

# Rancang Bangun Aplikasi Perencanaan Karir Berbasis *iOS* Menggunakan Metode *Rapid Application Development*

1<sup>st</sup> Doanta Aloycius Ginting  
Telkom University Purwokerto  
Purwokerto, Indonesia  
doantaaloycius@gmail.com

2<sup>nd</sup> Abednego Dwi Septiadi, S.Kom., M.Kom.  
Telkom University Purwokerto  
Purwokerto, Indonesia  
abednego@telkomuniversity.ac.id

**Abstrak** — Mahasiswa kerap mengalami kesulitan dalam merencanakan jalur karir yang sesuai dengan minat dan kompetensi mereka, akibat minimnya bimbingan, keterbatasan alat pendukung, dan kurangnya struktur eksplorasi karir. Penelitian ini bertujuan merancang dan mengembangkan aplikasi perencanaan karir berbasis *iOS* menggunakan metode *Rapid Application Development (RAD)* guna memberikan panduan terstruktur bagi mahasiswa dalam mengenal dan mengeksplorasi berbagai jalur karir. Proses pengembangan meliputi analisis kebutuhan pengguna, desain antarmuka secara iteratif, hingga implementasi fitur utama seperti eksplorasi karir dan roadmap pembelajaran. Validasi dilakukan melalui workshop dan pengujian bersama mahasiswa untuk memastikan aplikasi sesuai kebutuhan. Hasil pengujian menunjukkan bahwa aplikasi mudah digunakan dan efektif dalam meningkatkan pemahaman serta kesiapan mahasiswa menghadapi dunia kerja. Aplikasi ini memberikan kontribusi nyata dalam mendukung proses perencanaan karir mahasiswa, dengan potensi pengembangan lebih lanjut di masa depan.

**Kata kunci** — perencanaan karir, pengembangan aplikasi *iOS*, *Rapid Application Development*, *Swift*, mahasiswa, roadmap karir

## I. PENDAHULUAN

Perencanaan karir merupakan proses penting yang dilakukan oleh mahasiswa untuk menentukan jalur profesi yang sesuai dengan minat, kompetensi, dan aspirasi mereka. Namun, dalam praktiknya, banyak mahasiswa mengalami hambatan dalam merancang masa depan karir secara sistematis. Hambatan tersebut meliputi kurangnya bimbingan dari lingkungan kampus, terbatasnya akses terhadap alat pendukung perencanaan karir, serta minimnya struktur eksplorasi karir yang tersedia. Studi terdahulu menunjukkan bahwa perencanaan karir yang matang dapat meningkatkan motivasi mahasiswa dalam menyelesaikan studi dan memberikan gambaran lebih jelas mengenai prospek kerja setelah lulus [1], [2].

Survei yang dilakukan terhadap 108 mahasiswa di Telkom University Purwokerto mengungkapkan bahwa sebagian besar responden kesulitan mengenali minat, nilai, dan kemampuan dirinya, serta belum menemukan jalur karir

yang tepat. Selain itu, mayoritas mahasiswa belum pernah memanfaatkan aplikasi perencanaan karir, sehingga proses eksplorasi dan penentuan tujuan karir berjalan tanpa arah yang jelas. Kondisi ini menunjukkan adanya kebutuhan akan solusi berbasis teknologi yang dapat membantu mahasiswa dalam merancang dan mengelola perencanaan karir secara mandiri dan terstruktur.

Untuk menjawab kebutuhan tersebut, penelitian ini mengusulkan pengembangan aplikasi perencanaan karir berbasis *iOS* yang dirancang khusus bagi mahasiswa. Aplikasi ini diharapkan dapat menjadi sarana pendukung yang efektif dalam membantu mahasiswa mengenali potensi diri, mengeksplorasi berbagai jalur karir, dan merancang roadmap pembelajaran menuju dunia kerja. Metode *Rapid Application Development (RAD)* dipilih sebagai kerangka kerja pengembangan aplikasi karena memungkinkan proses iterasi desain yang cepat dan partisipasi aktif pengguna dalam memberikan umpan balik, sehingga aplikasi yang dihasilkan dapat lebih responsif terhadap kebutuhan nyata mahasiswa [3]. Dengan demikian, penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi perencanaan karir yang relevan, mudah digunakan, dan mampu memberikan dampak positif terhadap kesiapan mahasiswa dalam menghadapi dunia kerja.

## II. KAJIAN TEORI

Kajian teori berikut membahas landasan metodologis, teknologi, serta konsep arsitektur perangkat lunak yang menjadi dasar utama dalam penelitian ini.

### A. *Rapid Application Development*

*RAD* adalah metodologi pengembangan perangkat lunak yang menekankan kecepatan dan kualitas melalui penggunaan prototipe, iterasi, dan keterlibatan intensif pengguna [4]. *RAD* terdiri dari tiga fase utama: perencanaan kebutuhan pengguna, proses desain dan implementasi, serta *cutover* untuk validasi akhir [5].

### B. Platform *iOS* dan Bahasa pemrograman *Swift*

Platform *iOS* menyediakan infrastruktur teknis untuk pengembangan, pengujian, dan implementasi aplikasi berkualitas tinggi [6]. *Swift*, bahasa pemrograman utama untuk *iOS* dan *macOS*, menawarkan sintaks yang efisien

dan modern, serta menjadi standar industri untuk pengembangan aplikasi Apple [7].

### C. Clean Architecture

Clean Architecture merupakan pendekatan desain perangkat lunak yang memisahkan kode berdasarkan tanggung jawab ke dalam beberapa lapisan, seperti presentation, domain, dan data. Pendekatan ini meningkatkan skalabilitas, kemudahan pemeliharaan, dan memungkinkan pengujian unit secara terisolasi [8], [9], [10].

### D. Firebase

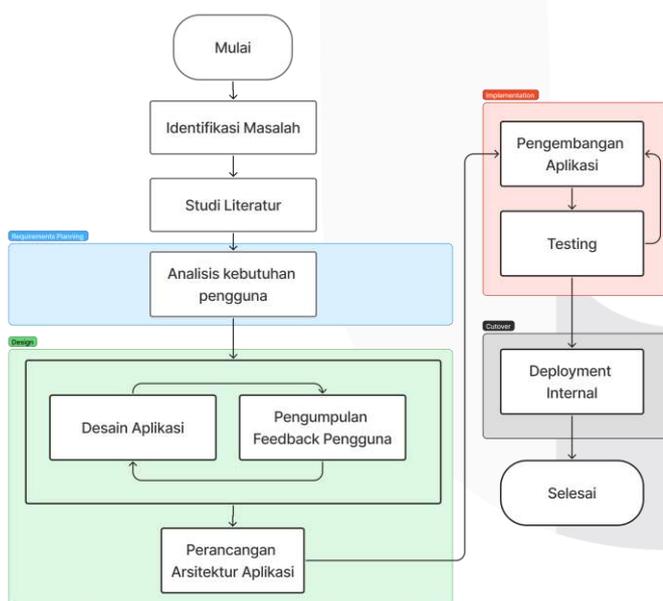
Firebase adalah platform backend yang menyediakan layanan seperti database real-time, autentikasi, dan penyimpanan data untuk aplikasi mobile. Integrasi Firebase mendukung keamanan data, update real-time, serta kemudahan pengelolaan data pengguna [11].

### E. SwiftUI

SwiftUI adalah framework deklaratif untuk membangun antarmuka pengguna pada ekosistem Apple. SwiftUI menyediakan komponen visual, event handler, dan alat pengelolaan aliran data yang modern dan responsif [12], [13].

## III. METODE

Penelitian ini menggunakan metode RAD, dimulai dari identifikasi masalah, studi literatur, analisis kebutuhan pengguna, desain aplikasi, perancangan arsitektur, pengembangan, pengujian, hingga deployment internal melalui TestFlight.



GAMBAR 1.  
(ALUR PENELITIAN)

Analisis kebutuhan dilakukan melalui wawancara dan survei terhadap mahasiswa, diikuti dengan iterasi desain bersama pengguna. Pengembangan aplikasi menggunakan Swift dan SwiftUI, dengan backend Firebase dan penerapan clean architecture untuk mendukung skalabilitas dan kemudahan pemeliharaan. Pengujian dilakukan dengan metode black box untuk memastikan seluruh fitur berjalan sesuai kebutuhan pengguna.

### A. Identifikasi Masalah

Pada tahap ini, peneliti mengidentifikasi permasalahan utama yang dihadapi mahasiswa dalam merencanakan karir, menentukan target pengguna, serta ruang lingkup dan ekosistem aplikasi yang akan dikembangkan.

### B. Studi Literatur

Studi literatur dilakukan untuk mengumpulkan referensi dan solusi dari penelitian terdahulu yang relevan, sehingga dapat ditemukan pendekatan dan fitur yang sesuai untuk aplikasi yang akan dikembangkan.

### C. Rapid Application Development

#### 1) Analisis Kebutuhan Pengguna

Pada tahap ini, peneliti melakukan wawancara terhadap mahasiswa sebagai calon pengguna untuk mengidentifikasi kebutuhan, permasalahan, dan fitur penting yang harus ada dalam aplikasi perencanaan karir. Hasil analisis ini menjadi dasar dalam merancang aplikasi agar benar-benar sesuai dengan kebutuhan nyata pengguna.

#### 2) Design Workshop

Tahap ini melibatkan proses perancangan desain aplikasi secara iteratif, mulai dari pembuatan sketsa low-fidelity hingga prototype high-fidelity. Pengguna dilibatkan dalam workshop untuk memberikan feedback langsung terhadap desain, sehingga aplikasi yang dikembangkan memiliki alur, tampilan, dan fitur yang sesuai harapan dan kebutuhan mereka.

#### 3) Perancangan Arsitektur Aplikasi

Setelah desain aplikasi disepakati, peneliti merancang arsitektur aplikasi untuk menentukan struktur teknis, pemilihan teknologi, dan pengaturan modul-modul utama. Perancangan arsitektur ini bertujuan agar proses pengembangan berjalan efisien, terstruktur, dan aplikasi mudah dikembangkan serta dipelihara di masa depan.

#### 4) Pengembangan Aplikasi

Pada tahap ini, pengembangan aplikasi dilakukan berdasarkan desain dan arsitektur yang telah dibuat. Proses ini mencakup pembuatan antarmuka pengguna, implementasi fitur utama, integrasi backend, serta dokumentasi teknis. Seluruh proses pengembangan dikelola menggunakan version control untuk memastikan perubahan tercatat dengan baik.

#### 5) Testing

Pada tahap ini, aplikasi yang telah dikembangkan diuji menggunakan metode blackbox untuk memastikan seluruh fitur berjalan sesuai fungsi dan bebas dari bug. Pengujian dilakukan dengan berbagai skenario penggunaan untuk memastikan aplikasi stabil, mudah digunakan, dan memenuhi kebutuhan pengguna.

#### 6) Cutover/Deployment

Tahap akhir adalah deployment aplikasi secara internal menggunakan TestFlight, di

mana aplikasi diuji secara terbatas oleh tim internal sebelum dirilis ke publik. Proses ini juga mencakup persiapan metadata, dokumentasi, dan validasi akhir agar aplikasi siap dipublikasikan secara luas di App Store.

IV. HASIL DAN PEMBAHASAN

Pada tahap ini, metode Rapid Application Development (RAD) diimplementasikan secara utuh mulai dari analisis kebutuhan, perancangan, pengembangan prototipe, hingga pengujian bersama pengguna untuk menghasilkan solusi aplikasi perencanaan karir berbasis iOS yang tepat guna dan sesuai dengan kebutuhan mahasiswa.

A. Analisis Kebutuhan Pengguna

Tahap ini diawali dengan wawancara kepada mahasiswa dari beberapa jurusan di Telkom University Purwokerto untuk memahami kendala dan kebutuhan mereka dalam merencanakan karir. Hasil analisis menunjukkan perlunya fitur eksplorasi karir yang terstruktur dan roadmap karir yang jelas. Kebutuhan aplikasi dirinci dalam dua kategori, yaitu kebutuhan fungsional (misalnya autentikasi, pemilihan karir, melihat detail tugas, dan pencatatan progres) dan non-fungsional (seperti keamanan data, kemudahan penggunaan, performa, dan skalabilitas). Tabel 1 (Tabel Fitur Utama) dan Tabel 2 (Kebutuhan Fungsional) merepresentasikan fitur-fitur inti yang harus ada dalam aplikasi.

TABEL 1. (FITUR UTAMA APLIKASI)

No.	Fitur	Deskripsi
1	Eksplorasi Karir	User dapat melakukan eksplorasi karir dalam lingkup IT yang terstruktur
2	Roadmap Karir	User dapat melihat roadmap karir yang dipilih dari fundamental hingga siap untuk dunia kerja

Tabel 1 berisi daftar fitur utama yang dikembangkan dalam aplikasi perencanaan karir berbasis iOS. Fitur-fitur ini dirancang untuk membantu mahasiswa melakukan eksplorasi karir secara terstruktur dan menyediakan roadmap karir yang jelas, sehingga pengguna dapat mengenal dan mempersiapkan jalur karir yang sesuai dengan minat dan kompetensi mereka. Adapun rincian kebutuhan fungsional dari setiap fitur utama tersebut dapat dilihat lebih detail pada Tabel 2 berikut ini.

TABEL 2. (KEBUTUHAN FUNGSIONAL)

No.	Kebutuhan Fungsional	Detail Kebutuhan Fungsional
1	Autentikasi	User dapat login ke dalam sistem
2		User dapat logout dari sistem
3	Pilih Rencana karir	User dapat mengakses list karir yang tersedia
4		User dapat memilih karir yang ingin dipelajari
5	Lihat Karir	User dapat melihat detail karir
6	Lihat Milestone	User dapat melihat roadmap karir
7		User dapat melihat keseluruhan milestone ketika commit ke salah satu karir
8	Lihat Tugas	User dapat melihat detail tugas
9	Lihat Resource	User dapat mengakses materi eksternal ( <i>external resource</i> ) yang disediakan

No.	Kebutuhan Fungsional	Detail Kebutuhan Fungsional
10	Komitmen Rencana Karir	User dapat commit ke salah satu karir setelah menyelesaikan trial task
11	Ganti Rencana Karir	User dapat mencoba karir yang lain setelah menjalani trial task
12		User dapat mengubah karir walaupun sudah commit terhadap satu karir
13	Selesaikan Tugas	User dapat menyelesaikan tugas yang ada
14	Buat Catatan	User dapat menyimpan catatan mengenai tugas yang dikerjakan
15	Lihat Profil	User dapat melihat progress melalui statistik pengguna

Tabel 2 merangkum kebutuhan fungsional utama aplikasi, yang meliputi fitur-fitur inti seperti autentikasi, pemilihan karir, pengelolaan tugas, dan pemantauan progres pengguna.

B. Design Workshop

Desain aplikasi dikembangkan melalui tiga iterasi workshop bersama mahasiswa dan tim pengembang, dengan tujuan menyempurnakan alur dan tampilan aplikasi berdasarkan umpan balik pengguna.

1) Iterasi 1

Pada iterasi pertama, dibuat desain low-fidelity yang menampilkan alur kerja aplikasi dari login hingga melihat roadmap karir, termasuk fitur tes kemampuan karir dan rekomendasi karir. Desain Low fidelity dapat dilihat pada gambar di bawah.

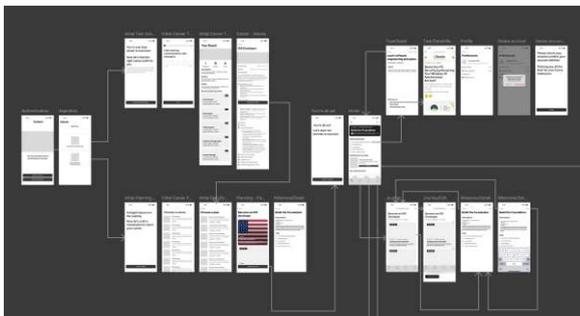


GAMBAR 2. (DESAIN LOW FIDELITY)

Setelah itu diberikan kepada user dan hasil pengujian menunjukkan alur aplikasi masih terlalu kompleks dan fitur tes karir dianggap membingungkan pengguna.

2) Iterasi 2

Desain disederhanakan dan dikembangkan menjadi mid-fidelity, di mana pengguna dapat memilih untuk mengikuti tes karir atau langsung memilih karir yang diinginkan. Tampilan aplikasi difokuskan pada aktivitas utama, yaitu mengerjakan tugas dan melihat progres.

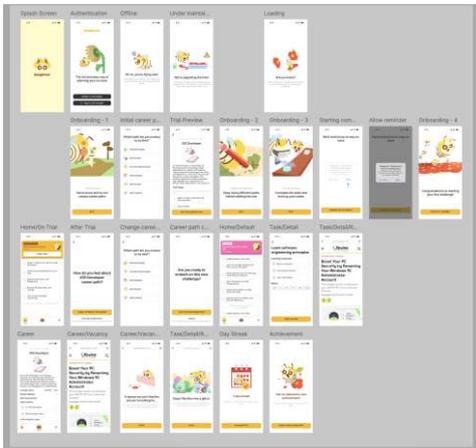


GAMBAR 3.  
(DESAIN MID FIDELITY)

Setelah dilakukan pengujian kepada pengguna, tes minat karir masih dianggap kurang membantu dan membatasi eksplorasi pengguna.

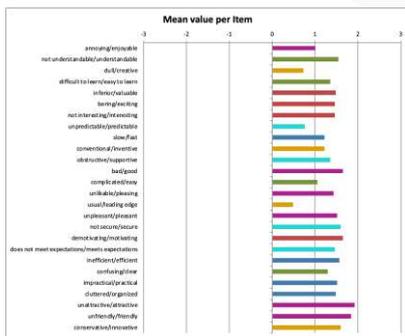
3) Iterasi 3

Pada iterasi ketiga, fitur tes minat karir dihilangkan sepenuhnya. Desain high-fidelity difokuskan pada kebebasan eksplorasi karir, dengan user interface yang lebih siap diimplementasikan dan fitur utama yang lebih sederhana. Berikut merupakan Desain yang sudah dikembangkan menjadi High Fidelity.



GAMBAR 4.  
(DESAIN HIGH FIDELITY)

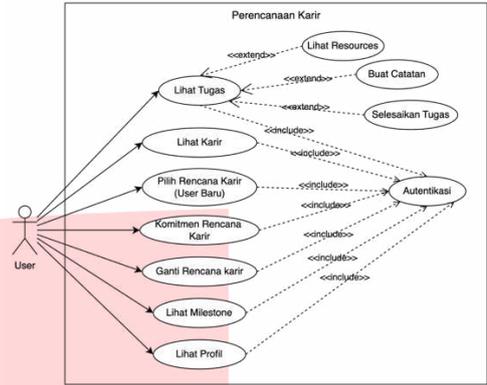
Pengujian menggunakan User Experience Questionnaire menunjukkan antarmuka aplikasi dinilai positif dari segi estetika, kemudahan, dan kejelasan. Gambar 5 menunjukkan skor persepsi pengguna terhadap aspek estetika, kenyamanan, kejelasan, dan kemudahan aplikasi, dengan mayoritas skor berada di rentang positif.



GAMBAR 5.  
(HASIL PENGUJIAN MENGGUNAKAN UEQ)

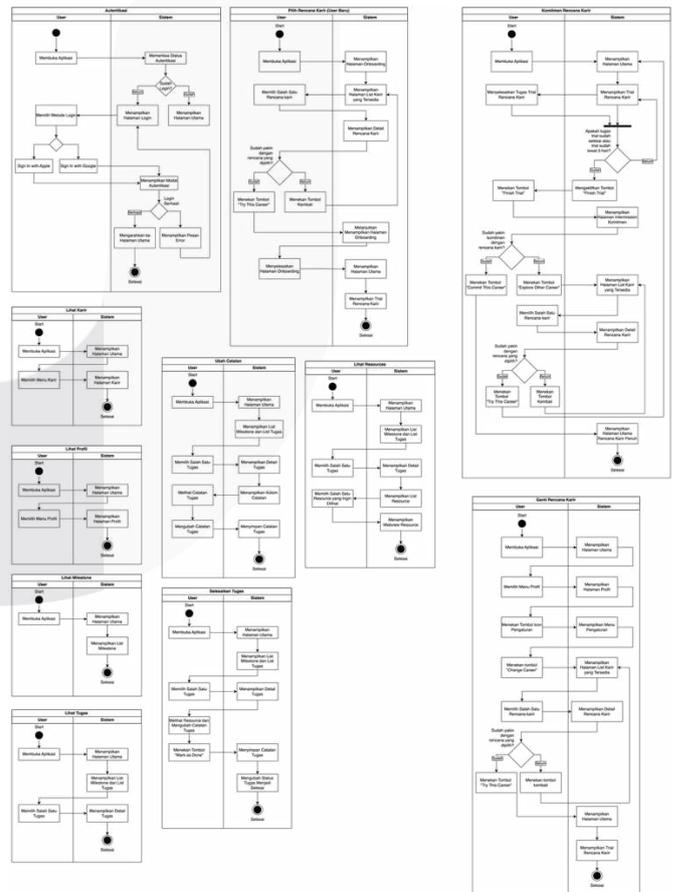
C. Perancangan Arsitektur Aplikasi

Setelah desain disetujui, langkah selanjutnya adalah menyusun use case diagram untuk memetakan interaksi utama antara pengguna dan sistem, seperti proses login, pemilihan karir, pengelolaan tugas, dan pemantauan progres. Use case diagram ini memberikan gambaran jelas mengenai fungsi-fungsi utama yang dapat diakses oleh pengguna dalam aplikasi.



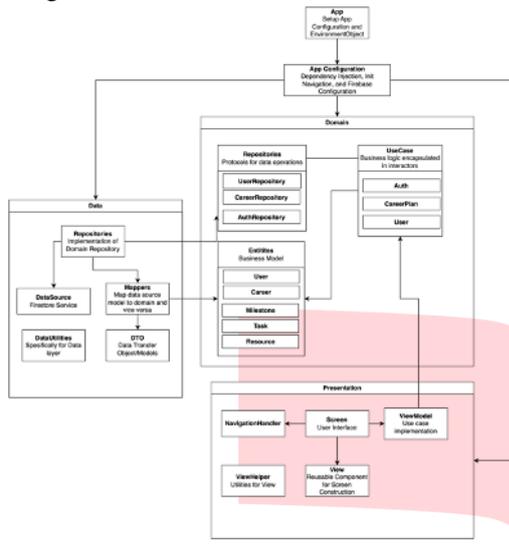
GAMBAR 6.  
(USE CASE DIAGRAM)

Selanjutnya, dilakukan perancangan arsitektur aplikasi menggunakan UML, seperti activity, class, dan sequence diagram. Activity Diagram dapat dilihat pada gambar di bawah.



GAMBAR 7.  
(ACTIVITY DIAGRAM)

Aplikasi menerapkan clean architecture yang memisahkan tanggung jawab ke dalam lapisan presentation, domain, dan data, sehingga mendukung skalabilitas dan kemudahan pemeliharaan. Diagram Arsitektur Aplikasi dapat dilihat pada gambar 6 di bawah.



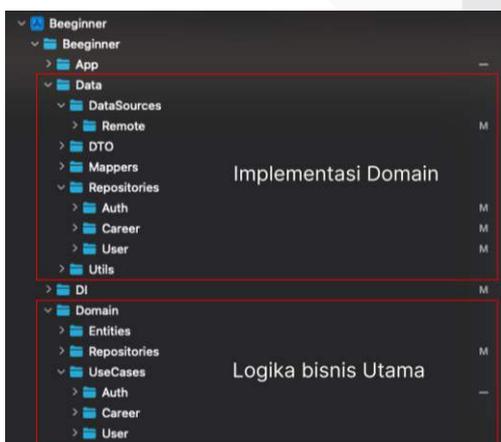
GAMBAR 8. (HIGH LEVEL ARCHITECTURE APLIKASI)

D. Pengembangan Aplikasi

Pengembangan aplikasi dilakukan secara bertahap, dimulai dari konfigurasi backend menggunakan Firebase untuk autentikasi dan penyimpanan data, lalu implementasi fungsi-fungsi utama dan user interface dengan Swift dan SwiftUI. Proses pengembangan didokumentasikan dengan version control di Github, dan deployment dilakukan secara otomatis melalui Xcode Cloud.

Tahap awal pengembangan dimulai dengan konfigurasi backend menggunakan Firebase. Layanan utama yang digunakan adalah Firestore Database untuk penyimpanan data dan Firebase Authentication untuk proses autentikasi pengguna.

Implementasi core logic aplikasi menggunakan bahasa pemrograman Swift. Komunikasi antar lapisan dilakukan melalui protokol, yang memungkinkan penggantian dependensi tanpa memengaruhi lapisan lain.



GAMBAR 9. (STRUKTUR FOLDER APLIKASI)

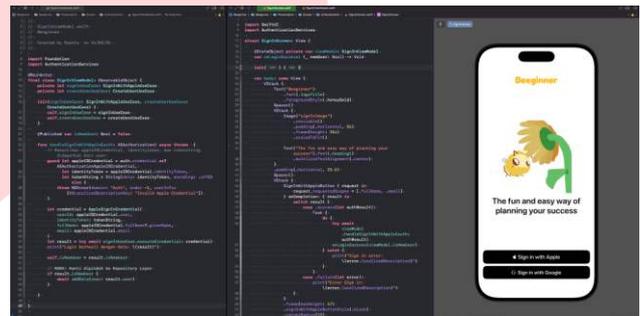
Gambar 9 merupakan struktur folder aplikasi yang dibuat agar pengembangan lebih terstruktur dan mudah dipelihara. Gambar 10 merupakan contoh penggunaan protokol dalam membuat sebuah repository.

```

10 protocol UserRepository {
11     func getUserData() async throws -> UserEntity
12     func createUserData(user: UserEntity) async
13     func createUserCareerPlan(careerID: String, trial: Bool) async
14     func getUserCareerPlan() async throws -> CareerEntity
15     func getAllUserMilestone() async throws -> [MilestoneEntity]
16     func getUserTask(milestoneID: String) async throws -> [TaskEntity]
17     func getUserDetailTask(milestoneID: String, taskID: String) async throws -> TaskEntity
18     func updateUserTask(milestoneID: String, taskID: String, task: TaskEntity) async
19     func getAllResources(milestoneID: String) async throws -> [ResourceEntity]
20 }
    
```

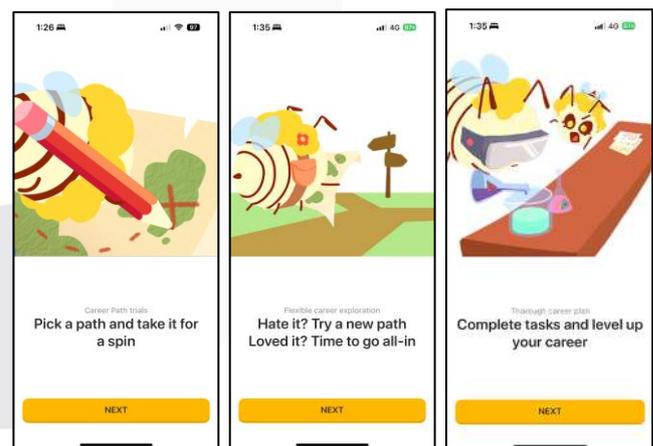
GAMBAR 10. (CONTOH PENGGUNAAN PROTOKOL)

Screen Login dirancang agar pengguna dapat masuk ke aplikasi dengan cepat dan aman menggunakan Sign In with Apple atau Google, tanpa perlu mengisi form secara manual. Tampilan login ini dapat dilihat pada Gambar 11.



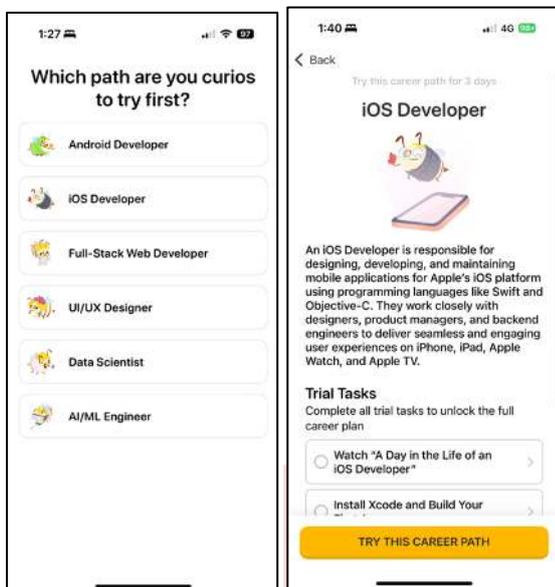
GAMBAR 11. (IMPLEMENTASI SCREEN LOGIN)

Selanjutnya, Screen Onboarding menyajikan serangkaian tampilan yang memandu pengguna baru dalam memahami fungsi dan alur aplikasi secara bertahap, sebagaimana diperlihatkan pada Gambar 12.



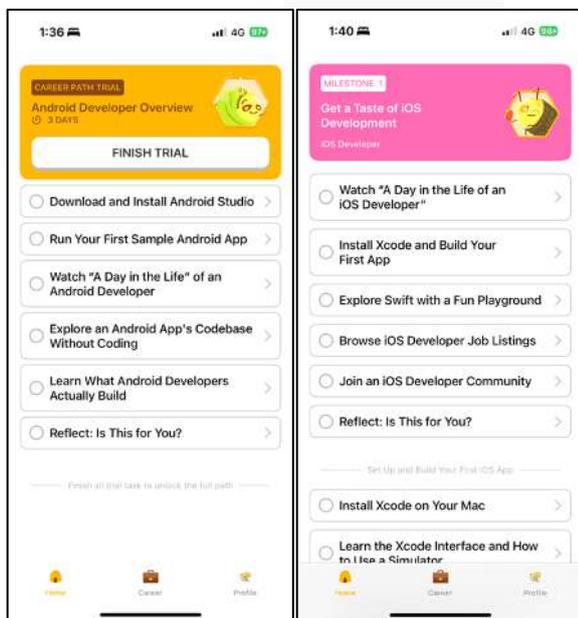
GAMBAR 12. (IMPLEMENTASI SCREEN ONBOARDING)

Pada Screen Career Setup, pengguna diberikan daftar karir yang dapat dipilih serta detail informatif mengenai masing-masing karir, sehingga pengguna dapat menentukan jalur yang sesuai dengan minat dan kompetensi mereka; tampilan ini tercermin pada Gambar 13.



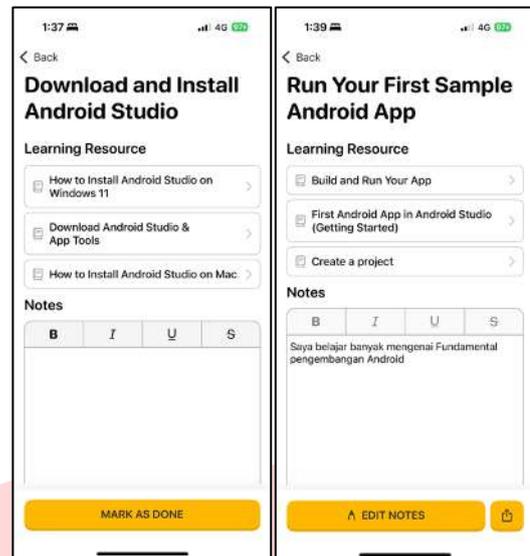
GAMBAR 13. (IMPLEMENTASI SCREEN CAREER SETUP)

Untuk mendukung proses eksplorasi dan pembelajaran, Screen Home menampilkan tugas trial serta roadmap pembelajaran yang terstruktur sesuai dengan karir yang dipilih pengguna. Perbedaan tampilan sebelum dan sesudah pengguna berkomitmen pada satu karir dapat dilihat pada Gambar 14.



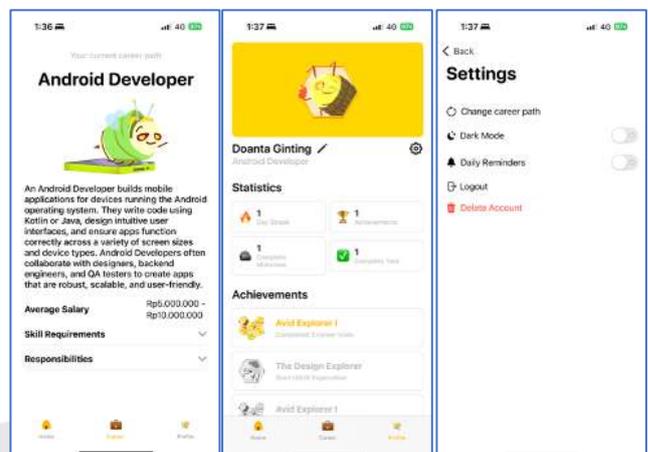
GAMBAR 14. (IMPLEMENTASI SCREEN HOME)

Selain itu, Screen Detail Task menyediakan informasi lengkap mengenai tugas yang harus diselesaikan, resource eksternal sebagai referensi pembelajaran, serta fitur catatan untuk mendokumentasikan proses belajar; tampilan detail tugas ini dapat dilihat pada Gambar 15.



GAMBAR 15. (IMPLEMENTASI SCREEN DETAIL TASK)

Aplikasi juga menyediakan Screen Tab Career, Profile, dan Setting yang memudahkan pengguna untuk memantau progres karir, mengelola data profil, serta melakukan pengaturan aplikasi sesuai kebutuhan. Tampilan ketiga fitur ini dapat didukung dengan Gambar 16.



GAMBAR 16. (IMPLEMENTASI SCREEN CAREER, PROFILE DAN SETTING)

E. Testing

Pengujian aplikasi dilakukan menggunakan metode blackbox untuk memastikan bahwa setiap fitur berjalan sesuai dengan fungsinya tanpa memeriksa struktur internal kode. Pengujian ini dilakukan langsung pada lingkungan produksi (App Store) dengan total 16 test case utama yang mencakup seluruh fitur inti aplikasi, seperti autentikasi, pemilihan dan perubahan karir, pengelolaan tugas, akses resource eksternal, pembuatan catatan, dan pemantauan progres pengguna. Seluruh test case berhasil dijalankan dan dinyatakan lulus, menandakan bahwa hampir seluruh fitur utama berfungsi dengan baik dan sesuai spesifikasi. Di bawah merupakan rangkuman hasil pengujian menggunakan blackbox testing

TABEL 3.  
(TEST REPORT MENGGUNAKAN BLACKBOX)

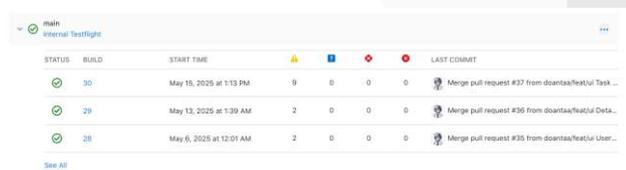
Item	Detail
Project	Beeginner version 1.0
Testing Period	5 Juni 2025 – 10 Juni 2025
Test Environment	Production (App Store)
Total Test Cases	16
Total Test Run	44
Passed	44
Failed	0
Anomalies Reported	2
Summary	Secara keseluruhan Test Case berhasil dijalankan. Namun Terdapat kasus dimana aplikasi force close dan ditemukan bahwa hal itu disebabkan ketika Aplikasi sedang menjalankan tugas yang berhubungan dengan data besar.

Isi Tabel 3 merinci hasil pengujian untuk setiap test case, mulai dari identifikasi item yang diuji, periode pengujian, lingkungan pengujian, hingga jumlah test case yang dijalankan, jumlah yang lulus, gagal, dan terblokir, serta anomali yang ditemukan. Dari 44 kali pengujian yang dilakukan terhadap 16 test case, seluruhnya dinyatakan lulus tanpa ada test case yang gagal atau terblokir. Namun, selama proses pengujian ditemukan dua anomali, salah satunya adalah aplikasi mengalami force close saat memproses data berukuran besar. Meskipun demikian, secara umum aplikasi dinyatakan stabil dan siap digunakan, dengan catatan bahwa anomali minor tersebut perlu diperhatikan dan diperbaiki pada pengembangan selanjutnya.

#### F. Cutover/Deployment

Proses yang dilakukan di tahap ini dilakukan dengan automasi menggunakan Xcode Cloud untuk mempermudah proses Integrasi dan Deployment ke Testflight menggunakan workflow. Workflow dipisah menjadi 2 bagian yaitu Internal Testing dan Eksternal Testing.

Workflow Internal Testing akan dijalankan ketika terjadi perubahan di branch main, kemudian Xcode Cloud akan melakukan testing, compile dan kemudian akan didistribusikan untuk pihak tester internal yaitu tim Developer. Berikut merupakan dokumentasi workflow yang telah dilakukan.



STATUS	BUILD	START TIME	LAST COMMIT
30	May 15, 2025 at 1:13 PM	9 0 0 0	Merge pull request #37 from doantaa/feat/Task...
29	May 13, 2025 at 1:39 AM	2 0 0 0	Merge pull request #36 from doantaa/feat/Data...
28	May 6, 2025 at 12:01 AM	2 0 0 0	Merge pull request #35 from doantaa/feat/User...

Gambar 17.

History Deployment menggunakan Internal Testflight Workflow

Sebelum aplikasi diajukan ke App Store, dilakukan persiapan metadata seperti deskripsi aplikasi, gambar layar, dan kata kunci sesuai persyaratan Apple. Dokumentasi lengkap juga disiapkan untuk memudahkan proses review oleh Apple. Setelah seluruh tahapan pengujian dan persiapan terpenuhi, aplikasi diajukan ke App Store Connect untuk proses review. Jika aplikasi lolos proses review, aplikasi akan tersedia secara publik di App Store dan dapat diunduh oleh pengguna secara luas. Dengan tahapan deployment yang terstruktur dan bertingkat ini, aplikasi dapat dipastikan layak,

stabil, dan siap digunakan secara optimal oleh pengguna akhir.

#### V. KESIMPULAN

Berdasarkan hasil analisis dan eksperimen yang telah dilakukan dalam tugas akhir ini, dapat disimpulkan bahwa rancang bangun aplikasi perencanaan karir berbasis iOS berhasil memenuhi tujuan penelitian, yaitu merancang dan mengembangkan aplikasi yang dapat membantu mahasiswa dalam melakukan perencanaan karir secara mandiri. Aplikasi ini menyediakan panduan terstruktur untuk mengenal dan mengeksplorasi karir sesuai dengan kompetensi dan aspirasi pengguna. Penerapan metode Rapid Application Development (RAD) terbukti efektif dalam mempercepat proses pengembangan serta memungkinkan penyesuaian fitur melalui iterasi dan umpan balik pengguna. Hasil pengujian menunjukkan aplikasi ini dapat memfasilitasi mahasiswa dalam merencanakan karir secara lebih terarah dan informatif. Dengan demikian, aplikasi yang dikembangkan telah mencapai tujuan yang dirumuskan dalam penelitian ini.

Untuk pengembangan penelitian dan pembuatan sistem ke depannya, disarankan agar dibuatkan sebuah sistem khusus untuk melakukan pengelolaan konten dalam aplikasi agar lebih fleksibel dikarenakan saat ini pengelolaan dilakukan menggunakan firebase. Selain itu, sistem sebaiknya terus menyesuaikan diri dengan perkembangan teknologi terbaru, seperti integrasi kecerdasan buatan dan fitur personalisasi yang lebih canggih. Perhatian juga perlu diberikan terhadap kelemahan-kelemahan yang ditemukan selama pengembangan, seperti keterbatasan pada cakupan data karir dan performa aplikasi pada perangkat tertentu. Dengan perbaikan pada aspek-aspek tersebut, diharapkan aplikasi perencanaan karir berbasis iOS ini dapat memberikan manfaat yang lebih luas dan optimal di masa mendatang.

#### REFERENSI

- [1] D. Masturina, "Pengaruh Kompetensi Diri Dan Kepercayaan Diri Terhadap Perencanaan Karir," *PSIKOBORNEO: Jurnal Ilmiah Psikologi*, vol. 6, no. 2, pp. 198–205, 2018, Accessed: May 19, 2025. [Online]. Available: <http://dx.doi.org/10.30872/psikoborneo.v6i2.4558>
- [2] G. Zhang, "A study of career planning and autonomy among university students: an investigation from the higher education learning crisis," 2023.
- [3] J. Martin, *Rapid application development*. USA: Macmillan Publishing Co., Inc., 1991.
- [4] B. James Martin *et al.*, "Rapid Application Development," 1991.
- [5] D. Brinendo, A. M. Mayestino, J. Arief, R. Hakim, and S. Surabaya, "Rancang bangun aplikasi pengaduan pelanggan menggunakan metode RAD (rapid application development)," 2024.
- [6] A. Rizkiriani, R. Dianah, and A. Kartinawati, "Development of an IOS-Based Baby Meal Planner Application," *E3S Web of Conferences*, vol. 454, May 2023, doi: 10.1051/e3sconf/202345403008.
- [7] W. Goodwill James and Matlock, "The Swift Programming Language," in *Beginning Swift Games*

*Development for iOS*, Berkeley, CA: Apress, 2015, pp. 219–244. doi: 10.1007/978-1-4842-0400-9\_17.

- [8] F. Sholichin, S. A. Halim, P. Quality, M. Adham, and M. Firdaus Bin Harun, “Review of iOS Architectural Pattern for Testability, Modifiability, and Performance Quality REVIEW OF IOS ARCHITECTURAL PATTERN FOR TESTABILITY,” *Article in Journal of Theoretical and Applied Information Technology*, vol. 15, p. 15, 2019, [Online]. Available: <https://www.researchgate.net/publication/335192719>
- [9] H. A. Epiloksa, D. S. Kusumo, and M. Adrian, “Effect Of MVVM Architecture Pattern on Android Based Application Performance,” *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 4, p. 1949, Oct. 2022, doi: 10.30865/mib.v6i4.4545.
- [10] A. Rahman Fajri and S. Rani, “Penerapan Design Pattern MVVM dan Clean Architecture pada Pengembangan Aplikasi Android (Studi Kasus: Aplikasi Agree Partner).”
- [11] S. K. Dirjen, P. Riset, D. Pengembangan, R. Dikti, and I. Firman Maulana, “Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 5, pp. 854–863, Oct. 2020, doi: 10.29207/RESTI.V4I5.2232.
- [12] C. Eidhof and F. Kugler, “Thinking in SwiftUI.”
- [13] “SwiftUI Documentation.”