



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN BACKEND APLIKASI KARIR IO BERBASIS WEB
MENGUNAKAN NESTJS: STUDI KASUS PT. QUANTUM TEKNOLOGI
NUSANTARA**

TUGAS AKHIR

**SABIQ MUHAMMAD ANTEBING MAME
0110220153**

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI
MARET 2024**



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**RANCANG BANGUN BACKEND APLIKASI KARIR IO BERBASIS WEB
MENGUNAKAN NESTJS: STUDI KASUS PT. QUANTUM TEKNOLOGI
NUSANTARA**

TUGAS AKHIR

SABIQ MUHAMMAD ANTEBING MAME
0110220153

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI
MARET 2024**

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.



Nama: Sabiq Muhammad Antebing Mame

NIM: 0110220153

STT - NF

Depok, 01 Maret 2024

Tanda Tangan

A handwritten signature in black ink, appearing to be the name Sabiq Muhammad Antebing Mame, written in a cursive style.

Sabiq Muhammad Antebing Mame

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

Nama : Sabiq Muhammad Antebing Mame

NIM : 0110220153

Program Studi : Teknik Informatika

Judul Tugas Akhir : Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing I

Signed by HILMY ABIDZAR TAWAKAL
(QM9913)

Signed at Aug 9, 2024 14:20:14



(Hilmi Abidzar Tawakal, S.T., M.Kom)

Penguji I



(Nasrul, S.Pd.I., S.Kom., M.Kom)

Ditetapkan di : Depok

Tanggal : 25 Juli 2024

KATA PENGANTAR

Alhamdulillah rabbilalamin, puji dan syukur penulis panjatkan kepada Allah Subhanahu Wa Ta'Ala, karena atas berkat, rahmat dan hidayah-Nya, penulis dapat menyelesaikan tugas akhir ini tepat pada waktu yang ditentukan. Dalam penyusunan tugas akhir, penulis berinisiatif untuk menyelesaikan tugas akhir yang berjudul “Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara”. Penulisan dokumen tugas akhir dilakukan untuk memenuhi persyaratan studi S1 pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri dan memperoleh gelar Sarjana Komputer. Penulis menyadari bahwa tanpa doa, bantuan, serta bimbingan berbagai pihak, dari masa perkuliahan hingga tiba penyusunan tugas akhir, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini tepat pada waktu yang ditentukan. Oleh karena itu, atas segala bantuan yang diberikan, penulis menyampaikan terima kasih kepada:

1. Allah Subhanahu Wa Ta'Ala
2. Kepada Bapak dan Ibu, serta semua anggota keluarga yang senantiasa memberikan bantuan doa, arahan, serta dukungan baik kepada penulis untuk menyelesaikan tugas akhir ini.
3. Ibu Nurul Janah, S.IIP., M.Hum., ibu dosen yang memberikan pengarahan kepada dan membimbing mahasiswa/i nya untuk menyelesaikan tugas akhir dengan lebih maksimal
4. Bapak Zaki Imaduddin, S.T., M.Kom., bapak dosen bimbingan akademik kelas TI05 2020 Sekolah Tinggi Teknologi Terpadu Nurul Fikri
5. Bapak Hilmi Abidzar Tawakal, S.T., M.Kom., bapak dosen pembimbing tugas akhir yang senantiasa memberikan pengarahan dalam penyusunan tugas akhir
6. Bapak dosen penguji tugas akhir yang telah memberikan pengalaman nyata kepada penulis dalam penyusunan tugas akhir
7. Bapak dan Ibu Dosen Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah memberikan pengetahuan, ilmu dan pengalamannya serta bersedia untuk membimbing penulis selama masa perkuliahan berlangsung

8. Beserta pihak yang tidak dapat disebutkan seluruhnya oleh penulis

Dalam penulisan laporan tugas akhir ini, tentu saja banyak kesalahan dan kekurangan yang terjadi yang mungkin saja disebabkan atas keterbatasan kemampuan dan pengetahuan yang dimiliki penulis dan memang kesempurnaan itu hanya milik Allah Subhanahu Wa Taa'ala. Meskipun demikian, penulis telah berusaha mengupayakan yang terbaik untuk menyelesaikan penulisan tugas akhir ini dengan sebaik mungkin. Karena hal itu, apabila dalam penulisan tugas akhir ini masih terdapat kekurangan maupun kekeliruan, dengan rendah hati penulis sangat menerima masukan dan saran dari pembaca. Akhir kata, semoga kedepannya penulis dapat mengembangkan penelitian ini menjadi lebih baik, masukan dan saran dari pembaca mendapatkan keberkahan oleh Allah dan penulisan tugas akhir ini dapat membawa manfaat positif untuk pembaca.

Depok, 01 Maret 2024



Sabiq Muhammad Antebing Mame

STT - NF

**HALAMAN PERNYATAAN
PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN
AKADEMIS**

Sebagai civitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Sabiq Muhammad Antebing Mame

NIM : 0110220153

Program Studi : Teknik Informatika

Jenis karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty - Free Right*) atas karya ilmiah saya yang berjudul :

RANCANG BANGUN BACKEND APLIKASI KARIR IO BERBASIS WEB MENGGUNAKAN NESTJS: STUDI KASUS PT. QUANTUM TEKNOLOGI NUSANTARA beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini STT-NF berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

STT - NF

Dibuat di : Depok

Pada tanggal : 01 Maret 2024

Yang menyatakan,



(Sabiq Muhammad Antebing Mame)

ABSTRAK

Nama : Sabiq Muhammad Antebing Mame
NIM : 0110220153
Program Studi : Teknik Informatika
Judul : Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web*
Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi
Nusantara

Manajemen tugas adalah kunci keberhasilan dalam meningkatkan produktivitas dan efektivitas operasional perusahaan. PT. Quantum Teknologi Nusantara merupakan perusahaan layanan jasa teknologi yang menghadapi tantangan dalam pengelolaan manajemen tugas kantor hingga menyebabkan keterlambatan, kesulitan pelacakan tugas dan kesulitan mengevaluasi kinerja karyawan karena manajemen tugas tidak terintegrasi dengan baik. Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara diharapkan dapat mengembangkan backend aplikasi manajemen tugas dengan fitur penugasan, pelacakan, dan evaluasi tugas untuk memfasilitasi komunikasi dan kolaborasi tim. Aplikasi ini memiliki fitur yang mempermudah penyelesaian tugas, pelacakan tugas dan mengevaluasi kinerja karyawan secara *real-time* karena dikembangkan dengan *NestJS*, metode *scrum*, menggunakan metode pengujian *blackbox testing* dan *skala likert*. Adapun hasil analisis yang dilakukan bersama tim pengembang memperoleh hasil bahwa *backend* aplikasi Karir Io telah meningkatkan koordinasi, mempercepat pelaporan tugas, memperbaiki pengelolaan waktu serta penggunaan sumber daya berdasarkan kebutuhan perusahaan yang dibuktikan oleh nilai maksimum *skala likert* dengan persentase pengujian fungsionalitas sebesar 89.79% dari *interval* 80-100. Hasil inilah yang menunjukkan bahwa aplikasi Karir IO pada penelitian ini layak digunakan oleh PT. Quantum untuk pengelolaan manajemen tugas kantor.

Kata Kunci: Karir IO, Manajemen Tugas, *NestJs*, PT. Quantum Teknologi Nusantara, *Scrum*.

ABSTRACT

Name : Sabiq Muhammad Antebing Mame
NIM : 0110220153
Study Program : Informatics Engineering
Title : Design and Development of the Backend for KarirIo Web Application Using NestJs: Case Study PT. Quantum Teknologi Nusantara

Task management is key to success in enhancing a company's productivity and operational effectiveness. PT. Quantum Teknologi Nusantara, a technology service provider, faces challenges in managing office tasks, leading to delays, difficulty in tracking tasks, and difficulties in evaluating employee performance due to poorly integrated task management. The design and development of the backend for the Karir Io web-based application using NestJS: Case Study PT. Quantum Teknologi Nusantara aims to develop a task management backend application with features for task assignment, tracking, and evaluation to facilitate team communication and collaboration. This application includes features that simplify task completion, task tracking, and real-time employee performance evaluation because it is developed with NestJS, the Scrum methodology, and tested using black box testing methods and Likert scale assessments. The analysis results conducted with the development team indicate that the Karir Io backend application has improved coordination, expedited task reporting, and improved time management and resource usage based on the company's needs. This is evidenced by the maximum Likert scale value, with a functional testing percentage of 89.79% from an interval of 80-100. These results demonstrate that the Karir Io application in this study is suitable for PT. Quantum's office task management.

Keywords: *Karir IO, NestJS, PT. Quantum Teknologi Nusantara, Scrum, Task Management,*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN DEPAN	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	3
BAB II KAJIAN LITERATURE	5
2.1 Tinjauan Pustaka	5
2.1.1 PT. Quantum Teknologi Nusantara	5
2.1.2 Karir Io	6
2.1.3 Rancang Bangun	7
2.1.4 Manajemen Tugas	7
2.1.5 Penelitian Kualitatif	7
2.1.6 <i>Web Service</i>	8
2.1.7 <i>Rest API</i>	9
2.1.8 <i>Framework NestJS</i>	10
2.1.9 Postman	11
2.1.10 <i>Socket.IO</i>	11
2.1.11 Git	12

2.1.12	MySQL	13
2.1.13	<i>Model View Controller (MVC)</i>	13
2.1.14	<i>Unified Modelling Language (UML)</i>	14
2.1.15	<i>Metode Agile</i>	16
2.1.16	<i>Scrum</i>	17
2.2	Testing	19
2.2.1	<i>Black Box Testing</i>	20
2.2.2	<i>Skala Likert</i>	21
2.3	Penelitian Terkait	22
2.3.1	Penelitian Terkait	22
2.3.2	Posisi Penelitian	24
BAB III ANALISIS DAN PERANCANGAN		28
3.1	Tahapan Penelitian	28
3.2	Rancangan Penelitian	35
3.2.1	Jenis Penelitian	36
3.2.2	Metode Analisis Data	36
3.2.3	Metode Pengumpulan Data	36
3.2.4	Metode Pengujian	37
3.2.5	Metode Implementasi dan Evaluasi	42
3.2.6	Lingkungan Pengembangan	42
3.2.7	Peran Sumber Daya Manusia	42
BAB IV IMPLEMENTASI DAN EVALUASI		44
4.1	Analisis dan Perancangan	44
4.1.1	Analisis Sistem Berjalan	44
4.1.2	Analisis Kebutuhan Sistem	57
4.1.3	Perancangan Sistem	58
4.1.4	Perancangan Pengujian	72
BAB V KESIMPULAN DAN SARAN		86
5.1	Kesimpulan	86
5.2	Saran	86
DAFTAR REFERENSI		887
DAFTAR REFERENSI GAMBAR		92

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Rest API.....	9
Gambar 2.2 Metode Agile.....	16
Gambar 2. 3 Metode Scrum	17
Gambar 2. 4 <i>Black Box</i> Testing.....	20
Gambar 3.1 Tahapan Penelitian	28
Gambar 3. 2 Arsitektur Sistem.....	32
Gambar 4.1 Alur Aplikasi Karir IO.....	45
Gambar 4.1.3.1 Use Case Diagram.....	60
Gambar 4.1.3.2 Actity Diagram.....	61
Gambar 4.1.3.3 Sequence Diagram.....	63
Gambar 4.1.3.4 Class Diagram	66
Gambar 4.1.3.5 Entity Relationship Diagram (ERD) Karir IO.....	71

STT - NF

DAFTAR TABEL

Table 2.3.1 Penelitian Terkait	24
Table 2.3.2 Posisi Penelitian	27
Tabel 3.1 Rancangan Pengujian - <i>Black Box Testing</i>	37
Tabel 3.2 Rancangan Pengujian - <i>Skala Likert</i>	41



STT - NF

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, efisiensi manajemen pengelolaan tugas menjadi kunci keberhasilan bagi perusahaan untuk menyelesaikan pekerjaan yang harus diselesaikan dengan lebih cepat dan efektif [1]. PT. Quantum Teknologi Nusantara (PT. Quantum) adalah perusahaan yang bergerak di bidang layanan jasa teknologi informasi, saat ini sedang menghadapi tantangan dalam mengelola tugas yang efisien dan efektif. PT. Quantum membutuhkan fitur-fitur teknologi canggih untuk menunjang pekerjaan dan mempermudah pengelolaan tugas perusahaan agar kualitas pekerjaan dipertahankan. PT. Quantum memutuskan merancang bangun aplikasi *web* yang dapat mempermudah pengelolaan manajemen tugas kantor, aplikasi ini kemudian diberi nama Karir Io.

Pada aplikasi Karir Io, arsitektur yang digunakan yaitu arsitektur *client-server*, dimana *client* dan *server* ditangani oleh *NestJS*. *NestJS* adalah *framework* berbasis *javascript* yang dibangun di atas *framework express.js* untuk pengembangan aplikasi *backend* terstruktur dan dapat diskalakan dengan beberapa layanan (*service*) terpisah atas area tertentu. *Database MySQL* adalah jenis *database* penyimpanan data aplikasi. Arsitektur *client-server* memisahkan tugas *client* dan *server*, sehingga skalabilitas, performa, dan keamanan aplikasi dapat meningkat. Adapun proses integrasi *backend* dilakukan melalui *framework NestJS*, menggunakan teknologi modern *AJAX* dan *WebSocket* untuk memastikan komunikasi *client* dan *server* dapat bergerak cepat [2].

Selain pemilihan *framework*, penelitian ini menggunakan metode *Agile* melalui pendekatan *Scrum* untuk mengelola proses pengembangan aplikasi Karir Io dan untuk memungkinkan pengembang aplikasi mampu bekerja secara fleksibel, mandiri, aktif dan memiliki kemampuan adaptasi cepat terhadap perubahan kebutuhan atau kondisi pasar yang berkembang saat ini. *Scrum* menjadi bagian dari *framework Agile* yang memfasilitasi tim pengembang aplikasi dalam melakukan iterasi cepat dan efisien melalui *sprint* yang terorganisir [3]. Metode *scrum* mudah dipahami karena mengatur penilaian aplikasi secara mandiri yang dikembangkan dalam periode pengembangan yang telah disepakati, memastikan pengembangan Karir Io berjalan responsif terhadap masukan, saran dan perubahan selama proses pengembangan aplikasi ini dilakukan.

Aplikasi Karir Io yang dikembangkan pada penelitian ini, melalui tahapan pengujian bersama dengan *Quality Assurance* (QA) berdasarkan setiap fitur yang terdapat pada aplikasi ini, tim penguji aplikasi menggunakan penelitian kualitatif dari wawancara dan survei sebagai metodologi pengujian. Karir Io yang dikembangkan ini, diharapkan dapat mengotomatisasi jadwal kerja PT. Quantum, mempermudah berbagai pihak dalam melakukan manajemen tugas, dan menjadi salah satu platform pelaksanaan tugas-tugas kantor PT. Quantum sesuai dengan jadwal yang ditentukan. Melalui pengembangan aplikasi Karir Io, penulis berharap pengelolaan sumber daya manusia (SDM), produktivitas karyawan, serta tugas yang harus di selesaikan ini pada akhirnya dapat dikelola baik, terstruktur dan lebih efektif [4]

Keberadaan aplikasi Karir Io tidak hanya untuk mengatasi permasalahan yang terjadi dalam mengelola manajemen tugas di PT. Quantum saja, aplikasi Karir Io juga dimanfaatkan untuk memberikan wawasan nyata kepada penulis tentang bagaimana pengembangan teknologi modern seperti *NestJs* dan metode *Agile* melalui pendekatan Scrum dapat diaplikasikan dengan baik sehingga efisiensi operasional dalam konteks bisnis teknologi dapat meningkat [5]. Penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap pengembangan aplikasi lainnya yang menggunakan *framework* yang dipilih agar menjadi efisien dan efektif, serta memberikan inspirasi penerapan manajemen tugas di lingkungan perusahaan lain yang membutuhkannya. Harapannya, aplikasi manajemen tugas ini dapat memberikan banyak kesempatan dan waktu untuk tim pengembang menyelesaikan pekerjaan perusahaan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas tersebut, maka didapatkan rumusan masalahnya yang akan diselesaikan dalam penelitian ini, diantaranya yaitu:

1. Bagaimana mengembangkan *backend* aplikasi Karir Io, aplikasi manajemen tugas kantor menggunakan *NestJS* dan metode pengembangan *scrum*?
2. Apakah aplikasi Karir IO yang dikembangkan dapat berjalan baik dan sesuai dengan kebutuhan pengguna?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah tersebut, tujuan yang akan dicapai dalam penelitian ini diantaranya yaitu:

1. Mengembangkan *backend* aplikasi Karir Io untuk melakukan pengelolaan manajemen tugas kantor dengan NestJS dan metode pengembangan *scrum*
2. Menguji aplikasi Karir IO yang dikembangkan telah berjalan baik dan sesuai dengan kebutuhan pengguna

1.4 Manfaat Penelitian

Selain memiliki tujuan penelitian, Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara memiliki manfaat penelitian untuk digunakan, manfaat dalam penelitian ini yaitu:

1. Memberikan solusi teknologi nyata untuk melakukan pengelolaan manajemen tugas kantor di PT. Quantum Teknologi Nusantara
2. Manfaat nyata untuk penulis karena menyelesaikan salah satu persyaratan lulus mata kuliah tugas akhir Sekolah Tinggi Teknologi Terpadu Nurul Fikri

1.5 Batasan Masalah

1. Penelitian ini dibatasi hanya terfokus pada pengembangan *backend* aplikasi Karir Io menggunakan *framework NestJs* untuk pengelolaan manajemen tugas kantor dan mengesampingkan *interface* aplikasi Karir Io yang mungkin juga menjadi kebutuhan yang harus diperhatikan.
2. Aplikasi Karir Io akan dikembangkan dan diuji khusus untuk kebutuhan PT. Quantum Teknologi Nusantara, sehingga akses pengguna dibatasi hanya dapat diakses oleh karyawan menggunakan *IP* kantor atau jaringan *WIFI* kantor saja.

1.6 Sistematika Penulisan

Dalam konteks penelitian Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs* di PT. Quantum Teknologi Nusantara yang dibahas dalam kesempatan ini, sistematika penulisan yang disusun penulis diantaranya yaitu:

1. **BAB I PENDAHULUAN**, bab pembuka yang penting dan mendapat perhatian khusus pada penelitian ini karena memberikan gambaran umum terkait hal-hal yang melatarbelakangi kebutuhan penelitian, membahas rumusan masalah, tujuan penelitian untuk menjawab rumusan masalah yang akan diselesaikan, manfaat penelitian untuk menguraikan dampak positif dan kontribusi nyata dari penelitian, batasan masalah yang membatasi cakupan pembahasan penelitian serta sistematika penulisan sebagai panduan dalam menguraikan pembahasan.

2. **BAB II KAJIAN LITERATUR**, berisi penjelasan mengenai definisi dan teori yang relevan dan berhubungan dengan penelitian, menyertakan kajian literatur sebagai bahan referensi utama dalam menyusun hasil penelitian, serta posisi penelitian yang sedang dikembangkan berdasarkan penelitian terkait yang relevan dengan penelitian ini dan dilakukan sebelumnya oleh ahli terdahulu.
3. **BAB III ANALISIS DAN PERANCANGAN**, mendetailkan alur tahapan penelitian untuk mempermudah proses analisis sistem yang diimplementasikan yaitu aplikasi Karir Io. Bab ini menganalisis proses bisnis perancangan aplikasi Karir Io, menguraikan alur penelitian, mendesain rancangan penelitian, serta melakukan analisis sistem terintegrasi. Bab III ini juga mencakup pembahasan rancangan pengujian untuk verifikasi prototipe yang dikembangkan.
4. **BAB IV IMPLEMENTASI DAN EVALUASI**, membahas implementasi dan evaluasi penelitian, menguraikan hasil dari pengujian sistem *backend* aplikasi Karir Io dan melakukan evaluasi terhadap efektivitas sistem aplikasi untuk mengetahui sejauh mana Karir Io diimplementasikan di PT. Quantum dan analisis tentang kekuatan serta kelemahan dari sistem yang dihasilkan.
5. **BAB V KESIMPULAN DAN SARAN**, berisi kesimpulan yang langsung menjawab point rumusan masalah dan tujuan dari penelitian ini. Berdasarkan analisis dan temuan yang diperoleh, bab ini juga menyediakan saran yang relevan untuk penelitian selanjutnya, yang bisa dilakukan untuk memperluas penelitian ini atau mengatasi kelemahan yang masih ada. Saran-saran ini diharapkan dapat memberikan arah yang jelas untuk penelitian di masa depan dan pengembangan lebih lanjut dari aplikasi atau teknologi yang telah diuji.

STT - NF

BAB II

KAJIAN LITERATUR

Kajian Literatur penelitian ini menyajikan uraian mendetail tentang definisi dan teori relevan, yang mendukung kerangka teoretis penelitian. Bab ini menguraikan teori-teori yang berkaitan langsung dengan topik penelitian, serta menyertakan analisis literatur yang menjadi bahan referensi utama dalam menyusun dan memformulasikan hasil penelitian. Selain itu, bab ini menggambarkan posisi penelitian yang berkaitan dengan penelitian terdahulu dengan penelitian yang sedang dikembangkan saat ini, sehingga kontribusi penelitian yang dilakukan ini memperoleh jangkauan lebih luas.

2.1 Tinjauan Pustaka

2.1.1 PT. Quantum Teknologi Nusantara

PT. Quantum Teknologi Nusantara adalah perusahaan layanan jasa teknologi informasi yang beralamat di Graha BIPJI. Gatot Subroto No.23, RT.2/RW.2, Karet Semanggi, Kecamatan Setiabudi, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12930. Perusahaan ini bekerjasama dengan 100+ mitra, *support* 273.5 juta populasi, membangun 10+ produk dan sistem terintegrasi sebanyak 250 dengan waktu relatif tinggi yakni 99.9%, bekerjasama dengan beberapa mitra nasional dan bekerjasama dengan beberapa mitra internasional terkenal. PT. Quantum bekerjasama dengan mitra nasional Kemenkes Indonesia, Kepolisian NKRI, Satgas Penanganan *Covid-19*, dan mitra lainnya. PT. Quantum juga bekerjasama dengan mitra internasional Postcard, Melengo, Rain dan perusahaan internasional lainnya. PT. Quantum berkomitmen pada visi dan misi jelas untuk mendukung upaya pemerintah mencapai digitalisasi efektif dengan berbagai proyek signifikan dan telah membangun berbagai aplikasi.

Portofolio konsumen PT. Quantum ini meliputi Unlimited, komunitas desain-sebagai-layanan (*Design-as-a-Service, DaaS*) pertama dari jenisnya, dan Kenangan, yang membuka era baru dalam social commerce. Selain itu, PT. Quantum telah mengembangkan IklanLive, sebuah platform iklan universal yang melayani pembuat konten langsung di berbagai platform. Semua produk yang di kembangkan didukung oleh inovasi dan keunggulan yang menarik. Berikut contoh aplikasi PT. Quantum:

1. Kenangan, aplikasi perdagangan sosial yang memfasilitasi pengalaman pribadi antar pengguna, komunitas, dan merek.

2. Meepo, pasar desain terpadu yang menawarkan desain grafis yang terjangkau.
3. Jualin, yang membantu pengguna mengelola iklan digital di berbagai platform media sosial melalui *influencer*.
4. Nusa, menyediakan serangkaian solusi pengembangan perangkat lunak termasuk kerangka kerja untuk aplikasi super dan *gateway* pembayaran.
5. Kelas Kode, kamp pelatihan pengkodean untuk lulusan baru dan peralihan karier, dipimpin oleh mantan insinyur Google dan Microsoft.
6. QuantumByte, aplikasi perdagangan sosial lain yang mengutamakan pengalaman pribadi antara pengguna, komunitas, dan merek.
7. UOS, sistem operasi terpadu yang membantu pemimpin dalam manajemen tugas dan meningkatkan akuntabilitas tim [6].

2.1.2 Karir Io

Karir Io adalah aplikasi manajemen tugas kantor untuk meningkatkan efisiensi dan efektivitas operasional di PT. Quantum. Aplikasi ini dirancang dengan *framework NestJs* melalui struktur modular dan efisien, sehingga ideal untuk pembangunan aplikasi *web* ber-skala besar. Karir Io juga memungkinkan pengguna mengelola tugas, melacak kemajuan, dan melakukan evaluasi kinerja secara *real-time*. Aplikasi Karir Io akan membantu pelacakan tugas secara *real-time* karena menyediakan fitur pelacakan tugas yang memungkinkan *project manager* monitoring status tugas secara aktual, memudahkan penyesuaian sumber daya dan prioritas yang cepat. Dan menerapkan evaluasi kinerja yang efektif, dimana aplikasi Karir Io ini menyediakan alat untuk evaluasi kinerja yang membantu manajemen dalam membuat keputusan berbasis data mengenai pengembangan karyawan dan alokasi sumber daya.

Proses pengembangan Karir Io menerapkan metode *Agile* melalui pendekatan *Scrum* yang memfasilitasi pengembangan aplikasi untuk mendukung alur pertumbuhan karir seluruh karyawan PT, Quantum dengan menyediakan visibilitas lebih besar atas kontribusi individu dan kemajuan proyek karena Karir Io menjadi langkah digitalisasi operasional di PT. Quantum untuk menggantikan sistem manual yang rentan terhadap kesalahan dan inefisiensi menjadi mudah. Dengan demikian, Karir Io berperan pada transformasi digital perusahaan dan peningkatan manajemen tugas serta proses bisnis. Dan dengan Karir Io, PT. Quantum dapat mengoptimalkan segala komunikasi dan kolaborasi yang dapat mempercepat proses pengembangan dan produktivitas kerja.

2.1.3 Rancang Bangun

Rancang bangun adalah suatu istilah pengembangan sistem untuk mendesain, merancang dan membuat suatu aplikasi dari hal yang bersifat *abstrak* hingga sampai aplikasi digunakan oleh pengguna [7]. Rancang bangun adalah 2 kata gabungan yang menerjemahkan hasil analisa ke dalam bentuk perangkat lunak untuk menciptakan sistem atau memperbaiki sistem yang sudah ada berdasarkan serangkaian prosedur perancangan sistem yang telah ditentukan [8]. Rancang bangun dalam penelitian ini adalah tahapan awal yang dilakukan oleh penulis mulai dari membuat skema gambaran rangkaian sistem yang belum pernah ditampilkan sama sekali kepada masyarakat menjadi suatu aplikasi yang memberikan manfaat bervariasi dan terstruktur.

2.1.4 Manajemen Tugas

Manajemen tugas adalah proses dan upaya pengelolaan tugas dan pemantauan tugas secara rutin oleh tim pengembang melalui berbagai tahapan seperti perencanaan, pelacakan, pelaporan dan evaluasi. Manajemen tugas melibatkan kinerja tim agar memastikan tim efektif, tugas ter-*manage* baik, alur kerja efisien, keputusan akhir pengembangan terhadap perubahan terjadi secara *real-time* dan diputuskan dengan cepat serta pengelolaan anggaran biaya terbentuk dengan sesuai [9]. Manajemen tugas dapat membantu pengelolaan waktu pengembangan sistem, sumber daya, cakupan dan pengulangan sistem terpenuhi karena penyelesaian tugas akan dilakukan berdasarkan tanggal mulai dan tanggal akhir pengembangan sistem. Tugas yang di *manage*, akan membantu proses perencanaan sistem yang ditentukan berjalan lancar, pemantauan kinerja mudah dilakukan, dan penutupan sistem berhasil mencapai sasaran akhir sesuai dengan konsep dan tenggat waktu yang tepat karena memiliki skala prioritas [10].

2.1.5 Penelitian Kualitatif

Penelitian kualitatif merupakan metode data yang digunakan dalam penelitian yang berfokus kepada peristiwa alami yang dialami responden karena menggambarkan dan menginformasikan peristiwa, fenomena serta situasi sosial yang akan dianalisis. Data kualitatif bersifat nyata, subjektif dan interaktif dengan partisipan, kualitatif menjadi prosedur penelitian yang menghasilkan data deskriptif berupa kata tertulis maupun lisan dari responden serta perilaku yang diamati kemudian disajikan dalam bentuk naratif dan terkonsep. Dengan kualitatif, strategi pencarian makna, konsep, karakteristik, hingga deskripsi fenomena dan kasus bisa ditampilkan dengan baik [11].

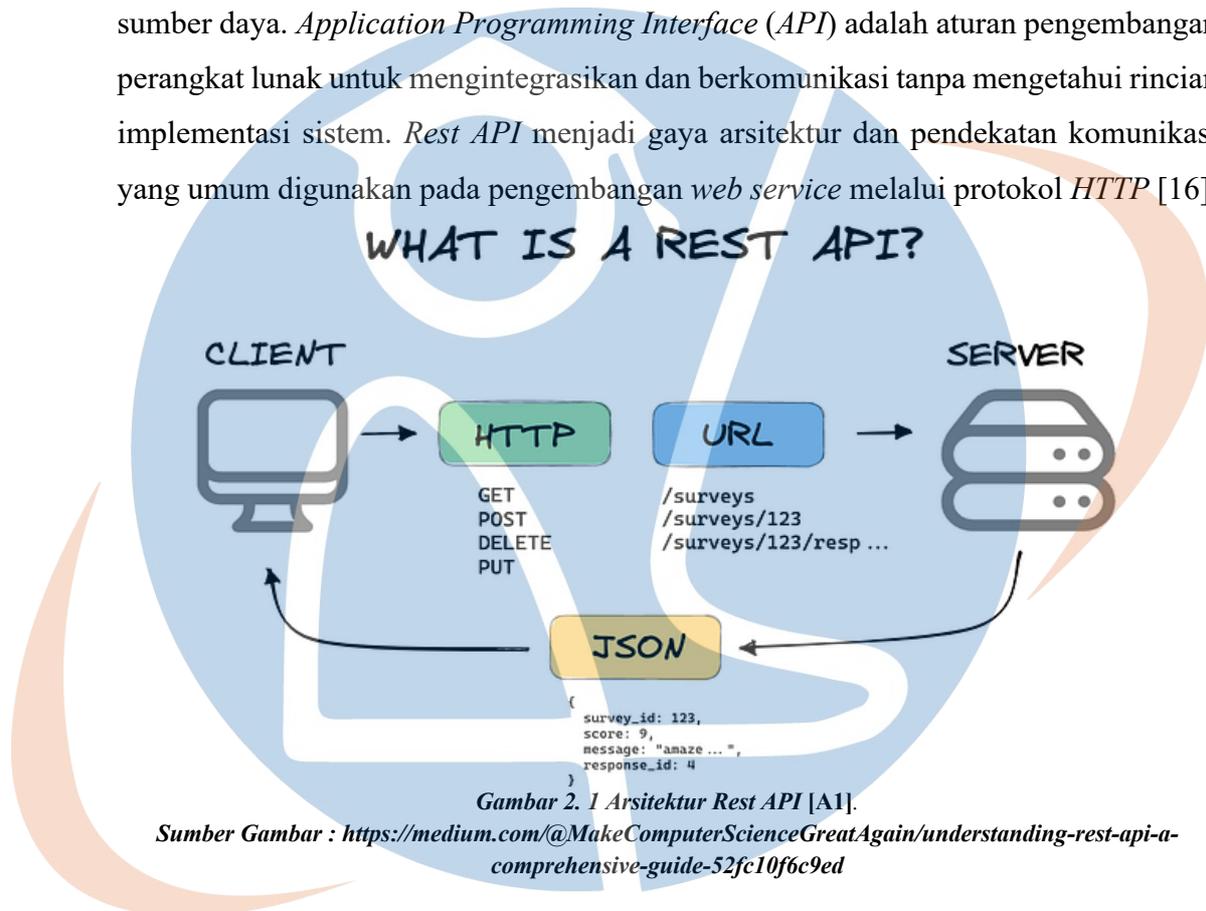
Penelitian kualitatif memiliki berbagai macam pendekatan yang bisa digunakan untuk menyesuaikan subjek dalam penelitian agar data-data yang dianalisis dapat di narasi kan dengan baik dan memperoleh hasil riset yang layak dan sesuai dengan situasi yang terjadi. Kualitatif menggunakan analisis mendalam sebagai fokus penelitian dengan latar alamiah guna membuat kejadian nyata yang mudah di mengerti dan memperoleh hipotesis terbaru [12]. Ciri-ciri utama penelitian kualitatif dapat di lihat dari prinsip penelitian yang mendalkan aspek deskriptif terhadap data-data yang diperoleh di lapangan. Karakteristik kualitatif memiliki lima aspek utama dimana penlitian kualitatif yang bersifat alamiah, kualitatif bersifat deskriptif, kualitatif yang mendahulukan alur proses penelitiannya, kualitatif yang bersifat deduktif dan kualitatif yang lebih menekankan kepada dimensi makna sebagai ciri dan karakter khususnya. Keakuratan hasil analisis metode kualitatif bergantung kepada penerapan karakteristik metode yang tepat serta hasil nyata yang diperoleh dari lapangan terjadinya peristiwa yang sedang dianalisis beserta hipotesis nyata terhadap peristiwa tersebut [13].

2.1.6 Web Service

Web service adalah sistem teknologi yang terhubung ke jaringan internet dan menjadi solusi nyata untuk mengoptimalisasikan kinerja *website* dalam perangkat internet. *Web service* menjembatani informasi tanpa mempermasalahkan perbedaan teknologi yang digunakan dari setiap sumber daya yang memungkinkan penggunaanya melakukan pertukaran data ke berbagai sistem untuk mengatasi permasalahan yang dialami [14]. Dengan *web service*, sekumpulan basis data dan perangkat lunak atau bagian dari program perangkat lunak seperti aplikasi maupun *website* dapat diakses dengan banyak protokol tanpa menggunakan banyak ruang dan waktu. Menggunakan *web service* yang terhubung ke internet, konektivitas di seluruh dunia bisa terhubung dengan lebih baik. Pada umumnya, *web service* memiliki ciri khusus berupa *URL* yang hanya menyampaikan informasi, perintah, dan konfigurasi untuk membangun fungsi tertentu dari aplikasi. Seiring dengan perkembangan teknologi yang terjadi, *web service* yang kerap disebut sebagai sekumpulan fungsi dan *method* yang berstandar protokol terbuka pada sebuah server untuk memungkinkan pertukaran data aplikasi dan sistem yang berbeda agar berjalan baik dan sesuai dengan bahasa pemrogramman yang digunakan. *Web service* memberikan keuntungan agar suatu program aplikasi dapat terhubung melalui pertukaran *web service* yang sesuai pada tiap perangkat [15].

2.1.7 Rest API

Rest API terdiri dari 2 kata utama yaitu *REST* dan *API*. *Representational State Transfer (REST)* adalah gaya arsitektur desain *API web* untuk berkomunikasi nyata dan merepresentasikan sumber daya dalam bentuk sederhana berformat *JSON* atau *XML*, mengusung *HTTP* dan menyediakan fitur *GET*, *POST*, *UPDATE* dan *DELETE* serta penggunaan *Uniform Resource Identifier (URI)* yang mempermudah identifikasi sumber daya. *Application Programming Interface (API)* adalah aturan pengembangan perangkat lunak untuk mengintegrasikan dan berkomunikasi tanpa mengetahui rincian implementasi sistem. *Rest API* menjadi gaya arsitektur dan pendekatan komunikasi yang umum digunakan pada pengembangan *web service* melalui protokol *HTTP* [16].



Pada **Gambar 2. 1**, *Rest API* memungkinkan *client request* data ke *server*, melakukan komunikasi dan betukar data *JSON* atau *XML* untuk mencapai *HTTP* yang menyediakan *GET* guna menerima data *server* tanpa mengubah data, *method POST* untuk mengirimkan data ke *server* dengan sumber daya baru, *Delete* untuk hapus data dan *PUT* kirim data ke *server* untuk memperbarui sumber daya dan membuat sumber daya baru [17]. Dalam penggunaannya, *URL* membutuhkan server agar mengirimkan *response* sesuai standar pengembangan, *API* sebagai penghubung klien, dan *server* merujuk pada arsitektur pengembangan yang membangun sistem terdistribusi secara sederhana melalui interaksi aplikasi *web* yang mendukung teknologi *caching* agar performa yang dihasilkan meningkat dan arsitektur *API* mampu berjalan baik [18].

2.1.8 Framework NestJS

Framework NestJS adalah kerangka kerja berbasis *server-side (backend)* yang bersifat efisien karena mempunyai konsep arsitektur yang terinspirasi dari *Angular* dan menggunakan *dependency injection, module, controller* untuk penerapan *middleware, pipes, guards, interceptors* dan lainnya yang *supportif*. Jika menggunakan berbagai basis data yang tersedia seperti *MySQL, PostgreSQL* dan *MongoDB* [19]. *Framework NestJS* memiliki pendekatan terintegrasi *TypeScript* yang menawarkan keamanan baik dan mendapat dukungan atas *Object-Oriented Programming (OOP)*.

Dalam perbandingan dengan *Express*, yang merupakan salah satu *framework backend JavaScript* yang paling populer dan menjadi standar mayoritas orang dalam pengembangan *Node.js*, *NestJS* menawarkan pendekatan terstruktur dan terorganisir. Meskipun *Express* cenderung lebih minimalis dan memungkinkan fleksibilitas yang tinggi dalam pembangunan aplikasi, *NestJS* menyediakan arsitektur yang lebih baku dengan konsep *dependency injection, module, dan controller* yang mirip dengan *framework frontend Angular*. Hal ini membuat *NestJS* lebih cocok untuk aplikasi berskala besar yang membutuhkan pengelolaan kode yang terstruktur dan terorganisir.

Dibandingkan dengan *Fastify* yang dikenal karena kecepatan eksekusi tinggi dan performanya, *NestJS* menawarkan fitur bawaan seperti *middleware, pipes, guards, dan interceptors* untuk mempercepat proses pengembangan. Meskipun *Fastify* adalah suatu aplikasi berforma tinggi, *NestJS* cocok untuk proyek dengan struktur terorganisir dan dukungan kuat untuk pengembangan aplikasi berskala besar. Jika dibandingkan oleh *HapiJS*, dengan pengembangan *API* konsisten dan mudah dibaca, *NestJS* ini juga memberikan pendekatan terintegrasi dengan *TypeScript* serta dukungan kuat untuk pembuatan aplikasi berbasis mikroservis. Meskipun *HapiJS* juga dapat digunakan untuk pembuatan aplikasi berbasis *mikroservis*, *NestJS* menyediakan lebih banyak fitur bawaan seperti *dependency injection, module, dan controller* yang mempercepat pengembangan aplikasi dan memungkinkan pengelolaan kode yang lebih terstruktur. Dan bila dibandingkan dengan *framework* lain, *NestJS* memiliki arsitektur berorientasi modul guna mengatasi *problem* arsitektur yang direktorinya asing dengan dukungan *library* yang tidak *initial project* sehingga mempermudah penghematan kapasitas *memory*, *NestJS* menggunakan *library* yang menghadirkan fitur *ORM* atau *Testing* dan *NestJS* untuk tim yang tidak mau belajar dan mendalami kegunaan *framework* ini.

Lingkungan *typescript* tidak cocok bila digunakan oleh tim yang belum pernah menggunakan bahasa program yang *strong typed* seperti *Java* karena *NestJS* bersifat *kompleks* [20]. Berdasarkan hal itu, penggunaan *framework* ini tepat untuk aplikasi berskala besar dan *kompleks* karena hanya perlu arsitektur yang solid, tepat untuk aplikasi modularitas berskala tinggi dan struktur yang jelas dan terorganisir sehingga pengelolaan kode proyek besar dapat terkelola dengan sesuai [21].

2.1.9 Postman

Postman adalah salah satu jenis alat testing *API* untuk menyederhanakan alur kerja sistem, membantu pengguna dalam membangun, menguji dan memodifikasi *API* karena menawarkan fitur dan fungsi penting sehingga penggunaan *postman* sederhana dan mudah. *Postman* digunakan untuk mengirimkan *request API* ke *server web* dan menerima segala jenis respon selayaknya tukang pos yang menerima surat untuk di antarkan. *Postman* mengakomodasi berbagai *request HTTP* yang menguji *API* dalam mengatur kerangka kerja saat mengirim dan menerima permintaan agar dikelola baik [22]. *Postman* menawarkan tampilan sederhana yang ramah pengguna untuk membuat permintaan *HTML* tanpa melalui kerumitan penulisan kode program.

Postman menawarkan pengelompokan permintaan berbeda yaitu *collection* untuk mengatur pengujian sistem aplikasi agar dapat dikonfigurasi sesuai dengan keinginan dan kebutuhan tim pengembang. *Postman* dapat membuat lingkungan berbeda melalui penggunaan alur variabel yang menargetkan lingkungan pengujian yang dikelola oleh *developer* dan *software quality assurance (SQA)* agar memenuhi permintaan pengujian. *Postman* memiliki fitur otomatisasi untuk mengatur rangkaian pengujian, menawarkan penggunaan aplikasi gratis dan versi berbayar dengan fitur-fitur profesional serta menyediakan banyak manfaat pada pengembangan sistem [23].

2.1.10 Socket.IO

Socket.IO merupakan *library JavaScript* untuk komunikasi *realtime*, responsif dan interaktif sesuai teknologi yang digunakan diantara klien dan *server* karena *Socket.IO* memanfaatkan protokol *WebSocket* dan *fallback* dengan teknik *polling* bila *WebSocket* tidak didukung oleh teknologi yang digunakan. Penggunaan *Socket.IO* memungkinkan pengiriman dan penerimaan data terjadi langsung disisi klien dan *server* agar *response* perubahan data diperoleh secara instan dan dapat digunakan oleh berbagai *browser*.

Socket.IO juga menggunakan platform yang memungkinkan model komunikasi berbasis *event* (*event-based communication*) agar klien dan *server* dapat melakukan pengiriman atau penerimaan *event* serta dirancang agar sistem mendapat skala di banyak koneksi [24]. *Socket.IO* juga memungkinkan komunikasi terjadi secara nyata, dua arah, dan berbasis *event* antara klien dan *server web*. Pustaka ini terdiri dari dua komponen yaitu klien yang berjalan di *browser* dan *server* pada sisi Node.js. *Socket.IO* populer untuk pengembangan aplikasi *web* yang memerlukan interaksi *real-time* dan pembaruan data langsung diantara *server* dan klien. Salah satu contoh penggunaan *library Socket.IO* digunakan dalam pengembangan *chat apps*, di mana latensi rendah dan komunikasi dua arah diperlukan untuk memberikan pengalaman pengguna. Untuk penggunaan *library* ini terbukti, ketika pengguna A mengirimkan suatu pesan, maka dengan memanfaatkan teknologi *Socket.IO* ini *server* akan segera meneruskan pesan ke pengguna B sesuai yang disampaikan pengguna A, tanpa *delay* yang berarti [25].

2.1.11 *Git*

Git adalah salah satu alat *version control system (VSC)* yang dimanfaatkan oleh tim pengembang aplikasi untuk mengelola berbagai perubahan kode yang tersedia pada suatu sistem. *Git* dimanfaatkan untuk pengelolaan perubahan kode dan pelacakan *source code* yang mempermudah proses perubahan kode dengan cabang yang terpisah dari versi utamanya, sehingga tim pengembang yang menggunakan *Git* melakukan perubahan kode tanpa mengganggu kode utama. *Distributed revision control* adalah aspek fundamental *Git*, dimana setiap salinan repositori *Git* tidak hanya salinan biasa, tetapi menjadi repositori penuh yang mencakup seluruh riwayat perubahan kode yang memungkinkan pengembangan sistem tanpa koneksi ke *server* pusat [26].

Git mempermudah kolaborasi banyak orang dalam mengerjakan proyek karena *Git* menjadi *tools* yang bersifat *opensource* sehingga seluruh tim dapat mengaksesnya dengan mudah. Dengan *Git*, penyimpanan proyek ke folder dapat terorganisir dengan mudah, *tools* ini menjadi layanan *hosting* yang fleksibilitas karena menjadi solusi *hosting* proyek yang tersedia. *Git* menjadi fitur *backup* data apabila terjadi kesalahan atau masalah pengembangan versi terbaru dan *Git* akan memproses pengembalian kode ke versi sebelumnya terlaksana dengan mudah dan kolaborasi berbagi kode antar anggota tim terlaksana. Kemampuan ini dapat mempermudah pengelolaan perubahan kode sumber yang efisien dan terdistribusi dalam pengembangan perangkat lunak [27].

2.1.12 MySQL

MySQL adalah sistem manajemen *database relational (RDBMS)* berbasis *SQL (Structured Query Language)* yang bersifat *open source*, tetapi juga menyediakan versi komersial yang diberikan oleh Oracle Corporation. MySQL populer di kalangan pengembang dan menjadi bagian integral dari *stack* teknologi LAMP (*Linux, Apache, MySQL, PHP/Python/Perl*). MySQL didirikan 3 orang Swedia yaitu David Axmark, Allan Larsson dan Michael Widenius, kemudian dikembangkan MySQL AB yang merupakan perusahaan asal Swedia tahun 1994, setelah melalui 1 th pengembangan, versi MySQL yang pertama stabil untuk terus digunakan sejak tanggal 23 Mei 1995 dan bertugas memberitahukan *server* terkait hal apa saja yang harus dilakukan [28]

MySQL menjadi pilihan *database* populer untuk berbagai tujuan karena fitur yang digunakan beragam dan dapat digunakan diberbagai platform *linux, windows, unix, os x*, dll. MySQL dikenal karena kecepatan eksekusi, performa tinggi, mudah digunakan, menawarkan mekanisme keamanan terenkripsi basis data dan kontrol akses pengguna serta didukung komunitas pengguna yang menyediakan dukungan dokumentasi untuk mempelajari *software* dan mengimplementasikannya. MySQL mendukung *stored procedures, triggers, views*, transaksi kompleks dan fitur lainnya. MySQL menjadi pilihan *database* populer untuk mencapai tujuan pembuatan dan pengelolaan *database*, penyimpanan data, transaksi *e-commerce*, pencatatan data dan *database* untuk *website* yang akan dikembangkan pada penelitian ini [29]

2.1.13 Model View Controller (MVC)

Model View Controller (MVC) adalah gambaran pola arsitektur aplikasi yang membantu pengembangan sistem dengan memisahkan kode program menjadi 3 bagian yaitu *model, view*, dan *controller* dimana diawali dengan *view* meminta data ditampilkan dalam bentuk grafis. Permintaan *view* akan diterima dan diproses oleh *controller* dan sistem akan meneruskan ke *model* untuk mengelolanya. Setelahnya *model* mencari dan mengolah data yang diminta pada *database* untuk di kirimkan kepada *controller*, setelah data diterima *controller* maka *controller* akan mengambil data pengolahan *model* dan mengaturnya kembali di bagian *view* untuk ditampilkan ke pengguna. Hal tersebut terjadi pada sistem yang menerapkan konsep MVC untuk mempermudah pengembangannya karena setiap data yang dikirimkan dikembalikan pada *view* untuk ditampilkan kepada pengguna aplikasi [30]

Berdasarkan pembahasan tersebut, *model* adalah bagian untuk mengelola dan berhubungan langsung dengan *database*, *view* sebagai bagian yang akan menyajikan tampilan desain aplikasi untuk menghasilkan beragam informasi yang dibutuhkan dan *controller* adalah penghubung *model* dan *view* untuk memenuhi setiap proses *request* yang diajukan *user*. *MVC* dapat menangani *error* atau *bug* pada sistem yang dikembangkan agar tim pengembang mengerjakan bagian *error* dahulu, *MVC* mudah dalam melakukan *testing*, bermanfaat untuk pengembangan *website*, mudah untuk proses *maintenance* aplikasi, memperbaiki penulisan kode program agar rapi, dan mempermudah pemeliharaan aplikasi agar tersusun dengan struktur yang baik [31]

2.1.14 Unified Modelling Language (UML)

UML adalah bahasa visual pengembangan sistem untuk memodelkan sistem dan dapat merepresentasikan berbagai aspek sistem seperti struktur sistem, perilaku, interaksi antara objek dan lingkungan pengoperasian. Grady Booch adalah ahli yang terlibat dalam pengembang *UML* dan mengatakan bahwa *UML* menjadi pemodelan visual sederhana ke kompleks untuk merepresentasikan sistem perangkat lunak dan menjadi pemodelan perangkat lunak untuk merancang berbagai diagram alur yang memfasilitasi pemahaman dan komunikasi struktur yang jelas serta perilaku sistem. *UML* dikembangkan Grady Booch, Ivar Jacobson, dan James Rumbaugh di Rational Software tahun 1994 dan dikelola oleh Object Management Group (OMG) [32]

UML digunakan untuk membuat diagram dan model yang mewakili sistem perangkat lunak dalam mendesain dan mengembangkan perangkat lunak, untuk menganalisis, merancang dan mengimplementasikan sistem, mampu meningkatkan komunikasi dan pemahaman mendalam melalui diagram yang mewakili berbagai aspek, memungkinkan penggunaan seluruh proses pengembangan perangkat lunak termasuk pengujian dan pemeliharaan sistem, menjadi bahasa pemodelan standar yang digunakan banyak industri dan membantu identifikasi masalah potensial di awal pengembangan *software* [33]. *UML* pengembangan aplikasi Karir Io terdiri dari beberapa diagram yang memberikan informasi pada aplikasi, seperti *use case*, *class*, *activity*, *sequence* diagram dll yang merepresentasikan aspek tertentu dari sistem agar lebih jelas dan sederhana sehingga produk akhir yang dihasilkan pada sistem dapat sesuai dengan ketentuan dan kebutuhan klien [34]. Berikut permodelan *UML* yaitu:

1. Use Case Diagram

Use Case Diagram menampilkan skenario penggunaan sistem, kebutuhan dan persyaratan pengguna yang memperlihatkan interaksi pengguna dengan sistem dalam situasi tertentu. Hal ini membantu pengembang mengerti dan memenuhi kebutuhan serta persyaratan, memastikan sistem berfungsi sesuai ekspektasi karena diagram akan memvisualisasikan hubungan interaktif antar sistem dan aktor. *Use Case* diagram juga mendetailkan jenis interaksi pengguna dengan sistem dan mudah dipelajari, *Use Case* sering kali menjadi titik awal dalam pemodelan yang memerlukan diagram untuk menggambarkan tindakan aktor beserta *respons* seperti pada *Use Case Diagram*.

2. Class Diagram

Class Diagram menggambarkan struktur sistem, mencakup kelas, atribut, dan metode yang dimiliki oleh kelas tersebut untuk menunjukkan hubungan antarkelas, seperti pewarisan dan asosiasi. Hal ini berguna untuk memodelkan struktur program serta memetakan hubungan antarkelas. Diagram ini dapat menampilkan kelas-kelas dalam kelompok paket yang memudahkan proses identifikasi pengaturan kelas agar berjalan sesuai kebutuhan yang diimplementasikan serta memfasilitasikan organisasi dan modularisasi kode, sehingga pengembangan dan pemeliharaan lebih mudah.

3. Activity Diagram

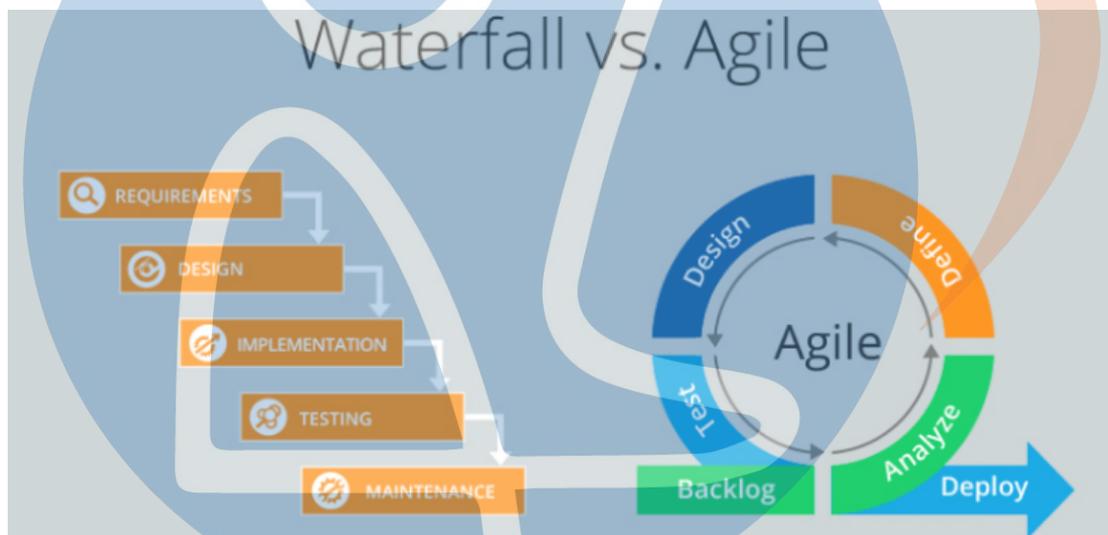
Activity Diagram merupakan diagram berjalannya sistem yang digambarkan secara vertikal dan menjadi contoh pengembangan *Use Case* yang menjelaskan alur kegiatan proses suatu sistem sehingga representasi alur kerja, termasuk tindakan dan keadaan selama proses agar berjalan baik. *Activity* diagram ini juga menunjukkan aktivitas dan tindakan yang harus dilakukan oleh sistem dalam menjalankan suatu proses sehingga dengan menganalisis diagram ini, tim pengembang dapat memahami alur kerja sistem dan mengidentifikasi masalah atau kesalahan dalam proses.

4. Sequence Diagram

Sequence Diagram digunakan untuk merepresentasikan interaksi objek dengan sistem, termasuk pesan dan *respons* yang dikirimkan karena diagram ini menunjukkan interaksi secara kronologis. Dengan *sequence* diagram, urutan pesan antara objek dan *respons* yang diberikan diperoleh mudah, sehingga tim pengembang dapat memahami interaksi dan memastikan sistem berfungsi. *Sequence* diagram juga dapat menjelaskan interaksi objek berdasarkan urutan waktu atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti yang tertera pada *Use Case* diagram.

2.1.15 Metode Agile

Metode *agile* adalah metode pengembangan yang berfokus pada fleksibilitas dan kecepatan manajemen proyek yang didasarkan pada iterasi pendek atau siklus pengembangan yang disebut dengan *sprint* dimana tim pengembang bekerja sesuai jangka waktu tetap (biasanya beberapa minggu) untuk mencapai tujuan. Metode *Agile* adalah respon kelemahan metode pengembangan lain yang kaku dan berisiko tinggi karena panjangnya siklus pengembangan. Metode *agile* ini seringkali disebut dengan *framework* gabungan berbagai kerangka kerja spesifik seperti *scrum* untuk memberikan panduan dan praktik konkret dalam pengembangan perangkat lunak. Metode ini membutuhkan alur kerja sesuai, peranan anggota tim, artefak yang akan dihasilkan serta metode komunikasi yang diterapkan selama pengembangan [35].



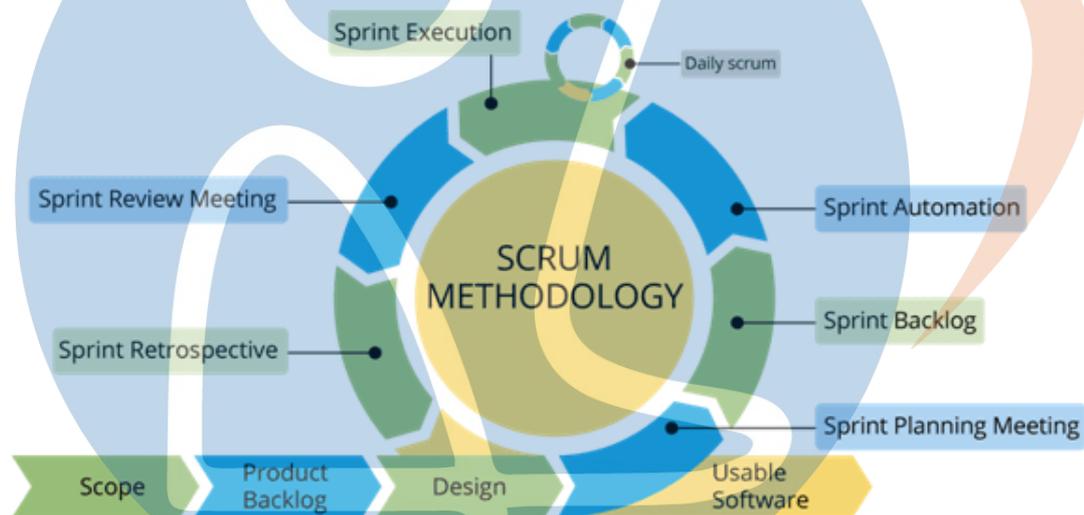
Gambar 2.2 Metode Agile [A2]

Sumber Gambar : <https://www.niagahoster.co.id/blog/agile-adalah/>

Gambar 2.2 adalah perbandingan *Agile VS Waterfall* dimana *agile* menjadi alternatif dari metode *waterfall* yang linear dan tidak dapat diubah di tengah proses pengembangan sistem. Metode *agile* menjadi landasan prioritas kepuasan konsumen tanpa menunggu waktu berbulan-bulan lamanya setelah proyek yang dikembangkan selesai. Tujuannya adalah menciptakan *software* yang berfungsi, bernilai jual tinggi dengan biaya pembuatan rendah, terbuka dalam mengatasi perubahan dan selalu memprioritaskan kepuasan konsumen melalui *software* yang dikembangkan. Dengan metode *agile*, diharapkan agar pribadi tim yang mau bekerjasama selama project dapat berjalan, project penuh motivasi terbtun, dan dapat menyampaikan informasi pada tim pengembang agar menjadi tolak ukur kesuksesan proyek dengan kesederhanaan [36].

2.1.16 Scrum

Scrum merupakan kerangka kerja *agile* untuk mengatur pengembangan sistem melalui *sprint* dan mempromosikan peran tertentu seperti *scrum master* dan *product owner*. Scrum adalah pendekatan yang cepat, efektif, fleksibel, iteratif, inkremental dan membutuhkan adaptasi tim untuk mengelola perubahan yang terjadi dalam menyelesaikan proyek dengan lingkungan yang berubah cepat dalam memberikan nilai selama pengembangan proyek sehingga kebutuhan pelanggan dapat menerapkan komunikasi transparan, tanggung jawab tim dapat terbentuk baik, dan pengembangan berkelanjutan berhasil diterapkan. Metode ini cocok untuk proyek kompleks dimana kebutuhan tidak sepenuhnya dipahami atau cenderung berubah-ubah [37].



Gambar 2.3 Metode Scrum [A3]

Sumber Gambar : <https://bif.telkomuniversity.ac.id/scrum-prinsip-agile-dan-tahapan-dalam-metode-scrum/>

Gambar 2.3 metode *scrum* dirancang dengan melalui beberapa metode untuk membantu tim pengembang mengelola proyek yang menggunakan metode *scrum* pada pengembangan sistem. Metode *scrum* unggul untuk digunakan dalam pengembangan sistem jika dibandingkan dengan penggunaan manajemen proyek tradisional karena metode ini memungkinkan tim pengembang mengambil keputusan dan melakukan perubahan *real-time* [38]. Berdasarkan **Gambar 2.3**, metode *scrum* yang digunakan melalui beberapa tahap seperti membuat *product backlog*, melaksanakan *sprint planning*, melakukan tahap *sprint*, menerapkan *sprint review* untuk memberikan masukan terhadap *sprint* yang dilakukan dan melaksanakan *retrospective process*. Adapun berikut adalah tahap yang harus dilakukan saat menggunakan metode *scrum*:

1. Product Backlog

Tahap ini diawasi oleh *Product Owner* atau *manager* proyek, yang bertanggung jawab menyusun daftar tugas berdasarkan prioritas kebutuhan perusahaan. Daftar ini dinamis; *Product Owner* bersama *stakeholder* harus memperbarui *product backlog* untuk menyesuaikan berbagai perubahan kondisi pasar, menghapus item yang tidak lagi relevan dalam pengembangan dan menambahkan kebutuhan baru yang muncul.

2. Sprint Planning

Selanjutnya adalah *sprint planning*, di mana item prioritas tinggi dari *product backlog* dipindahkan ke *sprint backlog*. Tim *scrum* menentukan strategi pelaksanaan tugas-tugas yang harus dilakukan berdasarkan dengan tahapan *sprint backlog*, atur alokasi sumber daya dan menetapkan *timeline* yang realistis untuk pengerjaan sistem.

3. Sprint

Fase ini dimulai setelah rencana telah ditetapkan, dengan tim melaksanakan tugas-tugas melalui pertemuan rutin oleh tim pengembang yang dikenal sebagai *daily scrum*. Dalam pertemuan ini, seluruh anggota tim harus melaporkan kemajuan dan mengidentifikasi hambatan, kendala atau kesalahan yang terjadi dan mungkin saja telah menghambat progres pengembangan sistem. *Scrum Master* memegang peran vital di sini, membantu tim untuk tetap fokus dan memecahkan masalah yang muncul.

4. Sprint Review

Di akhir *sprint*, tim pengembang harus tetap melaksanakan *sprint review* di mana produk atau proyek yang telah dikembangkan ini telah dipresentasikan pada pengguna. Tujuannya adalah untuk menilai hasil kerja pengembangan sistem dan memastikan semua fungsi telah berjalan seperti yang diharapkan. Ini juga menjadi kesempatan untuk menerima umpan balik yang digunakan untuk meningkatkan siklus berikutnya.

5. Retrospective Process

Tahap ini adalah tahapan dalam pengembangan sistem yang harus dilakukan yaitu tahap *retrospective*, di mana tim pengembang harus mengkaji kembali proses yang telah dilalui selama *sprint* berlangsung untuk mengetahui proses pengembangan. Tahap *retrospective* ini harus mengintrospeksi kesalahan pengembangan, memahami alur pengembangan, melakukan perbaikan sistem yang dikembangkan, dan bagaimana meningkatkan sistem agar dapat diimplementasikan dalam tahapan *sprint* berikutnya yang harus dilakukan oleh tim pengembang dalam proses pengembangan sistem.

Metode *scrum* pada umumnya bersifat iteratif dimana setiap pengalaman dari setiap sprint dipelajari dan diterapkan untuk meningkatkan efisiensi dan efektivitas dalam proyek-proyek mendatang. Dengan menerapkan setiap tahapan yang harus dilakukan dalam mengembangkan sistem secara disiplin dan konsisten, metode *scrum* memungkinkan organisasi untuk merespons dinamika pasar dengan lebih cepat dan efisien, seraya meningkatkan kolaborasi dan transparansi antar tim. Ini menciptakan lingkungan yang kondusif untuk inovasi dan perbaikan berkelanjutan. Metode *Scrum* menjadi teknik pengelolaan proyek yang terstruktur untuk dapat digunakan dalam mengembangkan sistem aplikasi dalam melalui beberapa fase tahapan yang dimulai dari pembuatan *product backlog* hingga pelaksanaan *sprint retrospective* [39].

2.2 Testing

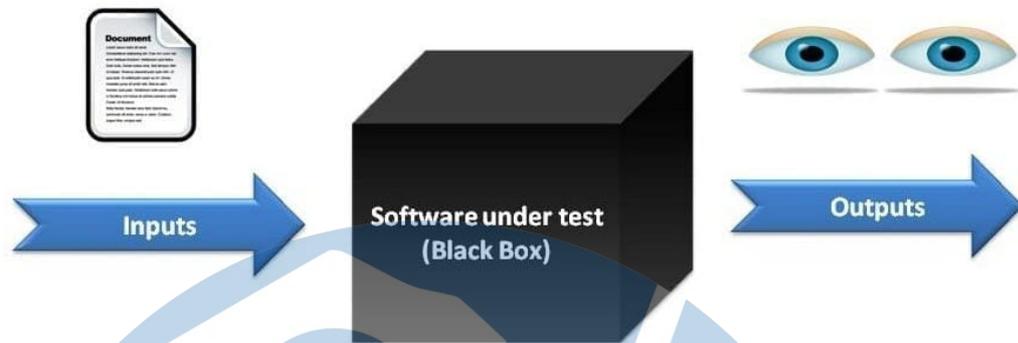
Testing disebut sebagai pengujian sistem dan evaluasi sistem perangkat lunak untuk menentukan apakah sistem yang dikembangkan berfungsi dan sesuai. *Testing* dilakukan untuk melakukan pengujian persyaratan atau spesifikasi yang telah ditetapkan untuk menemukan kesalahan atau kecatatan dalam perangkat lunak, terdapat jenis tes yang dapat dilakukan perangkat lunak seperti tes fungsional, tes integrasi, tes kinerja, tes keamanan dan penggunaan aplikasi. Dengan *testing*, kesalahan dan kecacatan sistem dapat ditemukan sehingga sistem dapat berkualitas tinggi, memenuhi kebutuhan pengguna untuk memastikan kualitas produk, menantikan keandalan sistem dengan pengujian efektif, dapat diandalkan di lingkungan produksi serta dapat memberikan penjamin keamanan untuk menghemat biaya dan waktu yang digunakan [40].

Testing menjadi serangkaian proses atau aktivitas yang dilakukan bersama oleh tim pengembang untuk melakukan verifikasi lanjutan apakah sistem telah berfungsi baik, sesuai persyaratan dan spesifikasi. Pada pengembangan *software*, pengujian dapat dilakukan dengan 3 teknik pengujian yaitu *black box testing*, *whitebox testing* dan *graybox testing* untuk memastikan fungsi pada aplikasi beroperasi dengan baik serta menjamin kualitas, keandalan, dan performa aplikasi sesuai dengan perencanaan. Melalui proses *testing*, kualitas produk dapat terjamin, keamanan *software* terjaga, kompatibilitas *software* terjamin, dan fitur pada aplikasi memenuhi persyaratan yang telah ditentukan serta bebas dari *bug* atau kesalahan yang bisa mempengaruhi kinerja sistem aplikasi. Oleh karena itu, *software testing* sangat penting untuk memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditetapkan [41]

2.2.1 Black Box Testing

Requirements Document

Validate output



Gambar 2. 4 Black Box Testing [A4]

Sumber Gambar : <https://fit.labs.telkomuniversity.ac.id/user-acceptance-test-uat/>

Berdasarkan Gambar 2. 4, dalam melakukan pengujian sistem aplikasi dapat dilakukan melalui beberapa jenis pengujian, dalam pengembangan aplikasi Karir IO ini, *black box* testing menjadi salah satu jenis pengujian program yang tidak terfokus kepada kinerja internal program saja tetapi pengujian ini berfokus pada kesesuaian fungsionalitas yang telah disediakan pada program terhadap proses bisnis organisasi dengan melibatkan pengguna nya tanpa mengetahui kode program [42]. *Black box* testing fokus pada pemeriksaan *input* dan *output* perangkat lunak tanpa memahami *detail internal* program karena akan menguji program sesuai dengan alur kerja fungsi.

Pengujian *black box* testing yang umum digunakan oleh tim pengembang hanya memperhatikan apa yang dimasukkan ke dalam sistem dan apa yang dihasilkan, tanpa mengetahui bagaimana proses *internal*-nya berlangsung. Dalam *black box testing*, tiga aspek utama yang diuji adalah antarmuka pengguna (*user interface*), *input*, dan *output* sistem. Meskipun dibuat untuk mencari kesalahan pada aplikasi, *black box testing* digunakan juga untuk melihat beberapa fungsi perangkat lunak yang dapat dioperasikan dengan baik agar *output* program tepat dan dapat dipelihara secara baik. Dengan *black box testing*, fokus utama adalah *input* dan *output* yang berinteraksi satu sama lain agar terfokus pada masukan dan keluaran program tanpa memperhatikan struktur *internal* atau *detail* implementasi perspektif. Dalam pengujian aplikasi, jenis *black box testing* memiliki 2 tahapan proses yang dapat dilakukan untuk memenuhi standarisasi sistem yaitu pengujian fungsional dan pengujian non-fungsional [43].

Black box testing tidak memerlukan pengetahuan teknis bahasa pemrograman, penguji sistem dengan tim *developer* dapat bekerjasama tanpa mengganggu tugas utama di pekerjaan masing-masing pengembang tanpa melakukan suatu pengecekan *structured code* berdasarkan perspektif pengguna untuk menemukan inkonsistensi. *Black box testing* dapat mengidentifikasi masalah atau *bug* sebelum diluncurkan yang dapat digunakan dengan 3 jenis pengujian utama seperti pengujian fungsional (*functional testing*), pengujian non-fungsional (*non-functional testing*), dan pengujian regresi (*regression testing*). Metode yang dilakukan tidak memerlukan keahlian teknis dari penguji, penguji tidak perlu memahami kode program serta tidak diharuskan untuk menguasai struktur *internal* dari kode. Akan tetapi, *black box testing* juga memiliki kekurangan, termasuk risiko terhadap berbagai kesalahan yang tidak terdeteksi dalam kode karena kurangnya pemahaman mendalam tentang arsitektur *internal* aplikasi [44]

2.2.2 Skala Likert

Skala Likert adalah pengukuran skala psikometrik yang umum digunakan dalam penelitian untuk mengukur seberapa banyak tingkat persetujuan, pendapat, atau sikap seseorang terhadap pernyataan atau pertanyaan yang berhubungan dengan penelitian yang kemudian diimplementasikan dalam bentuk angket maupun survei penelitian. Dengan *skala likert*, proses pengukuran kesepakatan atau ketidaksetujuan terhadap suatu pernyataan dapat ditentukan, gambaran yang jelas tentang sikap atau pendapat responden terhadap isu yang diteliti juga dapat ditentukan. Pernyataan *skala likert* akan dinyatakan dalam alternatif respon seperti sangat setuju (SS), setuju (S), ragu-ragu (R), tidak setuju (TS) dan sangat tidak setuju (STS). Meskipun item *skala likert* tidak secara langsung menunjukkan hubungan dengan sikap yang diteliti, item tersebut masih dapat dimasukkan dalam skala untuk menilai kepuasan pelanggan dengan lebih efektif [45]

Skala likert ini digunakan untuk mengukur sikap, pendapat, dan persepsi yang terjadi terhadap fenomena sosial dengan variabel yang dijabarkan menjadi indikator variabel. Indikator ini akan dijadikan sebagai titik tolak ukur penyusunan *item-item* instrumen yang dapat berupa pernyataan atau pertanyaan. Jawaban atas setiap *item* instrumen yang menggunakan *skala likert* mempunyai gradasi item dari sangat positif hingga sangat negatif. Dengan *skala likert*, pengukuran intensitas perasaan terkait ide, masalah, pengalaman yang membantu dalam analisis statistik lebih lanjut yang sangat berharga dan serbaguna dalam mengukur kesepakatan dan persasaan responden [46]

2.3 Penelitian Terkait

2.3.1 Penelitian Terkait

Pada penulisan dokumen pengembangan sistem, penelitian terkait merujuk kepada studi atau literatur yang berkaitan dengan topik penelitian. Berikut penelitian terkait:

1. Pengembangan *Backend Server* Berbasis Arsitektur *REST API* pada Sistem Transfer Dompet Digital untuk memfasilitasi sistem transfer uang pada lingkup *e-commerce*. Studi ini direspon atas rendahnya tingkat loyalitas pengguna pada sistem *backend*, dengan harapan dapat mempermudah proses transfer uang. Dalam konteks luas, pembayaran tunai bergeser cepat menuju pembayaran elektronik atau *cashless*. Namun, proses transfer saldo dompet digital kurang efisien karena melibatkan penarikan dana dari rekening bank. Maka dari itu, pengembangan ini menggunakan *NestJS*, *MySQL*, *Postman*, metode *Extreme Programming (XP)*, dan *black box testing* untuk menciptakan sistem *server* transfer dana antar dompet digital berbasis *REST API* yang terintegrasi.
2. Perancangan Sistem Informasi Manajemen Proyek Menggunakan *Software NestJS* Berbasis *Web* di PT. Mitra Pajakku dilakukan untuk mengembangkan manajemen proyek dan menggantikan sistem manual yang kurang terintegrasi. Metodologi penelitiannya adalah kualitatif dimana data dikumpulkan dari studi pustaka, observasi, dan wawancara. Pendekatan *Extreme Programming (XP)* memfasilitasi pengembangan, dimana pencatatan tugas per proyek, pencatatan *timeline*, dan *sprint* dilakukan terorganisir. Sebagai langkah awal, aplikasi diperkenalkan ke pengguna untuk mendapatkan masukan awal sebelum proses penggantian sistem informasi yang telah dirancang. Lalu dilakukan uji coba untuk memvalidasi fungsi yang akan dikembangkan. Penelitian ini mencakup pengembangan fitur laporan dan fitur impor untuk meningkatkan kemudahan proyek. Tujuan penelitian ini adalah menciptakan sistem informasi manajemen proyek yang efisien dan terintegrasi dengan platform *NestJS* berbasis *web*.
3. Penelitian Rancang Bangun *Backend* Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja *Laravel* Berbasis *Restful API* dilaksanakan dengan *framework Laravel* untuk mengelola data akreditasi secara efisien serta meminimalisir kesalahan manusia yang mungkin dapat terjadi. Penelitian ini menggunakan *waterfall* dengan menganalisis kebutuhan, merancang sistem, implementasi sistem, pengujian, dan evaluasi sistem berbasis *Restful API*.

Sistem ini memudahkan operasi tambah, lihat, edit, hapus, dan persetujuan data yang diuji dengan menggunakan metode *black box testing* untuk memastikan hasil pengembangan terpenuhi dan telah berfungsi sesuai dengan kebutuhan.

4. Pengembangan Sistem Manajemen Proyek Menggunakan Metode *Scrum* Berbasis *Android* mengimplementasikan *Rapid Application Development* untuk menciptakan sistem manajemen proyek pada platform *Android*. Inisiatif pengembangan yang memenuhi kebutuhan klien dengan menyediakan struktur jadwal yang terorganisir dan monitoring proyek yang efisien, sehingga hasil dan kemajuan tugas dapat dilacak dan dicapai. Mengadopsi metode *Scrum* menjadi keputusan strategis karena memanfaatkan kerangka iteratif dan incremental untuk pengembangan berorientasi objek. Dalam konteks *Rapid Application Development*, sistem yang dikembangkan bisa melewati pengujian komprehensif, meliputi pengujian unit, pengujian validasi, dan pengujian kompatibilitas. Hasilnya, pengujian unit mengidentifikasi 13 kasus uji dengan status valid, pengujian validasi mencatat 61 kasus uji berstatus valid dan pengujian kompatibilitas memastikan bahwa sistem beroperasi efektif pada berbagai versi sistem operasi *android*, mulai dari *lollipop* hingga *nougat*.
5. Rancang Bangun Sistem Informasi Manajemen Proyek Departemen IT PT. Pertamina UPMS V Surabaya dikembangkan dengan bahasa pemrograman C# oleh departemen IT PT. Pertamina UPMS V Surabaya. Tujuan utama sistem ini adalah mendukung proses pengawasan dan penjadwalan proyek, baik yang akan dilaksanakan maupun yang sedang berlangsung. Sebelumnya, Departemen IT M&T belum memanfaatkan aplikasi sistematis untuk pengawasan dan penjadwalan, mengandalkan proses manual seperti pembuatan laporan dengan file *Excel* untuk setiap capaian dari proyek yang dikerjakan, yang seringkali tidak efisien dan rentan terhadap kesalahan manusia. Sebagai solusi, telah dibuat aplikasi sistem informasi manajemen proyek berbasis *web*, yang dikembangkan menggunakan bahasa pemrograman C# dan *database Microsoft SQL Server*. Berdasarkan evaluasi yang telah dilakukan, fitur-fitur yang tersedia dalam aplikasi ini berfungsi dengan baik dan telah memenuhi kebutuhan pengguna, menawarkan peningkatan signifikan dalam efisiensi dan akurasi dalam pengawasan serta penjadwalan proyek.

Table 2.3.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	Subjek	Hasil
1	Irfan Rizq Dzaky Muhammad, Irving V. Papatungan [47]	Pengembangan <i>Backend Server</i> Berbasis Arsitektur <i>Rest API</i> pada Sistem Transfer Dompet Digital	<i>NestJS</i>	Dompet Digital	Sistem Transfer Dompet Digital
2	Acep Muhammad Saepuloh, Seliwati 2022 [48]	Perancangan Sistem Informasi Manajemen Proyek Dengan Menggunakan <i>Software NestJS</i> Berbasis <i>Web</i> di PT. Mitra Pajakku	<i>NestJS</i>	PT. Mitra Pajakku	Sistem Informasi Manajemen Proyek
3	Galih Saputra, Eko Didik Widiyanto, Adian Fatchur Rahim, 2023 [49]	Rancang Bangun <i>Backend</i> Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja <i>Laravel</i> Berbasis <i>Restful API</i>	<i>Laravel</i>	Institusi Pendidikan atau Program Studi	Sistem Pemantauan Akreditasi
4	Rizky Suhaimi, Nurudin Santoso, Reza Andria Siregar, 2020 [50]	Pengembangan Sistem Manajemen Proyek Menggunakan Metode <i>Scrum</i> Berbasis <i>Android</i>	<i>Rapid Application</i>	<i>Software Warehouse</i>	Sistem Manajemen Proyek
5	Yunita Ardilla, Shinta Pramesti, Dwi Sunaryono [51]	Rancang Bangun Sistem Informasi Manajemen Proyek Departemen IT PT. Pertamina UPMS V Surabaya	C3	Departemen IT PT. Pertamina UPMS V Surabaya	Sistem Informasi Manajemen Proyek Departemen IT

Sumber Tabel: Dokumentasi Pribadi

2.3.2 Posisi Penelitian

Penelitian ini memiliki posisi yang unik dalam literatur karena fokusnya pada pengembangan aplikasi *microservices* dan arsitektur modular. Penelitian ini memberikan kontribusi baru pada bidang ilmu pengembangan *web backend* dengan cara menunjukkan bagaimana *NestJS* atau *framework backend* lain dapat digunakan untuk membangun aplikasi *microservices* yang skalabel dan mudah dipelihara, berikut penjelasannya:

1. Pengembangan *Backend Server* Berbasis *Rest API* pada Sistem Transfer Dompet Digital adalah sistem yang dikembangkan agar alur proses *transfer* dapat terlaksana dengan baik, penelitian ini bukan merupakan aplikasi manajemen, tapi sistem *transfer* dompet digital yang dapat mempermudah pengguna nya. Penelitian ini dikembangkan berbasis *web* yang fokus kepada *backend* dengan menggunakan *NestJS* sebagai *framework* utamanya. Hasil dari penelitian ini adalah sistem transfer dompet digital yang menjadi lebih baik. Bila dibandingkan dengan penelitian ini, Rancang Bangun *Backend* Aplikasi Karir IO Berbasis *Web* Menggunakan *NestJS*: Studi Kasus PT. Quantum Teknologi Nusantara, penelitian terkait ini memiliki keterkaitan yang tidak signifikan namun tetap memiliki kesamaan dimana kedua penelitian ini sama-sama mengembangkan aplikasi berbasis *web* dengan fokus pengembangan kepada *backend* aplikasi (dalam hal ini arsitektur *Rest API*) dan menggunakan *NestJS* sebagai *framework* utama dalam pengembangan sistem.
2. Perancangan Sistem Informasi Manajemen Proyek Dengan Menggunakan *Software NestJS* Berbasis *Web* di PT. Mitra Pajakku merupakan sistem yang dikembangkan oleh PT. Mitra Pajakku untuk mengembangkan sistem informasi manajemen proyek untuk digunakan dengan baik oleh perusahaan, penelitian ini menggunakan *NestJS* sebagai *framework* utama agar terbentuk sistem informasi manajemen proyek. Bila dibandingkan dengan penelitian ini, Rancang Bangun *Backend* Aplikasi Karir IO Berbasis *Web* Menggunakan *NestJS*: Studi Kasus PT. Quantum Teknologi Nusantara, penelitian terkait ini memiliki keterkaitan yang tidak signifikan namun tetap memiliki kesamaan dengan penelitian yang dikembangkan, kedua penelitian ini mengembangkan aplikasi manajemen tugas proyek kantor berbasis *web* dengan fokus kepada *backend* aplikasi dan menggunakan *NestJS* sebagai *framework* utama yang digunakan oleh tim pengembang dalam pengembangan sistem aplikasi.
3. Rancang Bangun *Backend* Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja *Laravel* Berbasis *Restful API* adalah penelitian yang ditujukan untuk institusi pendidikan atau program studi untuk merancang sistem pemantauan akreditasi berbasis *Restful API* dengan *framework Laravel*. Bila dibandingkan dengan penelitian ini, Rancang Bangun *Backend* Aplikasi Karir IO Berbasis *Web* Menggunakan *NestJS*: Studi Kasus PT. Quantum

Teknologi Nusantara, penelitian terkait ini memiliki keterkaitan yang tidak signifikan namun tetap memiliki kesamaan dengan penelitian saat ini, kedua penelitian ini mengembangkan aplikasi manajemen tugas berbasis *web* dengan fokus pengembangan kepada *backend* aplikasi dalam hal ini rancang bangun *backend* sistem pemantauan yang berbasis *Restful API*.

4. Pengembangan Sistem Manajemen Proyek Menggunakan Metode *Scrum* Berbasis *Android* adalah penelitian yang ditujukan untuk merancang sistem manajemen proyek berbasis *android* dengan menggunakan *rapid application*. Penelitian ini ditujukan kepada *software warehouse*. Bila dibandingkan dengan penelitian ini, Rancang Bangun *Backend* Aplikasi Karir IO Berbasis *Web* Menggunakan *NestJS*: Studi Kasus PT. Quantum Teknologi Nusantara, pada penelitian terkait ini memiliki keterkaitan yang tidak signifikan namun tetap memiliki kesamaan dengan penelitian yang dikembangkan saat ini, kedua penelitian ini mengembangkan aplikasi manajemen proyek yang menerapkan metode *agile* melalui pendekatan *scrum* dalam pengembangan sistem aplikasi.
5. Rancang Bangun Sistem Informasi Manajemen Proyek Departemen IT PT. Pertamina UPMS V Surabaya adalah penelitian yang dibangun khusus untuk Departemen IT PT. Pertamina UPMS V Surabaya menggunakan C3 sebagai *framework* utamanya. Penelitian ini menghasilkan suatu sistem informasi manajemen proyek yang mempermudah Departemen IT dalam menyelesaikan proyek yang harus diselesaikan pada departemen tersebut. Bila dibandingkan dengan penelitian ini, Rancang Bangun *Backend* Aplikasi Karir IO Berbasis *Web* Menggunakan *NestJS*: Studi Kasus PT. Quantum Teknologi Nusantara, penelitian terkait ini memiliki keterkaitan yang tidak signifikan namun tetap memiliki kesamaan dengan penelitian yang dikembangkan saat ini, kedua penelitian ini mengembangkan aplikasi manajemen proyek berbasis *web*. Namun, kedua penelitian memiliki perbedaan pada fokus penelitian, teknologi yang digunakan dalam penelitian dan beberapa hal lainnya pun berbeda.

Berdasarkan pembahasan posisi penelitian, perbedaan penelitian terkait dengan Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara terasa begitu nyata, namun di setiap penelitian terkait memiliki kesamaan yang dapat menjadi acuan pengembangan sistem.

Table 2.3.2 Posisi Penelitian

No	Nama dan Tahun	Judul Penelitian	Aplikasi Manajemen	Aplikasi Berbasis Web	Backend	Scrum	NestJS
1	Irfan Rizq Dzaky Muhammad, Irving V. Papatungan	Pengembangan <i>Backend Server</i> Berbasis Arsitektur <i>Rest API</i> pada Sistem Transfer Dompot Digital	-	✓	✓	-	✓
2	Acep Muhammad Saepuloh, Seliwati 2022	Perancangan Sistem Informasi Manajemen Proyek Dengan Menggunakan <i>Software NestJS</i> Berbasis <i>Web</i> di PT. Mitra Pajakku	✓	✓	✓	-	✓
3	Galih Saputra, Eko Didik Widiyanto, Adian Fatchur Rahim, 2023	Rancang Bangun <i>Backend</i> Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja <i>Laravel</i> Berbasis <i>Restful API</i>	✓	✓	✓	-	-
4	Rizky Suhaimi, Nurudin Santoso, Reza Andria Siregar, 2020	Pengembangan Sistem Manajemen Proyek Menggunakan Metode <i>Scrum</i> Berbasis <i>Android</i>	✓	-	-	✓	-
5	Yunita Ardilla, Shinta Pramesti, Dwi Sunaryono	Rancang Bangun Sistem Informasi Manajemen Proyek Departemen IT PT. Pertamina UPMS V Surabaya	✓	✓	-	-	-
6	Sabiq Muhammad Antebing Mame (2024)	Rancang Bangun <i>Backend</i> Aplikasi Karir Io Berbasis <i>Web</i> Menggunakan <i>NestJs</i> : Studi Kasus PT. Quantum Teknologi Nusantara	✓	✓	✓	✓	✓

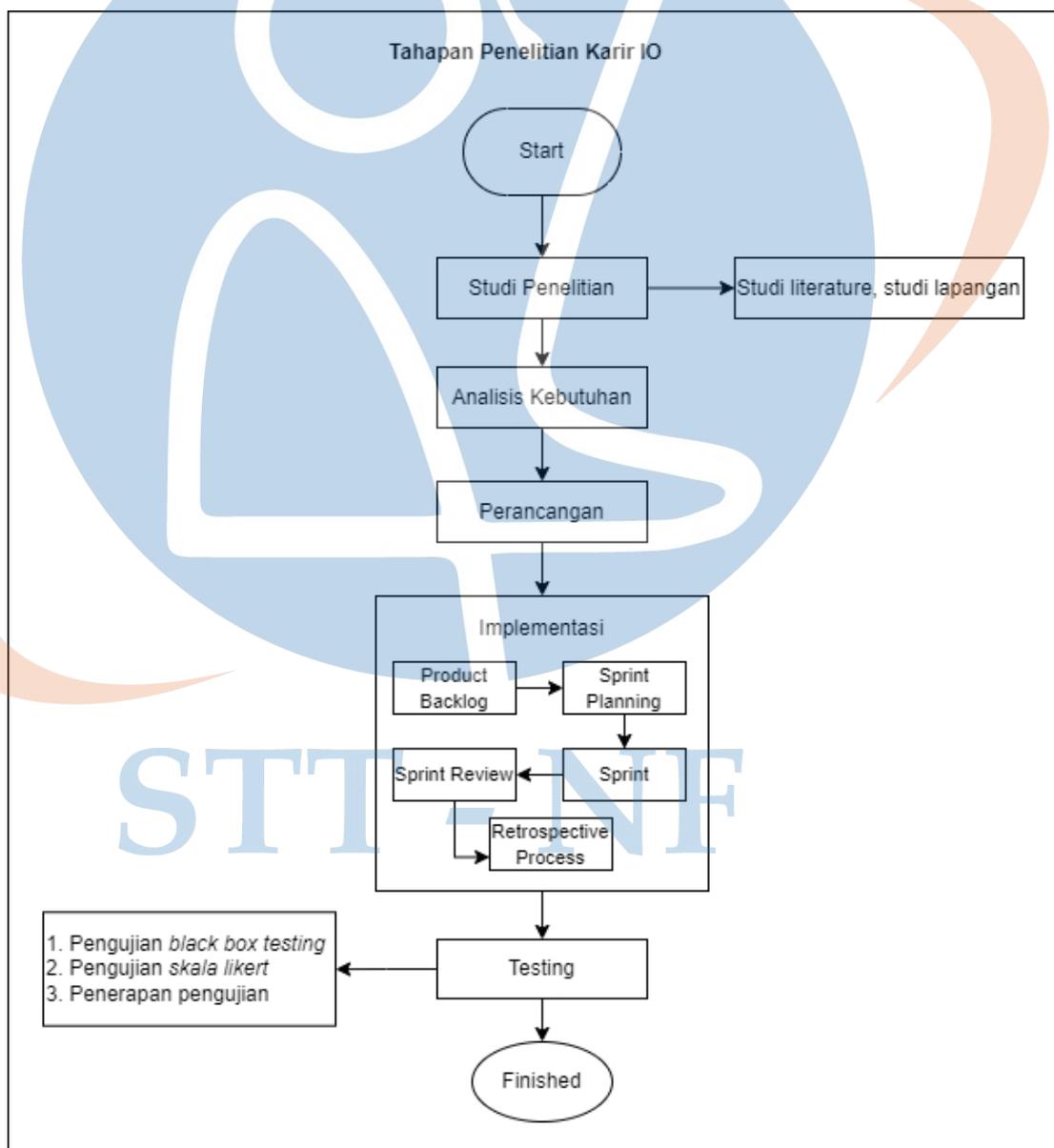
Sumber Tabel: Dokumentasi Pribadi

BAB III

ANALISIS DAN PERANCANGAN

Bab analisis dan perancangan membahas tahapan penelitian yang mempermudah proses analisis sistem yang diimplementasikan. Bab III ini akan menganalisis berbagai proses bisnis yang dilakukan dalam perancangan aplikasi Karir Io untuk menguraikan alur penelitian, mendesain rancangan penelitian, melakukan analisis sistem aplikasi serta membahas rancangan pengujian untuk verifikasi prototipe yang dikembangkan.

3.1 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian [A5]
Sumber Gambar: Dokumentasi Pribadi

Berdasarkan **Gambar 3.1**, Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara dikembangkan tim pengembang dengan melalui beberapa tahap penelitian berikut:

1. Studi Penelitian

Dalam pengembangan aplikasi Karir IO, penulis menyusun perancangan aplikasi dimana data-data yang dibutuhkan dalam perancangan aplikasi ini diperoleh melalui referensi artikel, jurnal dan materi penelitian yang berkaitan dengan rancang bangun aplikasi Karir IO, aplikasi manajemen tugas kantor. Sumber referensi studi literatur yang digunakan penulis adalah informasi pengembangan *backend* aplikasi, rancang bangun aplikasi manajemen tugas kantor / aplikasi manajemen, *framework NestJS*, metode pengembangan *agile* dengan pendekatan *scrum*. Setelah memperoleh beberapa data yang dibutuhkan, didapatkan pula hasil studi literatur yang berhubungan dengan teori penelitian untuk memberikan pemahaman model penelitian yang jelas sehingga tujuan penelitian dapat dicapai. Adapun untuk studi lapangan ini dilakukan penulis dimana penulis melakukan pengamatan dan mengelola langsung aplikasi Karir IO karena penulis menjadi tim pengembang aplikasi yang bertugas pula untuk mengamati keadaan di lapangan. Pengamatan langsung ini juga dilakukan dengan melakukan pernyataan kuesioner saat pengembangan aplikasi berlangsung.

2. Analisis Kebutuhan

Setelah melakukan studi penelitian pengembangan aplikasi menggunakan studi literature dan studi lapangan, penulis kemudian melaksanakan analisis kebutuhan agar mengetahui hal-hal apa saja yang dibutuhkan dalam pengembangan sistem. Penulis menyiapkan berbagai kebutuhan *tools* dan teknologi yang akan digunakan seperti menentukan penelitian ini akan mengembangkan *backend* untuk membangun aplikasi Karir IO yakni aplikasi untuk melakukan pengelolaan manajemen tugas-tugas kantor. Aplikasi Karir.IO yang dikembangkan pada penelitian ini menggunakan bahasa pemrograman *JavaScript* khususnya penggunaan *framework NestJS*, menggunakan metode *agile* melalui pendekatan *scrum*, menentukan RDBMS yang tepat dan sesuai pada penelitian ini adalah *MySQL*, menggunakan tempat penyimpanan data kode program yang dapat digunakan untuk berkolaborasi dengan tim pengembang yaitu *GitHub*, menerapkan library *JavaScript* untuk melakukan komunikasi secara *realtime* dan responsif yakni *Socket.IO* dan menerapkan *postman* sebagai alat *testing API*.

3. Perancangan

Tahapan perancangan yang dilalui oleh penulis melibatkan perencanaan yang terstruktur terkait dengan alur tahapan penelitian untuk pengembangan aplikasi Karir IO. Setelah melakukan penyusunan rencana awal, penulis melakukan diskusi lebih lanjut bersama dengan tim pengembang aplikasi lainnya untuk memastikan bahwa semua aspek teknis dan fungsional aplikasi Karir IO telah dipertimbangkan dengan baik dan secara menyeluruh. Diskusi yang dilakukan ini bertujuan untuk dapat mengintegrasikan berbagai perspektif dan keahlian yang dimiliki oleh anggota tim, sehingga menghasilkan strategi yang koherensif dan efektif dalam menghadapi segala tantangan yang mungkin muncul selama proses pengembangan aplikasi. Selanjutnya, hasil diskusi ini diimplementasikan dalam tahap implementasi aplikasi memastikan setiap langkah sesuai dengan tujuan penelitian dan kebutuhan pengguna akhir.

Tahapan perancangan yang dilalui oleh tim pengembang aplikasi dalam mengembangkan aplikasi Karir IO. Penulis melibatkan perencanaan terstruktur dan memiliki keterkaitan dengan alur tahapan penelitian untuk pengembangan aplikasi Karir IO. Aplikasi Karir IO menggunakan arsitektur *client-server*, di mana bagian *client* ditangani oleh *NestJS* dan bagian *server* ditangani oleh *NestJS*. *NestJS* merupakan *framework* berbasis *JavaScript* yang dibangun di atas *Express.js* dan dirancang khusus untuk pengembangan aplikasi *backend* terstruktur dan dapat diskalakan. *Database MySQL* digunakan untuk menyimpan data aplikasi. Arsitektur ini memisahkan tanggung jawab antara *client* dan *server*, sehingga meningkatkan skalabilitas, performa, dan keamanan aplikasi.

- **Client Next.js:** Bertanggung jawab untuk menampilkan *UI* dan menangani interaksi pengguna.
- **Server NestJS:** Bertanggung jawab untuk pemrosesan data, logika bisnis, komunikasi dengan *database*, dan komunikasi *real-time* dengan *client Next.js* menggunakan *Socket.io*.
- **Database MySQL:** Menyimpan data aplikasi, seperti informasi pengguna, proyek, tugas, komentar, dan notifikasi.

a. Penggunaan *Socket.io*

Socket.io adalah *library real-time* yang memungkinkan komunikasi dua arah antara *client* dan *server*. Dalam aplikasi Karir IO, *Socket.io* digunakan untuk:

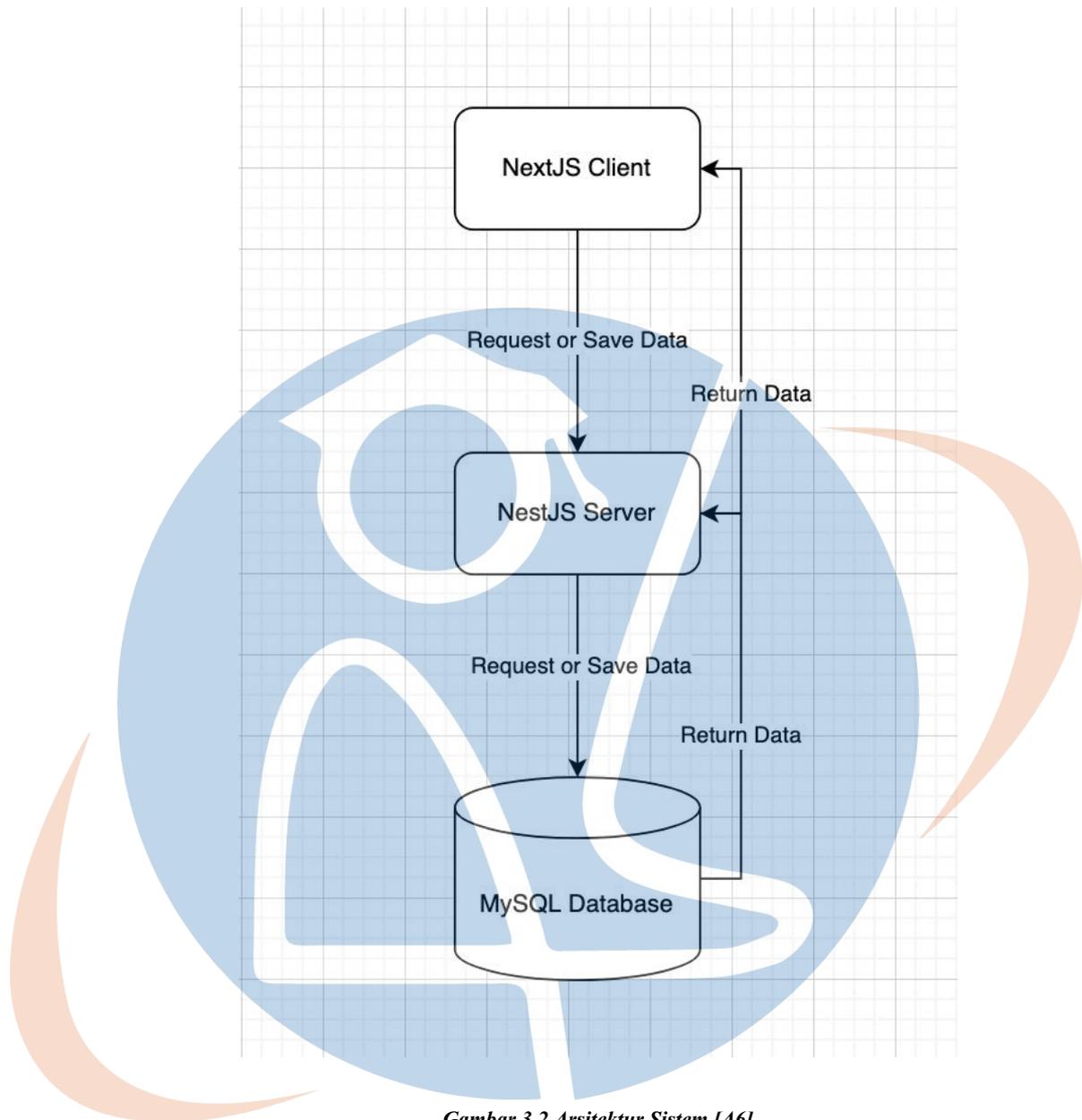
- **Komentar *Real-time*:** Pengguna dapat melihat komentar baru dan membalas komentar secara *real-time* tanpa perlu *me-refresh* halaman.

b. Penggunaan *API*

API (Application Programming Interface) adalah *interface* yang memungkinkan aplikasi lain untuk berinteraksi dengan aplikasi Karir IO. Dalam aplikasi Karir IO, *API* dapat digunakan untuk:

- **Manajemen Pengguna:** Mengelola akun pengguna, seperti membuat, membaca, dan memperbarui.
- **Manajemen Proyek:** Mengelola proyek, seperti membuat, membaca, memperbarui, dan menghapus proyek.
- **Manajemen Tugas:** Mengelola tugas, seperti membuat, membaca, memperbarui, dan menghapus tugas.

STT - NF



Gambar 3.2 Arsitektur Sistem [A6]

Sumber Gambar: Dokumentasi Pribadi

4. Implementasi

Pada tahap implementasi sistem, perencanaan proses dilakukan dengan *scrum* yang mengharuskan tim pengembang melakukan analisis *product backlog* sesuai pada penelitian, merancang alur *sprint planning*, melakukan *sprint*, melaksanakan tahapan *sprint review* terkait *sprint* yang telah dilakukan, dan melakukan *retrospective process* untuk mengevaluasi dan memperbaiki proses pengembangan aplikasi Karir IO ke depannya. Dalam melaksanakan setiap *sprint*, tim pengembang akan berfokus pada pencapaian tujuan yang ditetapkan dan memastikan fitur dikembangkan sesuai dengan kebutuhan yang telah diidentifikasi bersama saat awal pengembangan sistem.

Sprint review yang dilakukan pada penelitian ini bertujuan untuk menilai hasil kerja dan efektivitas solusi yang akan dikembangkan. Pelaksanaan *sprint review* juga mengharuskan semua *stakeholder* terlibat aktif dalam memberikan umpan balik yang digunakan pada perbaikan berikutnya dan memungkinkan tim untuk menyesuaikan prioritas serta teknik kebutuhan proyek yang terus berkembang dan respons pengguna. Tahap *retrospective process* dilakukan sebagai upaya untuk merenungkan proses yang berlangsung selama *sprint* untuk mengidentifikasi kekuatan, kelemahan, peluang, dan ancaman yang dapat hadir dalam metode kerja tim, sehingga setiap iterasi yang akan digunakan dapat menjadi lebih efisien dan efektif. Proses yang dilakukan ini esensial untuk memastikan adaptasi secara terus menerus terhadap perubahan kondisi dan peningkatan berkelanjutan dalam pengembangan aplikasi Karir IO.

a) *Backlog*

Selama tahap pembuatan *backlog*, seluruh tim pengembang aplikasi Karir IO ini melakukan analisis mendalam terhadap proses pengembangan sistem dengan memanfaatkan alur *user stories* yang telah dianalisis dan dirancang sebelumnya. Alur ini dirancang untuk memastikan bahwa setiap aspek dari *user stories* telah terintegrasi dengan baik dalam sistem yang sedang dikembangkan, sehingga menghasilkan *output* yang diharapkan untuk aplikasi Karir IO. Analisis ini digunakan untuk melakukan validasi kesesuaian yang dibutuhkan antara kebutuhan pengguna dan fungsionalitas yang akan dikembangkan, dengan tujuan akhirnya adalah menciptakan aplikasi Karir IO yang tidak hanya efektif tapi intuitif dan mudah digunakan oleh pengguna akhir.

b) *Sprint Planning*

Setelah merumuskan alur *user stories* yang akan digunakan untuk tahapan *product backlog*, dilanjutkan dengan melalui proses tahapan *sprint planning*, di mana tahapan ini dilakukan oleh tim pengembang untuk melakukan serangkaian uji coba yang tepat pada aplikasi Karir IO sehingga fitur-fitur yang telah dikembangkan dapat diverifikasi kebenaran dan kesesuaiannya. *Sprint planning* penelitian ini dilakukan bersama tim *SQA* untuk memastikan bahwa fitur-fitur pada aplikasi Karir IO, yang dirancang untuk meningkatkan efisiensi manajemen tugas kantor, beroperasi dengan baik dan bebas dari *bug* atau kecacatan. Uji coba ini esensial untuk mengidentifikasi dan memperbaiki kecacatan sebelum fitur tersebut diterapkan secara luas, sehingga integritas dan keandalan aplikasi dalam pengelolaan tugas sehari-hari terkelola baik.

c) *Sprint*

Sprint dilaksanakan sebagai serangkaian aktivitas pengembangan sistem aplikasi di mana tim pengembang aplikasi akan bekerja untuk mencapai berbagai tujuan yang telah ditetapkan sebelumnya dalam tahapan *sprint planning*. *Sprint* pada proses pengembangan sistem akan melibatkan kerja intensif dari seluruh tim pengembang untuk dapat mengimplementasikan fitur-fitur yang telah direncanakan. Dalam konteks pengembangan aplikasi Karir IO, tahapan *sprint* akan digunakan untuk mengelola dan melaksanakan proyek pengembangan perangkat lunak karena *sprint* dapat dikatakan sebagai periode waktu terbatas yang dimiliki tim pengembang selama pengembangan aplikasi berlangsung untuk menyelesaikan serangkaian fitur yang telah ditetapkan.

d) *Sprint Review*

Tahapan ini merupakan bagian integral dari metodologi *scrum* yang digunakan dalam pengembangan aplikasi Karir IO sebuah aplikasi manajemen tugas kantor untuk memastikan aplikasi Karir IO dapat berkembang sesuai dengan kebutuhan pengguna aplikasi dan membantu pengelolaan tugas kantor menjadi lebih efektif. Dalam *sprint review* yang dilakukan ini, seluruh tim pengembang aplikasi dengan tanggung jawabnya sebagai *tech lead*, *frontend*, *backend*, *QA*, *project manager*, dan *asistant project manager* berkumpul untuk mengevaluasi hasil kerja yang telah diselesaikannya selama proses *sprint* berlangsung dan memberikan penilaian tentang kemajuan proyek yang dikembangkan. Dengan *sprint review*, hasil *sprint* selanjutnya dapat dikembangkan menjadi lebih sejalan dengan kebutuhan bisnis serta ekspektasi pengguna aplikasi.

e) *Retrospective*

Tahap *retrospective* ini dilakukan oleh penulis untuk lebih menanggapi pertanyaan umum dan masalah yang hadir selama proses pengembangan aplikasi manajemen tugas, aplikasi Karir IO ini. Tahap *retrospective* ini dapat meningkatkan aspek-aspek yang masih kurang pada aplikasi dan menemukan solusi yang tepat untuk mengatasi permasalahan serta kendala yang terjadi. Tahapan ini juga mencakup improvisasi terbaik yang dapat dilakukan oleh tim pengembang aplikasi Karir IO dan penentuan solusi yang efektif untuk menghadapi masalah yang mungkin timbul pada aplikasi.

5. *Testing*

Tahapan *testing* ini dilakukan bersama tim pengembang aplikasi Karir IO untuk mengevaluasi sistem perangkat lunak telah berfungsi baik dimana pengujian aplikasi Karir IO ini akan dilakukan secara keseluruhan. Setelah pelaksanaan testing aplikasi, tim pengembang melalui proses evaluasi sistem untuk mencapai kesimpulan yang sesuai dengan aplikasi yang dikembangkan. Saat akan melakukan pengujian Karir IO terdapat beberapa jenis tes yang dilakukan bersama seperti tes fungsional aplikasi dengan menghasilkan hasil pengujian *black box testing* dan *skala likert*, tes integrasi sistem bersama dengan tim pengembang yang dilakukan berdasarkan *product backlog*, tes kinerja aplikasi, tes keamanan dan tes penggunaan aplikasi yang dilakukan sesuai dengan tahapan *sprint*. Melihat dari testing yang dilakukan ini, maka kesalahan dan kecacatan sistem aplikasi Karir IO dapat ditemukan, sehingga kebutuhan pengguna akan sistem manajemen tugas yang berkualitas tinggi dapat terpenuhi dengan sesuai.

3.2 Rancangan Penelitian

Rancangan penelitian adalah kunci penelitian untuk memastikan data-data yang diperoleh dan tujuan pada penelitian dapat tercapai. Rancangan penelitian menjadi pedoman logis untuk melaksanakan penelitian sesuai dengan data yang dikumpulkan sejak awal saat merencanakan objek penelitian [52]. Dalam pengembangan aplikasi Karir IO, penelitian kualitatif adalah jenis penelitian yang digunakan berdasarkan kesimpulan pengelolaan data dan pengumpulan wawasan yang telah ditindaklanjuti.

Adapun hal yang melatar belakangi rancang bangun aplikasi Karir IO yaitu manajemen tugas kantor di PT. Qauantum masih menjadi hal krusial, dimana tugas mengalami kendala pengerjaan karena tugas yang harus dikerjakan banyak namun tenggat waktu begitu cepat sehingga banyak tugas kantor mengalami permasalahan. Karena itu, penulis mengembangkan aplikasi Karir IO untuk dimanfaatkan oleh PT. Quantum, aplikasi Karir IO berbasis *web* yang dikembangkan dengan menggunakan bahasa pemrograman *JavaScript* khususnya menggunakan manfaat dari *framework NestJS*. Karir IO menggunakan metode *agile* melalui pendekatan *scrum* sebagai metode pengembangan sistem dan menggunakan metode kualitatif sebagai metode penelitian pada sistem yang dikembangkan saat ini. Penelitian ini mengumpulkan data dengan melaksanakan kuesioner pada karyawan PT. Quantum dan melakukan pengujian *black box testing* dan *skala likert* untuk menghasilkan aplikasi Karir IO.

3.2.1 Jenis Penelitian

Jenis penelitian yang diterapkan dalam pengembangan aplikasi Karir IO adalah penelitian pengembangan *R&D* untuk menciptakan aplikasi terbaru berbasis *website*. Perancangan aplikasi Karir IO membutuhkan riset mendalam dan terstruktur untuk memastikan fitur yang dikembangkan memenuhi seluruh kebutuhan pengguna. Jenis penelitian pengembangan *R&D* aplikasi Karir IO diharapkan dapat mempermudah manajemen tugas kantor PT. Quantum, memberikan pemahaman terkait keefektifan teknologi pada operasi sehari-hari, memberikan solusi nyata yang bisa diukur hasilnya agar mempermudah kinerja karyawan PT. Quantum, serta meningkatkan efisiensi dan produktivitas karyawan melalui penggunaan teknologi yang berkualitas tinggi.

3.2.2 Metode Analisis Data

Metode analisis data yang diadopsi pada pengembangan aplikasi Karir IO adalah metode kualitatif yang mendetailkan alur penelitian berdasarkan pertimbangan nyata dari berbagai sisi. Metode kualitatif dilakukan melalui pengisian survei kuesioner secara *online* oleh karyawan PT. Quantum. Metode ini menggunakan perincian dan observasi data lanjutan oleh pengguna aplikasi yang menjadi populasi utama pada penelitian. Kuesioner yang diisi karyawan PT. Quantum mencakup pertanyaan yang berhubungan dengan pengembangan aplikasi agar mencapai tujuan pengembangan. Hasil yang diperoleh dari pengisian survei kuesioner ini kemudian akan dilakukan pengolahan data kembali dengan *black box testing* dan melakukan analisis *skala likert* secara lanjut untuk memastikan penelitian ini memberikan manfaat yang signifikan.

3.2.3 Metode Pengumpulan Data

Pengembangan aplikasi Karir IO, metode pengumpulan data bergantung pada tujuan spesifik dan sumber daya yang memberikan pandangan dan data sesuai untuk mendukung pengembangan, berikut metode pengumpulan data pada penelitian, yaitu:

1. **Kuesioner:** Kuesioner digunakan untuk mengumpulkan data berjumlah besar dari seluruh responden. Berdasarkan permasalahan, penulis mengembangkan aplikasi Karir IO, dimana data dikumpulkan dengan kuesioner untuk menilai efektivitas fitur dan fungsi aplikasi, mengetahui kepuasan pengguna aplikasi, dan aspek operasional apakah telah sesuai dan memenuhi tujuan aplikasi atau belum. *Skala likert* dalam penelitian ini diharapkan dapat mempermudah pengukuran tingkat kepuasan atau pendapat karyawan tentang fitur aplikasi.

2. **Wawancara:** Wawancara ini dilakukan dengan pengguna aplikasi, dalam hal ini karyawan PT. Quantum untuk memberikan wawasan mendalam mengenai pengalamannya dalam menggunakan aplikasi Karir IO berbasis *web*. Metode wawancara dilakukan secara individual untuk memperoleh *feedback* yang lebih detail yang mungkin tidak terkumpul melalui metode kuesioner.

3.2.4 Metode Pengujian

Metode pengujian fungsionalitas yang digunakan dalam pengembangan aplikasi Karir IO adalah metode *black box testing* tanpa mengetahui struktur *internal* kode atau cara kerja program aplikasi Karir IO tetapi menerapkan desain pengujian yang sesuai dan menggunakan metode pengujian *skala likert* berupa kuesioner penelitian yang berisi pertanyaan yang membutuhkan jawaban dari pengguna aplikasi. Pada penelitian ini, metode pengujian *black box testing* melibatkan *input* sesuai dengan alur rancangan penelitian dan pengamatan *output* aplikasi Karir IO berbasis *web* dengan fitur dan fungsionalitas aplikasi yang telah berjalan sesuai, untuk pengujian *skala likert* membutuhkan pernyataan pengguna dan berisi *review* aplikasi Karir IO dalam bentuk kuesioner penelitian. Pengujian *black box testing* dan pengujian *skala likert* digunakan untuk menyesuaikan desain yang telah dirancang sebelumnya dan terfokus pada pengujian *backend* aplikasi Karir IO yang akan dikembangkan.

3.2.4.1 Metode Pengujian *Black Box Testing*

Pada penelitian ini, metode pengujian *black box testing* merupakan rancangan pengujian pertama yang dilakukan untuk menyesuaikan gambaran desain yang telah dibangun sebelumnya dan terfokus pada alur pengujian *backend* aplikasi Karir IO. Dalam hal ini, *black box testing* yang dilakukan diharapkan dapat menunjukkan fungsi dan manfaat nyata program aplikasi Karir IO agar bekerja sesuai dengan tahapan pengoperasiannya dan berjalan sebagaimana alur kode program.

Tabel 3.1 Rancangan Pengujian - *Black Box Testing*

No	Pengujian	<i>HTTP Method/Endpoint</i>	Hasil Yang Diharapkan	Kesimpulan
1	Register dengan <i>body</i> sesuai	[POST] https://karir-api.quantumteknologi.com/auth/register	Akan menghasilkan <i>response</i> : Status : Created, Code : 201 Dimana register pengguna berhasil dan pengujian ini berhasil	Pengguna Baru Berhasil Dibuat

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			dilakukan	
2	Register dengan body tidak sesuai	[POST] https://karir-api.quantumteknologi.com/auth/register	Akan menghasilkan <i>response</i> : <i>Status Code</i> : 400 Dengan <i>message</i> : "Password Must Be A String", "Password Should Not Be Empty", "Whatsapp Must Be A String", "Whatsapp Should Not Be Empty" Dan status <i>Error</i> : "Bad Request" Dimana data-data register tidak berhasil diinputkan dan pengujian ini tidak berhasil dilakukan apabila <i>request body</i> saat melakukan <i>register</i> tidak sesuai dengan ketentuan yang berlaku	Pengguna Baru Gagal Dibuat
3	<i>Login</i> dengan <i>body</i> sesuai	[POST] https://karir-api.quantumteknologi.com/auth/login	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana data-data <i>login</i> dengan <i>email</i> dan <i>password</i> yang telah di <i>input</i> oleh pengguna aplikasi sesuai dan pengujian ini berhasil dilakukan	Pengguna Berhasil <i>Login</i>
4	<i>Login</i> dengan <i>body</i> tidak sesuai	[POST] https://karir-api.quantumteknologi.com/auth/login	Akan menghasilkan <i>response</i> : <i>Status Code</i> : 400 <i>Message</i> : "Password must be a string", "Password should not be empty" Dan status <i>Error</i> : "Bad Request" Dimana data-data <i>login</i> tidak berhasil diinputkan dan pengujian ini tidak berhasil dilakukan apabila <i>login</i> tidak sesuai dengan ketentuan yang berlaku	Pengguna Gagal <i>Login</i>
5	Membuat <i>project</i> baru dengan <i>body</i> sesuai	[POST] https://karir-api.quantumteknologi.com/projects	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 Dimana proyek baru PT. Quantum dengan <i>input</i> data yang sesuai pada aplikasi Karir IO berhasil dibuat dan pengujian ini berhasil dilakukan	<i>Project</i> Berhasil Ditambahkan

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
6	Menampilkan semua <i>project</i> yang dibuat	[GET] https://karir-api.quantumteknologi.com/projects	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana seluruh proyek PT. Quantum yang tersedia pada aplikasi Karir IO berhasil ditampilkan dan pengujian ini berhasil dilakukan	<i>Project</i> Berhasil Ditambahkan
7	Menampilkan <i>project detail</i> dengan id 7	[GET] https://karir-api.quantumteknologi.com/projects/7	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana seluruh proyek PT. Quantum yang tersedia pada aplikasi Karir IO dan memiliki id 7 maka telah berhasil ditampilkan <i>detail</i> -nya dan pengujian ini berhasil dilakukan	<i>Project Detail</i> Dengan Id 7 Berhasil Didapatkan
8	Mengupdate <i>project</i> dengan id 7	[PUT] https://karir-api.quantumteknologi.com/projects/7	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana seluruh proyek PT. Quantum yang tersedia pada aplikasi Karir IO dan memiliki id 7 maka telah berhasil di <i>update project</i> nya dan pengujian ini berhasil dilakukan	<i>Project Detail</i> Dengan Id 7 Berubah Nama Dari INA Digital Menjadi Ina Digital Superapp
9	Menambahkan member ke <i>project</i> dengan id 1 dengan <i>body</i> sesuai	[POST] https://karir-api.quantumteknologi.com/projects/1-members	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 Dimana proyek PT. Quantum berhasil ditambahkan member nya ke <i>project</i> dengan id 1 berdasarkan <i>body</i> yang sesuai dan pengujian ini berhasil dilakukan	Member Berhasil Ditambahkan Ke <i>Project</i> Id 1
10	Menampilkan semua member yang ada di <i>project</i> dengan id 7	[GET] https://karir-api.quantumteknologi.com/projects/7-members	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana semua member yang terdaftar pada <i>project</i> PT. Quantum dengan id 7 telah berhasil ditampilkan dan pengujian ini berhasil dilakukan	Semua Member Berhasil Dimunculkan Di <i>Project</i> Id 7
11	Membuat tugas pada <i>project</i> id 7 kepada <i>user</i> dengan id 584	[POST] https://karir-api.quantumteknologi.com/tasks	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 Dimana proyek baru PT. Quantum berhasil membuat	Tugas Berhasil Diberikan

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			tugas pada project id 7 kepada user dengan id 584 berdasarkan ketentuan yang sesuai dan pengujian ini berhasil dilakukan	
12	Dapatkan semua tugas yang <i>active</i> pada <i>project</i> id 7	[GET] https://karir-api.quantumteknologi.com/tasks/projects/7?status=active&page=1&limit=10	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana seluruh tugas yang aktif berjalan pada PT. Quantum berhasil didapatkan pada <i>project</i> id 7 dan pengujian ini berhasil dilakukan	Tugas Berhasil Di Dapatkan Pada <i>Project</i> Id 7
13	Membuat komentar pada <i>task</i> id 635	[POST] https://karir-api.quantumteknologi.com/task-comments/tasks/635	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 Dimana pembuatan komentar pada <i>task</i> id 635 berhasil ditambahkan sesuai keadaan sebenarnya dan pengujian ini berhasil dilakukan	Komentar Berhasil Dibuat
14	Menampilkan komentar pada <i>task</i> 635	[GET] https://karir-api.quantumteknologi.com/task-comments/tasks/635	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana komentar pada <i>task</i> 635 berhasil ditampilkan dan pengujian ini berhasil dilakukan	Komentar Berhasil Didapatkan Pada <i>Task</i> Id 635
15	Membalas komentar pada <i>tasks</i> 635 dengan comment id 586	[POST] https://karir-api.quantumteknologi.com/task-comments/tasks/635	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 <i>Message</i> : “Deadline nanti ya hari minggu” atau <i>message</i> yang lainnya Dimana membalas komentar pada <i>task</i> 635 dengan <i>comment</i> id 586 dan pengujian pun berhasil	Berhasil Membalas Komentar
16	Melihat balasan pada komentar id 586	[GET] https://karir-api.quantumteknologi.com/task-comments/586/replies	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 <i>Message</i> : “Baik siapp bos” Dimana seluruh balasan komentar id 586 dapat ditampilkan, pengujian ini pun telah berhasil	Berhasil Melihat Balasan Pada Komentar Id 586

Sumber Tabel: Dokumentasi Pribadi

3.2.4.2 Metode Pengujian *Skala Likert*

Metode pengujian *skala likert* yang digunakan pada penelitian ini melalui suatu kuesioner penelitian untuk memberikan gambaran pengalaman pengguna aplikasi Karir IO. *Skala likert* yang digunakan ini, memiliki 4 macam pernyataan yang harus diisi oleh pengguna, pilihan tersebut yakni Sangat Tidak Setuju (STS) dengan skor 1 point, Tidak Setuju (TS) dengan skor 2 point, Setuju (S) dengan skor 3 point, dan Sangat Setuju (SS) dengan skor 4 point. Pilihan pernyataan yang telah dipilih oleh pengguna aplikasi akan diakumulasikan dan dibuat hipotesis penelitian yang sesuai, pengujian *skala likert* ini digambarkan dalam bentuk tabel sebagai berikut:

Tabel 3. 2 Rancangan Pengujian - Skala Likert

No	Pernyataan	STS	TS	S	SS
1	<i>API</i> Karir IO berjalan baik sebagaimana fungsinya				
2	<i>API</i> memberikan <i>response</i> yang cepat dan kinerja baik				
3	Struktur dan format <i>response API</i> mudah dimengerti				
4	<i>API</i> memberikan <i>response</i> yang informatif bila terjadi kesalahan atau permasalahan <i>request</i> yang diajukan				
5	Struktur <i>endpoint API</i> mudah untuk dipahami pengguna				
6	Integrasi <i>API</i> dengan aplikasi pihak ketiga sesuai dengan ketentuan yang diberlakukan				
7	Penggunaan <i>API</i> mempermudah karyawan PT. Quantum dalam mengelola dan manajemen tugas kantor				
8	<i>API</i> ini memberikan fitur keamanan yang supportif terhadap data-data pengguna aplikasi				
9	Struktur format dan <i>endpoint API</i> mempermudah proses perkembangan aplikasi Karir IO yang dikembangkan				
10	Secara keseluruhan, penggunaan <i>API</i> telah memenuhi ekspektasi dan kebutuhan PT. Quantum dalam mengelola tugas kantor				

Sumber Tabel: Dokumentasi Pribadi

3.2.5 Metode Implementasi dan Evaluasi

Metode implementasi dan evaluasi dilakukan oleh tim pengembang aplikasi Karir IO dengan memastikan pengembangan aplikasi berjalan baik. Metode ini dilakukan melalui observasi dan pengamatan langsung terhadap cara karyawan menggunakan aplikasi Karir IO dan memastikan bagaimana aplikasi Karir IO diintegrasikan dalam pekerjaan sehari-hari dan interaksi pengguna dengan aplikasi tersebut. Implementasi dan evaluasi pengembangan Karir IO membantu pengumpulan data perbaikan dan analisis log aplikasi dalam membantu identifikasi pengguna, masalah keamanan, dan potensi area peningkatan berdasarkan interaksi pengguna yang sebenarnya terjadi.

3.2.6 Lingkungan Pengembangan

1. **Perangkat:** Selama proses rancang bangun aplikasi Karir IO berbasis *web* PT. Quantum, penulis menggunakan perangkat keras laptop Macbook Air 2022 Apple M2 CPU, RAM 8.0GB dan MacOS Sonoma 14.1.1 *Operating System*
2. **Alat Desain:** Rancang bangun aplikasi Karir IO ini menggunakan *software online diagrams.net* untuk mempermudah penulis dalam mendesain diagram yang dapat menjelaskan proses pengembangan aplikasi pada penelitian ini seperti mendesain *class diagram, use case, activity diagram*, dan lainnya.
3. **Alat Kolaborasi:** Alat kolaborasi yang digunakan untuk *discuss* bersama membahas fitur aplikasi adalah dengan *whatsapp, telegram, dan github*.

3.2.7 Peran Sumber Daya Manusia

Dalam pengembangan aplikasi terdapat beberapa peran yang membantu yaitu:

1. **Tech Lead :** *Tech lead* dalam pengembangan aplikasi Karir IO PT. Quantum Teknologi Nusantara bertanggung jawab untuk merancang arsitektur karir io dan memberikan arahan untuk programmer dalam pengembangan. Dalam hal ini, Mirzha Ikhsan sebagai *teach leader* yang menjalankan tugas tersebut.
2. **Frontend :** Aplikasi Karir IO membutuhkan tampilan aplikasi yang bersifat interaktif, jelas, dan fleksibel untuk dapat memudahkan seluruh pengguna aplikasi mengakses fitur-fitur yang terdapat di aplikasi. *Frontend* aplikasi Karir IO ada 2 yaitu Billy Julius yang berfokus pada pembuatan fitur *create, read, update, delete project, login* dan *register*. Dan ada pula Agung Prabowo yang merupakan *frontend* dengan fokus kepada pembuatan *create, read, update, delete task*, dan pembuatan komentar di *task* aplikasi Karir IO.

3. **Backend** : Rancang bangun aplikasi Karir IO juga membutuhkan *server backend* yang dapat menangani logika dan operasi pemrosesan data yang tidak terlihat langsung pengguna aplikasi tetapi bersifat krusial untuk menghasilkan fungsi aplikasi yang sesuai. Sama halnya seperti *frontend*, *backend* aplikasi Karir IO ada 2 yaitu Sabiq Muhammad Antebing Mame yang tidak lain adalah penulis sendiri dan Nathan. Penulis bertugas sebagai *backend* yang berfokus pada pembuatan *crud project*, *crud task*, komentar di *task*. Sedangkan Nathan, bertugas sebagai *backend* yang berfokus pada pembuatan *login*, *register*.
4. **Quality Assurance (QA)** : Aplikasi Karir IO membutuhkan pengujian untuk mengetahui kecatatan fitur dan fungsionalitas aplikasi telah berjalan dengan baik atau belum. *QA* bertugas sebagai *QA* melakukan pengecekan tiap fitur di aplikasi. Dalam hal ini, Franky adalah *QA* yang menjalankan tugas tersebut.
5. **Project Manager** : Pada aplikasi Karir IO, project manager berperan penting untuk keberhasilan proyek, project manager ini bertugas untuk melakukan penyusunan rencana proyek, menentukan *timeline*, fitur apa saja yang harus dikelarkan terlebih dahulu, melakukan pemantauan dan pengendalian untuk mencapai kemajuan proyek aplikasi Karir IO, melakukan *management* tim pengembang, dan membuat keputusan kunci yang mempengaruhi arah dan hasil pengembangan aplikasi Karir IO. Dalam hal ini, Kevin bertugas sebagai project manager yang akan menjalankan tanggung jawab tersebut.
6. **Assistant Project Manager** : Dalam aplikasi Karir IO yang dikembangkan dalam penelitian ini, *assistant project manager* bertugas untuk mendukung dan mendampingi *project manager*. Figo adalah *assitant project manager* aplikasi Karir IO yang bertugas sebagai pendamping Kevin dan membantunya dalam meelakukan penyusunan rencana-reencanaa proyek, membantu memikirkan *timeline* yang sesuai dengan pengembangan aplikasi Karir IO, memikirkan fitur apa saja yang harus dikelarkan pada periode pengembangan dan lainnya.

BAB IV

IMPLEMENTASI DAN EVALUASI

Implementasi dan evaluasi pada penelitian ini menguraikan hasil dari pengujian sistem backend aplikasi Karir Io dan melakukan evaluasi terhadap efektivitas sistem aplikasi untuk mengetahui sejauh mana Karir Io diimplementasikan di PT. Quantum dan analisis tentang kekuatan serta kelemahan dari sistem yang dihasilkan. Bab ini menjadi suatu sarana untuk merencanakan alur pengembangan aplikasi Karir IO dan penerapan tindakan yang terencana dan sesuai untuk mencapai tujuan penelitian.

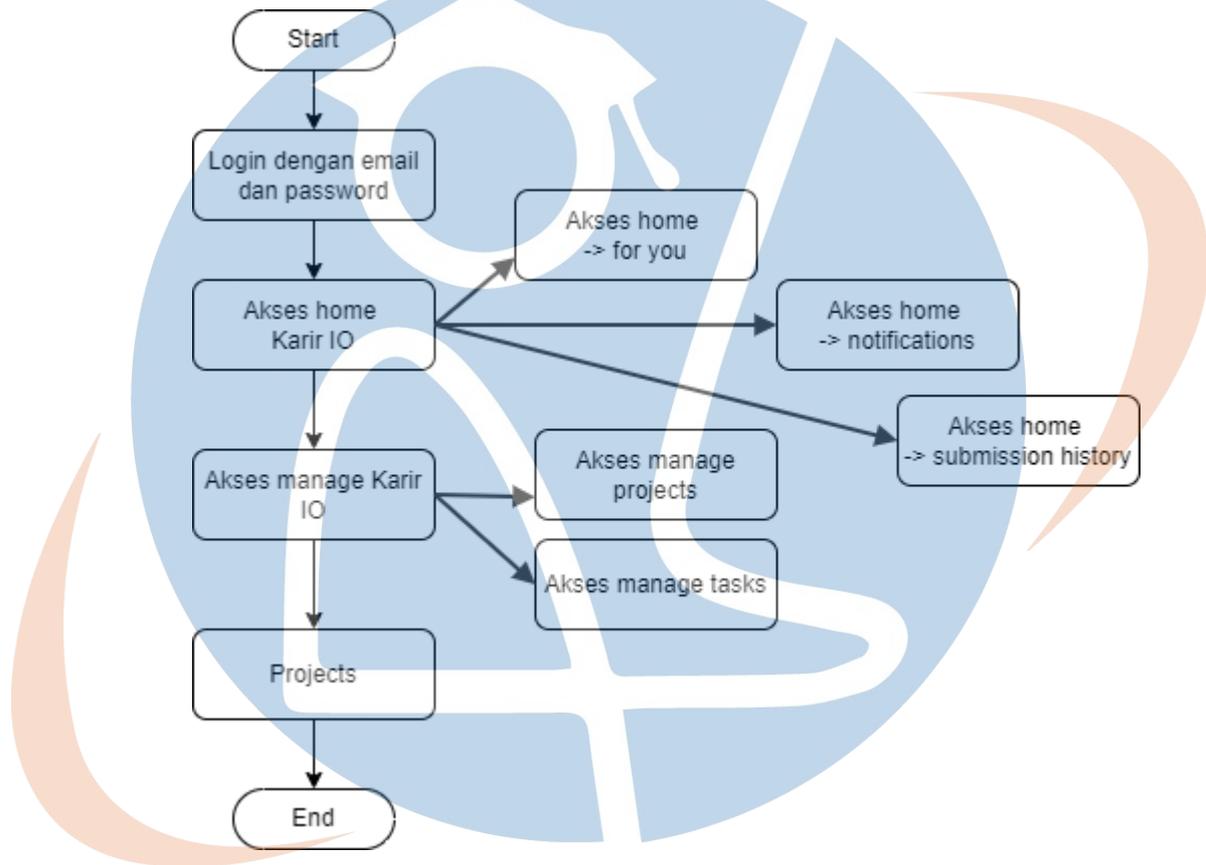
4.1 Analisis dan Perancangan

Analisis dan perancangan sistem rancang bangun aplikasi Karir IO adalah proses yang dilakukan untuk menentukan perencanaan sistem yang sesuai, melakukan proses pembuatan sistem dan menguraikan kondisi pengembangan sistem yang dibutuhkan. Pada penelitian ini, terdapat analisis kebutuhan (*requirement analysis*) yang menjadi instrumen dan bahan penelitian untuk menganalisis kebutuhan sistem lebih lanjut dengan melakukan wawancara, studi literature dari sumber-sumber yang terpercaya dan melakukan pembahasan lanjutan terhadap perangkat sistem yang akan digunakan. Dan untuk analisis sistem yang akan digunakan pada penelitian ini, harus melakukan analisis mendalam untuk memperoleh hasil rancangan yang dapat dipresentasikan dan diperoleh hasilnya pada penelitian [53].

4.1.1 Analisis Sistem Berjalan

Analisis sistem berjalan merupakan metode yang dilakukan oleh peneliti untuk memecahkan permasalahan atau hal-hal yang sedang terjadi pada penelitian, analisis sistem berjalan disebut juga sebagai teknik pemecahan masalah yang menguraikan sistem ke dalam komponen pembentuk agar mengetahui kinerja komponen tersebut telah berinteraksi satu sama lainnya atau tidak pada suatu penelitian [54]. Proses sistem manajemen tugas yang terjadi pada PT. Quantum masih menggunakan sistem manual, tugas dan proyek yang harus diselesaikan oleh karyawan masih disampaikan melalui *message* pribadi sehingga tugas dan proyek yang harus diselesaikan lebih dulu menjadi hal yang diselesaikan di akhir, tentu saja hal tersebut membuat PT. Quantum menerima berbagai permasalahan terhadap tenggat waktu penyelesaian proyek aplikasi. Melihat dari permasalahan tersebut, penulis kemudian berinisiatif untuk rancang bangun backend aplikasi Karir IO berbasis *web* agar mempermudah manajemen tugas kantor.

Rancang bangun *backend* aplikasi Karir IO ini terdiri dari 2 *role* utama dengan fungsi dan tugas yang berbeda yaitu *role coordinator* dan *role member*. *Coordinator* mempunyai tanggung jawab untuk membuat *task*, membuat *project*, memasukkan *member* pada *project* tertentu, memberikan *task* pada *user* tertentu dan *review task*. Sedangkan *role member* memiliki hak yang dibatasi hanya pada kegiatan *member* itu sendiri seperti halnya *member* berhak untuk mengerjakan *task* dimana *user* aplikasi Karir IO dapat melakukan perubahan status tugas dari “*active*” menjadi “*to review*”.



Gambar 4.1 Alur Aplikasi Karir IO [A7].

Sumber Gambar : Dokumentasi Pribadi

Berdasarkan Gambar 4.1, saat akan melakukan *login* ke aplikasi Karir IO, maka pengguna aplikasi (*role coordinator* atau *role member*) harus melakukan *login* dahulu dengan menginputkan *email* dan *password* sesuai. Setelah berhasil *login*, pengguna dengan *role coordinator* dapat mengakses *home* aplikasi yang berisikan fitur *for you*, *notifications* dan *submission history*. Fitur *for you* akan menginformasikan proyek yang tersedia, *notifications* akan menyampaikan notifikasi dan *submission history* bertujuan untuk memantau riwayat penyerahan proyek yang terselesaikan. Informasi *manager* yang berisi fitur *manage projects*, *manage tasks* dan menu *projects* juga ditampilkan.

Pengguna aplikasi dengan *role* akses *member* ketika mengakses aplikasi Karir IO juga melakukan *login* namun dengan hak akses yang dibatasi penggunaannya karena setiap *role* memiliki tanggung jawabnya. Setelah *login*, *role member* akan memperoleh informasi seluruh proyek dan melakukan perubahan status proyek pada aplikasi seperti perubahan atas status *task* dari status “*active*” menjadi “*under review*” atau “*done*”.

4.1.1.1 User Stories

User stories dilakukan setelah pengisian survei *user* aplikasi dan wawancara khusus pada *teach leader* mengenai kebutuhan rancang bangun aplikasi Karir IO. Setelah memperoleh data, penulis menganalisis lebih lanjut kebutuhan, fitur dan fungsi aplikasi untuk mencapai kebutuhan *user stories* serta mencapai tujuan perancangan aplikasi Karir IO dan kemudian disusun dalam tabel *user stories* berikut dibawah ini:

Table 4.1.1.1 User Stories

No	Sebagai	Saya Ingin	Sehingga
1	<i>Coordinator</i>	Melihat daftar semua <i>member</i> yang terdaftar	Saya mengelola data <i>member</i> , verifikasi, dan memasukkan <i>member</i> pada proyek
2	<i>Member</i>	Melakukan <i>login</i> dengan <i>input email</i> dan <i>password</i>	Saya dapat mengakses aplikasi Karir IO setelah <i>input email</i> dan <i>password</i>
3	<i>Coordinator</i>	Membuat proyek baru dengan <i>body</i> yang sesuai	Saya telah membuat suatu proyek baru dengan <i>body</i> proyek yang sesuai
4	<i>Coordinator</i>	Memberikan tugas ke <i>member</i> sesuai pekerjaan	Saya mampu memberikan tugas berupa “pekerjaan” kepada <i>member</i>
5	<i>Coordinator</i>	Menampilkan semua <i>member</i> yang ada	Saya bisa menampilkan dan mengetahui semua data <i>member</i>
6	<i>Coordinator</i>	Menampilkan semua proyek yang telah dibuat	Saya dapat menampilkan berbagai proyek yang telah dibuat pada aplikasi
7	<i>Member</i>	Menampilkan proyek <i>detail</i> yang telah diberikan	Saya dapat mengetahui <i>detail</i> dari setiap proyek yang tersedia di aplikasi
8	<i>Coordinator</i>	Melakukan <i>update</i> data proyek setiap <i>progress</i>	Saya dapat melakukan <i>update</i> data proyek dengan keadaan yang sebenarnya
9	<i>Member</i>	Melihat seluruh tugas dan melakukan pencarian tugas	Saya mengetahui tugas berdasarkan status tugas dan melakukan pencarian tugas dengan menuliskan nama tugas

No	Sebagai	Saya Ingin	Sehingga
10	<i>Member</i>	Ubah status tugas dari “ <i>active</i> ” menjadi “ <i>under review</i> ” atau “ <i>done</i> ”	Saya dapat melakukan berbagai perubahan status pada tugas yang sudah dikerjakan berdasarkan status tugas
11	<i>Member</i>	Memberikan komentar dan melihat komentar orang lain pada proyek	Saya dapat memberikan komentar dan melihat komentar yang disampaikan pada proyek yang akan dikerjakan
12	<i>Coordinator</i>	Membuat tugas untuk <i>member</i> berdasarkan id	Saya mampu membuat tugas baru yang diberikan pada <i>member</i> berdasarkan id
13	<i>Coordinator</i>	Melakukan perubahan data proyek (<i>edit data</i>)	Saya dapat melakukan berbagai perubahan data proyek yang terdaftar pada aplikasi sesuai dengan kebenaran datanya
14	<i>Member</i>	Mendapatkan notifikasi bila terdapat tugas baru	Saya mendapatkan notifikasi bila ada tugas baru yang harus diselesaikan
15	<i>Coordinator</i>	Memperoleh notifikasi terkait <i>progress</i> tugas	Saya menerima notifikasi bila <i>member</i> menyelesaikan <i>task</i> yang diberikan
16	<i>Member</i>	Mengakses <i>submission history</i> / riwayat proyek	Saya mendapatkan informasi seputar riwayat proyek yang telah terselesaikan
17	<i>Coordinator</i>	Mengakses <i>submission history</i>	Saya dapat memantau seluruh riwayat proyek yang selesai dikerjakan <i>member</i>
18	<i>Coordinator</i>	Melakukan pemantauan proyek pada fitur <i>manage projects</i>	Saya bisa menambahkan proyek, melihat detail proyek (nama proyek, email, nomor <i>whatsapp</i> , nomor telepon) dan melakukan hapus proyek yang tidak dibutuhkan
19	<i>Coordinator</i>	Melakukan pemantauan seluruh tugas pada fitur <i>manage tasks</i>	Saya dapat menambahkan <i>task</i> , melihat <i>detail task</i> (nama <i>task</i> , email, nomor <i>whatsapp</i> , nomor telepon) dan melakukan hapus tugas yang tidak dibutuhkan
20	<i>Coordinator</i>	Mengakses menu <i>projects</i>	Saya dapat melihat detail proyek dan memperoleh informasi yang saya butuhkan pada aplikasi Karir IO

Sumber Tabel: Dokumentasi Pribadi

4.1.1.2 Product Backlog

Berdasarkan alur *user stories*, penulis membuat rangkuman penjelasan tugas dan fitur yang harus tersedia, prioritasnya dan estimasi *story points* yang menggambarkan ukuran kompleksitas fitur serta waktu setiap pengerjaan *task*. *Product backlog* ini juga menjelaskan penilaian prioritas setiap *task* berdasarkan kebutuhan aplikasi. Mengukur kompleksitas membutuhkan skala *story points* dan estimasi waktu pengerjaan. Adapun perhitungan nilai *story points* tersebut disampaikan pada tabel berikut ini:

Table 4.1.1.2.1 Estimasi Story Points

No	Nilai	Keterangan Nilai	Estimasi Waktu
1	1 Nilai	Tugas yang sangat sederhana, diketahui penyebab permasalahannya dan tidak memerlukan banyak penelitian, perubahan dokumentasi atau koordinasi dengan tim	2-4 Jam
2	2 Nilai	Tugas yang sedikit lebih kompleks tetapi masih bisa diprediksi, memerlukan analisis atau diskusi anggota tim, memerlukan penyelidikan atau penambahan fitur minor	4-8 Jam
3	3 Nilai	Tugas yang memiliki kompleksitas sedang, perlu banyak usaha dan koordinasi karena melibatkan suatu komponen atau modul yang memerlukan analisis mendalam	8-16 Jam
4	5 Nilai	Tugas yang cukup kompleks karena mencakup berbagai aspek dari proyek, memerlukan perencanaan, koordinasi antar anggota tim dan iterasi atas berbagai tahap	16-24 Jam

Sumber Tabel: <https://images.app.goo.gl/BEH994ppqT49co2NU7>

Setiap *product backlog* akan memiliki estimasi waktu pengerjaan yang sesuai dengan kebutuhan aplikasi dan tergantung pada kesepakatan tim pengembang. Apabila *product backlog* memiliki prioritas tinggi maka nilai *story points* pun akan disesuaikan kembali. Berdasarkan informasi yang disampaikan ini, penulis berharap agar rancang bangun *backend* aplikasi Karir IO dapat diselesaikan dengan lebih terstruktur, jelas, dan efisien. Adapun berikut adalah tabel *product backlog* yang akan diselesaikan:

Table 4.1.1.2.2 Product Backlog

No	Task	Prioritas	Story Points	Deskripsi
1	Penentuan tahapan dan rancangan penelitian	Sedang	1	Menyesuaikan tahapan dan rancangan penelitian yang sesuai untuk mempermudah perancangan
2	Penyusunan <i>user requirement</i>	Sedang	1	Merangkum kebutuhan pengguna secara keseluruhan
3	Pembentukan <i>user stories</i>	Sedang	1	Membentuk serangkaian alur <i>user stories</i> yang dimanfaatkan untuk panduan perancangan sistem aplikasi
4	Penggambaran <i>use case diagram</i>	Sedang	2	Membentuk alur interaksi aktor dengan fitur sistem yang dibentuk
5	Penggambaran <i>class diagram</i>	Sedang	2	Menentukan desain struktur serta hubungan antar fitur aplikasi yang memodelkan logika dan relasi data
6	Penggambaran <i>activity diagram</i>	Sedang	2	Menggambarkan alur proses yang terjadi dari setiap fitur aplikasi
7	Penggambaran <i>sequence diagram</i>	Sedang	2	Menggambarkan skenario dan rangkaian langkah yang dilakukan dengan <i>output</i> berupa suatu respon
8	Pembentukan <i>schema database</i>	Tinggi	3	Membentuk <i>schema database</i> yang dapat menyimpan informasi data
9	Menerapkan <i>API login</i>	Tinggi	3	Menerapkan <i>API</i> untuk pengelolaan proses <i>login</i> pengguna aplikasi
10	Menerapkan <i>API register</i>	Tinggi	3	Mengimplementasikan <i>API</i> yang bisa mengizinkan pengguna melakukan pendaftaran akun baru
11	Menerapkan <i>API projects</i>	Tinggi	3	Menerapkan <i>API projects</i> untuk membuat projek baru dengan <i>body</i> yang sesuai serta menampilkan semua projek yang telah dibuat

No	Task	Prioritas	Story Points	Deskripsi
12	Menampilkan <i>API projects detail</i> dengan id	Sedang	2	Menerapkan <i>API projects detail</i> dengan id sesuai untuk menampilkan <i>detail</i> proyek berdasarkan id
13	Menerapkan <i>API update project</i> dengan id	Sedang	2	Menerapkan <i>API update project</i> dengan id untuk mempermudah <i>update</i> data proyek berdasarkan id
14	Menerapkan <i>API project members</i>	Tinggi	3	Menambahkan <i>API project members</i> untuk menambahkan data <i>member</i> ke proyek dengan id dan <i>body</i> sesuai
15	Menerapkan <i>API project members</i> dengan id	Tinggi	3	Menambahkan <i>member</i> ke <i>project</i> dengan id dan <i>body</i> sesuai untuk mempermudah penambahan <i>member</i>
16	Menerapkan <i>API project members</i> yang ada di <i>project</i>	Sedang	2	Menampilkan <i>member</i> yang ada pada <i>project</i> dengan id sesuai untuk mempermudah pendataan <i>member</i>
17	Menerapkan <i>API tasks</i>	Tinggi	3	Membuat tugas-tugas pada proyek id pada pengguna agar mempermudah pengelolaan dan manajemen tugas
18	Menerapkan <i>API tasks</i> yang <i>active</i>	Sedang	2	Memperoleh seluruh data tugas yang <i>active</i> pada <i>project</i> berdasarkan dengan id yang dipilih
19	Menerapkan <i>API comments tasks</i>	Sedang	2	Membuat <i>API comments tasks</i> pada aplikasi Karir IO untuk membuat suatu komentar pada <i>task</i>
20	Menerapkan <i>API comments tasks</i> dengan id	Sedang	2	Membentuk <i>API comments tasks</i> dengan id untuk menampilkan komentar pada <i>task</i> berdasarkan id

No	Task	Prioritas	Story Points	Deskripsi
21	Menerapkan <i>API comments tasks</i> dengan <i>comment id</i>	Sedang	2	Menerapkan <i>API comments tasks</i> untuk membalas komentar pada suatu <i>task</i> dengan <i>comment id</i>
22	Menerapkan <i>API comments replies</i>	Sedang	2	Menerapkan <i>API comments tasks</i> untuk melihat balasan yang tersedia pada komentar proyek aplikasi

Sumber Tabel: Dokumentasi Pribadi

Melihat dari informasi yang disampaikan pada **Table 4.1.1.2.2** tersebut, dapat disimpulkan bahwa *product backlog* pada penelitian ini dibuat berdasarkan alur *user stories* rancang bangun aplikasi Karir IO. Dalam hal ini, *product backlog* diharapkan dapat memberikan informasi mendalam kepada pengembang aplikasi dan pembaca penelitian terkait dengan fitur, fungsi dan modul aplikasi yang harus diselesaikan oleh tim pengembang sesuai dengan periode waktu tertentu. Dari *product backlog*, tujuan rancang bangun backend aplikasi Karir IO dapat dicapai dengan mudah dan terstruktur.

4.1.1.3 *Sprint Planning*

Tahapan yang selanjutnya dilakukan adalah menentukan alur perancangan *sprint planning* berdasarkan dengan *product backlog* yang telah disampaikan. Tahap *sprint planning* ini dilakukan dalam 5 tahapan dengan waktu pengerjaan sekitar 14 hari dari setiap tahapan *sprint* yang dilakukan. Adapun berikut adalah pembahasan lebih lanjut:

Table 4.1.1.2 3 *Sprint Planning*

No	<i>Sprint</i>	<i>Product Backlog</i>	Waktu
1	<i>Sprint 1</i>	Penentuan tahapan dan rancangan penelitian, penyusunan <i>user requirement</i> , pembentukan <i>user stories</i>	14 Hari
2	<i>Sprint 2</i>	Penggambaran <i>use case diagram</i> , penggambaran <i>class diagram</i> , penggambaran <i>activity diagram</i> , penggambaran <i>sequence diagram</i> , pembentukan <i>schema database</i>	14 Hari
3	<i>Sprint 3</i>	Menerapkan <i>API login</i> , menerapkan <i>API register</i> , menerapkan <i>API projects</i> , menampilkan <i>API projects detail</i> dengan id, menerapkan <i>API update project</i> dengan id	14 Hari

No	<i>Sprint</i>	<i>Product Backlog</i>	Waktu
4	<i>Sprint 4</i>	Menerapkan <i>API project members</i> , menerapkan <i>API project members</i> dengan id, menerapkan <i>API project members</i> yang ada di <i>project</i> , menerapkan <i>API tasks</i> , menerapkan <i>API tasks</i> yang <i>active</i>	14 Hari
5	<i>Sprint 5</i>	Menerapkan <i>API comments tasks</i> , menerapkan <i>API comments tasks</i> dengan id, menerapkan <i>API comments tasks</i> dengan <i>comment id</i> , menerapkan <i>API comments replies</i>	14 Hari

Sumber Tabel: Dokumentasi Pribadi

Tahap *sprint planning* dilakukan untuk menentukan pekerjaan yang diselesaikan selama masa *sprint* yang berlangsung. Tujuan pelaksanaan *sprint planning* ini dapat dicapai oleh tim selama masa *sprint* berlangsung apabila tim memahami dengan jelas apa yang harus dicapai dalam *sprint*, turut berkolaborasi dan berkomitmen atas setiap *sprint* yang dilakukan, dan melalui *sprint* dengan perencanaan yang tepat dan estimasi waktu yang jelas. Setelah menentukan bagian atas *sprint planning* maka yang harus dilakukan selanjutnya adalah melaksanakan *sprint* dengan baik dan melakukan *review*.

4.1.1.4 *Sprint Review*

Sprint review dilakukan pada setiap akhir pelaksanaan *sprint* untuk memastikan fitur dan fungsi aplikasi telah berjalan sesuai dengan alur *product backlog*. Tujuan dari *sprint review* adalah untuk memeriksa hasil kerja yang diselesaikan tim pengembang selama *sprint*, memperoleh umpan balik dari pemimpin kepentingan (*teach leader*) dan menentukan langkah yang seharusnya diambil selama masa perancangan aplikasi. Berikut adalah hasil evaluasi *sprint review* dari setiap tahap *sprint* yang dilakukan tim:

1. *Sprint 1*

a. *Product Backlog*

Selama masa pengembangan aplikasi Karir IO, *sprint* dilakukan secara bertahap bersama dengan tim pengembang dan menghasilkan hasil *sprint* yang akan dievaluasi. Dalam hal ini, pelaksanaan *sprint* aplikasi Karir IO memperoleh hasil *product backlog* yang disampaikan pada tabel berikut ini:

Table 4.1.1.4.1 Sprint 1 - Sprint Review

<i>Product Backlog</i>	Deskripsi	Hasil
Penentuan tahapan dan rancangan penelitian	Menyesuaikan tahapan dan rancangan penelitian yang sesuai untuk mempermudah perancangan	Sesuai
Penyusunan <i>user requirement</i>	Merangkum kebutuhan pengguna aplikasi Karir IO secara keseluruhan	Sesuai
Pembentukan <i>user stories</i>	Membentuk serangkaian alur <i>user stories</i> yang dimanfaatkan untuk panduan perancangan sistem	Sesuai

Sumber Tabel: Dokumentasi Pribadi

b. Hasil Sprint Review

Berdasarkan **Table 4.1.1.4.1**, disimpulkan bahwa sebelum pengembangan aplikasi Karir IO, tim pengembang aplikasi harus menentukan pondasi utama dalam mencapai tujuan pengembangan aplikasi dengan menentukan tahapan dan rancangan penelitian yang sesuai, kemudian menyusun *user requirement* berdasarkan kebutuhan pengguna secara keseluruhan, serta membentuk rangkaian alur *user stories* sebagai panduan dalam perancangan sistem. Berdasarkan *sprint* yang dilakukan, maka didapatkan hasil bahwa seluruh tahapan yang tersedia pada *sprint 1* berhasil dilakukan dan mencapai hasil desain yang sesuai dengan deskripsi *product backlog* yang telah dibuat.

2. *Sprint 2*

a. Product Backlog

Setelah berhasil menyelesaikan *product backlog* pada *sprint 1*, seluruh tim harus melaksanakan *sprint 2* sesuai dengan deskripsi *product backlog* yang telah ditentukan. Berikut pembahasan lebih lanjut terkait *sprint 2* rancang bangun aplikasi Karir IO:

Table 4.1.1.4. 2 Sprint 2 - Sprint Review

<i>Product Backlog</i>	Deskripsi	Hasil
Penggambaran <i>use case</i> diagram	Membentuk alur interaksi aktor dengan fitur sistem yang dibentuk	Sesuai
Penggambaran <i>class</i> diagram	Menentukan desain struktur serta hubungan antar fitur aplikasi yang memodelkan logika dan relasi data	Sesuai

Penggambaran <i>activity</i> diagram	Menggambarkan alur proses yang terjadi dari setiap fitur aplikasi	Sesuai
Penggambaran <i>sequence</i> diagram	Menggambarkan skenario dan rangkaian langkah yang dilakukan dengan <i>output</i> berupa suatu respon	Sesuai
Pembentukan <i>schema database</i>	Membentuk <i>schema database</i> yang dapat menyimpan informasi data	Sesuai

Sumber Tabel: Dokumentasi Pribadi

b. Hasil Sprint Review

Berdasarkan *product backlog* yang telah dihasilkan dalam perancangan aplikasi Karir IO, diperoleh hasil bahwa *use case* diagram berfungsi untuk menggambarkan dan membentuk alur interaksi antar aktor dengan fitur sistem yang akan dibangun, penggambaran *class* diagram untuk menentukan desain struktur dan hubungan antar fitur aplikasi yang memodelkan logika dan relasi data, penggambaran *activity* diagram untuk menggambarkan alur proses aplikasi yang terjadi dari setiap fitur aplikasi agar mudah dipahami, penggambaran *sequence* diagram untuk menggambarkan skenario dan langkah yang harus dilakukan guna mencapai *output* berupa respon yang sesuai, serta pembentukan *schema database* yang dimanfaatkan untuk menyimpan informasi data aplikasi Karir IO yang sesuai dengan desain dan deskripsi *product backlog*.

3. Sprint 3

a. Product Backlog

Setelah berhasil menyelesaikan *product backlog* pada *sprint 1* dan *sprint 2*, maka tahap *sprint 3* harus dilakukan oleh tim pengembang aplikasi. Adapun berikut adalah tabel *sprint 3* rancang bangun aplikasi Karir IO berdasarkan dengan *product backlog*:

Table 4.1.1.4.3 Sprint 3 - Sprint Review

<i>Product Backlog</i>	Deskripsi	Hasil
Menerapkan <i>API login</i>	Menerapkan <i>API</i> untuk pengelolaan proses <i>login</i> pengguna aplikasi	Sesuai
Menerapkan <i>API register</i>	Mengimplementasikan <i>API</i> yang bisa mengizinkan pengguna melakukan pendaftaran akun baru	Sesuai

Menerapkan <i>API projects</i>	Menerapkan <i>API projects</i> untuk membuat proyek baru dengan <i>body</i> yang sesuai serta menampilkan semua proyek yang telah dibuat	Sesuai
Menampilkan <i>API projects detail</i> dengan id	Menerapkan <i>API projects detail</i> dengan id sesuai untuk menampilkan <i>detail</i> proyek berdasarkan id	Sesuai
Menerapkan <i>API update project</i> dengan id untuk menampilkan detail proyek berdasarkan id	Menerapkan <i>API update project</i> dengan id untuk mempermudah <i>update</i> data proyek berdasarkan id	Sesuai

Sumber Tabel: Dokumentasi Pribadi

b. Hasil Sprint Review

Setelah melaksanakan *sprint planning* tiba saatnya untuk melakukan evaluasi *sprint* dan merangkum catatan evaluasinya pada *sprint review*, pada *sprint review 3* diperoleh hasil bahwa penerapan *API* untuk pengelolaan proses *login* pengguna aplikasi berjalan dengan seharusnya, implementasi *API* yang dapat mengizinkan pengguna melakukan pendaftaran akun baru sudah dapat diterapkan, *API projects* untuk membuat proyek baru dengan *body* yang sesuai serta menampilkan semua proyek yang telah dibuat telah berhasil di bentuk, *API projects detail* dengan id sesuai untuk menampilkan *detail* proyek berdasarkan id telah diterapkan, dan *API update project* dengan id untuk mempermudah *update* data proyek berdasarkan id sudah disesuaikan dengan *product backlog* yang dibentuk.

4. Sprint 4

a. Product Backlog

Tim pengembang aplikasi Karir IO melaksanakan proses *sprint* berdasarkan dengan tahapan *product backlog* yang telah dibuat sebelumnya bersama dengan tim pengembang aplikasi. Setelah melaksanakan *sprint*, tiba saatnya kepada tim untuk melakukan *review* atau yang dikenal dengan kegiatan *sprint review* untuk memastikan setiap langkah pengembangan di setiap tahap *sprint* yang dilakukan, telah sesuai oleh kebutuhan dan tujuan rancang bangun aplikasi. Adapun berikut dibawah adalah tabel *sprint 4* berdasarkan dengan *product backlog* yang telah disusun sebelumnya:

Table 4.1.1.4.4 Sprint 4 - Sprint Review

<i>Product Backlog</i>	Deskripsi	Hasil
Menerapkan <i>API project members</i>	Menambahkan <i>API project members</i> untuk menambahkan data <i>member</i> ke proyek dengan id dan <i>body</i> sesuai	Sesuai
Menerapkan <i>API project members</i> dengan id	Menambahkan <i>member</i> ke <i>project</i> dengan id dan <i>body</i> sesuai untuk mempermudah penambahan <i>member</i>	Sesuai
Menerapkan <i>API project members</i> yang ada di <i>project</i>	Menampilkan <i>member</i> yang ada pada <i>project</i> dengan id yang sesuai untuk mempermudah pendataan <i>member</i>	Sesuai
Menerapkan <i>API tasks</i>	Membuat tugas-tugas pada proyek id pada pengguna agar mempermudah pengelolaan dan manajemen tugas	Sesuai
Menerapkan <i>API tasks</i> yang <i>active</i>	Memperoleh seluruh data tugas yang <i>active</i> pada <i>project</i> berdasarkan dengan id	Sesuai

Sumber Tabel: Dokumentasi Pribadi

b. Hasil Sprint Review

Berdasarkan *product backlog*, diperoleh hasil bahwa *sprint 4* ini berhasil dicapai tim pengembang dengan menambahkan *API project members* untuk menambahkan data *member* ke proyek dengan id dan *body* sesuai, penambahan *member* ke *project* dengan id dan *body* sesuai membantu *coordinator* dalam mempermudah penambahan *member*, *member* pada *project* dengan id yang sesuai ditampilkan dengan mudah untuk membantu pendataan *member*, *sprint 4* ini membantu pembuatan tugas pada proyek id guna pengelolaan data serta manajemen seluruh data tugas yang *active* pada *project* berdasarkan id yang dipilih sehingga data *member* dapat diperoleh dengan baik.

5. *Sprint 5*

a. Product Backlog

Tim pengembang aplikasi Karir IO melaksanakan proses *sprint* berdasarkan dengan tahapan *product backlog* yang telah dibuat. Adapun berikut adalah tabel *sprint 5* rancang bangun aplikasi Karir IO berdasarkan dengan *product backlog*:

Table 4.1.1.4.5 Sprint 5 - Sprint Review

<i>Product Backlog</i>	Deskripsi	Hasil
Menerapkan <i>API comments tasks</i>	Membuat <i>API comments tasks</i> pada aplikasi Karir IO untuk membuat suatu komentar pada <i>task</i>	Sesuai
Menerapkan <i>API comments tasks</i> dengan id	Membentuk <i>API comments tasks</i> dengan id untuk menampilkan komentar pada <i>task</i> berdasarkan id	Sesuai
Menerapkan <i>API comments tasks</i> dengan <i>comment id</i>	Menerapkan <i>API comments tasks</i> untuk membalas komentar pada suatu <i>task</i> dengan <i>comment id</i>	Sesuai
Menerapkan <i>API comments replies</i>	Menerapkan <i>API comments tasks</i> untuk melihat balasan pada komentar	Sesuai

Sumber Tabel: Dokumentasi Pribadi

b. Hasil Sprint Review

Sprint review 5 dilakukan bersama dengan tim pengembang dan menyesuaikan *product backlog* yang telah dibentuk sebelumnya. Dalam kesempatan *sprint review 5*, terdapat beberapa tahapan yang telah selesai dikerjakan, dimana pada *sprint* ini tim telah berhasil menerapkan *API comments tasks* pada aplikasi Karir IO untuk membuat suatu komentar pada *task*, berhasil membentuk *API comments tasks* dengan id untuk menampilkan komentar pada *task* berdasarkan id, telah menerapkan *API comments tasks* untuk membalas komentar pada suatu *task* dengan *comment id* dan menerapkan *API comments tasks* untuk melihat balasan pada komentar.

4.1.2 Analisis Kebutuhan Sistem

Kebutuhan sistem pengembangan perangkat lunak adalah salah satu pemenuhan persyaratan, kriteria, serta kemampuan yang harus dipenuhi oleh tim pengembang aplikasi agar dapat membangun sistem aplikasi dan memenuhi kebutuhan sistem yang seharusnya. Analisis kebutuhan sistem disebut sebagai proses kritis yang dilakukan untuk mengidentifikasi, mendokumentasikan dan memvalidasi kebutuhan pengguna bahwa sistem yang dikembangkan bisa memenuhi ekspektasi dan kebutuhan bisnis. Dengan analisis tepat, siklus pengembangan perangkat lunak dapat mencapai keberhasilan proyek dan mengoptimalkan penggunaan sumber daya suatu sistem [55].

Analisis kebutuhan sistem rancang bangun backend aplikasi Karir IO berbasis *web* menggunakan *NestJS* dengan studi kasus PT. Quantum Teknologi Nusantara pada penelitian ini melibatkan identifikasi dan pendokumentasian kebutuhan yang lebih terperinci guna membangun aplikasi manajemen tugas yang efektif, efisien, dan sesuai dengan tujuan bisnis. Berikut adalah rincian dari analisis kebutuhan sistem aplikasi Karir IO untuk membangun *backend* aplikasi yang andal dan *scalable*:

1. Kebutuhan Teknologi

Rancang bangun backend aplikasi Karir IO membutuhkan *framework NestJS* sebagai *framework backend* karena arsitektur *framework* yang bersifat modular dan memperoleh dukungan baik untuk *typescript*, menggunakan *database MySQL* untuk manajemen data relasional yang andal dan tepat serta menggunakan *server* resmi dari PT. Quantum Teknologi Nusantara untuk menjamin keamanan penggunaan.

2. Kebutuhan Pengembangan

Rancang bangun backend aplikasi Karir IO juga membutuhkan suatu metodologi pengembangan yang tepat guna, memberikan fungsi yang relevan dan bisa diandalkan. Dalam hal ini penulis menggunakan metodologi *agile* dengan pendekatan *scrum* untuk mencapai iterasi cepat dan fleksibilitas terhadap berbagai perubahan yang terjadi.

3. Kebutuhan Pengujian

Pengujian yang dilakukan untuk mencapai tujuan rancang bangun backend aplikasi Karir IO adalah dengan mengimplementasikan unit testing yang memastikan setiap komponene bekerja sesuai dengan yang diharapkan dan menguji integrasi antar setiap komponen untuk memastikan sistem berfungsi secara keseluruhan dimana pengujian aplikasi Karir IO menggunakan metode *black box testing* dan *skala likert*.

4.1.3 Perancangan Sistem

Tim pengembang melaksanakan proses *sprint* berdasarkan dengan tahapan *product backlog* yang dibuat sebelumnya bersama tim pengembang aplikasi. Proses ini dilakukan untuk memastikan setiap langkah pengembangan *sprint* yang dilakukan berhasil mencapai kebutuhan dan tujuan rancang bangun aplikasi. Perancangan sistem adalah penggambaran, pembuatan alur sketsa dan perencanaan proses kegiatan dengan teknik bervariasi yang melibatkan deskripsi arsitektur sistem serta *detail* komponen yang berkaitan [56]. Perancangan sistem menjadi jenis kebutuhan yang harus dipenuhi untuk mempermudah pengembangan dan memastikan kebutuhan bisnis terpenuhi.

4.1.3.1 Identifikasi Aktor

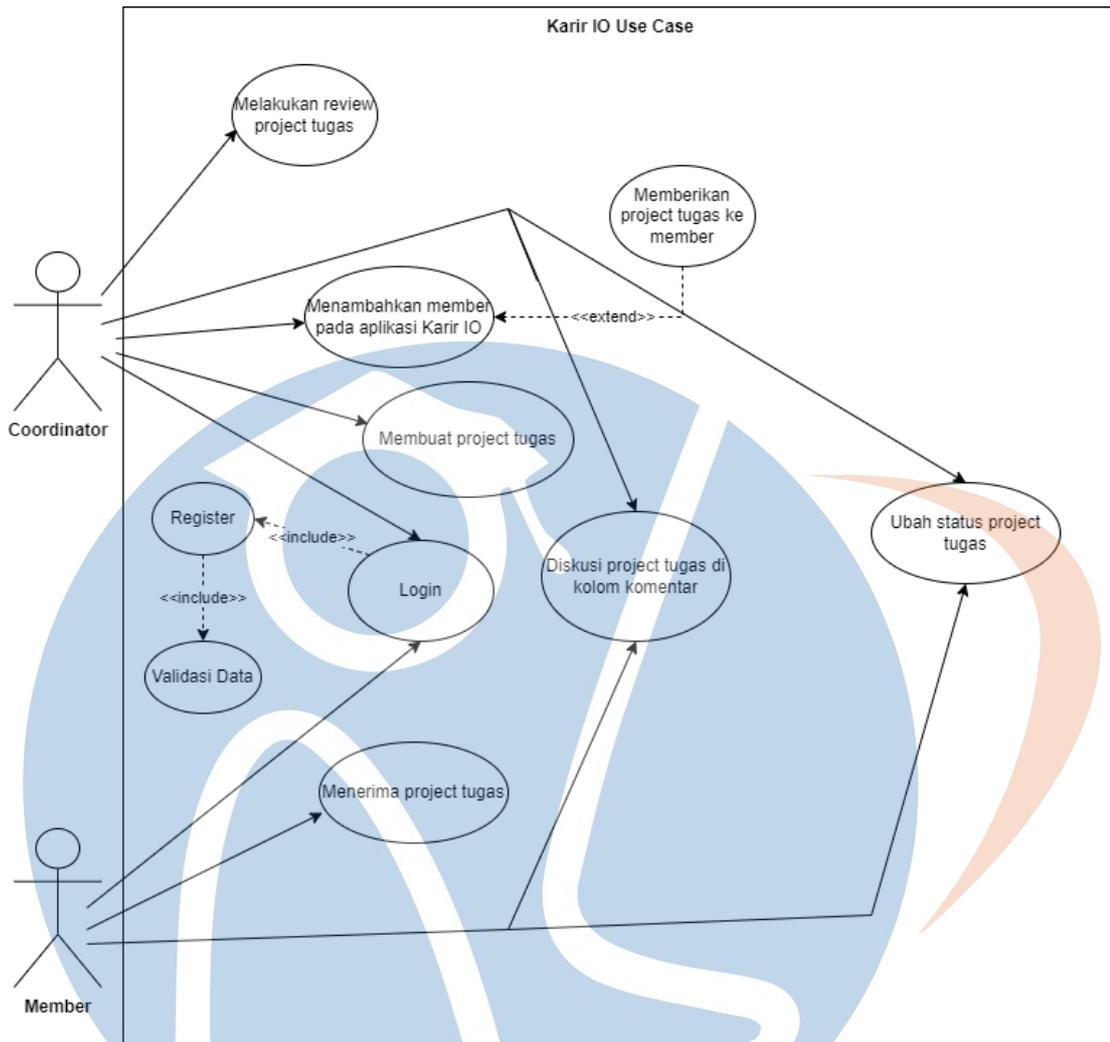
Rancang Bangun Backend Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara adalah judul penelitian ini yang menerangkan bagaimana perancangan aplikasi manajemen tugas kantor yang diberi nama aplikasi Karir IO PT. Quantum Teknologi Nusantara dapat memenuhi tujuan pengembangan. Pada penelitian ini, terdapat 2 *role* dengan hak akses yang berbeda dimana satu *role* memiliki banyak akses dan *role* lainnya memiliki akses yang dibatasi.

Pada aplikasi Karir IO, terdapat 2 *role* utama (*Coordinator* dan *Member*) berikut:

1. **Role Coordinator** memiliki tanggung jawab untuk membuat *task* baru pada aplikasi, membuat *project* yang akan diselesaikan, memasukkan *member* ke dalam *project* tertentu, memberikan *task* pada *user* tertentu yang memang memiliki hak untuk mengakses nya dan melakukan *review task* tanpa melihat tim yang mengembangkan *task* tersebut karena *review* tersebut akan dianalisis lebih lanjut oleh tim pengembang dan diselesaikan dalam kesempatan *sprint* rutin selama proses rancang bangun aplikasi berjalan.
2. **Role Member** memiliki hak akses yang dibatasi (tidak bisa mengakses beberapa fitur aplikasi seperti *role coordinator*). Pada aplikasi Karir IO, *role member* hanya dapat mengakses fitur aplikasi sesuai dengan kegiatan yang *member* tersebut lakukan sendiri seperti hal-nya *member* memiliki hak akses untuk mengerjakan *task* yang diberikan oleh *coordinator*, *role member* juga dapat melakukan perubahan status tugas yang tersedia pada aplikasi dari status “*active*” menjadi “*to review*” atau ketika “*done*”.

4.1.3.2 Use Case Diagram

Use case diagram pada *project* aplikasi Karir IO akan memberikan gambaran nyata yang berkaitan dengan fungsionalitas aplikasi yang akan digunakan oleh pengguna aplikasi Karir IO dalam melakukan manajemen tugas kantor pada PT. Quantum secara menyeluruh. Melalui *use case* diagram, tim pengembang berharap agar aplikasi Karir IO yang dikembangkan dalam kesempatan ini mampu menginterpretasikan rangkaian alur interaksi yang terjadi di antara aktor dengan sistem aplikasi. Selain itu, *use case* diagram ini juga diharapkan dapat membantu proses identifikasi kebutuhan pengguna secara lebih *detail*, sehingga pengembangan fitur-fitur aplikasi Karir IO menjadi tepat sasaran dan sesuai dengan kebutuhan operasional perusahaan PT. Quantum.



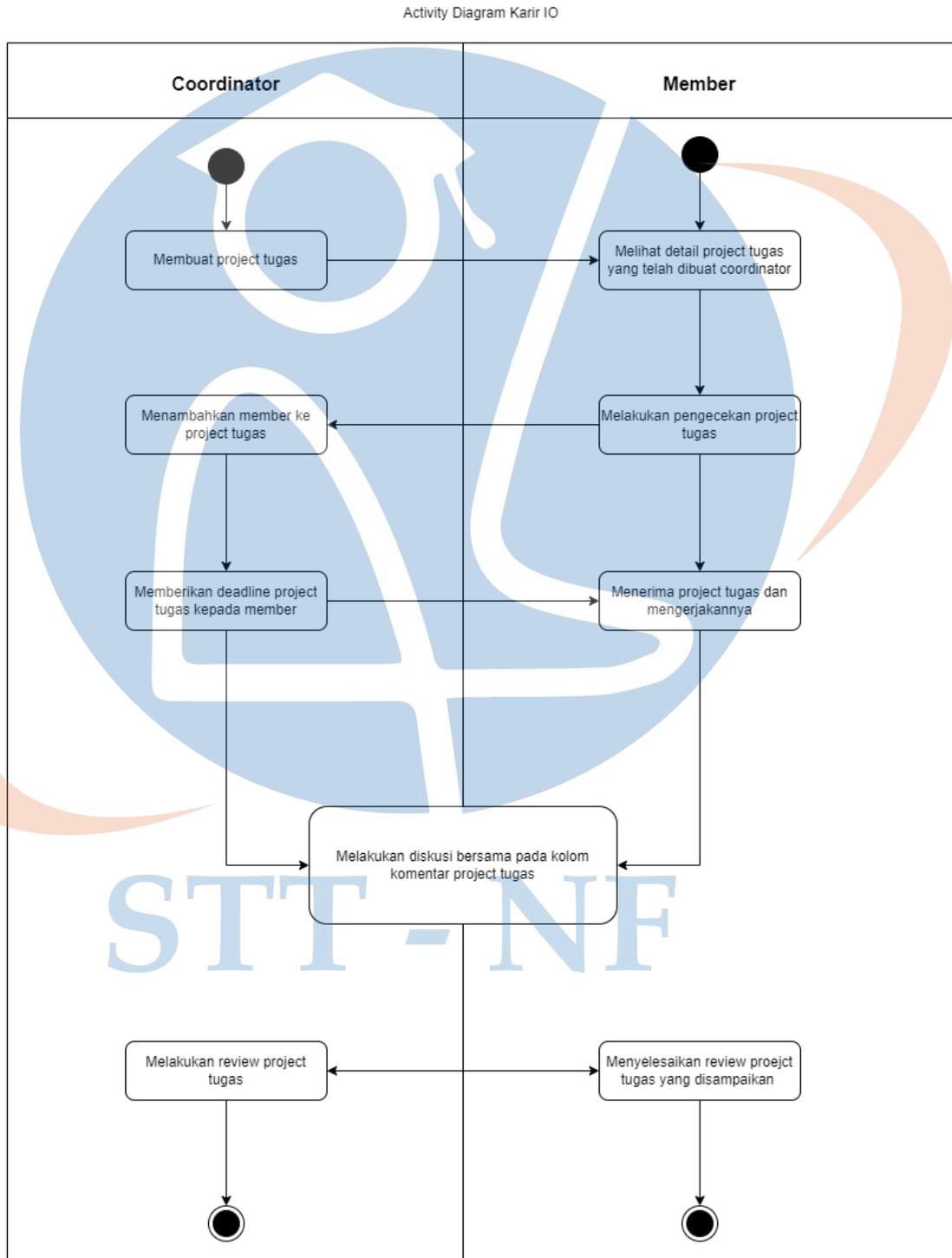
Gambar 4.1.3.1 Use Case Diagram [A8]

Sumber Gambar : Dokumentasi Pribadi

Pada aplikasi Karir IO, *role* yang dapat mengakses aplikasi yaitu *coordinator* dan *member*. Saat mengakses aplikasi, kedua *role* harus melakukan login dengan *email* dan *password*. Setelah *login*, *coordinator* dapat membuat *project* tugas, menambahkan *member* yang berkompeten pada *project* tugas yang dikerjakan, memberikan *project* tugas pada *member* sesuai keahlian dan tanggung jawab. *Member* memiliki hak dalam menerima *project* tugas yang diberikan *coordinator* dalam pengembangan aplikasi aplikasi dan mulai menyelesaikan *project* tugas sesuai dengan ketentuan yang berlaku. *Role coordinator* akan melakukan *review project* tugas sesuai catatan dan hasil proyek yang diselesaikan oleh pihak *member*. Apabila ditemukan ketidaksesuaian fitur, maka kedua *role* dapat melakukan diskusi *project* tugas pada kolom komentar penugasan, bila sudah sesuai pengerjaan maka kedua *role* dapat melakukan perubahan status *project* tugas dari “*active*” menjadi “*to review*” dan atau status proyek “*done*”.

4.1.3.3 Activity Diagram

Activity diagram adalah representasi grafis dari alur kerja, aktivitas yang terjadi pada suatu sistem, kegiatan yang dilakukan serta urutan pelaksanaannya untuk dapat membantu penggambaran proses bisnis atau alur kerja yang kompleks dengan cara yang mudah dipahami. Berikut dibawah alur proses activity diagram aplikasi Karir IO:

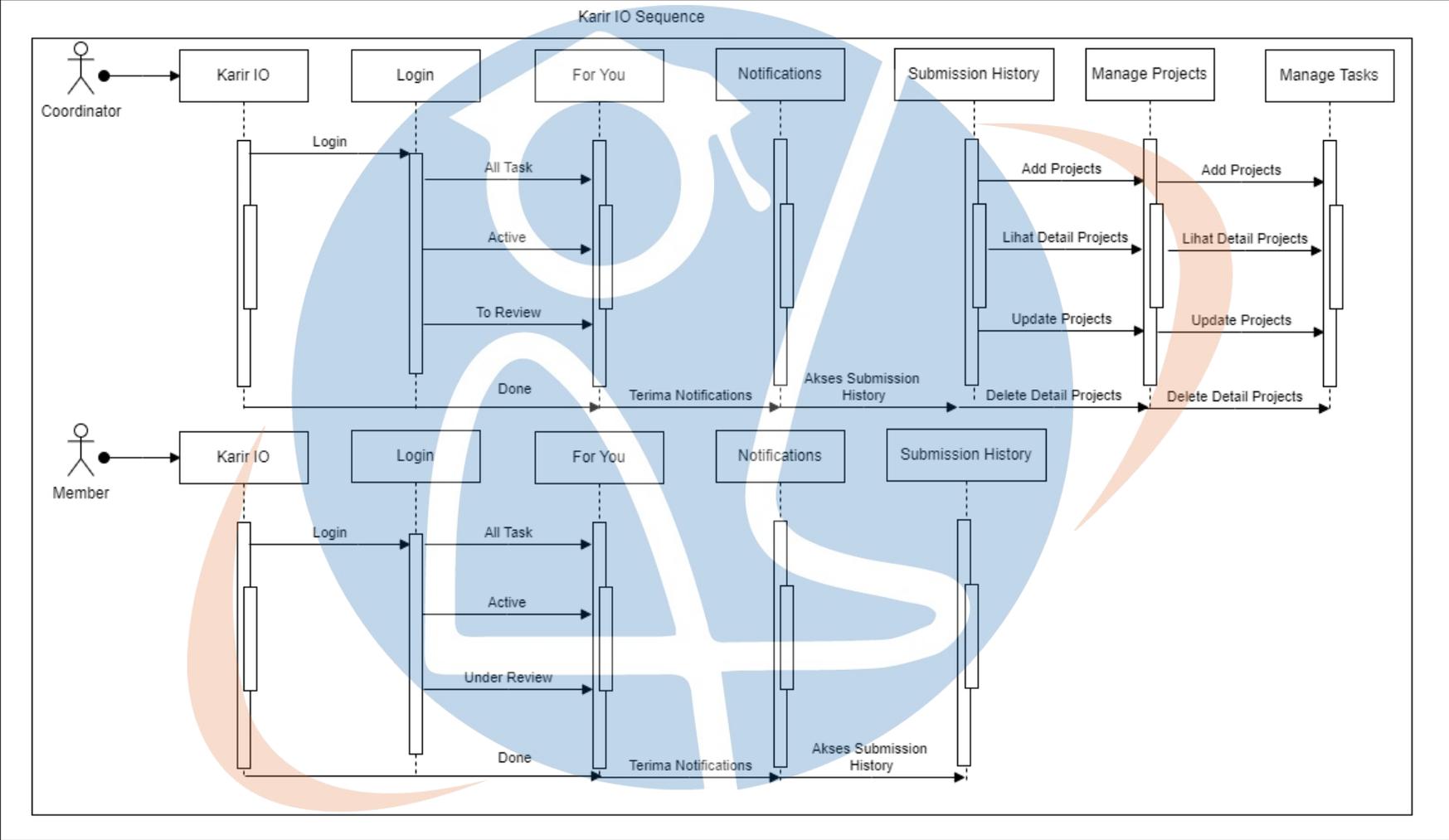


Gambar 4.1.3.2 Actity Diagram [A9]
Sumber Gambar : Dokumentasi Pribadi

Dengan *activity* diagram, pemetaan aktivitas pengguna pada aplikasi Karir IO dapat mencapai semua fitur dan fungsionalitas yang dibutuhkan dalam pengembangan aplikasi. Melalui *activity* diagram, aplikasi Karir IO dapat dirancang dengan baik dan alur kerja yang dirancang berjalan dengan terstruktur, kebutuhan pengguna dapat terpenuhi, serta kualitas dan efektivitas aplikasi secara keseluruhan dapat ditingkatkan. Berdasarkan **Gambar 4.1.3.2**, *coordinator* harus membuat *project* tugas dahulu, bila *project* tugas sudah terdaftar pada aplikasi, *member* dapat melihat detail *project* tugas yang dibuat tersebut dan melakukan pengecekan *project* tugas yang telah diberikan *coordinator* tersebut seperti ketentuan pengerjaan tugas, metode yang digunakan dan lainnya. Setelah itu, *coordinator* dapat menambahkan *member* ke dalam *project* tugas yang tersedia dan memberikan *deadline project* tugas kepada *member* dan *member* akan menerima *project* tugas yang diberikan *coordinator* lalu mulai mengerjakannya. Selama pengerjaan *project* tugas, baik *coordinator* ataupun *member* dapat melakukan diskusi bersama pada kolom komentar *project* tugas untuk mempertanyakan hal-hal yang membingungkan, kurang tepat atau sekadar berdiskusi tentang *project* tugas yang dikerjakan. Lalu setelah itu, *coordinator* dapat melakukan *review project* tugas dan *member* dapat menyelesaikan *review project* tugas yang disampaikan *coordinator*.

4.1.3.4 Sequence Diagram

Sequence diagram adalah menggambarkan interaksi sistem melalui serangkaian pesan yang menunjukkan urutan kronologis. Untuk aplikasi Karir IO, *sequence* diagram menggambarkan fitur aplikasi dan kegunaan fitur tersebut, proses login, bagaimana kedua *role* (*coordinator* dan *member*) melakukan pengelolaan tugas dan *manage* tugas sesuai tujuan pengembangan. **Gambar 4.1.3.3** dibawah, menggambarkan bahwa pada aplikasi Karir IO terdapat 2 *role* berbeda, *role coordinator* dapat mengakses fitur *for you*, *notifications*, *submission history*, *manage projects*, *manage tasks* dan fitur lain. *Role member*, hanya dapat mengakses *for you*, *notifications* dan *submission history* saja. *Coordinator* harus membuat *project* tugas, menambahkan *member* ke *project* melakukan diskusi bersama pada kolom komentar *project* tugas serta melakukan *review project*. Sedangkan *member*, hanya dapat mengakses aplikasi seputar tanggung jawab *member* saja, seperti melihat *detail project*, menerima *project* dari *coordinator* dan mengerjakannya, melakukan diskusi bersama pada kolom komentar *project* tugas, menyelesaikan *project* sesuai *review* yang disampaikan serta mengubah status *project*.



Gambar 4.1.3.3 Sequence Diagram [A10]
Sumber Gambar : Dokumentasi Pribadi

4.1.3.5 Class Diagram

Class diagram adalah diagram terstruktur pemodelan objek berorientasi untuk menggambarkan struktur statis sistem yang di bangun, menunjukkan informasi kelas sistem, atribut, metode serta hubungan yang terjadi antar kelas. Pada aplikasi Karir IO, *class* diagram mendukung berbagai fitur dan fungsi seperti manajemen pengguna dan manajemen tugas. Dengan menggunakan *class* diagram sebagai gambaran tabel yang tersedia pada aplikasi, maka aplikasi Karir IO diharapkan dapat dikembangkan dengan desain yang terstruktur dan memastikan bahwa semua komponen serta interaksi yang terjadi di antara tabel satu dengan tabel lainnya telah dipertimbangkan dan dirancang dengan baik hingga menghasilkan alur yang mudah dipahami oleh pengguna aplikasi.

Pada aplikasi Karir IO, terdapat 10 tabel yang berelasi yang diharapkan dapat memastikan data pengguna, proyek yang diselesaikan, serta tugas yang tersedia dapat berjalan dengan baik dan mendukung fungsionalitas manajemen tugas PT. Quantum. Relasi-relasi ini juga memungkinkan pengembang melakukan *query* dan manipulasi data dengan cara konsisten. Adapun relasi yang terjadi berikut:

1. **Tabel *notifications*** memiliki relasi dengan tabel *users*, dimana relasi yang terjalin diantara tabel *notifications* dengan tabel *users* memungkinkan agar pengguna aplikasi menerima notifikasi terkait aktivitas tertentu
2. **Tabel *project_members*** berelasi dengan tabel *users* dan tabel *projects*, relasi tabel yang terjalin ini dimanfaatkan untuk mengelola seluruh anggota dalam proyek tertentu karena relasi ini akan mengaitkan pengguna dengan proyek yang diikuti dan diberikan oleh *coordinator* sesuai dengan kemampuannya
3. **Tabel *projects_labels*** memiliki relasi dengan tabel *task_labels* dan tabel *projects*, relasi yang terjalin ini memungkinkan pengelolaan label yang bisa digunakan baik pada proyek maupun tugas pada aplikasi, memberikan akses yang fleksibilitas dalam mengkategorikan proyek dan mengorganisir data
4. **Tabel *users*** memiliki relasi dengan tabel *project_members*, tabel *tasks*, tabel *task_reminders*, tabel *task_comments*, dan tabel *task_assignness*, relasi yang ini dapat menghubungkan akses pengguna aplikasi Karir IO dengan berbagai elemen dan fitur dalam aplikasi, mempermudah pengelolaan tugas, sebagai pengingat tugas, mempermudah kedua *role* dalam memberikan komentar inti pada tugas yang dikerjakan, serta memantau tugas yang telah diberikan

5. **Tabel *task_labels*** hanya memiliki satu relasi saja, yaitu dengan tabel *task*, relasi yang terjadi ini menghubungkan secara langsung label dengan tugas. Relasi ini memungkinkan pengorganisasian dan pengelompokan tugas pada aplikasi dapat berurutan berdasarkan dengan label yang seharusnya
6. **Tabel *projects*** sama seperti tabel *task_labels* dimana hanya memiliki relasi dengan satu tabel saja, yaitu berelasi dengan tabel *tasks*, relasi yang terjadi di antara tabel ini menghubungkan proyek dengan tugas, memastikan bahwa setiap tugas yang dikerjakan telah dikaitkan dengan proyek tertentu untuk dapat dilakukan pemantauan lanjutan dan meminimalisir kesalahan proyek
7. **Tabel *task*** memiliki relasi dengan tabel *users*, tabel *task_reminders*, tabel *task_comments*, tabel *task_assignness*, tabel *projects* dan berelasi dengan tabel *task_labels*. Berbagai relasi yang terjalin ini menghubungkan tugas yang tersedia pada aplikasi Karir IO dengan berbagai aspek yang seharusnya, seperti *role* yang akan membuat tugas berjalan atau mereka yang ditugaskan untuk tugas tersebut, menjadi suatu pengingat tugas yang akan dikerjakan, membantu komentar pada tugas dapat tersampaikan, memberikan hubungan yang nyata antar tugas yang terkait, dan membantu pemberian label tugas
8. **Tabel *task_reminders*** memiliki satu relasi saja yaitu dengan tabel *tasks*. Relasi yang terjadi ini menghubungkan pengingat langsung dengan tugas tertentu serta memastikan pengguna diingatkan tentang tugas yang harus diselesaikan sesuai dengan tenggat waktu tugas yang ditentukan (*deadline*)
9. **Tabel *task_comments*** berelasi dengan tabel *users* dan tabel *tasks*, relasi ini memungkinkan pengguna meninggalkan komentar pada tugas, yang bisa dilihat dan dikelola oleh pengguna lain yang terkait dengan tugas tersebut.
10. **Tabel *task_assignness*** memiliki relasi dengan tabel *users* dan tabel *tasks*, relasi yang terjalin ini menghubungkan berbagai tugas yang terdapat pada aplikasi Karir IO dengan *role* pengguna yang ditugaskan agar menyelesaikan tugas tersebut sesuai dengan tenggat waktu yang ditentukan, relasi ini juga dapat memastikan tugas dikelola dan diselesaikan oleh pengguna yang tepat

Berikut adalah gambaran *class diagram* aplikasi Karir IO yang akan dikembangkan dalam penelitian ini sesuai dengan pembahasan yang telah disampaikan sebelumnya

Karir IO Class Diagram



Gambar 4.1.3.4 Class Diagram [A11]
 Sumber Gambar : Dokumentasi Pribadi

4.1.3.6 Entity Relationship Diagram (ERD)

Entity relationship diagram (ERD) Karir IO memberikan gambaran visualisasi dari struktur *database* yang digunakan oleh tim pengembang pada aplikasi Karir IO. Tabel yang terdapat pada *ERD* ini mencakup objek pada sistem serta relasi yang telah terjadi di antara satu tabel dengan tabel lainnya yang saling berkaitan. *ERD* memiliki peranan penting dalam meminimalisir relasi yang tidak diperlukan diantara tabel berdasarkan implementasi *database* yang tersedia. *ERD* menggambarkan hubungan entitas basis data yang berelasi. Pada aplikasi Karir IO, terdapat **entitas dan atribut** yang terjadi:

1. Tabel *Notifications*, memiliki entitas dan atribut sebagai berikut:
notificationID: int (Primary Key)
userID: int (Foreign Key)
notificationText: string
notificationDate: Date
2. Tabel *Users*, memiliki entitas dan atribut sebagai berikut:
userID: int (Primary Key)
username: string
password: string
name: string
email: string
3. Tabel *Task_Assignees*, memiliki entitas dan atribut sebagai berikut:
taskAssigneeID: int (Primary Key)
taskID: int (Foreign Key)
userID: int (Foreign Key)
4. Tabel *Task_Reminders*, memiliki entitas dan atribut sebagai berikut:
reminderID: int (Primary Key)
taskID: int (Foreign Key)
reminderDate: Date
5. Tabel *Task_Comments*, memiliki entitas dan atribut sebagai berikut:
commentID: int (Primary Key)
taskID: int (Foreign Key)
userID: int (Foreign Key)
commentText: string
commentDate: Date

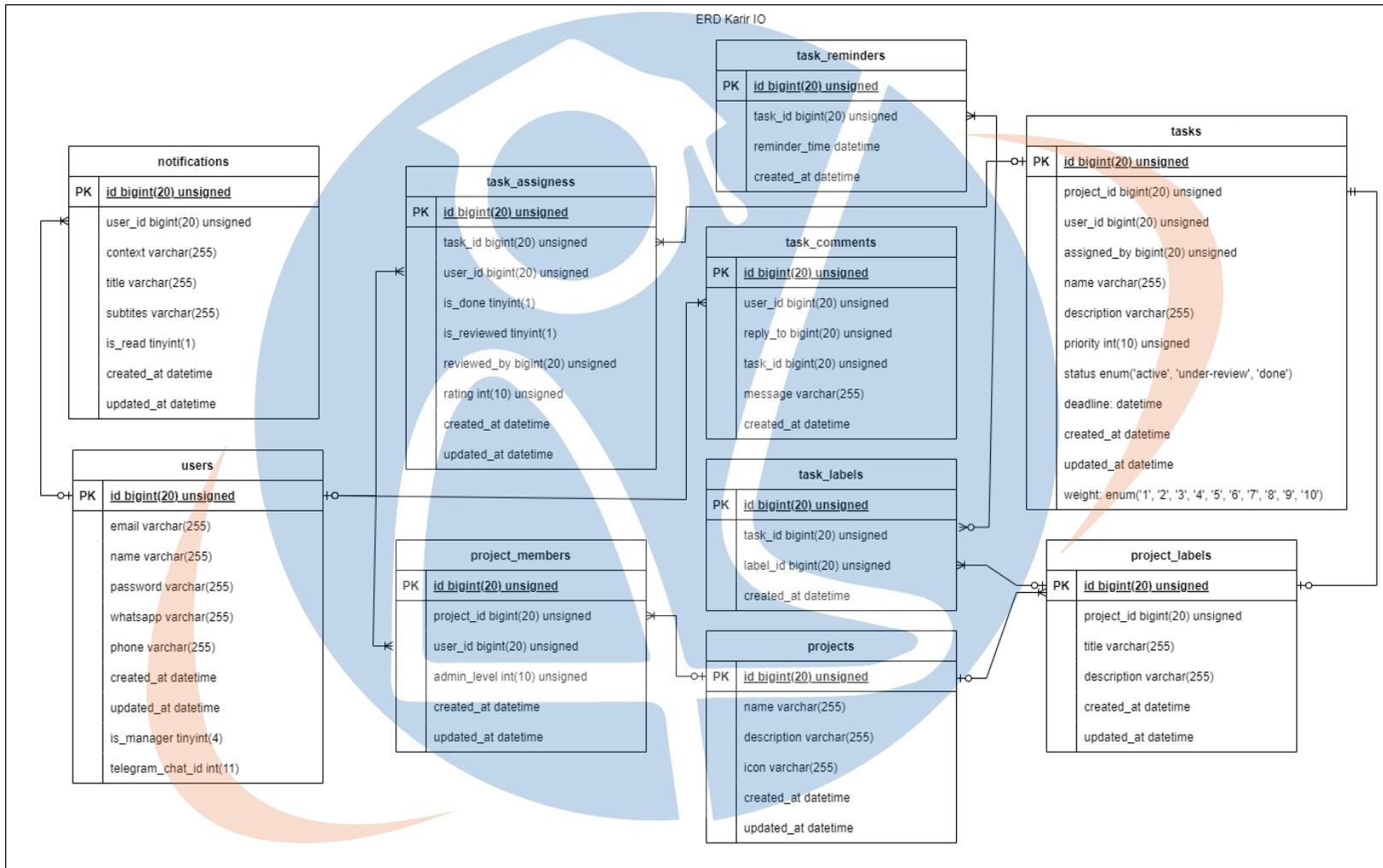
6. Tabel *Tasks*, memiliki entitas dan atribut sebagai berikut:
 - taskID: int (Primary Key)
 - title: string
 - description: string
 - dueDate: Date
 - status: string
 - assignedTo: userID (Foreign Key)
7. Tabel *Task_Labels*, memiliki entitas dan atribut sebagai berikut:
 - taskLabelID: int (Primary Key)
 - taskID: int (Foreign Key)
 - label: string
8. Tabel *Projects*, memiliki entitas dan atribut sebagai berikut:
 - projectID: int (Primary Key)
 - projectName: string
 - projectDescription: string
9. Tabel *Project_Labels*, memiliki entitas dan atribut sebagai berikut:
 - projectLabelID: int (Primary Key)
 - projectID: int (Foreign Key)
 - label: string
10. Tabel *Project_Members*, memiliki entitas dan atribut sebagai berikut:
 - projectMemberID: int (Primary Key)
 - projectID: int (Foreign Key)
 - userID: int (Foreign Key)

Selain memiliki entitas dan atribut yang menggambarkan relasi antara satu tabel dengan tabel lain, aplikasi Karir IO juga memiliki relasi antar entitas dimana satu tabel memiliki entitas dengan tabel lainnya untuk mendapatkan gambaran yang jelas dan terperinci bagaimana alur entitas yang terjadi pada aplikasi Karir IO. Dengan entitas ini, interaksi dan relasi yang terjadi antara satu tabel dengan tabel lain dapat dipastikan tersimpan secara terstruktur dan terorganisir, berikut adalah pembahasan lebih lanjut:

1. **Tabel *Notifications*** adalah suatu entitas data aplikasi Karir IO yang menyimpan notifikasi untuk pengguna dan setiap notifikasi yang terdapat pada aplikasi, memiliki keterkaitan dengan satu pengguna melalui *userID*.

2. **Tabel *Notifications*** – *Users* satu pengguna bisa memiliki banyak notifikasi yang terjadi dengan relasi: *One-to-Many* (*Users* → *Notifications*)
3. **Tabel *Project_Members*** - *Users*, *Projects* menggambarkan bahwa pada aplikasi Karir IO, satu proyek bisa memiliki banyak anggota yang memiliki wewenang untuk mengerjakan proyek dan satu pengguna bisa menjadi anggota banyak proyek. Adapun relasi yang terjadi adalah *Many-to-Many* dengan dipecah menjadi dua relasi *One-to-Many*, (*Projects* → *Project_Members*, *Users* → *Project_Members*)
4. **Tabel *Project_Labels*** - *Projects*, *Task_Labels* menjelaskan bahwa dalam entitas yang terjadi tersebut, satu proyek bisa memiliki banyak label, dan satu label bisa digunakan di banyak proyek yang dapat terjalin dengan relasi: *One-to-Many* (*Projects* → *Project_Labels*)
5. **Tabel *Task_Labels*** dan *Project_Labels* adalah entitas terpisah tetapi bisa memiliki keterkaitan dalam aplikasi.
6. *Users* - *Project_Members*, *Tasks*, *Task_Reminders*, *Task_Comments*, *Task_Assignees* adalah entitas yang terjadi pada aplikasi Karir IO dimana satu pengguna bisa memiliki banyak tugas, pengingat tugas, komentar pada tugas, dan ditugaskan pada banyak tugas dan relasi yang terjadi adalah *One-to-Many* (*Users* → *Tasks*, *Users* → *Task_Reminders*, *Users* → *Task_Comments*, *Users* → *Task_Assignees*)
7. **Tabel *Task_Labels*** – *Tasks* memiliki entitas dimana satu tugas bisa memiliki banyak label dengan relasi: *One-to-Many* (*Tasks* → *Task_Labels*)
8. **Tabel *Projects*** – *Tasks* memiliki entitas yang nyata dimana satu proyek bisa memiliki banyak tugas. Relasi: *One-to-Many* (*Projects* → *Tasks*)
9. **Tabel *Tasks*** - *Task_Reminders*, *Task_Comments*, *Task_Assignees* memiliki entitas dimana satu tugas bisa memiliki banyak pengingat, komentar, dan penugasan. Relasi: *One-to-Many* (*Tasks* → *Task_Reminders*, *Tasks* → *Task_Comments*, *Tasks* → *Task_Assignees*)
10. **Tabel *Users*** memiliki entitas data yang dapat menyimpan berbagai macam informasi tentang pengguna aplikasi. Relasi dengan *Tasks*, *Task_Reminders*, *Task_Comments*, *Task_Assignees*, *Notifications*, dan *Project_Members* menunjukkan bahwa pengguna dapat memiliki banyak tugas, pengingat, komentar, penugasan, notifikasi, dan keanggotaan proyek.

11. **Tabel *Tasks*** adalah salah satu entitas data pada aplikasi Karir IO yang juga menyimpan informasi tentang tugas-tugas yang harus diselesaikan. Relasi dengan *Task_Reminders*, *Task_Comments*, *Task_Assignees*, *Task_Labels*, dan *Projects* menunjukkan bahwa setiap tugas bisa memiliki banyak pengingat, komentar, penugasan, label, dan berhubungan dengan satu proyek.
12. **Tabel *Task_Assignees*** adalah suatu entitas data yang digunakan untuk dapat mengelola hubungan banyak-ke-banyak antara pengguna dan tugas serta menghubungkan *Users* dan *Tasks* melalui *taskID* dan *userID*.
13. **Tabel *Task_Reminders*** menjadi salah satu entitas data aplikasi Karir IO yang menyimpan pengingat untuk tugas tertentu dan setiap pengingat terkait dengan satu tugas melalui tabel *taskID*.
14. **Tabel *Task_Comments*** adalah salah satu entitas yang menyimpan komentar pada tugas dimana setiap komentar terkait dengan satu tugas dan satu pengguna (*user*) melalui *taskID* dan *userID*.
15. **Tabel *Projects*** adalah suatu entitas data yang menyimpan informasi tentang proyek dimana setiap proyek dapat memiliki banyak tugas dan anggota proyek melalui *projectID*.
16. **Tabel *Project_Members*** merupakan salah satu entitas data aplikasi Karir IO yang dimanfaatkan untuk mengelola hubungan banyak-ke-banyak diantara proyek dan pengguna aplikasi. Relasi ini menghubungkan *Users* dan *Projects* melalui tabel *projectID* dan *userID*.
17. **Tabel *Project_Labels*** menjadi suatu entitas data yang menyimpan label untuk digunakan pada proyek yang tersedia pada aplikasi Karir IO. Adapun setiap label yang tersedia, memiliki keterkaitan dengan 1 proyek melalui *projectID*.
18. **Tabel *Task_Labels*** menjadi salah satu entitas data yang menyimpan label untuk dapat digunakan pada setiap tugas yang tersedia di aplikasi Karir IO. Setiap label yang tersedia, memiliki keterkaitan dengan 1 tugas melalui *taskID*.



Gambar 4.1.3.5 Entity Relationship Diagram (ERD) Karir IO [A12]

Sumber Gambar : Dokumentasi Pribadi

4.1.4 Perancangan Pengujian

Pengujian sistem merupakan serangkaian pengujian yang dilakukan pada suatu sistem oleh tim pengembang berdasarkan skenario pengujian yang telah disusun rinci, jelas dan menyampaikan kronologi yang sesuai dengan penelitian. Pada penelitian ini, rancangan pengujian Karir IO dilakukan untuk memperoleh hasil evaluasi yang sesuai dengan tujuan penelitian dan menyatakan bahwa aplikasi Karir IO layak digunakan sebagai aplikasi yang mempermudah PT. Quantum mengelola tugas dan proyek kantor. Adapun rancangan pengujian yang dilakukan dalam perancangan aplikasi Karir IO pada penelitian ini adalah dengan menggunakan metode pengujian *black box testing* dan *skala likert* berdasarkan dengan alur *user stories* dan *product backlog* yang telah disusun dimana hasil pengujian tersebut diharapkan dapat menjadi suatu pertimbangan perusahaan untuk menggunakan aplikasi ini dalam mengelola tugas kantor.

4.1.4.1 Black Box Testing

Pada penelitian ini, rancangan pengujian pertama yang dilakukan adalah dengan menggunakan metode *black box testing* yang menyesuaikan dengan alur *user stories* dan *product backlog* yang telah disusun sebelumnya untuk mencapai fungsionalitas aplikasi yang seharusnya. Adapun berikut adalah beberapa kasus yang dianalisis:

Tabel 4.1.4.1 Perancangan Pengujian - Black Box Testing

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
1	Register dengan <i>body sesuai</i> <pre>{ "email": "sabiqmuhammad9 8@gmail.com", "name": "Muhammad Sabiq", "password": "123456", "whatsapp": "085691550726", "phone": "085691550726" }</pre>	[POST] https://karir-api.quantumteknologi.com/auth/register	Akan menghasilkan <i>response</i> : <i>Status : Created,</i> <i>Code : 201</i> Dimana register pengguna berhasil dan pengujian ini berhasil dilakukan <pre>{ "status": "CREATED", "created": true, "statusCode": 201, "data": { "email": "muhammadsabiq44@gmail.com", "name": "Muhammad Sabiq",</pre>	Pengguna Baru Berhasil Dibuat

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
	}		<pre> "password": "\$2b\$10\$12NGrvjCLRjvHJQVD UO/HOrw/L0FFCbaUZMEARC EnjWWZ8tkQLAQ.", "whatsapp": "085691550726", "phone": "085691550726", "is_manager": 0, "created_at": "2024-05- 29T10:48:32.000Z", "updated_at": "2024-05- 29T10:48:32.000Z", "id": "584" } </pre>	
2	<p>Register dengan body tidak sesuai</p> <pre> { "email": "muhammadsabiq4 4@gmail.com", "name": "Muhammad Sabiq", "password_test": "123456", "whatsapp_test": "085691550726", "phone": "085691550726" } </pre>	<p>[POST] https://karir-api.quantumteknologi.com/auth/register</p>	<p>Akan menghasilkan <i>response</i>: <i>Status Code</i> : 400 Dengan <i>message</i>: "Password Must Be A String", "Password Should Not Be Empty", "Whatsapp Must Be A String", "Whatsapp Should Not Be Empty" Dan status <i>Error</i> : "Bad Request" Dimana data-data register tidak berhasil diinputkan dan pengujian ini tidak berhasil dilakukan apabila <i>request body</i> saat melakukan <i>register</i> tidak sesuai dengan ketentuan yang berlaku</p> <pre> { "statusCode": 400, "message": ["password must be a string", "password should not be empty", "whatsapp must be a string", "whatsapp should not be empty"], "error": "Bad Request" } </pre>	<p>Pengguna Baru Gagal Dibuat</p>

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
3	<p>Login dengan body sesuai</p> <pre>{ "email": "sabiqmuhammad98@gmail.com", "password": "*****" }</pre>	<p>[POST] https://karir-api.quantumteknologi.com/auth/login</p>	<p>Akan menghasilkan <i>response</i>: <i>Status</i> : OK, <i>Code</i> : 200 Dimana data-data <i>login</i> dengan <i>email</i> dan <i>password</i> yang telah di input oleh pengguna aplikasi sesuai dan pengujian ini berhasil dilakukan</p> <pre>{ "status": "OK", "statusCode": 200, "data": { "results": { "id": "584", "email": "sabiqmuhammad98@gmail.com", "name": "Sabiq Muhammad", "whatsapp": "085691550726", "phone": "085691550726", "is_manager": 1, "created_at": "2024-04-09T06:38:28.000Z", "updated_at": "2024-04-09T06:38:28.000Z" }, "auth_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjU4NCIsIm5hbWUiOiJTYTY" } }</pre>	<p>Pengguna Berhasil Login</p>
4	<p>Login dengan body tidak sesuai</p> <pre>{ "email": "sabiqmuhammad98@gmail.com", "password_test": "Sabiq7392/" }</pre>	<p>[POST] https://karir-api.quantumteknologi.com/auth/login</p>	<p>Akan menghasilkan <i>response</i>: <i>Status Code</i> : 400 <i>Message</i>: "Password must be a string", "Password should not be empty" Dan status <i>Error</i> : "Bad Request" Dimana data-data login tidak berhasil diinputkan dan pengujian ini tidak berhasil dilakukan apabila <i>login</i> tidak sesuai dengan ketentuan yang berlaku</p>	<p>Pengguna Gagal Login</p>

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			<pre>{ "statusCode": 400, "message": ["password must be a string", "password should not be empty"], "error": "Bad Request" }</pre>	
5	<p>Membuat project baru dengan <i>body</i> sesuai</p> <pre>{ "name": "INA Digital", "description": "INA Digital adalah aplikasi untuk menyatukan semua layanan pemerintahan di indonesia", "icon": "example-icon" }</pre>	<p>[POST] https://karir-api.quantumteknologi.com/projects</p>	<p>Akan menghasilkan <i>response</i>: <i>Status</i> : <i>CREATED</i>, <i>Code</i> : <i>201</i> Dimana proyek baru PT. Quantum dengan input data yang sesuai pada aplikasi Karir IO berhasil dibuat dan pengujian ini berhasil dilakukan</p> <pre>{ "status": "CREATED", "created": true, "statusCode": 201, "data": { "name": "INA Digital", "description": "INA Digital adalah aplikasi untuk menyatukan semua layanan pemerintahan di indonesia", "icon": "example-icon", "created_at": "2024-05-29T11:14:50.000Z", "updated_at": "2024-05-29T11:14:50.000Z", "id": "7" } }</pre>	Project Berhasil Ditambahkan
6	<p>Menampilkan semua <i>project</i> yang dibuat</p>	<p>[GET] https://karir-api.quantumteknologi.com/projects</p>	<p>Akan menghasilkan <i>response</i>: <i>Status</i> : <i>OK</i>, <i>Code</i> : <i>200</i> Dimana seluruh proyek PT. Quantum yang tersedia pada aplikasi Karir IO berhasil ditampilkan dan pengujian ini berhasil dilakukan</p> <pre>{</pre>	Project Berhasil Ditambahkan

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			<pre> "status": "OK", "statusCode": 200, "data": { "projects": [{ "id": "7", "name": "INA Digital", "description": "INA Digital adalah aplikasi untuk menyatukan semua layanan pemerintahan di indonesia", "icon": "example-icon", "created_at": "2024-05-29T11:14:50.000Z", "updated_at": "2024-05-29T11:14:50.000Z" }], "total": 1 } </pre>	
7	Menampilkan <i>project detail</i> dengan id 7	[GET] https://karir-api.quantumteknologi.com/projects/7	<p>Akan menghasilkan <i>response</i>: <i>Status : OK</i>, <i>Code : 200</i> Dimana seluruh proyek PT. Quantum yang tersedia pada aplikasi Karir IO dan memiliki id 7 maka telah berhasil ditampilkan <i>detail</i>-nya dan pengujian ini berhasil dilakukan</p> <pre> { "status": "OK", "statusCode": 200, "data": { "id": "7", "name": "INA Digital", "description": "INA Digital adalah aplikasi untuk menyatukan semua layanan pemerintahan di indonesia", "icon": "example-icon", "created_at": "2024-05-29T11:14:50.000Z", "updated_at": "2024-05-29T11:14:50.000Z" } } </pre>	Project Detail Dengan Id 7 Berhasil Didapatkan

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			} }	
8	Mengupdate project dengan id 7 { "name": "INA Digital SuperApp" }	[PUT] https://karir-api.quantumteknologi.com/projects/7	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana seluruh projek PT. Quantum yang tersedia pada aplikasi Karir IO dan memiliki id 7 maka telah berhasil di <i>update project</i> nya dan pengujian ini berhasil dilakukan { "status": "OK", "statusCode": 200, "data": { "generatedMaps": [], "raw": [], "affected": 1 } }	Project Detail Dengan Id 7 Berubah Nama Dari INA Digital Menjadi Ina Digital Superapp
9	Menambahkan member ke <i>project</i> dengan id 1 dengan <i>body</i> sesuai { "project_id": 7, "user_id": 584, "admin_level": 0 }	[POST] https://karir-api.quantumteknologi.com/projects/t-members	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 Dimana projek PT. Quantum berhasil ditambahkan <i>member</i> nya ke <i>project</i> dengan id 1 berdasarkan <i>body</i> yang sesuai dan pengujian ini berhasil dilakukan { "status": "CREATED", "created": true, "statusCode": 201, "data": { "admin_level": 0, "project_id": 1, "user_id": 584, "created_at": "2024-05-29T11:10:04.000Z", "updated_at": "2024-05-29T11:10:04.000Z", "id": "30" } }	Member Berhasil Ditambahkan Ke Project Id 1

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
10	Menampilkan semua member yang ada di <i>project</i> dengan id 7	[GET] https://karir-api.quantumteknologi.com/projects/7	Akan menghasilkan <i>response</i> : <i>Status</i> : OK, <i>Code</i> : 200 Dimana semua <i>member</i> yang terdaftar pada <i>project</i> PT. Quantum dengan id 7 telah berhasil ditampilkan dan pengujian ini berhasil dilakukan <pre>{ "status": "OK", "statusCode": 200, "data": { "project": { "id": "7", "name": "INA Digital SuperApp", "description": "INA Digital adalah aplikasi untuk menyatukan semua layanan pemerintahan di indonesia" }, "members": [{ "id": "31", "user_id": "584", "name": "Sabiq", "email": "sabiqmuhammad98@gmail.com", "whatsapp": null, "phone": "085691550726", "created_at": "2024-05-29T11:23:55.000Z", "update_at": "2024-05-29T11:23:55.000Z" }], "total_members": 1 } }</pre>	Semua Member Berhasil Dimunculkan Di Project Id 7
11	Membuat tugas pada <i>project</i> id 7 kepada <i>user</i> dengan id 584	[POST] https://karir-api.quantumteknologi.com/tasks	Akan menghasilkan <i>response</i> : <i>Status</i> : CREATED, <i>Code</i> : 201 Dimana proyek baru PT. Quantum berhasil membuat	Tugas Berhasil Diberikan

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
	<pre>{ "project_id": 7, "user_id": 584, "name": "Fix bug oracle", "priority": 1, "status": 1, "weight": 2 }</pre>		<p>tugas pada project id 7 kepada user dengan id 584 berdasarkan ketentuan yang sesuai dan pengujian ini berhasil dilakukan</p> <pre>{ "status": "CREATED", "created": true, "statusCode": 201, "data": { "name": "Fix bug oracle", "priority": 1, "status": "active", "weight": 2, "project_id": 7, "user_id": 584, "assigned_by": 584, "created_at": "2024-05-29T11:32:51.000Z", "updated_at": "2024-05-29T11:32:51.000Z", "id": "635" } }</pre>	
12	Dapatkan semua tugas yang <i>active</i> pada <i>project</i> id 7	<p>[GET] https://karir-api.quantumteknologi.com/tasks/projects/7?status=active&page=1&limit=10</p>	<p>Akan menghasilkan <i>response</i>: Status : OK, Code : 200 Dimana seluruh tugas yang aktif berjalan pada PT. Quantum berhasil didapatkan pada project id 7 dan pengujian ini berhasil dilakukan</p> <pre>{ "status": "OK", "statusCode": 200, "data": { "tasks": [{ "id": "635", "name": "Fix bug oracle", "description": null, "priority": 1, "status": "active", "deadline": null,</pre>	Tugas Berhasil Di Dapatkan Pada Project Id 7

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			<pre> "created_at": "2024-05-29T11:32:51.000Z", "updated_at": "2024-05-29T11:32:51.000Z", "weight": 2, "project": { "id": "7", "name": "INA Digital SuperApp", "labels": [] }, "user": { "id": "584", "name": "Sabiq" }, "assigned_by": { "id": "584", "name": "Sabiq" }, "labels": [] }], "total": 1 } </pre>	
13	<p>Membuat komentar pada <i>task</i> id 635</p> <pre> { "message": "Deadline nanti ya hari minggu" } </pre>	<p>[POST] https://karir-ologi.com/task-comments/tasks/635</p>	<p>Akan menghasilkan <i>response</i>: Status : CREATED, Code : 201 Dimana pembuatan komentar pada task id 635 berhasil ditambahkan sesuai keadaan sebenarnya dan pengujian ini berhasil dilakukan</p> <pre> { "status": "CREATED", "created": true, "statusCode": 201, "data": { "message": "Deadline nanti ya hari minggu", "user_id": "584", "task_id": "635", "created_at": "2024-05-29T11:38:34.000Z", "id": "586" } } </pre>	<p>Komentar Berhasil Dibuat</p>

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			<pre> } } </pre>	
14	Menampilkan komentar pada <i>task</i> 635	<p>[GET]</p> <p>https://karir-api.quantumteknologi.com/task-comments/tasks/635</p>	<p>Akan menghasilkan <i>response</i>: Status : OK, Code : 200 Dimana komentar pada <i>task</i> 635 berhasil ditampilkan dan pengujian ini berhasil dilakukan</p> <pre> { "status": "OK", "statusCode": 200, "data": { "task_comments": [{ "id": "586", "message": "Deadline nanti ya hari minggu", "user_id": { "id": "584", "name": "Sabiq", "is_manager": 1 }, "replies": [], "created_at": "2024-05-29T11:38:34.000Z" }], "total": 1 } } </pre>	Komentar Berhasil Didapatkan Pada Task Id 635
15	<p>Membalas komentar pada <i>tasks</i> 635 dengan comment id 586</p> <pre> { "message": "Oke siappp", "reply_to": 586 } </pre>	<p>[POST]</p> <p>https://karir-api.quantumteknologi.com/task-comments/tasks/635</p>	<p>Akan menghasilkan <i>response</i>: Status : CREATED, Code : 201 <i>Message</i> : “Deadline nanti ya hari minggu” atau <i>message</i> yang lainnya Dimana membalas komentar pada <i>task</i> 635 dengan <i>comment id</i> 586 dan pengujian pun berhasil</p> <pre> { "status": "CREATED", "created": true, </pre>	Berhasil Membalas Komentar

No	Pengujian	HTTP Method/ Endpoint	Hasil Yang Diharapkan	Kesimpulan
			<pre>"statusCode": 201, "data": { "message": "Deadline nanti ya hari minggu", "user_id": "584", "reply_to": 586, "task_id": "635", "created_at": "2024-05- 29T11:42:34.000Z", "id": "587" } }</pre>	
16	Melihat balasan pada komentar id 586	<p>[GET] https://karir-api.quantumteknologi.com/task-comments/586/replies</p>	<p>Akan menghasilkan <i>response</i>: Status : OK, Code : 200 Message : “Baik siapp bos” Dimana seluruh balasan komentar id 586 dapat ditampilkan, pengujian ini pun telah berhasil</p> <pre>{ "status": "OK", "statusCode": 200, "data": { "task_comment_replies": [{ "id": "588", "message": "Oke siappp", "created_at": "2024-05- 29T11:45:41.000Z", "user_id": { "id": "584", "name": "Sabiq", "is_manager": 1 } }], "total": 2 } }</pre>	Berhasil Melihat Balasan Pada Komentar Id 586

Sumber Tabel: Dokumentasi Pribadi

4.1.4.2 Skala Likert

Pada penelitian ini, metode pengujian yang dilakukan selanjutnya untuk mengukur pencapaian tujuan rancang bangun aplikasi Karir IO adalah dengan menggunakan metode pengujian *skala likert* melalui kuesioner penelitian yang disampaikan kepada pengguna aplikasi Karir IO untuk menggambarkan pengalaman pengguna aplikasi saat menggunakan fitur dan fungsi aplikasi. Pilihan pernyataan yang telah dipilih pengguna aplikasi Karir IO diakumulasikan keseluruhan lalu dibuat hipotesis penelitian, hingga membentuk tabel pengujian *skala likert* yang digambarkan pada tabel berikut:

Tabel 4.1.4.2 Perancangan Pengujian - Skala Likert Role Coordinator

No	Pernyataan	STS	TS	S	SS
1	<i>API</i> Karir IO berjalan baik sebagaimana fungsinya	0	0	0	2
2	<i>API</i> memberikan <i>response</i> yang cepat dan kinerja baik	0	0	0	2
3	Struktur dan format <i>response API</i> mudah dimengerti	0	0	2	0
4	<i>API</i> memberikan <i>response</i> yang informatif bila terjadi kesalahan atau permasalahan <i>request</i> yang diajukan	0	0	2	0
5	Struktur <i>endpoint API</i> mudah untuk dipahami pengguna	0	0	0	2
6	Integrasi <i>API</i> dengan aplikasi pihak ketiga sesuai dengan ketentuan yang diberlakukan	0	0	0	2
7	Penggunaan <i>API</i> mempermudah karyawan PT. Quantum dalam mengelola dan manajemen tugas kantor	0	0	0	2
8	<i>API</i> ini memberikan fitur keamanan yang supportif terhadap data-data pengguna aplikasi	0	0	1	1
9	Struktur format dan <i>endpoint API</i> mempermudah proses perkembangan aplikasi Karir IO yang dikembangkan	0	0	2	0
10	Secara keseluruhan, penggunaan <i>API</i> telah memenuhi ekspektasi dan kebutuhan PT. Quantum dalam mengelola tugas kantor	0	0	0	2

Sumber Tabel: Dokumentasi Pribadi

Pada penelitian ini, metode pengujian *skala likert* dilakukan secara terpisah kepada *role coordinator* dan *role member*. Hingga saat penelitian ini berlangsung, ditetapkan bahwa *role coordinator* berjumlah 2 orang karyawan PT. Quantum yakni karyawan yang bertanggung jawab dalam pengembangan aplikasi Karir IO sebagai *tech lead* dan *project manager*. Sedangkan untuk *role member* berjumlah 10 orang karyawan PT. Quantum yang bertanggung jawab dalam pengembangan aplikasi Karir IO.

Tabel 4.1.4.3 Tabel 4.1.4.2 Perancangan Pengujian - Skala Likert Role Member

No	Pernyataan	STS	TS	S	SS
1	<i>API</i> Karir IO berjalan baik sebagaimana fungsinya	0	0	5	5
2	<i>API</i> memberikan <i>response</i> yang cepat dan kinerja baik	0	1	5	4
3	Struktur dan format <i>response API</i> mudah dimengerti	0	0	6	4
4	<i>API</i> memberikan <i>response</i> yang informatif bila terjadi kesalahan atau permasalahan <i>request</i> yang diajukan	0	1	3	6
5	Struktur <i>endpoint API</i> mudah untuk dipahami pengguna	0	0	0	10
6	Integrasi <i>API</i> dengan aplikasi pihak ketiga sesuai dengan ketentuan yang diberlakukan	0	1	4	5
7	Penggunaan <i>API</i> mempermudah karyawan PT. Quantum dalam mengelola dan manajemen tugas kantor	0	0	0	10
8	<i>API</i> ini memberikan fitur keamanan yang supportif terhadap data-data pengguna aplikasi	1	2	2	5
9	Struktur format dan <i>endpoint API</i> mempermudah proses perkembangan aplikasi Karir IO yang dikembangkan	0	0	3	7
10	Secara keseluruhan, penggunaan <i>API</i> telah memenuhi ekspektasi dan kebutuhan PT. Quantum dalam mengelola tugas kantor	0	0	0	10

Sumber Tabel: Dokumentasi Pribadi

Pada penelitian ini, metode pengujian *skala likert* menentukan 4 macam interpretasi pilihan yang akan menentukan nilai perhitungan *skala likert*. Interpretasi interval *skala likert* tersebut yakni Sangat Tidak Setuju (STS) dengan skor 1 *point*, Tidak Setuju (TS) dengan skor 2 *point*, Setuju (S) dengan skor 3 *point*, dan Sangat Setuju (SS) dengan skor 4 *point*. Berikut adalah gambaran dari interval *skala likert*:

Tabel 4.1.4.4 Interval Skala Likert

No	Interval %	Interpretasi
1	0 – 30	Sangat Tidak Setuju
2	31 – 55	Tidak Setuju
3	56 – 80	Setuju
4	80 - 100	Sangat Setuju

Sumber Tabel: Dokumentasi Pribadi

Berdasarkan evaluasi metode pengujian *skala likert*, total perhitungan *skala likert* aplikasi Karir IO terhadap 2 *role* yang berbeda, dapat dilihat dari perhitungan yang telah diperoleh oleh penulis dibawah ini:

- Pertanyaan dengan jawaban **Sangat Setuju (SS)** = $79 \times 4 = 316$
- Pertanyaan dengan jawaban **Setuju (S)** = $35 \times 3 = 105$
- Pertanyaan dengan jawaban **Tidak Setuju (TS)** = $5 \times 2 = 10$
- Pertanyaan dengan jawaban **Sangat Tidak Setuju (STS)** = $0 \times 1 = 0$

Total skor perhitungan *skala likert* = 431

Rumus nilai maksimum *skala likert* = (Jumlah responden * Nilai tertinggi *skala likert* * Total pernyataan), maka didapatkan hasil nilai maksimum = $12 * 4 * 10 = 480$

Penyelesaian Akhir Indeks (%) = $(\text{Total skor} / \text{Skor Max}) * 100$ Indeks (%)

Berdasarkan rumus nilai maksimum maka diperoleh perhitungan = $(431 / 480) * 100 = 89,79\%$. Hasil *skala likert* **89,79%** yang menyatakan bahwa penggunaan aplikasi Karir IO untuk PT. Quantum berada diantara interval 80-100 yang menunjukkan responden sangat setuju terhadap topik penelitian yang dikembangkan ini.

BAB V

KESIMPULAN DAN SARAN

Bab V sekaligus bab penutup pada penelitian ini yang menyampaikan informasi kesimpulan untuk memberikan simpulan dari keseluruhan isi penelitian yang dibahas pada kesempatan ini dan memberikan pemahaman yang menyeluruh tentang temuan serta implikasi yang diperoleh. Adapun saran yang dibahas pada bab V menyampaikan rekomendasi serta saran yang sesuai untuk penelitian ini di masa mendatang, perbaikan sistem yang dapat dilakukan, dan potensi pengembangan terhadap aplikasi Karir IO.

5.1 Kesimpulan

Rancang Bangun *Backend* Aplikasi Karir Io Berbasis *Web* Menggunakan *NestJs*: Studi Kasus PT. Quantum Teknologi Nusantara yang menjadi judul penelitian ini memperoleh beberapa kesimpulan yang relevan dengan rumusan masalah, tujuan dan lingkup penelitian. Kesimpulan ini diharapkan dapat memberikan wawasan mengenai pengembangan *backend* aplikasi *web* dengan teknologi *NestJs*, kesimpulannya yaitu:

1. *Backend* aplikasi Karir IO berfungsi dengan baik karena menggunakan *NestJS* dan metode pengembangan *scrum* dengan pengembangan terstruktur, iteratif, dan adaptif terhadap perubahan kebutuhan selama proses pengembangan.
2. Aplikasi Karir IO berjalan baik dan sesuai kebutuhan pengguna yang kemudian dibuktikan penulis berdasarkan nilai maksimum *skala likert* dengan persentase pengujian fungsionalitas aplikasi Karir IO mencapai 89.79% dari *interval* 80-100. Hal ini menunjukkan bahwa aplikasi Karir IO yang dikembangkan layak digunakan PT. Quantum untuk pengelolaan manajemen tugas kantor.

5.2 Saran

Berdasarkan kesimpulan yang diperoleh dari penelitian ini, beberapa saran untuk pengembangan lebih lanjut dan peningkatan kualitas aplikasi Karir IO, diantaranya:

1. Peningkatan optimalisasi kinerja *backend* dengan melakukan *caching*, *load balancing*, dan optimasi *query* basis data untuk menangani peningkatan jumlah pengguna dan data karena saat ini kinerja *backend* masih belum optimal
2. Peningkatan keamanan aplikasi dengan enkripsi data, pengelolaan akses yang ketat, dan uji penetrasi secara rutin untuk mendeteksi dan menangani masalah
3. Mengembangkan versi *mobile* aplikasi Karir IO yang menjangkau karyawan PT. Quantum dalam menggunakan aplikasi di berbagai perangkat yang tersedia

DAFTAR REFERENSI

- [1] “Aplikasi Untuk Manajemen Kantor | Suteki Tech.” Accessed: Apr. 23, 2024. [Online]. Available: <https://suteki.co.id/aplikasi-untuk-manajemen-kantor/>
- [2] “Mengenal Apa Itu Next.js, Pengertian, Fitur, Cara Kerja, dan Kelebihannya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.biznetgio.com/news/apa-itu-next-js>
- [3] “Apa itu Scrum? Penjelasan Metodologi Scrum - AWS.” Accessed: Apr. 23, 2024. [Online]. Available: <https://aws.amazon.com/id/what-is/scrum/>
- [4] “Bagaimana Platform Manajemen Tugas dapat Mempermudah Perkembangan Bisnis | Cartrack Indonesia.” Accessed: Apr. 23, 2024. [Online]. Available: <https://cartrack.id/id/bagaimana-platform-manajemen-tugas-dapat-mempermudah-perkembangan-bisnis>
- [5] “Pentingnya Manajemen Tugas di Perusahaan - Presensi.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.presensi.co.id/blog/pentingnya-manajemen-tugas-di-perusahaan->
- [6] “Quantum Teknologi Nusantara.” Accessed: Apr. 23, 2024. [Online]. Available: <https://quantumteknologi.com/>
- [7] A. Josi, T. Fisika, J. Patra No, K. Sukaraja Kecamatan Prabumulih Selatan, and S. Selatan STMIK Prabumulih Rhieyah, “Rancang Bangun Sistem Informasi Tender Karet Desa Jungai Menggunakan Metode Waterfall,” *Jurnal SISFOKOM*, vol. 06, 2017.
- [8] “View of Rancang Bangun Aplikasi Pelayanan E-Trayek Berbasis Mobile Menggunakan Metode Extreme Programming.” Accessed: Apr. 23, 2024. [Online]. Available: <https://jurnal.ilmubersama.com/index.php/sudo/article/view/229/155>
- [9] “Apa yang dimaksud dengan Manajemen Tugas? - Definisi - Manfaat | Zoho Projects.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.zoho.com/id/projects/task-management.html>
- [10] “6 Tips Produktif dengan Aplikasi Manajemen Tugas.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.talenta.co/blog/6-tips-produktif-dengan-aplikasi-manajemen-tugas/>
- [11] “Pendekatan Penelitian Pendidikan: Metode Penelitian Kualitatif, Metode Penelitian Kuantitatif dan Metode Penelitian Kombinasi (*Mixed Method*).” Accessed: Mei. 30, 2024. [Online]. Available: <https://jptam.org/index.php/jptam/article/download/6187/5167/11729>
- [12] “Kualitatif: Memahami Karakteristik Penelitian Sebagai Metodologi.” Accessed: Mei.

- 30, 2024. [Online]. Available:
<https://jurnalstkipmelawi.ac.id/index.php/JPD/article/view/902>
- [13] “Kualitatif : Ciri dan Karakter Sebagai Metodologi.” Accessed: Mei. 30, 2024. [Online]. Available:
<https://journal.unismuh.ac.id/index.php/equilibrium/article/view/4489/3096>
- [14] “Implementasi Web Service Waktu Shalat Berbasis Teknologi Paas Cloud Computing.” Accessed: Mei. 30, 2024. [Online]. Available: <https://ojs.uniska-bjm.ac.id/index.php/JIT/article/view/8074>
- [15] “Implementasi Web Service Untuk Mendukung Interoperabilitas Pada Aplikasi E-Commerce.” Accessed: Apr. 23, 2024. [Online]. Available:
<https://core.ac.uk/download/pdf/12342414.pdf>
- [16] “Perancangan Dan Implementasi Restful API Pada Sistem Informasi Manajemen Dosen Universitas Udayana.” Accessed: Mei. 30, 2024. [Online]. Available:
<https://ojs.unud.ac.id/index.php/spektrum/article/download/92900/46519>
- [17] “Understanding Rest API: Comprehensive Guide.” Accessed: Mei. 30, 2024. [Online]. Available: <https://medium.com/@MakeComputerScienceGreatAgain/understanding-rest-api-a-comprehensive-guide-52fc10f6c9ed>
- [18] “Rest API adalah: Perbedaan Rest API dan Restful API - DomaiNesia.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.domainesia.com/berita/rest-api-adalah/>
- [19] “Mahir Nest JS & Typescript sampai Deploy ke VPS | Udemey.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.udemy.com/course/menjadi-node-developer-js-dengan-nest-js-typescript/?couponCode=ST6MT42324>
- [20] “Apa itu NestJS? Pengertian dan contoh 2024 | RevoU.” Accessed: Apr. 23, 2024. [Online]. Available: <https://revou.co/kosakata/nestjs>
- [21] “NestJS : Framework Bagus Yang Membingungkan!” Accessed: Apr. 23, 2024. [Online]. Available: <https://ambrizal.net/blog/read/nestjs-framework-bagus-yang-membingungkan>
- [22] “Terlengkap: Panduan Menggunakan Postman API.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.sekawanmedia.co.id/blog/postman-api/>
- [23] “5 Alternatif Postman API Test yang Open Source – UNNES ICT Center.” Accessed: Apr. 23, 2024. [Online]. Available: <https://unnes.ac.id/ictcenter/2022/10/03/5-alternatif-postman-api-test-yang-open-source/>
- [24] “Mengenal Socket.IO Komunikasi Real-Time antara Klien dan Server - CODEPOLITAN.” Accessed: Apr. 23, 2024. [Online]. Available:
<https://www.codepolitan.com/blog/mengenal-socketio-komunikasi-realtime-antara->

klien-dan-server/

- [25] “Membuat Aplikasi Web Real-Time dengan Flask dan Socket.IO.” Accessed: Apr. 23, 2024. [Online]. Available: <https://blog.unmaha.ac.id/membuat-aplikasi-web-real-time-dengan-flask-dan-socket.io>
- [26] “Apa itu Git? Ini Pengertian & Perbedaannya dengan Github.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.dewaweb.com/blog/mengenal-git/>
- [27] “Apa Itu Git? Ini Pengertian, Fungsi, Fitur, dan Manfaatnya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.jagoanhosting.com/blog/git-adalah/>
- [28] “Apa Itu MySQL? Pengertian MySQL, Cara Kerja, dan Kelebihannya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-mysql>
- [29] “Pengertian MySQL, Fungsi, Cara Kerja, dan Kelebihannya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.dewaweb.com/blog/apa-itu-mysql/>
- [30] “Apa Itu MVC? Pahami Konsepnya dengan Baik - Dicoding Blog.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-mvc-pahami-konsepnya/>
- [31] “Mengenal MVC, Pengertian, Manfaat, dan Contohnya .” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.biznetgio.com/news/apa-itu-mvc>
- [32] “Apa itu UML? Ini Pengertian, Fungsi, dan Contohnya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.sekawanmedia.co.id/blog/apa-itu-uml/>
- [33] “Apa itu UML? Pengertian dan contoh 2024 | RevoU.” Accessed: Apr. 23, 2024. [Online]. Available: <https://revou.co/kosakata/uml>
- [34] “Apa itu UML? Beserta Pengertian dan Contohnya - Dicoding Blog.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-uml/>
- [35] “Apa Itu Agile? Pengertian, Prinsip, Metode, dan Kelebihan [Terlengkap].” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.niagahoster.co.id/blog/agile-adalah/>
- [36] “Mengenal Agile Development, Metode yang Cocok Diterapkan Developer.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.biznetgio.com/news/apa-itu-agile-development>
- [37] “Apa Itu Scrum? Pengertian, Prinsip dan Prosesnya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://greatnusa.com/artikel/apa-itu-scrum/>
- [38] “Scrum Prinsip Agile dan Tahapan Dalam Metode Scrum.” Accessed: Apr. 23, 2024. [Online]. Available: <https://bif.telkomuniversity.ac.id/scrum-prinsip-agile-dan-tahapan-dalam-metode-scrum/>
- [39] “Apa itu Scrum? Ini Tahapan Metodenya dalam Project Management.” Accessed: Apr. 23, 2024. [Online]. Available: <https://info.populix.co/articles/apa-itu-scrum/>
- [40] IdMetafora.com, “Pentingnya Melakukan Testing: Apa Itu Testing? Pengertian, Jenis,

- Tujuan, Konsep, dan Tahapannya, Jasa Pembuatan Website - Metafora Indonesia Tehnology,” Apr. 2023.
- [41] “Unit Testing: Pengertian, Manfaat, Teknik dan Toolsnya - Glints Blog.” Accessed: Apr. 23, 2024. [Online]. Available: <https://glints.com/id/lowongan/unit-testing-adalah/>
- [42] “Blackbox Testing Adalah : Pengertian dan Contohnya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://itbox.id/blog/black-box-testing-adalah/>
- [43] “Blackbox Testing: Pengertian, Kelebihan, dan Kekurangannya.” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.dewaweb.com/blog/pengertian-black-box-testing/>
- [44] “Pengertian, Rumus, dan Cara Menghitung Skala Likert Halaman 2 - Edukasi Katadata.co.id.” Accessed: Apr. 23, 2024. [Online]. Available: <https://katadata.co.id/lifestyle/edukasi/6492a0d1a4b93/pengertian-rumus-dan-cara-menghitung-skala-likert?page=2>
- [45] “Pengertian Skala Likert.” Accessed: Apr. 23, 2024. [Online]. Available: <https://nsd.co.id/posts/10002-pengertian-skala-likert.html>
- [46] A. Mobile et al., “Aplikasi Mobile Absensi Karyawan Dan Pengajuan Cuti Berbasis GPS,” *CCIT (Creative Communication and Innovative Technology) Journal*, vol. 12, no. 2, pp. 241–247, Aug. 2019, doi: 10.33050/CCIT.V12I2.695.
- [47] I. Rizq, D. Muhammad, and I. V Papatungan, “Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System,” *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 3, no. 2, pp. 79–87, Jan. 2024, doi: 10.20885/SNATI.V3.I2.35.
- [48] “View of PERANCANGAN SISTEM INFORMASI MANAJEMEN PROYEK DENGAN MENGGUNAKAN SOFTWARE NEST.JS BERBASIS WEB DI PT. MITRA PAJAKKU.” Accessed: Apr. 23, 2024. [Online]. Available: <https://journal.piksi.ac.id/index.php/INFOKOM/article/view/818/482>
- [49] “Rancang Bangun Backend Sistem Pemantauan Dokumen Akreditasi Menggunakan Kerangka Kerja Laravel Berbasis RESTful API | Saputra | Jurnal Teknik Komputer.” Accessed: Apr. 23, 2024. [Online]. Available: <https://ejournal3.undip.ac.id/index.php/jtk/article/view/38320/30535>
- [50] R. Suhaimi, N. Santoso, and R. A. Siregar, “Pengembangan Sistem Manajemen Proyek Menggunakan Metode Scrum Berbasis Android,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 6, pp. 1897–1905, Aug. 2020, Accessed: Apr. 23, 2024. [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/7457>
- [51] Y. Ardilla, S. Pramesti, and D. Sunaryono, “Rancang Bangun Sistem Informasi

Manajemen Proyek Departemen IT PT. Pertamina UPMS V Surabaya,” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 5, no. 3, pp. 136–145, Nov. 2020, doi: 10.14421/jiska.2020.53-01.

- [52] “Pengertian Rancangan Penelitian dan Jenis-jenisnya | kumparan.com.” Accessed: Apr. 23, 2024. [Online]. Available: <https://kumparan.com/ragam-info/pengertian-rancangan-penelitian-dan-jenis-jenisnya-21Q2LcXMoal>
- [53] “View of Analisis Dan Perancangan Sistem Informasi Pengolahan Data Berobat Karyawan” Accessed: Mei. 30, 2024. [Online]. Available: <https://ejournal.undhari.ac.id/index.php/simtika/article/download/74/53>
- [54] “Perancangan Sistem Informasi Kepegawaian Di Kantor Bagian Administrasi Pembangunan Sekretariat Daerah Kabupaten Karimun Berbasis Website” Accessed: Jun. 01, 2024. [Online]. Available: https://ejournal.universitaskarimun.ac.id/index.php/teknik_informatika/article/download/318/198
- [55] “Analisa Kebutuhan Kebutuhan Sistem Informasi Manajemen Perusahaan Dagang” Accessed: Jun. 07, 2024. [Online]. Available: <https://jurnal.machung.ac.id/index.php/kurawal/article/download/400/199>
- [56] “View of Analisis Perancangan Sistem Informasi” Accessed: Jun. 07, 2024. [Online]. Available: <https://repository.penerbitwidina.com/publications/407171/analisis-perancangan-sistem-informasi>

STT - NF

DAFTAR REFERENSI GAMBAR

- [A1] “Understanding REST API: A Comprehensive Guide.” Accessed: Apr. 23, 2024. [Online]. Available: <https://medium.com/@MakeComputerScienceGreatAgain/understanding-rest-api-a-comprehensive-guide-52fc10f6c9ed>
- [A2] “Apa Itu Agile? Pengertian, Prinsip, Metode, dan Kelebihan [Terlengkap].” Accessed: Apr. 23, 2024. [Online]. Available: <https://www.niagahoster.co.id/blog/agile-adalah/>
- [A3] “Scrum Prinsip Agile dan Tahapan Dalam Metode Scrum.” Accessed: Apr. 23, 2024. [Online]. Available: <https://bif.telkomuniversity.ac.id/scrum-prinsip-agile-dan-tahapan-dalam-metode-scrum/>
- [A4] “User Acceptance Test (UAT).” Accessed: Apr. 23, 2024. [Online]. Available: <https://fit.labs.telkomuniversity.ac.id/user-acceptance-test-uat/>
- [A5] “Dokumentasi Pribadi: Tahapan Penelitian Karir IO.” Created: Apr. 20, 2024.
- [A6] “Dokumentasi Pribadi: Arsitektur Sistem Karir IO.” Created: Apr. 24, 2024.
- [A7] “Dokumentasi Pribadi: Alur Aplikasi Karir IO.” Created: Apr. 30, 2024.
- [A8] “Dokumentasi Pribadi: Use Case Diagram Karir IO.” Created: Mei. 05, 2024.
- [A9] “Dokumentasi Pribadi: Activity Diagram Karir IO.” Created: Mei. 15, 2024.
- [A10] “Dokumentasi Pribadi: Sequence Diagram Karir IO.” Created: Mei. 25, 2024.
- [A11] “Dokumentasi Pribadi: Class Diagram Karir IO.” Created: Jun. 10, 2024.
- [A12] “Dokumentasi Pribadi: Entity Relationship Diagram (ERD) Karir IO.” Created: Jun. 18, 2024.

STT - NF