digunakan untuk menjalankan berbagai jenis perangkat lunak seperti aplikasi, IDE (*Integrated Development Environment*), atau bahkan untuk mengakses data dan sumber daya online. Dalam konteks penelitian, laptop umumnya digunakan untuk melakukan pengolahan data, analisis, pengembangan perangkat lunak, penulisan laporan, dan tugas-tugas lainnya terkait penelitian. Spesifikasi laptop yang digunakan untuk penelitian yaitu AMD Athlon Gold 3150u dengan ram 4gb.

2. Internet

Dalam konteks penelitian, akses internet memiliki peran krusial. Internet digunakan untuk mengumpulkan informasi, data, dan literatur relevan dengan topik penelitian. Hal ini memungkinkan akses ke basis data, jurnal ilmiah, dan sumber daya online lainnya yang mendukung penelitian.

3. Android Studio IDE

Android Studio IDE merupakan komponen yang krusial dalam konteks penelitian yang berfokus pada pengembangan aplikasi Android. IDE ini menyediakan berbagai fitur yang memungkinkan peneliti untuk mendesain tampilan, merancang kode program, dan menganalisis aplikasi Android dengan efisien.

4. Perangkat Android

Perangkat android digunakan untuk melakukan uji coba atau testing aplikasi untuk mengetahui apakah aplikasi dapat berjalan dengan baik atau masih terdapat kesalahan atau bug pada aplikasi. Spesifikasi perangkat yang digunakan adalah perangkat dengan prosessor 8 core, ram 4gb, dan menggunakan versi android 10.

3.4.2 Spesifikasi Aplikasi

Bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa pemrograman Kotlin. Kemudian komponen-komponen yang ditambahkan ke dalam aplikasi yaitu firebase auth untuk proses autentikasi login dan register, firebase firestore untuk penyimpanan data secara cloud, *Card Swipe Layout* untuk tampilan utama aplikasi untuk menggeser *list* pengguna, okhttp untuk memanggil API hasil *training* data machine learning, *Recyclerview* untuk menampilkan list

data *favorite* pengguna, dan *Room Database* untuk menyimpan data pengguna yang disukai secara local.

3.5 Skenario Pengujian Aplikasi

Metode pengujian yang digunakan pada penelitian ini yaitu metode *system usability scale (SUS)*. Metode SUS bersifat *quick and dirty* dan berbasiskan pada kuesioner yang terdiri dari sepuluh pertanyaan dengan jawaban skala 1 sampai 5, yang dimulai dari Sangat Tidak Setuju sampai Sangat Setuju. Metode ini tidak membutuhkan banyak sampel sebagai sumber datanya sehingga metode SUS ini tergolong metode yang efektif, murah dan paling umum digunakan [42]. Pertanyaan yang digunakan dalam pengujian SUS ini adalah sebagai berikut.

Tabel 4.3 Tabel Pertanyaan SUS

No	Pertanyaan
Q1	"Saya berpikir akan menggunakan sistem ini lagi."
Q2	"Saya merasa sistem ini rumit untuk digunakan."
Q3	"Saya merasa sistem ini mudah digunakan."
Q4	"Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini."
Q5	"Saya merasa fitur-fitur sistem ini berjalan dengan semestinya."
Q6	"Saya merasa ada banyak hal yang tidak konsisten (tidak sesuai pada sistem ini)."
Q7	"Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat."
Q8	"Saya merasa sistem ini membingungkan."
Q9	"Saya merasa tidak ada hambatan dalam menggunakan sistem ini."
Q10	"Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini."

Kemudian dalam melakukan perhitungan skor SUS terdapat beberapa aturan sebagai berikut :

1. Pertanyaan Ganjil

Untuk setiap pertanyaan bernomor ganjil (1, 3, 5, 7, 9), nilai pengguna dikurangi dengan 1.

Rumus : $[Skor\ Pertanyaan\ Ganjil = Penilaian\ pengguna - 1]$.

2. Pertanyaan Genap

Untuk setiap pertanyaan bernomor genap (2, 4, 6, 8, 10), nilai maksimum (5) dikurangi dengan penilaian pengguna.

Rumus : $[Skor\ Pertanyaan\ Genap = 5 - Penilaian\ Pengguna]$.

3. Menghitung Skor Responden

Jumlahkan skor dari semua pertanyaan (ganjil dan genap) untuk setiap responden kemudian hasilnya dikalikan dengan 2,5 untuk mendapatkan skor responden.

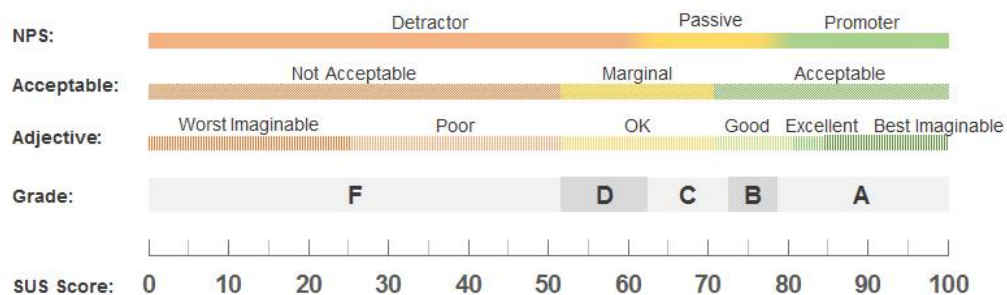
Rumus : $(Total\ Skor\ Semua\ Pertanyaan) \times 2,5 = Skor\ Responden$.

4. Menghitung Skor SUS

Untuk menghitung nilai rata-rata, jumlahkan semua skor responden dan bagi dengan jumlah responden.

Rumus : $\frac{Total\ Skor\ Responden}{Jumlah\ Responden} = Skor\ SUS$.

Setelah didapatkan hasil skor rata-rata SUS, dilanjutkan ke proses interpretasi skor SUS. Dalam menginterpretasikan hasil skor SUS, terdapat lima cara yang bisa digunakan yaitu dengan berdasarkan pada interpretasi perbandingan peringkat persentil, peringkat sifat (*adjective*), tingkat penerimaan (*acceptable*), dan NPS dari skor SUS itu sendiri [42]. Hasil akhirnya dapat di bandingkan dengan gambar skala interpretasi berikut.



Gambar 3.4 Skala Interpretasi Hasil Skor SUS [42].

3.6 Tempat Penelitian

Lokasi penelitian adalah tempat atau objek untuk diadakan suatu penelitian. Lokasi penelitian berada di rumah penulis, yaitu Kelurahan Blokan, Kecamatan Bandung, Kabupaten Serang, Provinsi Banten. Lokasi penelitian ini dipilih berdasarkan pelaksanaan penelitian yang diadakan oleh mitra Kampus Merdeka, yaitu Yayasan Dicoding Indonesia, yang dilaksanakan secara daring atau online dengan menggunakan perangkat komputer/laptop.

3.7 Pengambilan Data

Pengambilan data merupakan tahap kunci dalam proses penelitian Perancangan Aplikasi Collabolio Sebagai Platform Sosial Kolaboratif Berbasis Android. Penelitian ini membutuhkan data yang akurat mengenai fitur yang perlu di masukkan ke aplikasi, metode pemrograman yang di gunakan agar bisa berjalan secara optimal di perangkat android, serta desain UI/UX yang menarik agar pengguna dapat merasakan pengalaman yang impresif saat menggunakan aplikasi. Pengambilan data dilakukan dengan mengambil beberapa sumber yaitu referensi aplikasi Tinder, Bumble dan LinkedIn, referensi dari jurnal penelitian terkait terdahulu, panduan developer android yang disediakan oleh google serta panduan yang diberikan oleh platform Dicoding. Kemudian dilakukan juga pengujian keandalan aplikasi lewat metode *System Usability Scale* (SUS) yaitu pengujian langsung kepada beberapa responden untuk menjalankan aplikasi yang sudah dibuat dan kemudian responden akan mengisi form yang disediakan.

BAB IV

PEMBAHASAN

4.1 Analisis Kebutuhan dan Konsep Awal Aplikasi

Konsep penelitian ini muncul dari analisis mendalam terhadap permasalahan meningkatnya angka pengangguran di Indonesia, khususnya di kalangan lulusan perguruan tinggi yang kurang memiliki pengalaman. Salah satu hambatan utamanya adalah kurangnya keahlian dalam hal kolaborasi bersama rekan tim. Dengan memahami tantangan tersebut, maka dibuatlah aplikasi Collabolio sebagai solusi yang dapat memfasilitasi para lulusan dan profesional muda untuk berkolaborasi dalam suatu *project*.

Nama "Collabolio" diambil dari gabungan kata "*collaboration*" dan "*portfolio*". Hal ini mencerminkan tujuan utama dari aplikasi tersebut, yaitu memfasilitasi kolaborasi antara individu dalam berbagai proyek serta memungkinkan mereka untuk membangun dan memamerkan portofolio profesional mereka. Dengan demikian, nama "Collabolio" tidak hanya mencerminkan esensi dan fokus utama aplikasi, tetapi juga agar mudah diingat dan memiliki daya tarik yang kuat.

Dari Konsep awal tersebut, kemudian dilakukan analisis fitur yang tersedia dalam *dating apps* dengan mengambil dari referensi aplikasi Tinder atau Bumble dan juga aplikasi LinkedIn sehingga dapat di implementasikan pada aplikasi Collabolio. Berikut adalah beberapa fiturnya.

Tabel 4.1 Analisis Fitur Aplikasi

No	Fitur	Deskripsi	Tingkat Kesulitan	Implementasi yang Digunakan
1	Sistem Pencocokan (Matching System)	Menggunakan algoritma untuk mencocokkan pengguna berdasarkan keahlian, minat, dan profil proyek.	Tinggi	Integrasi Algoritma <i>Machine Learning</i> , Android Retrofit
2	Profil Pengguna Terperinci	Memungkinkan pengguna untuk membuat dan mengedit profil dengan informasi	Sedang	Firebase Firestore (CRUD), Android MVVM

		terperinci seperti pendidikan, pengalaman, dan keahlian.		
3	Fitur Swipe	Memungkinkan pengguna untuk mengevaluasi profil rekan potensial dengan cepat dengan menggeser ke kanan (minat) atau ke kiri (tidak cocok).	Sedang	Android RecyclerView dan MVVM
4	Pesan Langsung	Memungkinkan pengguna untuk berkomunikasi secara langsung dengan rekan kolaborasi potensial.	Tinggi	Firebase Cloud Messaging
5	Notifikasi Aktivitas	Memberi tahu pengguna tentang aktivitas baru seperti match baru, pesan masuk, atau pembaruan proyek yang relevan.	Sedang	Firebase Cloud Messaging
6	Verifikasi Profil	Memungkinkan pengguna untuk memverifikasi identitas mereka untuk meningkatkan keamanan dan kepercayaan.	Sedang	Firebase Authentication
7	Pengaturan Privasi	Memungkinkan pengguna untuk mengatur preferensi privasi mereka, termasuk siapa yang dapat melihat profil dan mengirim pesan.	Sedang	Android SharedPreferences API

Berdasarkan tabel 4.1 diatas, diambil beberapa fitur yang dapat diimplementasikan dalam aplikasi Collabolio yaitu sistem pencocokan (Matching System) dengan algoritma *Machine Learning*, Fitur profil pengguna yang terperinci, Fitur Swipe, serta fitur verifikasi profil dan edit profil. Beberapa fitur lain seperti Notifikasi Aktivitas, Pengaturan Privasi, dan Pesan Langsung tidak akan diimplementasikan karena beberapa alasan yaitu tingkat kesulitan yang tinggi

serta ketersinambungan antara ketiganya sehingga menambah kompleksitas dari aplikasi. Sebagai alternatif, percakapan pesan akan dialihkan ke aplikasi lain seperti ke aplikasi pesan atau ke aplikasi email menggunakan *Implicit Intent* Android. *Implicit Intent* sendiri merupakan sebuah mekanisme yang dapat memungkinkan aplikasi untuk berpindah dan mengirim data ke aplikasi lain. Dengan demikian, pengguna masih dapat berkomunikasi dengan rekan kolaborasi mereka dengan mudah melalui aplikasi pesan yang sudah tersedia.

4.2 Desain Logo Aplikasi

Logo merupakan elemen penting dalam perancangan aplikasi karena merupakan representasi visual dari merek dan identitas aplikasi tersebut. Dalam konteks aplikasi Collabolio, logo akan menjadi elemen yang membedakan aplikasi ini dari yang lain, mencerminkan nilai-nilai seperti kolaborasi, inovasi, dan profesionalisme. Desain logo yang kuat dan menarik akan membantu memperkuat citra merek Collabolio di mata pengguna dan membangun koneksi emosional dengan mereka. Oleh karena itu, desain logo perlu memperhatikan aspek-aspek seperti kesan visual, keterbacaan, daya ingat, serta kesesuaian dengan tema dan tujuan aplikasi.



Gambar 4.1 Logo Aplikasi Collabolio

Gambar 4.1 merupakan tampilan desain logo dari aplikasi Collabolio. Bentuk dari logo tersebut merupakan gabungan dari beberapa potongan huruf yang

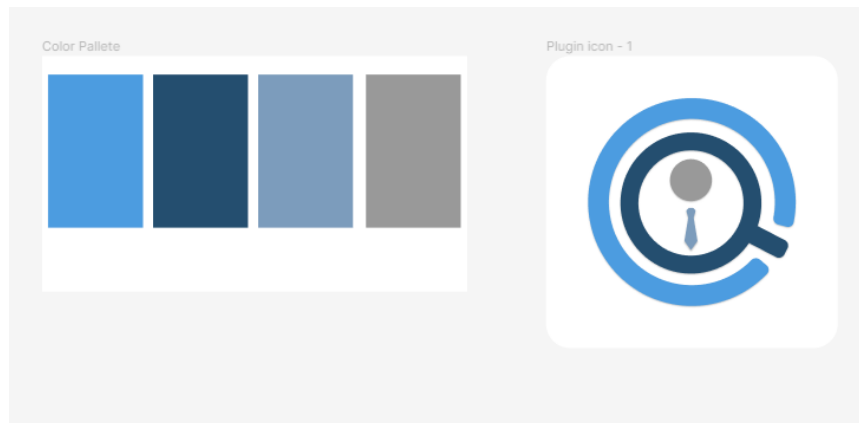
terdapat pada nama Collabolio. Berikut penjelasan makna dari setiap elemen yang terdapat pada logo.

Pertama, Lingkaran terluar dari logo yang membentuk huruf C memiliki makna kesatuan dan inklusivitas. Hal ini melambangkan sifat kolaboratif Collabolio, di mana individu-individu berkumpul dari berbagai latar belakang dan keahlian untuk berkolaborasi. bentuk lingkaran juga menunjukkan bentuk komunitas dan keterhubungan, menyoroti pentingnya kerja tim dan dukungan bersama dalam mencapai tujuan yang sama. Secara keseluruhan, lingkaran luar dalam logo memperkuat nilai-nilai inti kolaborasi dan kerjasama yang menandai Collabolio.

Kedua, Lingkaran yang membentuk gambar kaca pembesar pada logo memiliki makna pencarian, eksplorasi, dan penemuan. Hal tersebut mencerminkan semangat inovasi dan keingintahuan yang akan menjadi bagian dari pengalaman pengguna aplikasi Collabolio. Kaca pembesar sering kali diasosiasikan dengan pencarian informasi, pemahaman yang lebih dalam, dan penemuan hal-hal baru. Dalam konteks Collabolio, kaca pembesar melambangkan upaya pengguna untuk mengeksplorasi peluang kolaborasi, memperluas jaringan profesional, dan menemukan rekan *project* yang sesuai dengan minat dan keahlian mereka.

Terakhir, Lingkaran dan dasi yang terdapat di dalam kaca pembesar membentuk huruf I dan O, atau terlihat membentuk figur manusia berdasi, memiliki makna formalitas dan profesionalisme. Bentuk tersebut juga merupakan representasi dari portofolio pengguna aplikasi. Dengan demikian, simbol tersebut mencerminkan pentingnya portofolio dalam platform Collabolio sebagai cara untuk mempresentasikan keterampilan, proyek-proyek sebelumnya, dan pencapaian profesional pengguna kepada orang lain.

Penggunaan warna pada logo juga sangat berpengaruh terhadap elemen aplikasi. Warna yang dipilih tidak hanya mencerminkan identitas merek Collabolio tetapi juga dapat mempengaruhi persepsi pengguna terhadap aplikasi. Pemilihan warna yang tepat dapat membantu memperkuat pesan merek, menarik perhatian pengguna, dan menciptakan atmosfer yang sesuai dengan tujuan dan nilai-nilai aplikasi. Berikut palet warna yang digunakan pada logo.

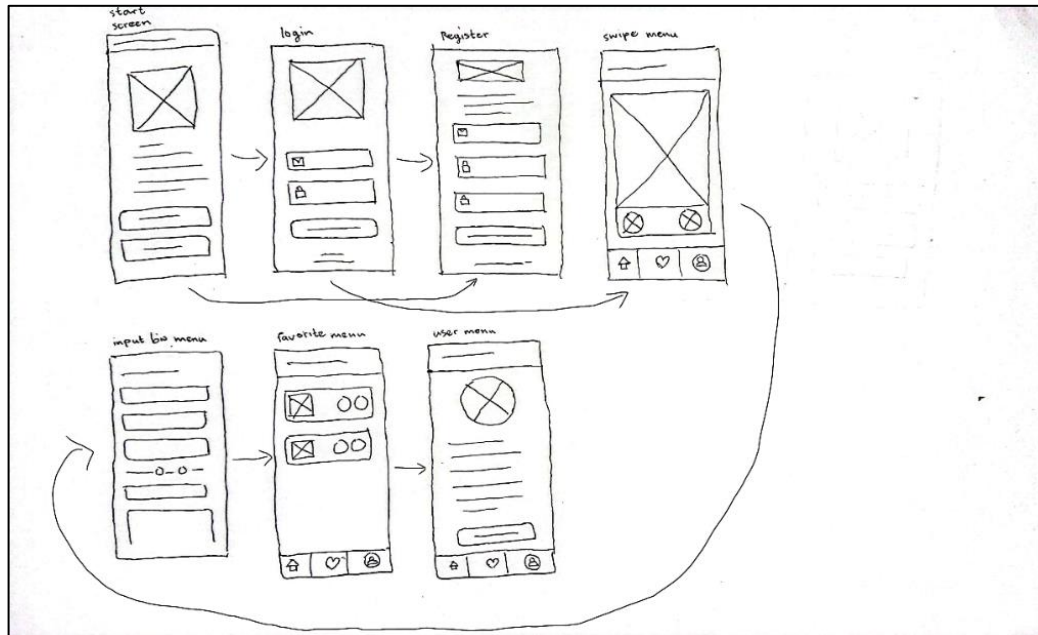


Gambar 4.2 Palet Warna Pada Logo

Penggunaan warna utama untuk aplikasi Collabolio adalah dari rentang warna biru cerah sampai biru gelap dengan tambahan warna abu-abu. Pemilihan warna ini tidak hanya mencerminkan unsur kepercayaan dan profesionalisme, tetapi juga menimbulkan kesan yang tenang dan stabil bagi pengguna. Warna biru cerah dapat memberikan kesan yang segar dan inovatif, sementara warna biru gelap menambahkan sentuhan elegan dan serius. Kombinasi dari kedua warna ini menciptakan harmoni visual yang dapat memperkuat identitas aplikasi dan menciptakan pengalaman pengguna yang konsisten dan menarik di seluruh aplikasi. Dengan demikian, penggunaan warna tersebut dalam desain aplikasi Collabolio diharapkan dapat menciptakan kesan yang positif dan menarik bagi pengguna serta memperkuat citra merek secara keseluruhan.

4.3 Desain Low Fidelity dan High Fidelity

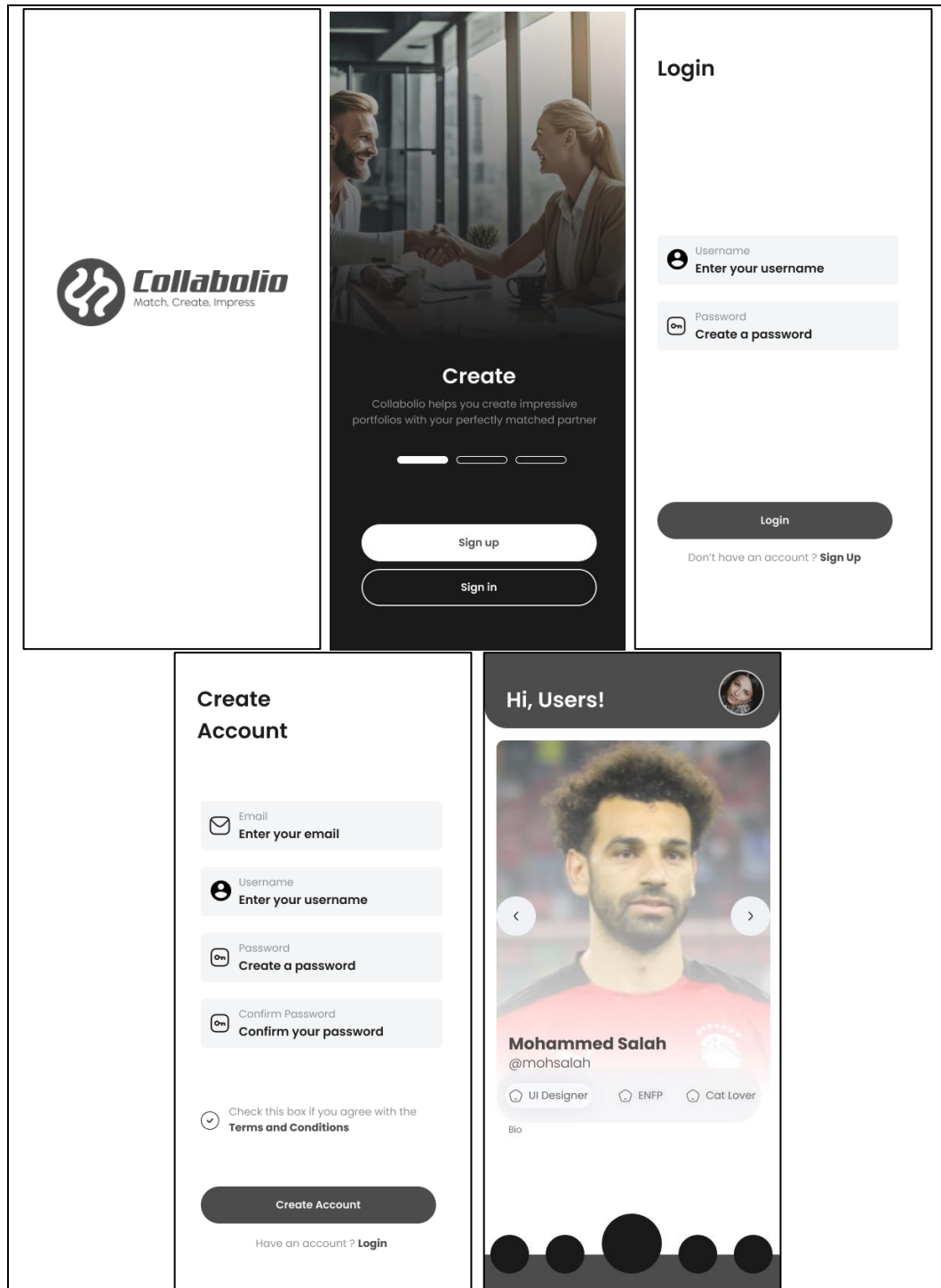
Untuk mendukung pengembangan aplikasi Collabolio, dibuat juga desain *low fidelity* (*Wireframe*) dan *high fidelity* (*Mockup*). Desain *low fidelity* akan memberikan gambaran awal tentang tata letak dan fungsi dasar aplikasi tanpa detail visual yang mendalam. Sementara itu, desain *high fidelity* akan menampilkan detail visual yang lebih lengkap, termasuk warna, tipografi, dan grafis yang lebih mendetail. Berikut untuk desain *low fidelity* yang dibuat menggunakan sketsa sederhana.



Gambar 4.3 Desain Wireframe

Pada gambar 4.3 merupakan desain *wireframe* aplikasi collabolio yang terdapat beberapa halaman yaitu *start/welcome screen* yang berfungsi sebagai tampilan awal atau sambutan kepada pengguna, halaman *login* Dimana pengguna diminta untuk memasukkan email dan kata sandi yang sudah terdaftar sebagai syarat masuk aplikasi, halaman *register* jika pengguna belum memiliki akun yang terdaftar ke dalam aplikasi, halaman menu utama yang memiliki 3 tab menu yaitu *swipe menu* sebagai menu utama aplikasi, *favourite menu* yaitu menu yang menampilkan rekan yang disukai atau diminati, dan *user menu* yang berisi biodata dari pengguna.

Setelah gambaran kasar dari tampilan aplikasi sudah didapatkan dan disetujui oleh tim pengembang, langkah berikutnya adalah mentransformasikan gambar tersebut menjadi desain *mockup* menggunakan *software* Figma. Desain *mockup* ini akan memperinci tampilan antarmuka pengguna (UI) dari aplikasi Collabolio, termasuk aspek-aspek seperti layout, warna, tipografi, dan ikon. Desain *mockup* ini akan menjadi dasar bagi tahap pengembangan selanjutnya, memastikan konsistensi desain dan kesesuaian dengan kebutuhan dan harapan pengguna. Berikut adalah hasilnya.



Gambar 4.4 Desain Mockup

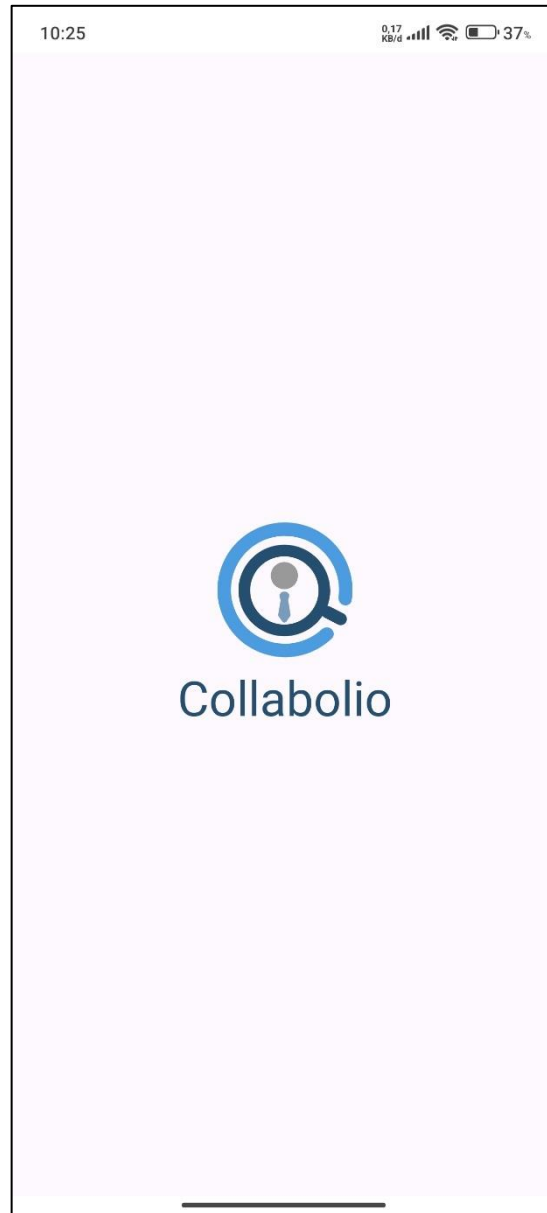
Gambar 4.4 merupakan hasil desain mockup aplikasi menggunakan software Figma. Pada gambar tersebut, sudah terlihat kemiripan dengan tampilan aplikasi yang asli dengan penambahan warna sementara dan penempatan posisi komponen aplikasi yang disempurnakan, seperti tombol yang lebih terstruktur. Hal

ini akan mempermudah dalam implementasi desain UI pada aplikasi. Selain itu, dengan desain *mockup* ini juga mengurangi kemungkinan kesalahan atau perbedaan antara desain awal dan hasil akhir yang akan dilihat oleh pengguna. Desain *mockup* yang akurat juga memungkinkan tim untuk lebih fokus pada pengembangan fitur dan fungsionalitas aplikasi tanpa terlalu banyak membuang waktu untuk menyesuaikan desain yang tidak sesuai.

4.4 Implementasi Desain ke Dalam Aplikasi

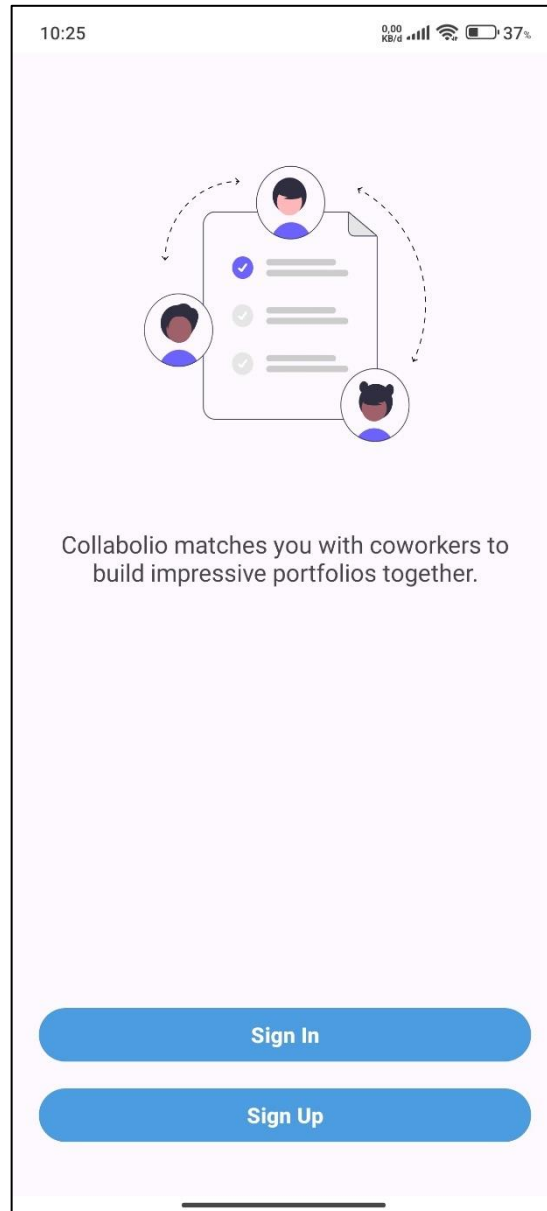
Langkah selanjutnya adalah mengimplementasikan desain *mockup* ke aplikasi Collabolio menggunakan Software Android Studio dalam bentuk file XML. XML sendiri merupakan bahasa markup yang digunakan untuk mendefinisikan struktur dan tata letak elemen-elemen antarmuka pengguna dalam pengembangan aplikasi Android. Dalam XML, UI didefinisikan dengan menggunakan elemen-elemen seperti *TextView*, *Button*, *ImageView*, *LinearLayout*, *RelativeLayout*, dan sebagainya. Pengaturan dan tata letak elemen-elemen UI juga ditentukan oleh atribut-atribut XML seperti *layout_width*, *layout_height*, *padding*, *margin*, dan atribut lainnya.

Implementasi tampilan aplikasi Collabolio terdiri dari tampilan *splash screen*, *start/welcome screen*, *login screen*, *register screen*, *input bio screen*, dan 3 tab menu utama yaitu *swipe menu*, *favorite menu*, dan *user profile menu*. Semua tampilan menggunakan *constraint layout* yang mana lebih fleksibel untuk digunakan. *Constraint layout* sendiri merupakan jenis tata letak (*layout*) yang digunakan dalam pengembangan aplikasi Android. Tata letak ini memungkinkan pengembang untuk menentukan posisi dan hubungan antara elemen-elemen *user interface* menggunakan batasan (*constraints*). Dalam *constraint layout*, elemen-elemen UI ditempatkan relatif terhadap elemen-elemen lainnya dan terhadap batasan-batasan yang ditetapkan, seperti batasan atas, bawah, kiri, dan kanan.



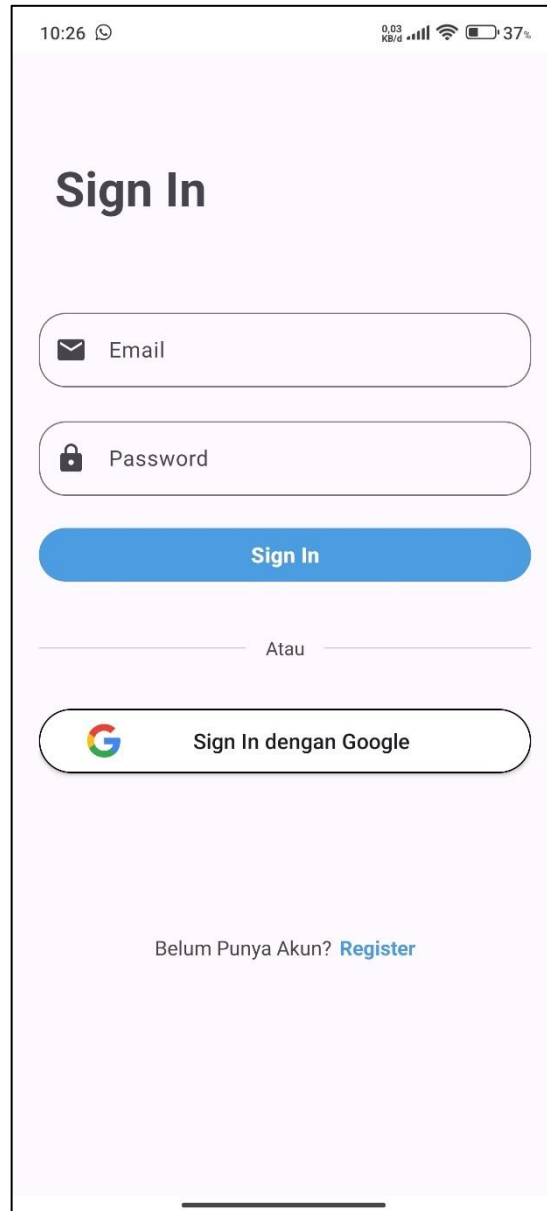
Gambar 4.5 Desain *Splash Screen Activity*

Gambar 4.5 merupakan tampilan *Splash screen activity* yang hanya berisi logo Collabolio. *Splash screen* berfungsi untuk memberikan pengalaman awal yang menarik kepada pengguna saat mereka membuka aplikasi. Selain itu juga, splash screen berfungsi untuk melakukan persiapan awal, seperti inisialisasi aplikasi, memuat data, atau melakukan tugas-tugas lain yang diperlukan sebelum aplikasi benar-benar dimuat. Hal tersebut akan memberikan pengalaman yang lebih responsif dan mengurangi kebingungan pengguna.



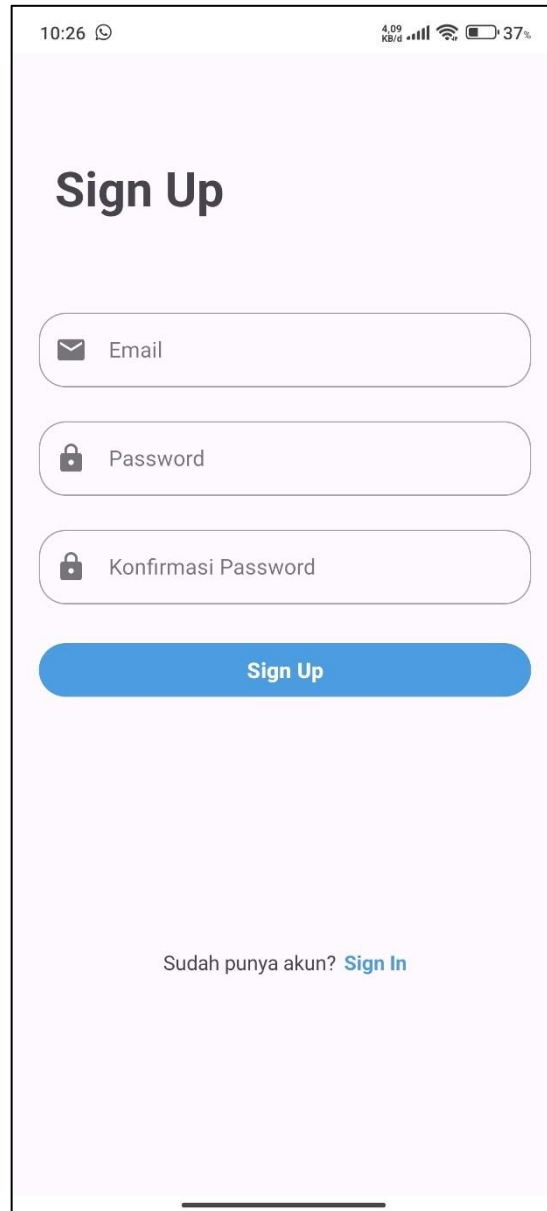
Gambar 4.6 Desain *Welcome Activity*

Gambar 4.6 Merupakan tampilan *Start/welcome activity* yang berfungsi sebagai tampilan pembuka dan sambutan kepada pengguna. Pada *start activity* ini di sematkan beberapa elemen seperti *text view* pembuka dan pengenalan aplikasi. Selain itu, terdapat juga gambar yang menarik dan sesuai dengan tema aplikasi. Pada *start screen* ini juga terdapat dua tombol yaitu tombol *sign in* dan *sign up*.

The image shows a mobile application login screen. At the top, the status bar displays the time 10:26, a clock icon, network speed 0.03 KB/d, signal strength, Wi-Fi, and a 37% battery level. The main heading is "Sign In" in a large, bold, black font. Below it are two input fields: "Email" with an envelope icon and "Password" with a lock icon. A blue "Sign In" button is positioned below the password field. A horizontal line with the word "Atau" (Or) in the center separates the manual login section from the Google login section. The Google login section features the Google "G" logo and the text "Sign In dengan Google". At the bottom, there is a link that says "Belum Punya Akun? Register".

Gambar 4.7 Desain *Login Activity*

Gambar 4.7 merupakan tampilan *login activity* Dimana pengguna diminta untuk memasukkan alamat email dan *password* agar bisa masuk ke dalam menu utama aplikasi. Elemen yang digunakan pada tampilan ini yaitu *edit text view* untuk input email dan *password*, dan juga dua tombol *sign in*. Tombol *sign in* pertama digunakan jika memasukkan alamat email dan kata sandi secara manual. Dan tombol *sign in* kedua digunakan jika ingin masuk secara otomatis dengan memasukkan akun google. Selain itu, terdapat *clickable text* yaitu “belum punya akun? Sign Up”.



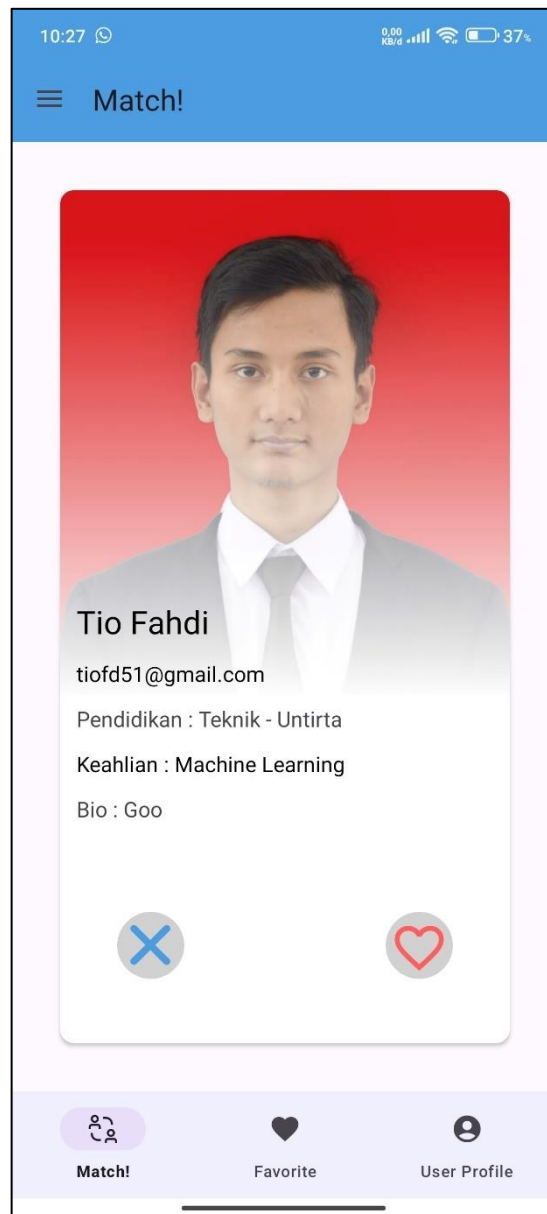
The image shows a mobile application interface for a 'Sign Up' screen. At the top, the status bar displays the time '10:26', a location icon, network speed '4.09 KB/d', signal strength, Wi-Fi, and battery level '37%'. The main heading 'Sign Up' is centered in a large, bold, black font. Below the heading are three input fields, each with a rounded rectangular border and a light purple background. The first field is labeled 'Email' with an envelope icon. The second field is labeled 'Password' with a lock icon. The third field is labeled 'Konfirmasi Password' with a lock icon. Below these fields is a prominent blue button with rounded corners labeled 'Sign Up' in white text. At the bottom of the screen, there is a link that says 'Sudah punya akun? [Sign In](#)' in a smaller, black font. The entire screen has a light purple background.

Gambar 4.8 Desain *Register Activity*

Gambar 4.8 merupakan tampilan *register activity* dimana pengguna diberi kesempatan untuk membuat akun baru dalam aplikasi. Pada tampilan ini, terdapat beberapa elemen yang sama seperti *login activity* namun dengan tambahan yaitu input untuk konfirmasi ulang *password*. Setelah pengguna berhasil membuat akun baru maka pengguna akan langsung diarahkan ke tampilan *InputBio Activity*.

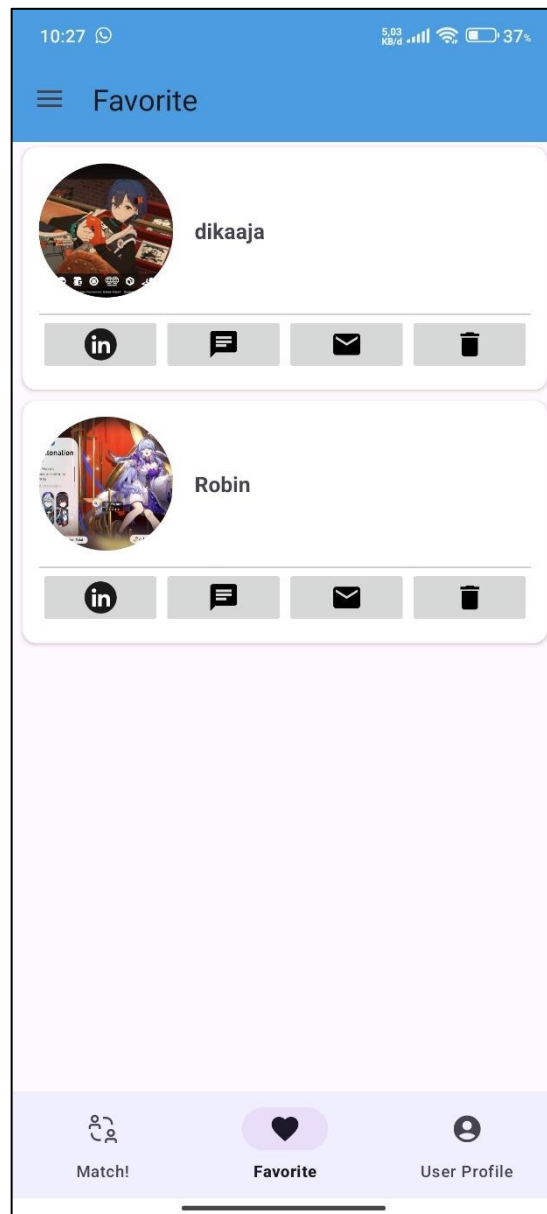
Gambar 4.9 Desain *InputBio Activity*

Gambar 4.9 merupakan tampilan *inputbio activity* yaitu tampilan untuk memasukkan biodata diri pengguna. Proses input pada tampilan ini menggunakan beberapa elemen input seperti *edit text view* untuk input nama, nomor telepon, alamat, pendidikan, dan deskripsi singkat. Selain itu terdapat juga elemen *radio button* untuk input jenis kelamin dan juga *dropdown menu* untuk pemilihan *skill* dan *interest*. Pengguna akan diminta untuk mengisi formulir secara lengkap dan sesuai dengan input yang tersedia.



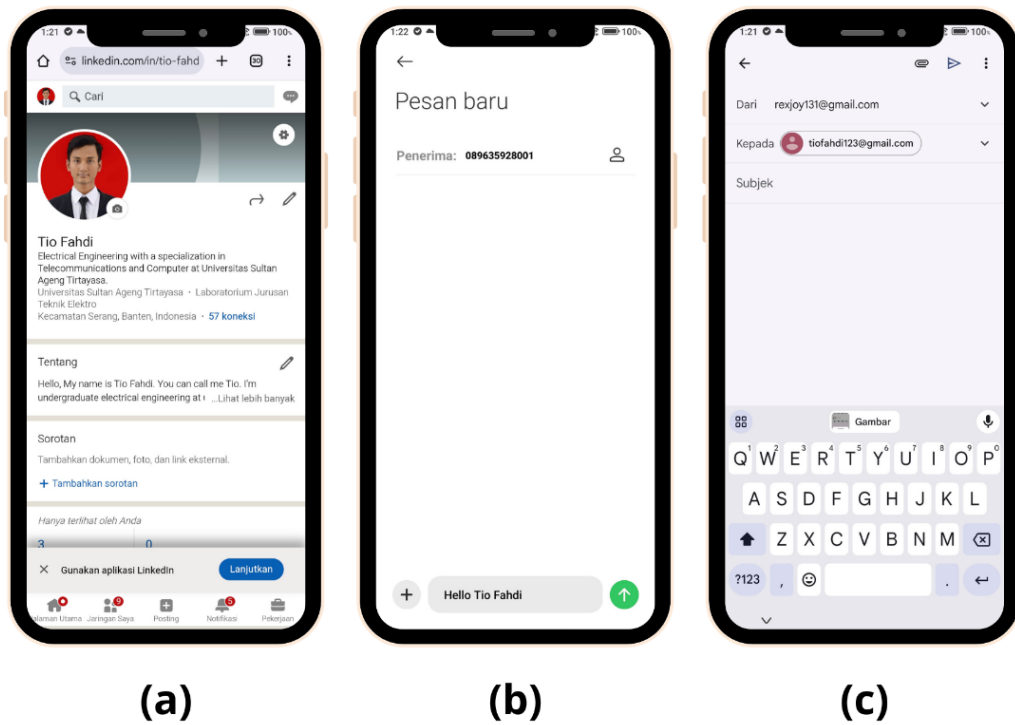
Gambar 4.10 Desain *Swipe Fragment*

Gambar 4.10 merupakan tampilan *tab swipe fragment* yang merupakan fitur utama aplikasi. Pada menu *swipe* ini mengimplementasikan elemen *recycler view* berupa *card stack view*. Setiap kartu mewakili profil pengguna dan memuat informasi yang relevan, seperti foto profil dan informasi singkat tentang pengguna tersebut. Pengguna dapat melihat profil pengguna lain dengan cara menggeser kartu ke kanan (jika tertarik) atau ke kiri (jika tidak tertarik). Hasil *swipe* kanan akan disimpan pada tab *favorite* sebagai pengguna yang disukai.



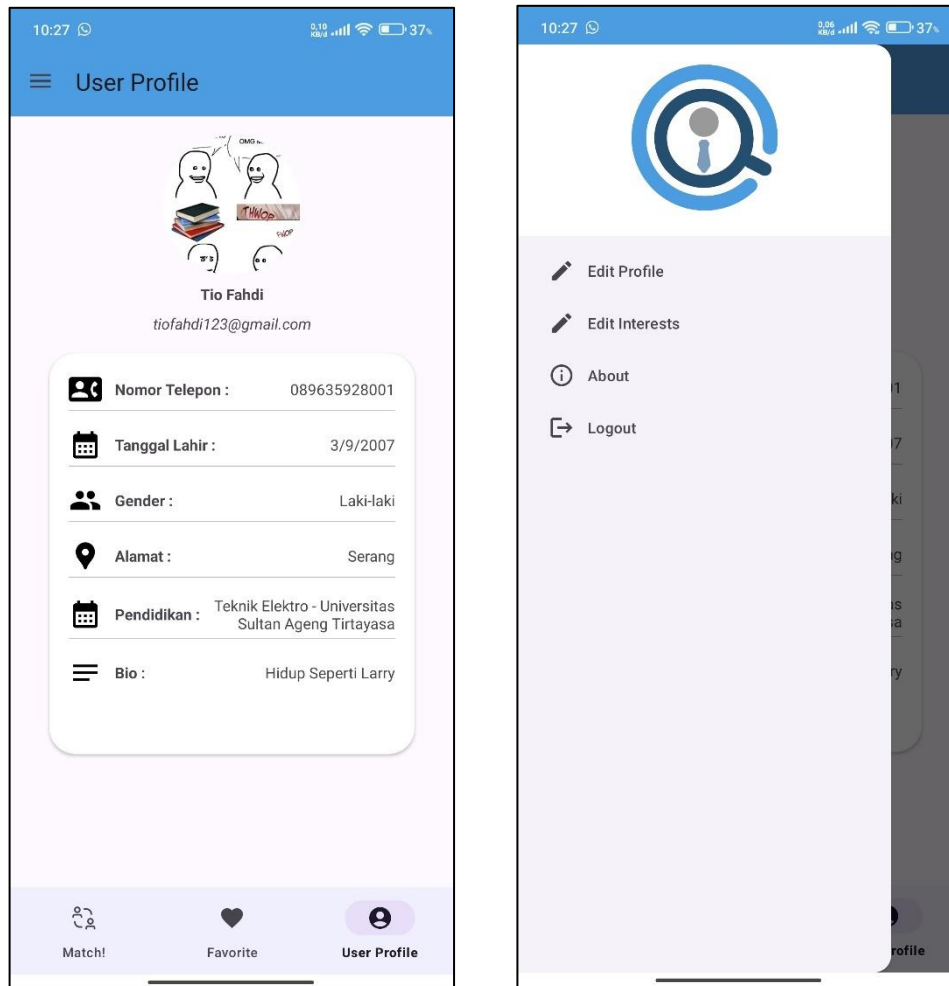
Gambar 4.11 Desain *Favorite Fragment*

Gambar 4.11 merupakan tab *favorite*, yang merupakan tampilan fragment untuk menyimpan hasil dari *swipe* kanan pada menu *swipe*. Elemen yang digunakan yaitu *Recycler view* untuk menampilkan daftar pengguna lain yang telah disukai. Di dalam *recycler view* ini, setiap item menampilkan informasi singkat tentang profil pengguna lain yang disukai, seperti foto profil, dan nama pengguna. Terdapat juga tombol-tombol aksi yang akan mengarah ke aplikasi lain seperti untuk membuka profile linked in pengguna yang disukai, mengirim pesan, mengirim email, dan terakhir yaitu menghapus pengguna dari daftar favorite.



Gambar 4.12 Tampilan ke Aplikasi Lain.

Pada gambar 4.12 bagian (a), ketika pengguna menekan tombol linkedin maka akan diarahkan ke web linkedin pengguna. Untuk gambar 4.12 bagian (b), ketika pengguna menekan tombol pesan maka pengguna akan diarahkan untuk mengirim pesan lewat whatsapp ataupun lewat aplikasi pesan seluler. Untuk gambar 4.12 bagian (c) ketika pengguna menekan tombol email maka pengguna akan diarahkan ke aplikasi gmail.



Gambar 4.13 Desain *Profile Fragment* dan *Navigation Drawer*

Gambar 4.12 merupakan tab user profile fragment yang menampilkan profil dari pengguna saat ini, yang memuat informasi seperti nama, nomor telepon, dan informasi lainnya. Kemudian pada navigation drawer, terdapat beberapa tombol yaitu tombol edit profil yang berfungsi untuk mengedit data profil, kemudian tombol edit interest, tombol about, dan terakhir tombol *logout* yang memungkinkan pengguna untuk keluar dari sesi saat ini. Fitur *logout* ini penting karena memungkinkan pengguna untuk mengakhiri akses mereka ke aplikasi dan memastikan keamanan data serta privasi mereka

4.5 Perancangan Program Aplikasi

Pada proses ini, dibutuhkan pengetahuan mendalam tentang komponen-komponen yang terdapat pada android studio. Hal tersebut disebabkan karena pada

android studio terdapat banyak sekali komponen *class* dan *function* yang bisa digunakan. Setiap bagian dari kode *class* dan *function* dapat disesuaikan dengan kebutuhan aplikasi yang sedang dikembangkan sehingga aplikasi bisa berjalan dengan lancar dan tidak membebani kinerja perangkat.

4.5.1 Program pada tampilan *Splash Screen*

```

override fun onStart() {
    super.onStart()

    val currentUser = auth.currentUser

    Handler(Looper.getMainLooper()).postDelayed({
        if (currentUser != null) {
            goToNextActivity()
        } else {
            val intent = Intent(this,
WelcomeActivity::class.java)
            startActivity(intent)
            finish()
        }
    }, 3000)
}

private fun goToNextActivity() {
    val user = auth.currentUser
    val firestore = FirebaseFirestore.getInstance()
    user?.let { currentUser ->
        val userId = currentUser.uid
        val userRef =
firestore.collection("users").document(userId)
        userRef.get()
            .addOnSuccessListener { documentSnapshot ->
                val phoneNumber =
documentSnapshot.getString("phoneNumber")
                if (phoneNumber.isNullOrEmpty()) {
                    val intent = Intent(this,
InputBioActivity::class.java)
                    startActivity(intent)
                    finish()
                } else {
                    val intent = Intent(this,
MainActivity::class.java)
                    startActivity(intent)
                    finish()
                }
            }
        finish()
    }
    .addOnFailureListener { exception ->
        Toast.makeText(this, "Error:
${exception.message}", Toast.LENGTH_SHORT).show()
    }
}
}

```

Pada kode program diatas, Ketika proses `onStart()` pada *Splash screen*, tampilan yang akan muncul setelahnya diatur dengan menggunakan *handler* dan pengecekan kondisi dengan percabangan *if else*. Jika pengguna baru menginstal aplikasi dan belum pernah *login* ke dalam aplikasi, maka akan diarahkan ke *welcome activity* yang kemudian pengguna akan diminta untuk melakukan *login* atau *register*. Selain itu, jika pengguna sudah pernah *login*, maka data *login* tersebut akan disimpan sehingga ketika pengguna masuk ke aplikasi akan langsung diarahkan ke menu utama yaitu menu *swipe*.

4.5.2 Autentikasi Login dan Register Menggunakan Firebase Auth

Dengan menggunakan Firebase Auth, pengguna dapat melakukan login ke aplikasi menggunakan alamat email dan *password*, atau melalui mekanisme autentikasi lainnya yang disediakan, seperti login menggunakan akun google atau akun facebook. Namun untuk implementasi ke aplikasi Collabolio ini hanya menggunakan dua cara saja yaitu login menggunakan email dan password atau login menggunakan akun Google. Untuk autentikasi menggunakan email dan *password* dapat dilihat pada program berikut.

```
private fun inputValidity(){
    binding.etLoginEmail.addTextChangedListener(object :
    TextWatcher {
        override fun afterTextChanged(s: Editable?) {
            checkInputValidity()
        }

        override fun beforeTextChanged(s: CharSequence?,
start: Int, count: Int, after: Int) {}

        override fun onTextChanged(s: CharSequence?, start:
Int, before: Int, count: Int) {}
    })

    binding.etLoginPassword.addTextChangedListener(object :
    TextWatcher {
        override fun afterTextChanged(s: Editable?) {
            checkInputValidity()
        }

        override fun beforeTextChanged(s: CharSequence?,
start: Int, count: Int, after: Int) {}

        override fun onTextChanged(s: CharSequence?, start:
Int, before: Int, count: Int) {}
    })
}
```

```

private fun checkInputValidity() {
    val email = binding.etLoginEmail.text.toString()
    val password = binding.etLoginPassword.text.toString()
    val emailIsValid =
Patterns.EMAIL_ADDRESS.matcher(email).matches()
    val passwordIsValid = password.length >= 8
    binding.btnLogin.isEnabled = emailIsValid &&
passwordIsValid
}

private fun signIn(email: String, password: String) {
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                goToNextActivity()
            } else {
                Toast.makeText(this,
getString(R.string.error_login), Toast.LENGTH_SHORT).show()
            }
        }
}

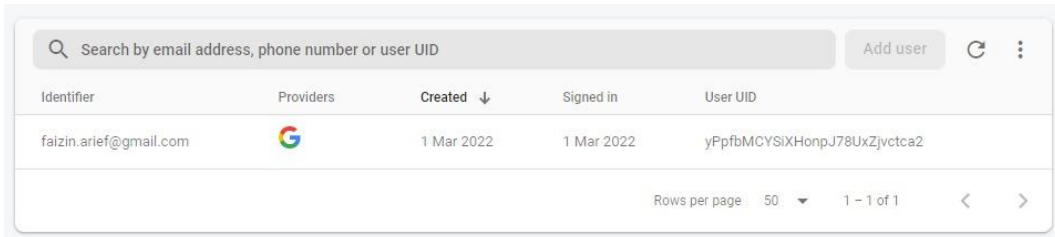
private fun actionClick(){
    binding.toRegister.setOnClickListener {
        val intent = Intent(this,
RegisterActivity::class.java)
        startActivity(intent)
    }
    binding.btnGoogle.setOnClickListener {
        signInGoogle()
    }
    binding.btnLogin.setOnClickListener {
        val email = binding.etLoginEmail.text.toString()
        val password =
binding.etLoginPassword.text.toString()
        signIn(email, password)
    }
}
}

```

Pengguna akan diminta untuk memasukkan alamat email dan password yang valid sesuai dengan input yang tersedia. Terdapat juga pengecekan input dari email dan *password* dengan syarat format email harus sesuai dan *password* harus berisi minimal 8 karakter. Jika syarat tidak terpenuhi maka tombol *sign in* tidak akan bisa ditekan.

Pada kode program juga, terdapat fungsi `signInGoogle()` dimana terdapat `ActivityResultLauncher` dengan parameter `signInIntent` untuk menampilkan pop up berisi pilihan akun Google. Pengguna akan diminta untuk memilih akun google yang tersedia pada perangkat. Apabila sudah dipilih, pengguna akan mendapatkan `idToken` dari `GoogleSignInAccount`. Untuk pengguna yang login menggunakan akun google ini, tidak perlu melakukan register

karena akunnya akan otomatis terdaftar ke firebase *auth*. Hasil dari *login* pengguna akan tercatat pada firebase *auth* dan dapat dilihat pada gambar berikut.



Identifier	Providers	Created ↓	Signed in	User UID
faizin.arief@gmail.com		1 Mar 2022	1 Mar 2022	yPfbMCYSiXHonpJ78UxZjvctca2

Gambar 4.14 Hasil Autentikasi Pada Firebase

Pada login activity juga terdapat proses pengecekan pada program yang dimana ketika biodata pengguna sudah tersedia pada *cloud firestore*, maka pengguna akan langsung diarahkan ke tampilan menu utama. Sedangkan ketika biodata pengguna belum tersedia, maka pengguna akan diarahkan ke *inputbio activity*.

Register activity digunakan jika pengguna ingin mendaftarkan akun baru ke aplikasi dengan menggunakan email dan *password*. untuk implementasinya mirip seperti *login activity* yaitu terdapat pengecekan untuk input dari email dan *password* dengan *tambahan input* berupa konfirmasi ulang *password*. Setelah pengguna menekan tombol *sign up* maka akan muncul *popup* pemberitahuan “Registrasi Berhasil” dan pengguna akan langsung diarahkan ke *inputbio activity*.

4.5.3 Mengirim Dan Menyimpan Data Pengguna ke Firestore

Proses mengirim dan menyimpan data pengguna terjadi pada *inputbio activity* dengan menggunakan firebase *firestore cloud*. Berikut adalah kode programnya.

```
private fun startGallery() {

    launcherGallery.launch(PickVisualMediaRequest(ActivityResultContracts.PickVisualMedia.ImageOnly))
}

private val launcherGallery = registerForActivityResult(
    ActivityResultContracts.PickVisualMedia()
) { uri: Uri? ->
    if (uri != null) {
        currentImageUri = uri
        showImage()
    } else {
        Log.d("Photo Picker", "No media selected")
    }
}
```

```

    }
}

private fun showImage() {
    currentImageUri?.let {
        Log.d("Image URI", "showImage: $it")
        binding.ivProfile.setImageURI(it)
    }
}

private fun saveProfile() {
    progressBar.visibility = View.VISIBLE
    val email = firebaseAuth.currentUser?.email ?: ""
    val uid = firebaseAuth.currentUser?.uid ?: ""
    val displayName = binding.inputName.text.toString()
    val phoneNumber = binding.inputNumber.text.toString()
    val birthDate = binding.inputDate.text.toString()
    val male = binding.radioButtonMale.isChecked
    val location = binding.inputAddress.text.toString()
    val degree = binding.inputDegree.text.toString()
    val school = binding.inputSchool.text.toString()
    val bio = binding.inputBio.text.toString()
    var userSkills = binding.mactvSkills.text.toString()
    userSkills = userSkills.trimEnd(',', ' ')
    val arraySkills = userSkills.split(",").map{it.trim()}
    val education = Education(degree, school)
    val linkedinURL = binding.inputLinkedin.text.toString()

    val skillsList = mutableListOf<Skills>()
    for (skillName in arraySkills) {
        val skill = Skills(skillName)
        skillsList.add(skill)
    }

    val connectionList = mutableListOf<Connections>()

    val currentUser = firebaseAuth.currentUser
    currentUser?.let {
        val userId = it.uid
        val userRef = firestore.collection("users").document(userId)
        when {
            displayName.isEmpty() -> {
                progressBar.visibility = View.GONE
                binding.inputName.error = "Masukkan Nama"
                Toast.makeText(this, "Masukkan Nama",
                    Toast.LENGTH_SHORT).show()
            }
            phoneNumber.isEmpty() -> {
                progressBar.visibility = View.GONE
                binding.inputNumber.error = "Masukkan Nomor
                Telepon"
                Toast.makeText(this, "Masukkan Nomor
                Telepon", Toast.LENGTH_SHORT).show()
            }
            birthDate.isEmpty() -> {
                progressBar.visibility = View.GONE

```

```

binding.inputNumber.error = "Masukkan
Tanggal Lahir"
        Toast.makeText(this, "Masukkan Tanggal
Lahir", Toast.LENGTH_SHORT).show()
    }

    userSkills.isEmpty() -> {
        progressBar.visibility = View.GONE
        binding.mactvSkills.error = "Masukkan
Keahlian"
        Toast.makeText(this, "Masukkan Keahlian",
Toast.LENGTH_SHORT).show()
    }

    location.isEmpty() -> {
        progressBar.visibility = View.GONE
        binding.inputAddress.error = "Masukkan
Alamat"
        Toast.makeText(this, "Masukkan Alamat",
Toast.LENGTH_SHORT).show()
    }
    degree.isEmpty() -> {
        progressBar.visibility = View.GONE
        binding.inputDegree.error = "Masukkan Gelar
dan Jurusan"
        Toast.makeText(this, "Masukkan Gelar dan
Jurusan", Toast.LENGTH_SHORT).show()
    }
    school.isEmpty() -> {
        progressBar.visibility = View.GONE
        binding.inputAddress.error = "Masukkan
Sekolah atau Universitas"
        Toast.makeText(this, "Masukkan Sekolah atau
Universitas", Toast.LENGTH_SHORT).show()
    }
    bio.isEmpty() -> {
        progressBar.visibility = View.GONE
        binding.inputAddress.error = "Masukkan Bio"
        Toast.makeText(this, "Masukkan Bio",
Toast.LENGTH_SHORT).show()
    }
    linkedinURL.isNotEmpty() &&
!linkedinURL.startsWith("http") -> {
        progressBar.visibility = View.GONE
        binding.inputLinkedin.error = "Masukkan link
URL dengan sesuai"
        Toast.makeText(this, "Link tidak valid",
Toast.LENGTH_SHORT).show()
    }
    currentImageUri == null -> {
        progressBar.visibility = View.GONE
        Toast.makeText(this, "Masukkan Foto Profil
Anda", Toast.LENGTH_SHORT).show()
    }
    else -> {
        val storageRef =
storage.getReference("users").child(userId).child("profile.jpg")
        storageRef.putFile(currentImageUri!!).addOnSuccessListener {

```

```

storageRef.downloadUrl.addOnSuccessListener { documents ->
    val users =
    UserProfileResponse(email, uid, displayName, phoneNumber,
    birthDate, education, male, bio, location, photoURL =
    documents.toString(), skillsList, connectionList, linkedinURL =
    linkedinURL)
    userRef.set(users,
    SetOptions.merge())
    .addOnSuccessListener {
        progressBar.visibility =
        View.GONE
        Toast.makeText(this, "Profil
        berhasil diperbarui", Toast.LENGTH_SHORT)
        .show()
        goToNextActivity()
    }
    .addOnFailureListener {
        Toast.makeText(this, "Profil
        gagal diperbarui", Toast.LENGTH_SHORT)
        .show()
    }
    }.addOnFailureListener {
        Toast.makeText(this, "Terjadi
        Masalah", Toast.LENGTH_SHORT).show()
    }
    }.addOnFailureListener{
        Toast.makeText(this, "Terjadi Masalah",
        Toast.LENGTH_SHORT).show()
    }
    }
    }
    }
    private fun fetchSkills() {
        val skillsCollection = firestore.collection("Skills")

        skillsCollection.get()
        .addOnSuccessListener { querySnapshot ->
            val skillsList = mutableListOf<String>()

            for (document in querySnapshot.documents) {
                val skill = document.getString("name")
                skill?.let { skillsList.add(it) }
            }

            val adapter = ArrayAdapter(this,
            R.layout.simple_dropdown_item_1line, skillsList)
            binding.mactvSkills.setAdapter(adapter)

            binding.mactvSkills.setTokenizer(MultiAutoCompleteTextView.Comma
            Tokenizer())
        }
        .addOnFailureListener {
        }
    }

    private fun goToNextActivity() {

```

```

        val user = firebaseAuth.currentUser
        val firestore = FirebaseFirestore.getInstance()
        user?.let { currentUser ->
            val userId = currentUser.uid
            val userRef = firestore.collection("users").document(userId)
            userRef.get().addOnSuccessListener { documentSnapshot ->
                val interests = documentSnapshot.getString("interests")
                if (interests.isNullOrEmpty()) {
                    val intent = Intent(this, InterestActivity::class.java)
                    startActivity(intent)
                    finish()
                } else {
                    val intent = Intent(this, MainActivity::class.java)
                    startActivity(intent)
                    finish()
                }
            }.addOnFailureListener { exception ->
                Toast.makeText(this, "Error: ${exception.message}", Toast.LENGTH_SHORT).show()
            }
        }

    private fun showDatePicker(editText: EditText) {
        val year = calendar.get(Calendar.YEAR)
        val month = calendar.get(Calendar.MONTH)
        val day = calendar.get(Calendar.DAY_OF_MONTH)

        datePickerDialog = DatePickerDialog(this, {_, year, month, day ->
            val selectedDate = "$day/${month+1}/$year"
            editText.setText(selectedDate)
        }, year, month, day)
        datePickerDialog.show()
    }
}

```

Pada listing kode, pengguna akan diminta untuk memasukkan data diri pengguna sesuai dengan *input* yang tersedia seperti nama lengkap, nomor telepon, jenis kelamin, Pendidikan, keahlian (*skill*), dan minat keahlian rekan yang dicari (*interest*). Pada *inputbio activity* terdapat pengecekan agar tidak ada kolom *input* yang kosong atau *null* data. Ketika pengguna selesai mengisi data kemudian menekan tombol “lanjut” maka data pengguna akan dikirim dan disimpan pada firebase firestore *cloud* dan pengguna akan diarahkan ke tampilan menu utama yaitu menu *swipe*.

4.5.4 Implementasi *machine learning* pada aplikasi

Implementasi *machine learning* pada aplikasi Collabolio adalah langkah penting karena hal tersebut merupakan fitur utama dalam aplikasi. Dengan memanfaatkan teknologi *machine learning*, aplikasi dapat memberikan rekomendasi rekan kerja yang sesuai dengan keterampilan dan minat pengguna. Untuk program algoritma *machine learning* sendiri sudah disiapkan oleh tim pengembang *machine learning* dan disimpan ke dalam sebuah API. Proses implementasi pada aplikasi dilakukan dengan cara melakukan panggilan API *machine learning* tersebut untuk melakukan *training* model dan hasilnya akan di tampilkan pada aplikasi. Berikut adalah kode program pemanggilan API *machine learning* dengan menggunakan okhttp.

```
class ApiRequest : OkHttpClient() {

    private val firebaseAuth = FirebaseAuth.getInstance()

    fun getUsers(): Array<UserApi>? {
        val uid = firebaseAuth.currentUser?.uid

        if (uid == null) {
            return arrayOf()
        }

        val request = Request.Builder()
            .url("https://model-endpoint-onhqnm5xvq-uc.a.run.app/api/users/$uid/30")
            .build()

        val response = newCall(request).execute()

        val user = response.body?.string()?.let {
            Gson().fromJson(it, Array<UserApi>::class.java)
        }

        if (user != null) {
            return user
        } else {
            return arrayOf()
        }
    }
}
```

Pada listing kode, okhttp akan melakukan panggilan api dengan endpoint yang sudah ditentukan. Setelah itu, akan terjadi training model pada cloud yang menghasilkan sebuah *list* rekomendasi untuk pengguna. List tersebut akan

ditangkap oleh okhttp dan kemudian list data tersebut disimpan dan diolah sesuai dengan tampilan yang dibuat.

4.5.5 Menampilkan Hasil Rekomendasi

Setelah mendapatkan hasil *training model* pada *machine learning* berupa *list* UID (User ID), langkah berikutnya adalah menampilkan daftar tersebut pada swipe menu. Swipe menu ini menggunakan RecyclerView khusus yang telah disesuaikan dengan bentuk *CardStackView*. Pada tampilan *CardStackView* ini, pengguna dapat melakukan swipe ke kiri (tidak minat) atau swipe kanan (minat). Berikut untuk program pada swipe menu.

```
private suspend fun apiReqForUids(): MutableList<String> {
    return withContext(Dispatchers.IO) {
        val apiRequest = ApiRequest()
        val uidsResult = mutableListOf<String>()
        val users = apiRequest.getUsers()
        for (user in users!!) {
            uidsResult.add(user.uid.toString())
        }
        uidsResult
    }
}

private fun fillingTheItemsByApi() {
    CoroutineScope(Dispatchers.Main).launch {
        val uids = apiReqForUids()
        val cardsQuery =
            db.collection("users").whereIn("uid", uids)
        try {
            val documents = withContext(Dispatchers.IO) {
                cardsQuery.get().await()
            }
            for (document in documents) {
                val user =
                    document.toObject(UserSwipe::class.java)
                if (!rowItems.contains(user)) {
                    rowItems.add(user)
                    cardsAdapter?.notifyDataSetChanged()
                }
            }
        } catch (exception: Exception) {
            Log.d("KARTU", "get failed with : ", exception)
        }
    }
}
```

Pertama diambil list rekomendasi pengguna berupa UID menggunakan fungsi `apiReqForUids()`, kemudian hasil list UID tersebut di masukkan ke dalam fungsi `fillingTheItemsByApi()` untuk proses pencarian list UID

tersebut pada database di firestore. Setelah didapatkan list pengguna berdasarkan UID tersebut pada firestore, maka *list* data pengguna akan diambil dan kemudian ditampilkan pada tampilan swipe menu.

Setelah pengguna melakukan swipe ke kanan pada swipe menu, rekan kerja yang dipilih akan disimpan ke dalam *list favourite* menu. Implementasi penyimpanan ini juga menggunakan firestore. Setiap kali pengguna melakukan swipe ke kanan pada kartu rekomendasi, data rekan kerja tersebut akan disimpan ke list connection pengguna. Informasi yang disimpan berupa data UID. Untuk menampilkannya pada favorite menu, list uid dari connection tersebut diambil untuk dijadikan parameter filter pada firestore sehingga didapatkan profile pengguna yang di favoritkan. Berikut adalah kode program pada menu favorite.

```
fun fetchConnections() {
    if (currentUserId != null) {
        firestore.collection("users")
            .document(currentUserId)
            .get()
            .addOnSuccessListener { document ->
                val userConnections =
document.toObject(UserProfileResponse::class.java)?.connections
?: listOf()

                if (userConnections.isNotEmpty()) {
                    fetchConnectionProfiles(userConnections)
                } else {
                    _connections.value = listOf()
                }
            }
            .addOnFailureListener { exception ->
                Log.d("FavoriteViewModel", "Error getting
user document: ", exception)
            }
    } else {
        Log.d("FavoriteViewModel", "Current user UID is
null")
    }
}

private fun fetchConnectionProfiles(connectionUids:
List<Connections>) {
    val profilesList = mutableListOf<UserProfileResponse>()
    firestore.collection("users")
        .whereIn("uid", connectionUids.map { it.uid })
        .get()
        .addOnSuccessListener { result ->
            for (document in result) {
                val profile =
document.toObject(UserProfileResponse::class.java)
                profilesList.add(profile)
            }
            _connections.value = profilesList
        }
}
```

```

        .addOnFailureListener { exception ->
            Log.d("FavoriteViewModel", "Error getting
connection profiles: ", exception)
        }
    }

    fun removeConnection(userProfile: UserProfileResponse) {
        if (currentUserId != null) {
            firestore.collection("users")
                .document(currentUserId)
                .update("connections",
FieldValue.arrayRemove(Connections(userProfile.uid)))
                .addOnSuccessListener {
                    fetchConnections()
                }
                .addOnFailureListener { exception ->
                    Log.d("FavoriteViewModel", "Error removing
connection: ", exception)
                }
        }
    }
}

```

Pengguna dapat melihat daftar rekan favorite dalam bentuk list *RecyclerView*. Selain itu, pengguna dapat melakukan tindakan seperti menghapus rekan kerja dari daftar favorit, mengirim pesan, ataupun mengirim email ke pengguna yang dipilih.

4.5.6 Menampilkan dan Mengedit User Profile

Pada *user profile* menu, aplikasi akan menampilkan informasi yang telah mereka berikan saat mendaftar, seperti nama, nomor telepon, alamat email dan lain-lain. Profile menu juga mengimplementasikan database firestore yang berfungsi untuk mengambil data pengguna dari database firestore. Berikut adalah kode program pada *user profile* menu.

```

private fun displayUserProfile(userProfile:
UserProfileResponse?) {
    if (userProfile != null) {
        binding.tvEmail.text = userProfile.email
        binding.tvName.text = userProfile.displayName
        binding.tvPhoneNumber.text = userProfile.phoneNumber
        binding.tvBirthDate.text = userProfile.birthDate
        binding.tvLocation.text = userProfile.location
        binding.tvBio.text = userProfile.bio
        binding.tvEducation.text =
"${userProfile.educations.degree} -
${userProfile.educations.school}"
        val genderText = if (userProfile.male) {
            "Laki-laki"
        } else {
            "Perempuan"
        }
    }
}

```

```

    }
    binding.tvIsMale.text = genderText
    Glide.with(requireContext())
        .load(userProfile.photoURL)
        .into(binding.userImageView)
    }
}

```

Pada listing code, pertama dilakukan pengambilan data pengguna dari firestore, kemudian dilakukan proses pengecekan data null pada `userProfile` agar terhindar dari error *null pointer exception* yang menyebabkan aplikasi *crash* atau *force close*. Setelah itu, data akan dapat ditampilkan pada tampilan *user profile* menu. Selain menampilkan informasi profil, pengguna juga diberikan opsi untuk mengedit profil. Pada tombol edit profil, diberikan perintah yang akan melakukan Intent ke *inputbio activity*. Pengguna dapat mengubah atau menambahkan informasi baru sesuai kebutuhan. Dan terakhir terdapat opsi *logout* yang berfungsi untuk keluar dari sesi saat ini dan kembali ke tampilan login activity. Opsi *logout* berguna untuk keamanan agar data pengguna terjaga ketika pengguna sedang tidak menggunakan aplikasi.

4.5.7 Navigation Drawer

Untuk navigation drawer sendiri merupakan menu tambahan yang membantu dalam melakukan navigasi ke tampilan lain. Berikut adalah kodenya.

```

navViewDrawer.setNavigationItemSelectedListener { menuItem ->
    when (menuItem.itemId) {
        R.id.nav_edit_profile -> {
            val intent = Intent(this,
InputBioActivity::class.java)
            startActivity(intent)
            true
        }
        R.id.nav_edit_interests -> {
            val intent = Intent(this,
InterestActivity::class.java)
            startActivity(intent)
            true
        }
        R.id.nav_about -> {
            val intent = Intent(this,
AboutActivity::class.java)
            startActivity(intent)
            true
        }
        R.id.nav_logout -> {
            signOut()
        }
    }
}

```

```

        true
    }
    else -> false
}

```

Pada menu navigasi terdapat beberapa pilihan untuk pindah ke tampilan lain dengan menggunakan `intent` seperti edit profile, edit interest, tampilan about, dan melakukan logout.

4.6 Pengujian Aplikasi

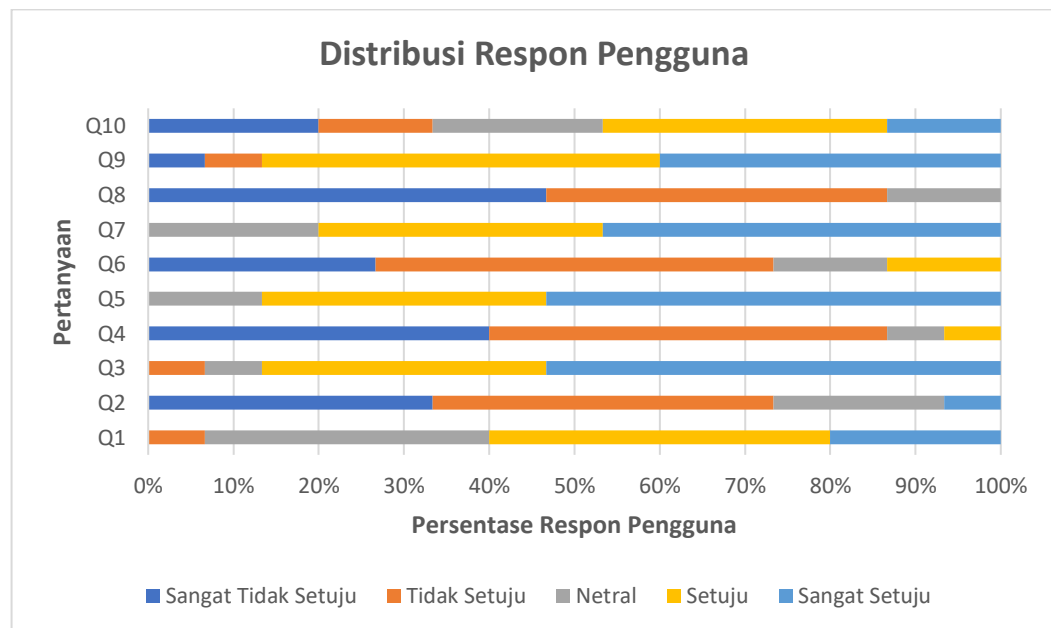
Tahap terakhir dalam proses perancangan aplikasi adalah pengujian aplikasi. Pengujian dilakukan dengan metode *System Usability Scale* (SUS). Pengujian *usability* berfungsi untuk mengetahui tingkat kelayakan aplikasi yang dinilai dari segi efektivitas, efisiensi, dan kepuasan pengguna.

Pengujian melibatkan 15 responden mahasiswa yang berasal dari berbagai bidang keahlian. Responden terbagi atas 13 orang laki-laki (86,7%) dan 2 orang Perempuan (13,3%). Hasil Penilaian Responden tersebut dapat dilihat pada tabel berikut.

Tabel 4.2 Hasil Penilaian Responden

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
R1	5	1	5	1	5	1	5	1	5	1
R2	4	2	5	1	4	3	4	2	4	3
R3	3	1	5	3	5	3	4	2	4	5
R4	4	2	5	2	5	1	5	1	5	2
R5	4	1	5	2	4	2	5	1	4	2
R6	5	3	4	2	5	2	3	3	5	3
R7	5	1	5	1	5	1	5	1	1	1
R8	4	3	3	2	4	2	3	2	4	5
R9	3	2	4	2	4	4	4	3	2	4
R10	2	2	4	2	3	4	4	2	4	3
R11	3	1	5	1	4	2	5	1	5	4
R12	4	2	4	4	5	2	5	2	4	4
R13	3	3	4	1	3	2	3	1	4	4
R14	4	5	5	1	5	1	5	1	5	1
R15	3	2	2	2	5	2	4	2	5	4
Keterangan: R1-R15 = Jumlah Responden Q1-Q10 = Jumlah Pertanyaan										

Berdasarkan data tabel 4.3 diatas, dapat dibuat sebuah grafik data distribusi respon pengguna pada kuisioner SUS. Untuk grafik distribusi respon pengguna dapat dilihat pada gambar berikut.



Gambar 4.15 Distribusi Respon Pengguna

Setelah didapatkan data penilaian responden terhadap aplikasi, langkah selanjutnya adalah proses perhitungan data skor SUS. Hasil perhitungan data skor SUS dapat dilihat pada tabel berikut.

Tabel 4.3 Hasil Rata-rata Skor SUS

Responden	Jumlah	Skor SUS
R1	40	100
R2	30	75
R3	27	67.5
R4	36	90
R5	34	85
R6	29	72.5
R7	36	90
R8	24	60
R9	22	55
R10	24	60
R11	33	82.5
R12	28	70
R13	26	65
R14	35	87.5
R15	27	67.5

Rata-rata	75.16666667
Keterangan: R1-R15 = Jumlah Responden	

Berdasarkan perhitungan tabel diatas, didapatkan nilai rata-rata skor SUS sebesar 75.16666667. Jika dilakukan interpretasi skor SUS berdasarkan skala pada gambar 3.4, maka dapat diambil kesimpulan bahwa aplikasi termasuk ke dalam kategori *Good* (bagus) dalam skala *adjective* (peringkat sifat), sudah memenuhi kriteria *acceptable* dalam skala tingkat penerimaan, namun mendapat tingkat loyalitas pengguna passive dalam skala NPS (*Net Promotore Score*) yang berarti pengguna tidak dalam kondisi menolak maupun menyukai aplikasi Collabolio ini.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan sebelumnya didapatkan kesimpulan yang diperoleh sebagai berikut.

1. Penelitian ini berhasil merancang dan mengembangkan aplikasi Collabolio dengan user experience aplikasi yang cukup baik dan dapat dengan mudah digunakan oleh pengguna.
2. Implementasi fitur, elemen, dan komponen yang sesuai dengan aplikasi yang akan dibuat berfungsi untuk memastikan bahwa aplikasi Collabolio memenuhi kebutuhan dan tujuan pengguna. Implementasi fitur pada aplikasi Collabolio diambil berdasarkan dari referensi aplikasi *dating apps* seperti aplikasi Tinder atau Bumble. Fitur-fitur tersebut diantaranya yaitu fitur verifikasi profil, fitur swipe, fitur profil pengguna terperinci, dan fitur *favorite*.
3. Hasil pengujian keandalan aplikasi menggunakan metode *System Usability Test* (SUS) didapatkan hasil aplikasi termasuk ke dalam kategori *Good* (bagus) dalam skala *adjective* (peringkat sifat), sudah memenuhi kriteria *acceptable* dalam skala tingkat penerimaan, namun mendapat tingkat loyalitas pengguna *passive* dalam skala NPS (*Net Promotore Score*) yang berarti pengguna tidak dalam kondisi menolak maupun menyukai aplikasi.

5.2 Saran

Adapun saran yang dapat diberikan dari hasil penelitian yang sudah dilakukan antara lain sebagai berikut.

1. Meningkatkan aksesibilitas dan pengembangan aplikasi lebih lanjut dengan menambahkan beberapa fitur yang membantu seperti fitur *in app message*, pengaturan privasi, dan lain-lain.
2. Membuat program aplikasi dengan *design pattern* yang baik sehingga meningkatkan efisiensi aplikasi dan mempermudah pengembang dalam membaca kode program.

3. Menambahkan parameter untuk tersambung ke sosial media pengguna seperti Facebook, Instagram, dan lain-lain.

DAFTAR PUSTAKA

- [1] M. Rajab Fachrizal dan G. W. Sukmo Aji, “Rancang Bangun Aplikasi Pencarian Kerja Berbasis Android Di Unikom Bandung,” *Jurnal Manajemen Informatika (JAMIKA)*, 2017, Diakses: 24 Oktober 2023. [Daring]. Tersedia pada:
<https://ojs.unikom.ac.id/index.php/jamika/article/view/483>
- [2] D. Silvi Purnia, A. Rifai, dan S. Rahmatullah, “Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android,” *Seminar Nasional Sains dan Teknologi 2019*, vol. 16, 2019, Diakses: 24 Oktober 2023. [Daring]. Tersedia pada:
<https://jurnal.umj.ac.id/index.php/semnastek/article/view/5238/3516>
- [3] Y. Kurniasari, B. Suseta, N. Hendiyani, dan A. M. Abadi, “Classification of Open Unemployment Rate in Indonesia with Mamdani Fuzzy Inference System,” dalam *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul 2020. doi: 10.1088/1742-6596/1581/1/012010.
- [4] Z. Hasan dan H. Sasana, “Determinants Of Youth Unemployment Rate In Asean,” *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, vol. 9, no. 03, 2020, [Daring]. Tersedia pada:
www.ijstr.org
- [5] N. A. Rahman dan N. Z. Riani, “Analisis Pengangguran Terbuka di Indonesia,” 2023. [Daring]. Tersedia pada:
<http://ejournal.unp.ac.id/students/index.php/epb/index>
- [6] A. M. Setyanti dan F. Finuliyah, “Pengangguran Terdidik Pada Masa Pandemi Covid-19: Analisis Pada Data Sakernas 2020,” *Jurnal Ketenagakerjaan*, vol. 17, no. 1, Jun 2022, doi: 10.47198/naker.v17i1.118.
- [7] Sherty Veronika dan A. Y. Mafruhah, “Pengaruh Pertumbuhan Ekonomi, Investasi dan Inflasi terhadap Pengangguran Terdidik di Provinsi Jawa Barat,” *Jurnal Riset Ilmu Ekonomi dan Bisnis*, hlm. 139–146, Des 2022, doi: 10.29313/jrieb.vi.1290.

- [8] Y. Safitri dan R. Syofyan, “Pengaruh Pengalaman Magang dan Future Time Perspective terhadap Kesiapan Kerja Mahasiswa Fakultas Ekonomi Universitas Negeri Padang,” 2023.
- [9] L. Auliya dan L. Agusalim, “Pengaruh COVID-19 dan Upah terhadap Pengangguran Terdidik di Indonesia: Analisis Data Panel 33 Provinsi Tahun 2010-2021,” *Jurnal Ketenagakerjaan*, vol. 17, no. 3, hlm. 285–303, Des 2022, doi: 10.47198/naker.v17i3.148.
- [10] O. Rizky, M. Fajar, O. R. Prasetyo, dan S. Nonalisa, “Forecasting unemployment rate in the time of COVID-19 pandemic using Google trends data (case of Indonesia),” *Munich Personal RePEc Archive*, 2020, Diakses: 19 November 2023. [Daring]. Tersedia pada: <https://mpra.ub.uni-muenchen.de/105042/>
- [11] A. M. Rif’a, E. Mi’rajatul Ummah, dan M. Rosyid, “Upaya Menurunkan Kecemasan Menjadi Pengangguran Melalui Strategi Konseling,” 2023, [Daring]. Tersedia pada: <https://jurnal.iaihnwpancor.ac.id/index.php/tauji>
- [12] K. Ngang TANG, “Beyond Employability: Embedding Soft Skills in Higher Education,” *TOJET: The Turkish Online Journal of Educational Technology*, vol. 18, no. 2, 2019, Diakses: 19 November 2023. [Daring]. Tersedia pada: <https://eric.ed.gov/?id=EJ1211098>
- [13] V. Vukmirović, I. Domazet, dan D. Pavlović, “Development of 21st Century Skills As A Response to Youth Unemployment,” *26th International Scientific Conference Strategic Management and Decision Support Systems in Strategic Management*, 2021.
- [14] D. Sufajar dan A. Qosyim, “Analisis Keterampilan Kolaborasi Siswa SMP Pada Pembelajaran IPA di Masa Pandemi Covid-19,” vol. 10, no. 2, hlm. 253–259, 2022, [Daring]. Tersedia pada: <https://ejournal.unesa.ac.id/index.php/pensa>
- [15] X. Li dan D. Li, “An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy,” *Mobile Information Systems*, vol. 2019, 2019, doi: 10.1155/2019/3560968.
- [16] K. Hidayatulloh, M. Komarudin MZ, dan A. Sutanti, “Perancangan Aplikasi Pengolahan Data Dana Sehat Pada Rumah Sakit Umum

- Muhammadiyah Metro,” *Jurnal Mahasiswa Ilmu Komputer (JMIK)*, vol. 1, no. 1, hlm. 18, Apr 2020, Diakses: 7 Januari 2025. [Daring]. Tersedia pada: <https://scholar.ummetro.ac.id/index.php/IlmuKomputer/article/download/122/73/>
- [17] Azis N, Pribadi G, dan Savitrie Nurchahya M, “Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android,” *Jurnal IKRA-ITH Informatika*, vol. 4, 2020, Diakses: 24 Oktober 2023. [Daring]. Tersedia pada: <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/article/view/815/618>
- [18] F. Fariyanto dan F. Ulum, “Perancangan Aplikasi Pemilihan Kepala Desa Dengan Metode Ux Design Thinking (Studi Kasus: Kampung Kuripan),” *Jurnal Teknologi dan Sistem Informasi (JTISI)*, vol. 2, no. 2, hlm. 52–60, 2021, [Daring]. Tersedia pada: <http://jim.teknokrat.ac.id/index.php/JTISI>
- [19] IBM, “What is mobile application development?,” International Business Machines. Diakses: 24 Oktober 2023. [Daring]. Tersedia pada: <https://www.ibm.com/topics/mobile-application-development>
- [20] AWS, “What is Mobile Application Development?,” Amazon Web Services. Diakses: 24 Oktober 2023. [Daring]. Tersedia pada: <https://aws.amazon.com/id/mobile/mobile-application-development/>
- [21] A. Eka dan Y. Wibawa, “Implementasi Platform Digital Sebagai Media Pembelajaran Daring di MI Muhammadiyah PK Kartasura Pada Masa Pandemi Covid-19,” *Berajah Journal*, 2021, Diakses: 21 Oktober 2023. [Daring]. Tersedia pada: <https://www.ojs.berajah.com/index.php/go/article/view/27>
- [22] R. Ramdani, A. Parlaungan Nasution, P. Ramanda, D. Darma Sagita, dan A. Yanizon, “Strategi Kolaborasi Dalam Manajemen Pelayanan Bimbingan dan Konseling di Sekolah,” 2020.
- [23] A. Endi, Afrina, dan P. H. Wahyudiono, “Perancangan Aplikasi Manajemen Proyek, Tugas dan Kolaborasi dalam Meningkatkan Kinerja Pegawai,” *Journal of Engineering, Technology, and Applied Science*, vol. 4, no. 1, hlm. 22–32, Apr 2022, doi: 10.36079/lamintang.jetas-0401.352.

- [24] R. Lara Ditha, S. Tita Faulina, dan Wisnumurti, “Rancang Bangun Aplikasi Layanan Pengaduan Pada Dinas Pendidikan Kabupaten Oku Berbasis Android Menggunakan Android Studio,” *Jurnal Informatika dan Komputer (JIK)*, vol. 14, hlm. 25, 2023, Diakses: 7 Januari 2025. [Daring]. Tersedia pada:
<https://journal.unmaha.ac.id/index.php/jik/article/download/252/214/896>
- [25] A. J. Kurniawan, C. Hermawan, dan S. Kom, “Rancang Bangun Aplikasi Game Edukasi Pengenalan Budaya Indonesia Berbasis Android,” *Jurnal Penelitian Dosen Fikom (UNDA)*, vol. 10, no. 2, 2019, Diakses: 24 Oktober 2023. [Daring]. Tersedia pada:
<https://www.jurnal.unda.ac.id/index.php/Jpdf/article/view/137/132>
- [26] W. Alexandra, A. Dwi Putra, dan A. S. Puspanigrum, “Penerapan Teknologi Augmented Reality Berbasis Android Untuk Pembelajaran Rantai Makanan Pada Hewan,” *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 3, no. 1, hlm. 1–24, 2022, [Daring]. Tersedia pada: <http://jim.teknokrat.ac.id/index.php/informatika>
- [27] A. Toyib Hidayat dan I. Gede Olka Santoso, “Membershiplication Berbasis Android Dengan Penerapan Kotlin Programming Language di Wijaya Fitness Center (WFC),” *Jusim : Jurnal SistemInformasiMusi Rawas*, vol. 8, no. 1, 2023, Diakses: 8 Januari 2025. [Daring]. Tersedia pada:
<https://www.jurnal.univbinainsan.ac.id/index.php/jusim/article/view/1952/1025>
- [28] Febriandiza A, “Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin,” *Jurnal Pseudocode*, vol. 7, 2020, Diakses: 24 Oktober 2023. [Daring]. Tersedia pada:
<https://ejournal.unib.ac.id/pseudocode/article/view/12946/6500>
- [29] F. Deka, S. Samudra, A. Rosadi, dan T. Novianti, “Rancang Bangun Aplikasi Jadwal Sholat Dan Pengantar Ke Masjid Terdekat Berbasis Android,” 2019. Diakses: 24 Oktober 2023. [Daring]. Tersedia pada:
<https://journal.um-surabaya.ac.id/index.php/CI/article/view/3833>

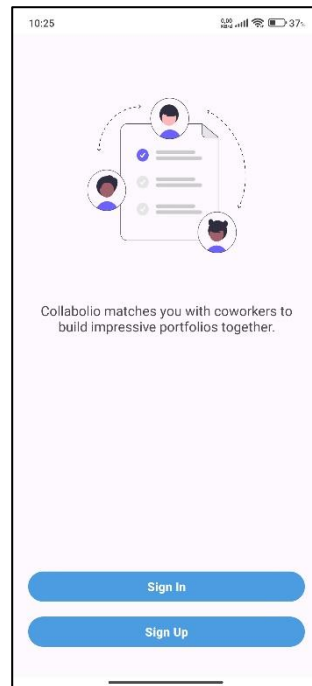
- [30] A. Agustin Muris, J. Ratu Penghulu No, K. Sari, K. Ogan Komerling Ulu, dan S. Selatan, “Pembuatan Company Profile Rutan Klas II B Baturaja Menggunakan Android Studio,” *JURNAL INTECH*, vol. 4, no. 1, hlm. 1–6, 2023, Diakses: 8 Januari 2025. [Daring]. Tersedia pada: <https://unbara.ac.id/journal/index.php/INTECH/article/view/1817/1272>
- [31] A. A. Makiolor A, A. E. Sinsuw A, dan B. N. Najoan Xaverius, “Rancang Bangun Pencarian Rumah Sakit, Puskesmas dan Dokter Praktek Terdekat di Wilayah Manado Berbasis Android,” *E-Journal TeknikInformatika*, vol. 10, 2017, Diakses: 24 Oktober 2023. [Daring]. Tersedia pada: <https://ejournal.unsrat.ac.id/index.php/informatika/article/view/16552>
- [32] R. Andrianto dan M. Haris Munandar, “Aplikasi E-Commerce Penjualan Pakaian Berbasis Android Menggunakan Firebase Realtime Database,” *Journal Computer Science and Information Technology(JCoInT) Program Studi Teknologi Informasi*, no. 1, 2022, [Daring]. Tersedia pada: <http://jurnal.ulb.ac.id/index.php/JCoInT/index>
- [33] Google, “Firebase,” Google Firebase. Diakses: 25 Oktober 2023. [Daring]. Tersedia pada: <https://firebase.google.com/products-build>
- [34] Y. A. Raisnaldi dkk., “Prototype Sistem Kamera Menggunakan EPS32 Dengan Modul Kamera OV2640 Yang Berintegrasi Firebase,” *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 4, 2024.
- [35] Fakultas Ilmu Komputer Universitas Brawijaya, “Programming Educational Scholarship 2023 From Dicoding Indonesia,” Fakultas Ilmu Komputer Universitas Brawijaya. Diakses: 7 Januari 2025. [Daring]. Tersedia pada: <https://filkom.ub.ac.id/2023/03/09/programming-educational-scholarship-2023-from-dicoding-indonesia/>
- [36] Dicoding, “Tentang Dicoding,” Dicoding. Diakses: 7 Januari 2025. [Daring]. Tersedia pada: <https://www.dicoding.com/about>
- [37] Dio Farrel Putra Rachmawan, “Menjadi Android Developer Profesional Bersama Bangkit Academy.” Diakses: 1 Februari 2024. [Daring]. Tersedia pada: <https://kumparan.com/duofarrel/menjadi-android-developer-profesional-bersama-bangkit-academy-1yXKYcADuN1/2>

- [38] L. D. Kumalasari dan A. Susanto, "Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement," *SISFORMA*, vol. 6, no. 2, hlm. 63–72, Feb 2020, doi: 10.24167/sisforma.v6i2.2240.
- [39] I. P. Saputra dan S. R. Nudin, "Rancang Bangun Aplikasi SSKA (Sistem Informasi Karier) Berbasis Android," *Jurnal Manajemen Informatika*, vol. 10, no. 2, hlm. 21–27, 2020.
- [40] E. B. Pramana dan A. Sunandar, "Analisa Rancang Bangun Aplikasi Biro Jodoh Islami Menggunakan Algoritma Profile Matching Berdasarkan Platform Android," *Arcitech: Journal of Computer Science and Artificial Intelligence*, vol. 2, no. 2, hlm. 123, Des 2022, doi: 10.29240/arcitech.v2i2.5834.
- [41] G. Nazari dan E. Lumba, "Aplikasi Pencarian Rekan dan Tempat Singgah saat Touring Berbasis Android," *KALBISIANA : Jurnal Mahasiswa Institut Teknologi dan Bisnis Kalbis*, vol. 8, no. 1, 2022.
- [42] D. P. Kesuma, "Penggunaan Metode System Usability Scale Untuk Mengukur Aspek Usability Pada Media Pembelajaran Daring Di Universitas XYZ," vol. 8, no. 3, hlm. 1615–1626, 2021, [Daring]. Tersedia pada: <http://jurnal.mdp.ac.id>

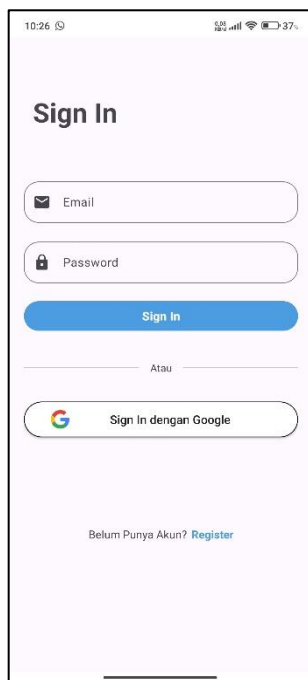
LAMPIRAN A HASIL PERANCANGAN APLIKASI



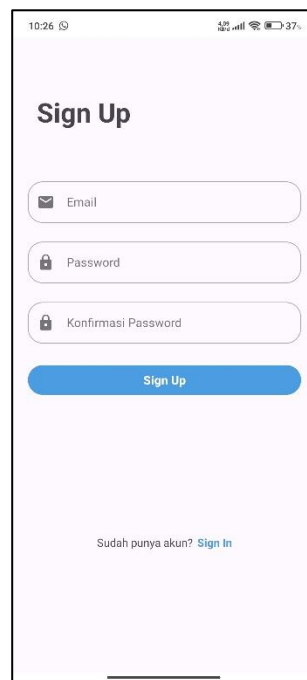
Gambar A.1 Tampilan *Splash Screen*



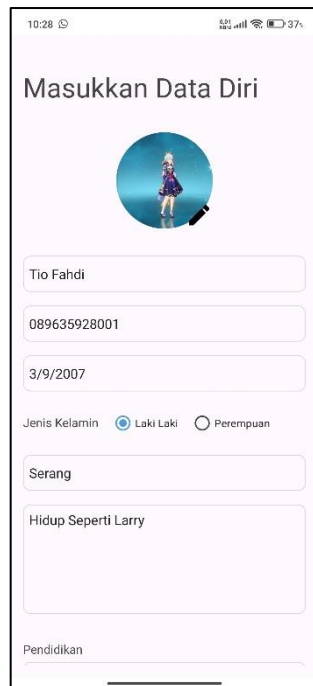
Gambar A.2 Tampilan *Welcome Activity*



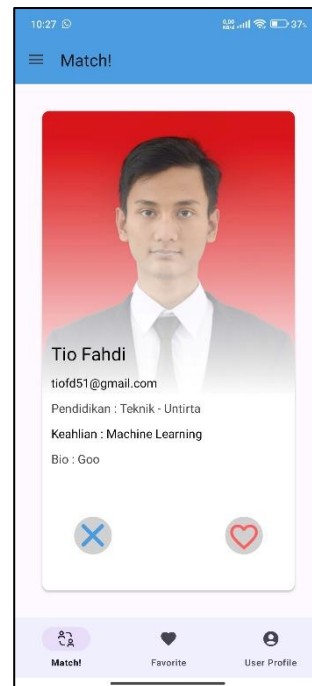
Gambar A.3 Tampilan *Login Activity*



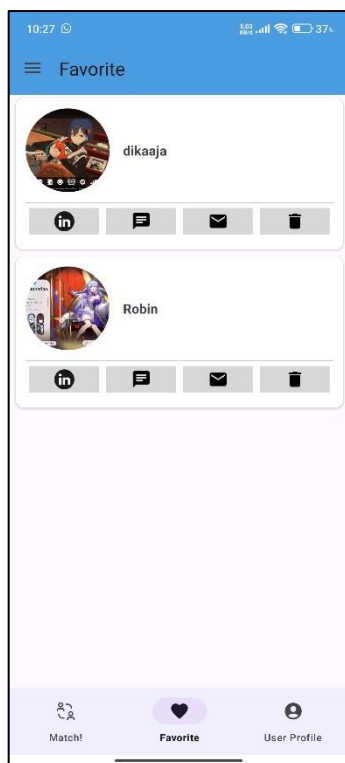
Gambar A.4 Tampilan *Register Activity*



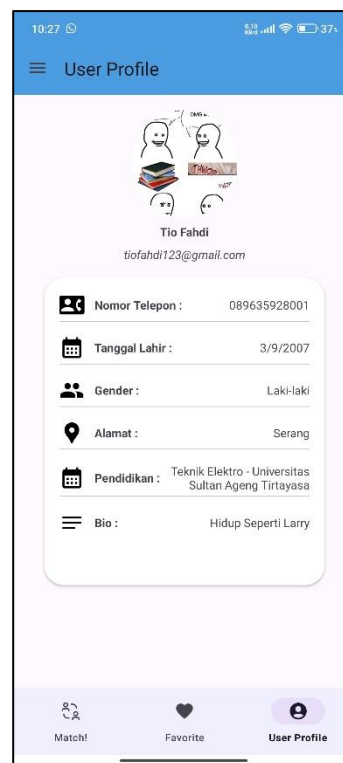
Gambar A.5 Tampilan *Input Bio Activity*



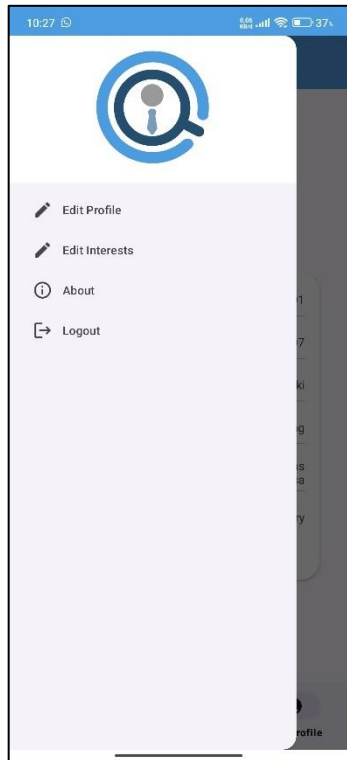
Gambar A.6 Tampilan *Swipe*



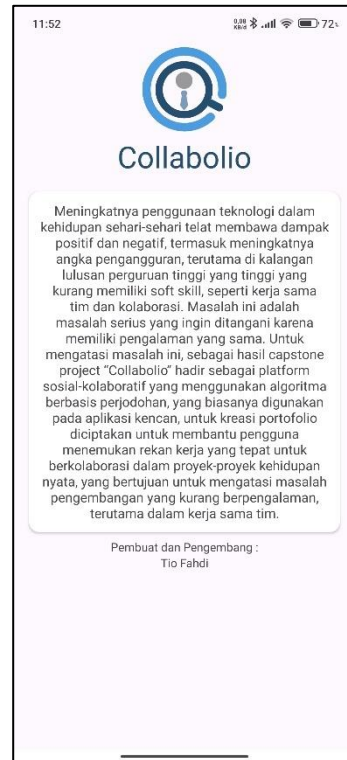
Gambar A.6 Tampilan *Favorite*



Gambar A.7 Tampilan *User Profile*



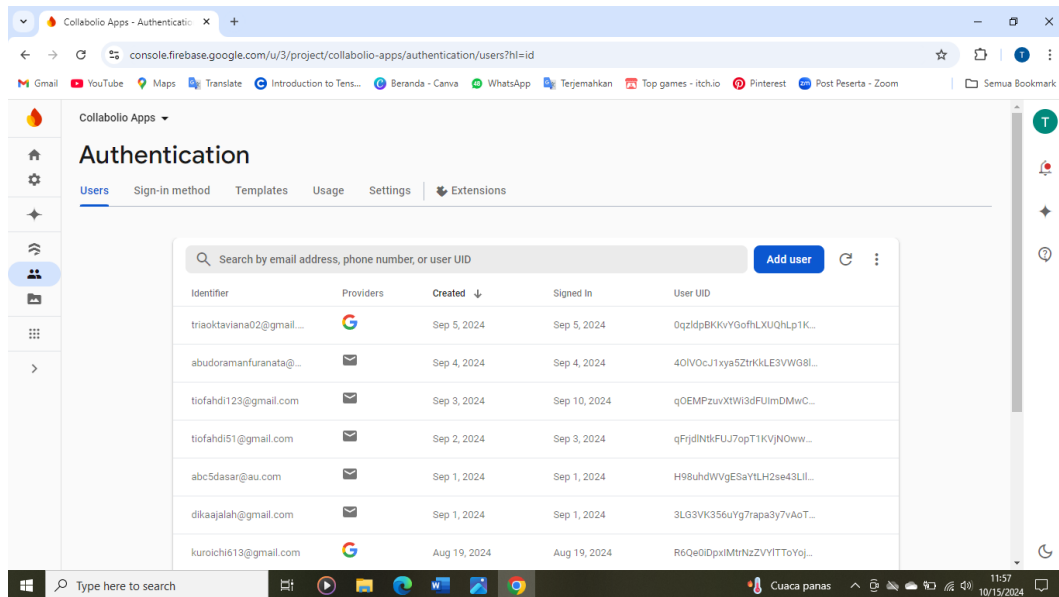
Gambar A.8 Tampilan *Navigation Drawer*



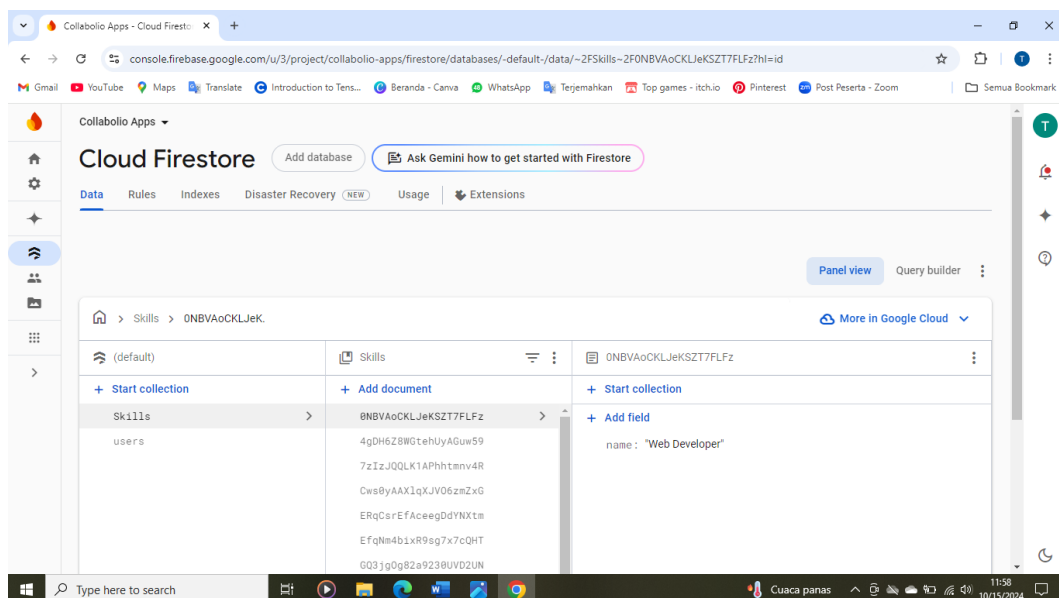
Gambar A.9 Tampilan *About*

Link Demo Aplikasi: https://drive.google.com/file/d/1aAncZodVln-6VKwG9A_hySC7guRA9qCW/view?usp=sharing

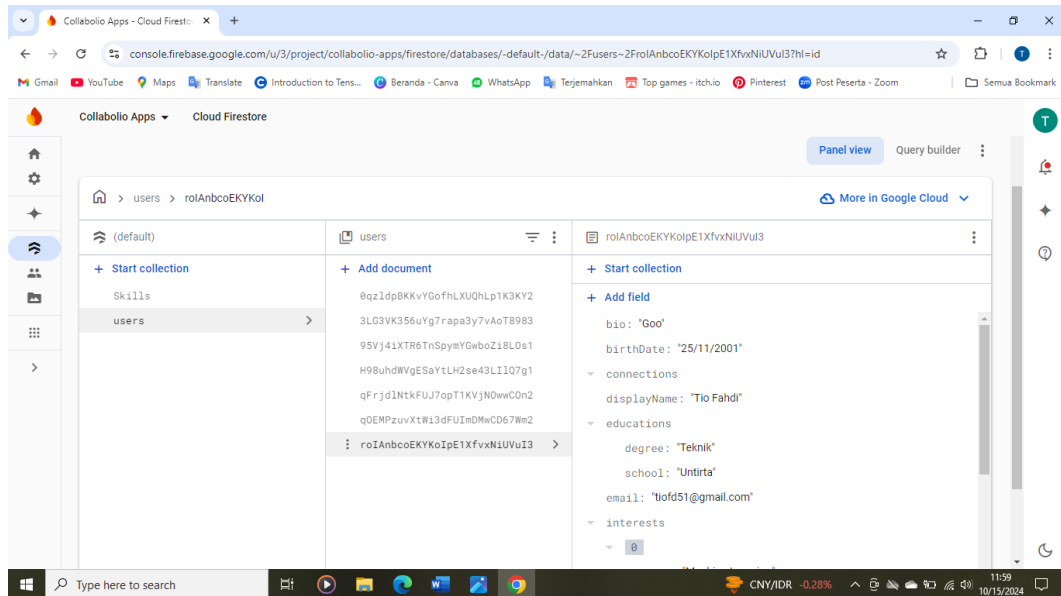
LAMPIRAN B TAMPILAN PADA FIREBASE



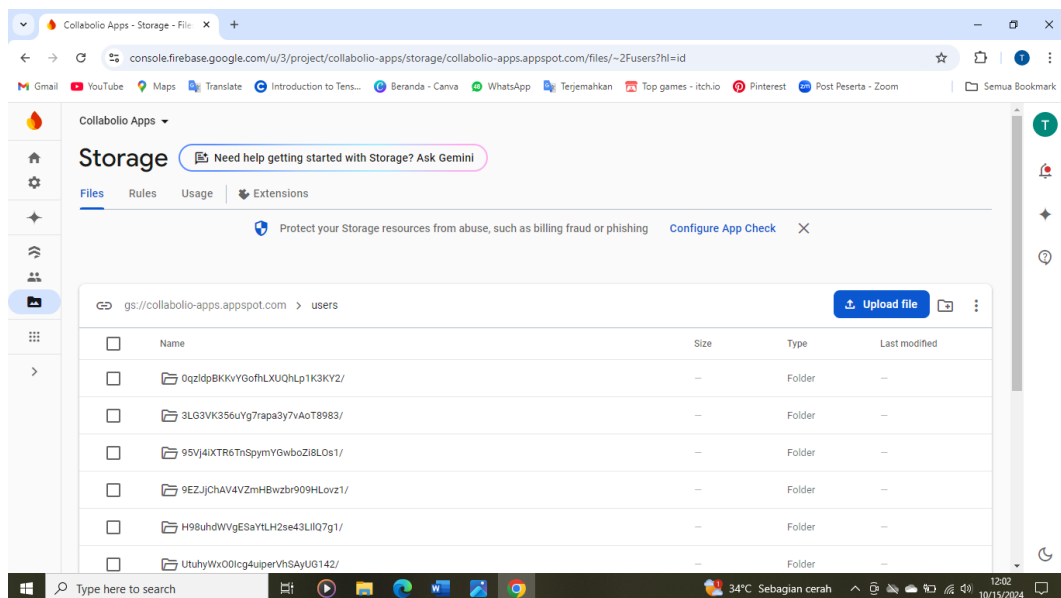
Gambar B.1 Tampilan Firebase Authentication



Gambar B.2 Tampilan Data Firestore Skills



Gambar B.3 Tampilan Data Firestore *Users*

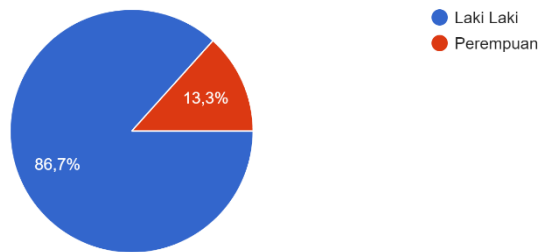


Gambar B.4 Tampilan Firebase *Storage*

LAMPIRAN C HASIL SURVEI APLIKASI

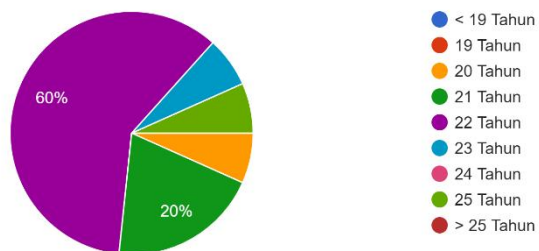
Link Survei Aplikasi <https://forms.gle/V94LzonbvmexRiot8>

Jenis Kelamin
15 jawaban



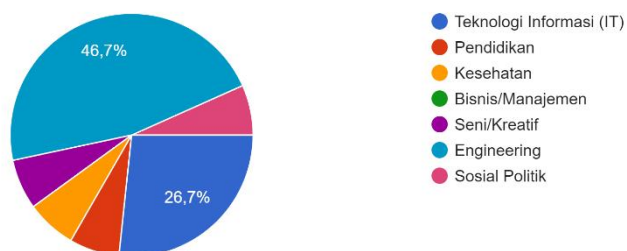
Gambar C.1 Data Survei Jenis Kelamin Responden

Usia
15 jawaban



Gambar C.2 Data Survei Usia Responden

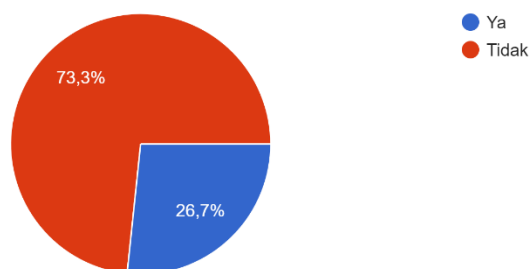
Bidang keahlian
15 jawaban



Gambar C.3 Data Survei Bidang Keahlian Responden

Apakah Anda pernah menggunakan aplikasi kencan online? (tinder, bumble, omi, etc)

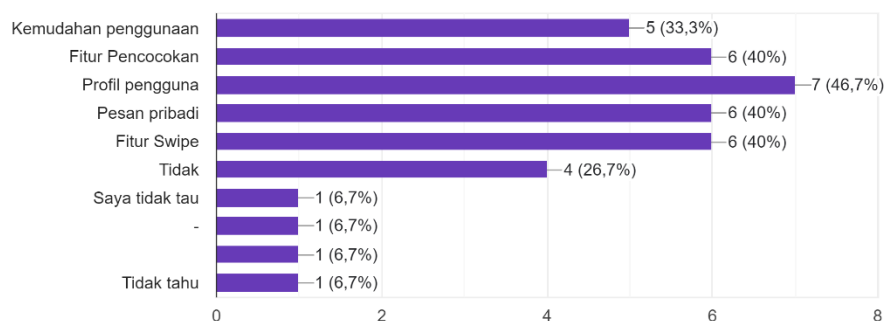
15 jawaban



Gambar C.4 Data Survei Pengalaman Menggunakan Aplikasi Kencan Online

Fitur apa saja yang Anda ketahui dari aplikasi kencan online? (Jika pada pertanyaan sebelumnya menjawab "Tidak" silahkan memilih opsi lainnya)

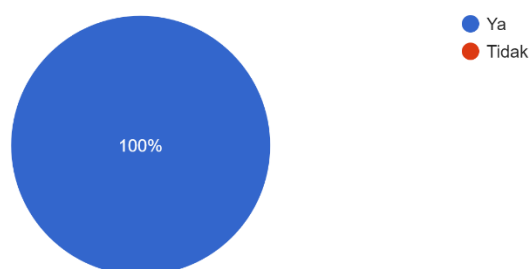
15 jawaban



Gambar C.5 Data Survei Fitur Kencan Online yang Diketahui Responden

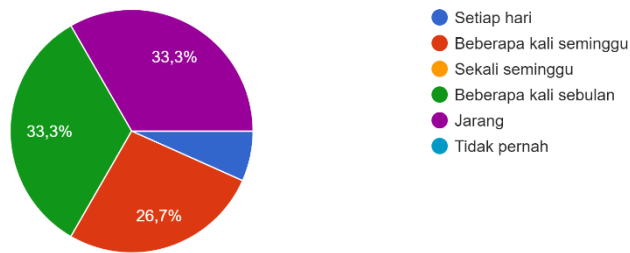
Apakah Anda Pernah Menggunakan Aplikasi/Web portofolio Digital (LinkedIn, etc)

15 jawaban



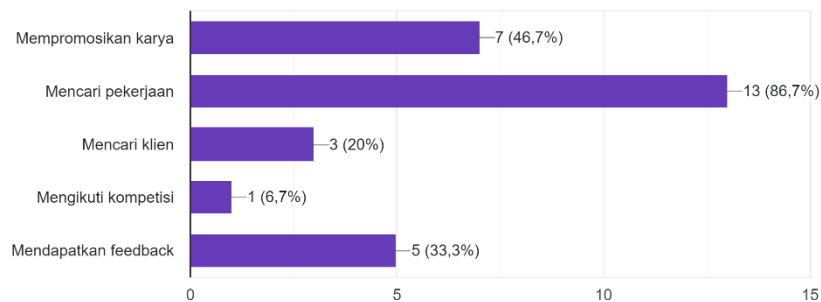
Gambar C.6 Data Survei Pengalaman Menggunakan Aplikasi/Web Portofolio Digital

Jika "Ya", Seberapa sering anda menggunakan aplikasi/web portofolio digital?
15 jawaban



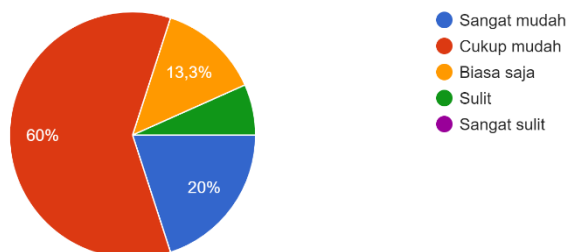
Gambar C.7 Data Survei Frekuensi Penggunaan Aplikasi/Web Portofolio Digital

Apa tujuan utama Anda menggunakan aplikasi/web portofolio digital?
15 jawaban



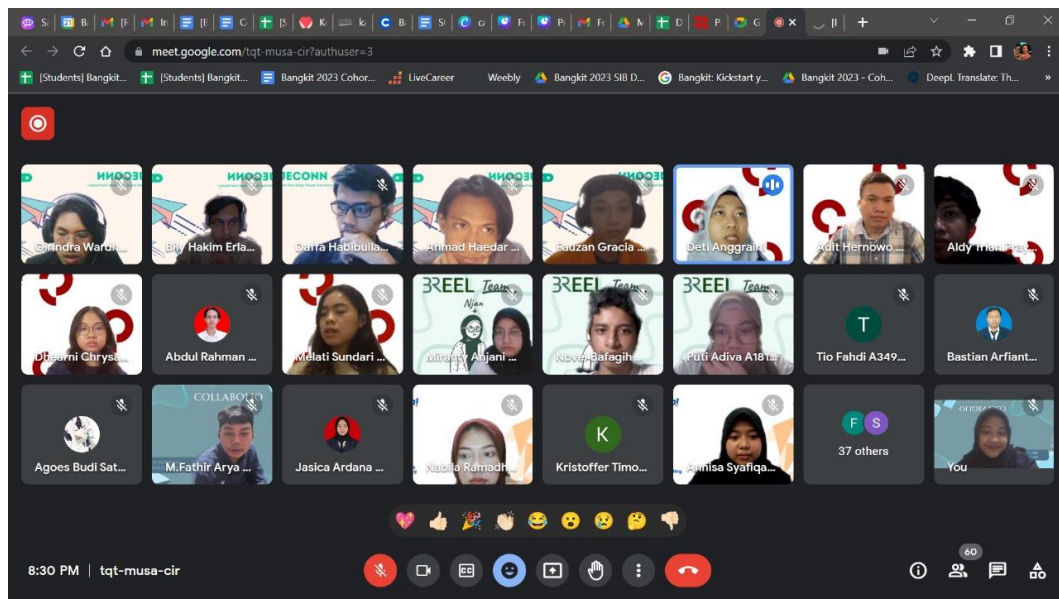
Gambar C.8 Data Survei Tujuan Penggunaan Aplikasi/Web Portofolio Digital

Seberapa mudah Anda menemukan informasi yang Anda butuhkan pada aplikasi/web portofolio digital?
15 jawaban

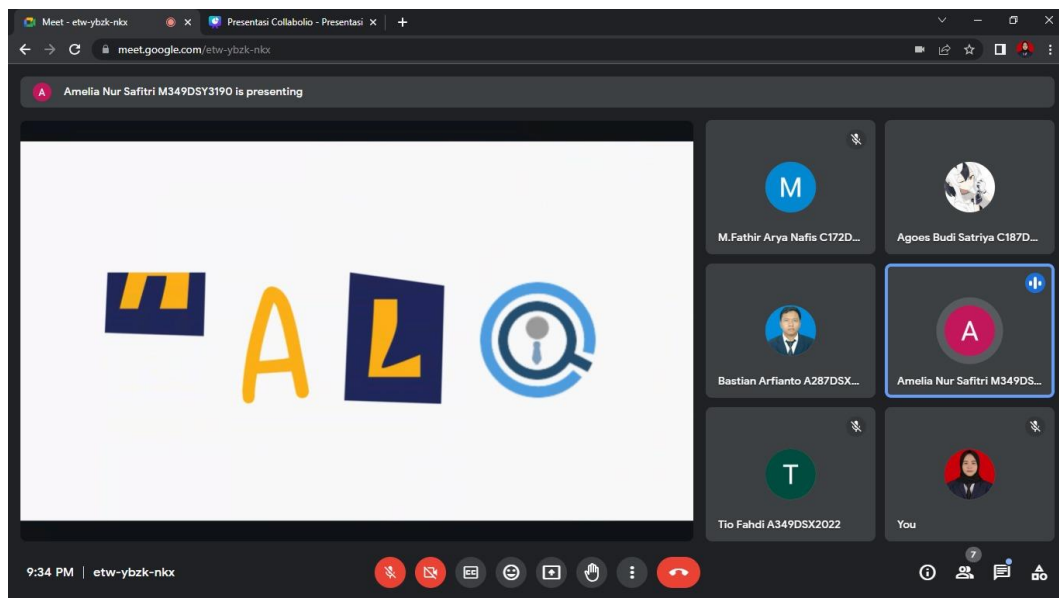


Gambar C.3 Data Survei Kemudahan Pengguna Dalam Menggunakan Aplikasi/Web Portofolio Digital

LAMPIRAN D DOKUMENTASI KEGIATAN



Gambar C.1 Presentasi *Capstone Project*



Gambar C.2 Meeting Mingguan *Capstone Project*