

**PENGEMBANGAN REST API PADA SISTEM *MONITORING* KPI
MENGUNAKAN *FRAMEWORK* LARAVEL
DI PT INDUSTRI KERETA API (PERSERO)**

(Skripsi)

Oleh

**SADDAM SURYA MARDIANSYAH
NPM 2017051014**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

PENGEMBANGAN REST API PADA SISTEM *MONITORING* KPI MENGGUNAKAN *FRAMEWORK* LARAVEL DI PT INDUSTRI KERETA API (PERSERO)

Oleh

SADDAM SURYA MARDIANSYAH

Dalam era bisnis yang semakin kompetitif dan dinamis, perusahaan-perusahaan modern semakin menyadari pentingnya memantau *Key Performance Indicators* (KPI) untuk memastikan pencapaian target kinerja. PT Industri Kereta Api (Persero) dihadapkan pada tantangan dalam mengelola dan memantau banyak KPI, sehingga penelitian ini diarahkan pada pengembangan sistem *monitoring* KPI dengan memanfaatkan REST API untuk menangani data dalam jumlah besar dan memastikan informasi yang disajikan dapat diandalkan. REST API diterapkan untuk menyederhanakan pengelolaan data yang kompleks dan memastikan operasional yang efisien, sehingga memberikan nilai tambah optimal bagi pengambilan keputusan manajemen. Dengan REST API, sistem pemantauan KPI dapat diakses melalui berbagai platform, termasuk *web* dan *mobile*, yang meningkatkan fleksibilitas dan kemudahan akses informasi bagi pengguna. Penelitian ini bertujuan untuk mengembangkan sistem *monitoring* KPI yang terintegrasi, memanfaatkan REST API untuk menyediakan akses data yang efisien dan *real-time*, serta merancang dan menguji aplikasi *mobile*. Berbagai pengujian dilakukan, termasuk *security testing* yang menunjukkan sistem memiliki tingkat keamanan yang tinggi, *performance testing* yang menunjukkan aplikasi mampu menangani beban berat dengan waktu respons yang konsisten, *black box testing* yang memastikan sistem bekerja sesuai spesifikasi tanpa adanya kesalahan besar, serta *User Acceptance Testing* (UAT) yang menghasilkan tingkat kepuasan pengguna sebesar 88,75% dengan kategori “Sangat Memuaskan.” Penelitian ini menggunakan metode *Personal Extreme Programming* (PXP), suatu proses iteratif yang sesuai untuk pengembangan perangkat lunak oleh individu, yang memastikan fleksibilitas dan akuntabilitas dalam menanggapi kebutuhan yang berubah.

Kata Kunci: *key performance indicators* (KPI); *mobile*; *real time*; REST API.

ABSTRACT

DEVELOPMENT OF REST API FOR KPI MONITORING SYSTEM USING LARAVEL FRAMEWORK AT PT INDUSTRI KERETA API (PERSERO)

By

SADDAM SURYA MARDIANSYAH

In an increasingly competitive and dynamic business environment, modern companies are becoming more aware of the importance of monitoring Key Performance Indicators (KPIs) to ensure that performance targets are met. PT Industri Kereta Api (Persero) faces challenges in managing and monitoring numerous KPIs, prompting this research to focus on developing a KPI monitoring system that leverages REST API to handle large volumes of data and ensure the reliability of the information presented. REST API is implemented to streamline the management of complex data and ensure efficient operations, thereby providing optimal value for management decision-making. By utilizing REST API, the KPI monitoring system can be accessed across various platforms, including web and mobile, enhancing flexibility and ease of information access for users. This research aims to develop an integrated KPI monitoring system, employing REST API for efficient and real-time data access, and designing and testing a mobile application. Comprehensive testing is conducted, including security testing demonstrating high system security, performance testing showing the application's capability to handle heavy loads with consistent response times, black box testing confirming that the system operates according to specifications without major errors, and user acceptance testing (UAT) achieving a user satisfaction rate of 88.75% with a "Very Satisfactory" rating. The research employs Personal Extreme Programming (PXP), an iterative process suitable for individual software development, ensuring flexibility and accountability in addressing changing requirements.

Keyword: *key performance indicators (KPI); mobile; real time; REST API.*

**PENGEMBANGAN REST API PADA SISTEM *MONITORING* KPI
MENGUNAKAN *FRAMEWORK* LARAVEL
DI PT INDUSTRI KERETA API (PERSERO)**

Oleh

SADDAM SURYA MARDIANSYAH

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

Judul Skripsi : **PENGEMBANGAN REST API PADA SISTEM MONITORING KPI MENGGUNAKAN FRAMEWORK LARAVEL DI PT INDUSTRI KERETA API (PERSERO)**

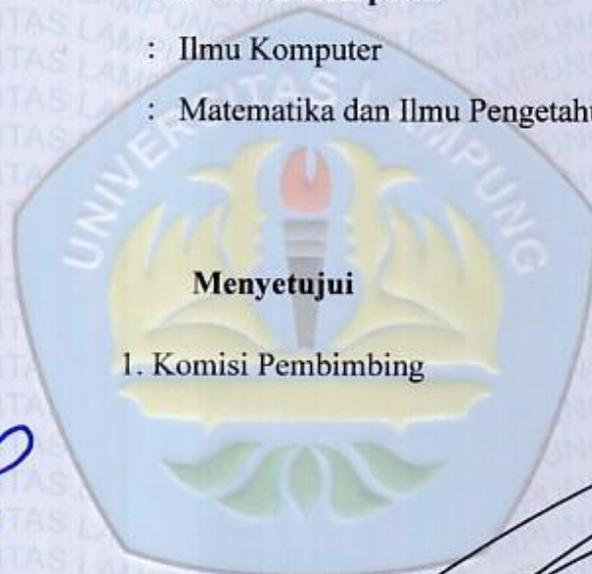
Nama Mahasiswa : **Saddam Surya Mardiansyah**

Nomor Pokok Mahasiswa : 2017051014

Program Studi : **S1 Ilmu Komputer**

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



Menyetujui

1. Komisi Pembimbing

Dewi Asiah Shofiana, S.Komp., M.Kom.
NIP. 199509292020122030

M. Iqbal Parabi, S.SI., M.T.
NIP. 199011302015041002

2. Ketua Jurusan Ilmu Komputer

Dwi Sakethi, S.Si., M.Kom
NIP. 196806111998021001

MENGESAHKAN

1. Tim Penguji

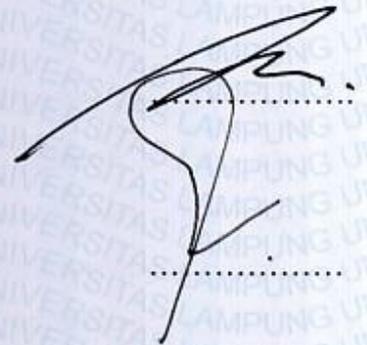
Ketua

: Dewi Asiah Shofiana, S.Komp., M.Kom.



Sekretaris Penguji

: M. Iqbal Parabi, S.SI., M.T.



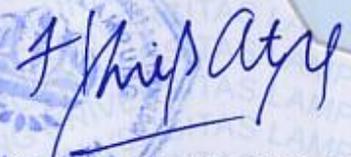
Penguji

Bukan Pembimbing

: Prof. Admi Syarif, Ph.D.



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M. Si

NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: 9 Agustus 2024

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Saddam Surya Mardiansyah

NPM : 2017051014

Menyatakan bahwa skripsi saya yang berjudul **“Pengembangan REST API Pada Sistem Monitoring KPI Menggunakan Framework Laravel di PT Industri Kereta Api (Persero)”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 20 Agustus 2024



Saddam Surya Mardiansyah
NPM. 2017051014

RIWAYAT HIDUP



Lahir di Ibukota Jakarta, pada hari Rabu, 19 Maret 2003.

Anak ketiga dari empat bersaudara, dari Bapak Chazairin dan Ibu Rochmah. menyelesaikan pendidikan di SDN 02 Jagabaya 1 Kota Bandar Lampung pada tahun 2014, kemudian menyelesaikan pendidikan menengah pertama di SMPN 05 Bandar Lampung pada tahun 2017, dan lulus dari pendidikan menengah atas di SMAN 10 Bandar Lampung pada tahun 2020.

Pada tahun 2020, terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Kegiatan-kegiatan yang dilakukan selama menjadi mahasiswa yaitu sebagai berikut.

1. Menjadi Asisten Dosen mata kuliah Matematika di Jurusan Ilmu Komputer pada tahun ajaran Ganjil 2021/2022.
2. Menjadi Asisten Dosen mata kuliah Matematika Diskrit di Jurusan Ilmu Komputer pada tahun ajaran Genap 2021/2022.
3. Menjadi Asisten Dosen mata kuliah Komunikasi Data dan Jaringan Komputer di Jurusan Ilmu Komputer pada tahun ajaran Ganjil 2022/2023.
4. Menjadi Anggota Bidang Badan Khusus Himpunan Mahasiswa Jurusan IlmuKomputer periode 2022/2023.

5. Menjadi Anggota Divisi Kemitraan dan Kerjasama Generasi Baru Bank Indonesia (GenBI) Universitas Lampung Periode 2021/2022.
6. Menjadi Kepala Divisi Kemitraan dan Kerjasama Generasi Baru Bank Indonesia (GenBI) Universitas Lampung Periode 2022/2023.
7. Menjadi Ketua Pelaksana Pekan Raya Jurusan Ilmu Komputer X Himakom Tournament FMIPA Unila periode 2022.
8. Melaksanakan Program Short Course Desain Interaksi Untuk UI/UX Designer Pemula Kredensial Mikro Mahasiswa Indonesia (KMMI) di jurusan Ilmu Komputer Fakultas MIPA Unila periode 2021/2022.
9. Melaksanakan Kerja Praktik pada bulan Desember – Januari periode 2022/2023 di Badan Pengawasan Keuangan dan Pembangunan Provinsi Lampung.
10. Melaksanakan Kerja Praktik pada bulan Oktober periode 2022/2023 di Kpw Bank Indonesia Provinsi Lampung.
11. Melaksanakan Magang Merdeka Belajar – Kampus Merdeka (MBKM) pada bulan Februari – Juni periode 2022/2023 di Departemen IT PT Industri Kereta Api (Persero).
12. Melaksanakan Kuliah Kerja Nyata Kota Bandar Lampung Pada tahun 2022/2023.

MOTO

“Sesungguhnya bersama kesulitan ada kemudahan.”

(Al-Insyirah ayat 6)

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

(Al-Baqarah ayat 286)

“Tidak ada yang bisa menolak takdir kecuali doa. Dan tidak ada yang bisa menambah umur melainkan perbuatan baik.”

(HR. Ahmad dari Tsauban radhiyallahu 'anhu)

“Never give up, because challenges do not require validation from others.”

(Saddam Surya)

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji dan syukur tercurahkan kepada Allah Subhanahu Wa Ta'ala atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan Kepada Nabi Muhammad SAW.

Kupersembahkan karya ini kepada:

Keluarga Tercinta

Ayah, Ibu, Kakak, dan Adik

Yang senantiasa memberikan yang terbaik dan melantunkan do'a yang selalu Menyertaiku. Kuucapkan pula terimakasih sebesar-besarnya kepada Ayah dan Ibu karena telah mendidik dan membesarkanku dengan cara yang dipenuhi kasih sayang, dukungan, dan pengorbanan. Kupersembahkan semua ini untuk kalian, Tanpa kalian, semua ini tidak akan mungkin terwujud. Setiap langkah yang kuambil adalah hasil dari cinta dan kerja keras kalian.

Seluruh Keluarga Besar Ilmu Komputer 2020

Yang selalu memberikan semangat dan dukungan.

Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat menimba ilmu, untuk menjadi bekal hidup dunia dan akhirat.

SANWACANA

Alhamdulillah Rabbi ‘Alamin. Segala puji dan syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta’ala yang telah memberikan nikmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini. Sholawat serta salam penulis sanjungkan kepada Baginda Nabi Muhammad Shallallahu ‘Alaihi Wasallam yang penulis harapkan syafaatnya di hari akhir kelak.

Skripsi yang berjudul “Pengembangan REST API Pada Sistem Monitoring KPI Menggunakan Framework Laravel di PT Industri Kereta Api (Persero)” ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana jurusan Ilmu Komputer Universitas Lampung. Selesaiannya Skripsi ini tidak terlepas dari bantuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, ucapan terima kasih ditujukan kepada:

1. Allah Subhanahu Wa Ta’ala yang telah memberikan nikmat kesehatan dan kemampuan untuk menyelesaikan skripsi ini.
2. Ayahanda tercinta Chazairin, Ibunda tercinta Rochmah, kakak - kakakku Fajar Apriansyah dan Fikri Ismail Abdullah, Adikku Destiara Amirah Nabilah serta Keluarga besar yang selalu memberikan do'a dan dukungan.
3. Bapak Febi Eka Febriansyah, M.T. sebagai Dosen Pembimbing Akademik yang telah memberikan arahan, ide, motivasi, dan dukungan akademik penulis.
4. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom. sebagai Dosen Pembimbing Utama yang telah memberikan arahan, ide, motivasi, kritik serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.

5. Bapak M. Iqbal Parabi, S.SI., M.T. sebagai Dosen Pembimbing 2 yang telah memberikan masukan yang bermanfaat dalam perbaikan skripsi ini.
6. Bapak Prof. Admi Syarif, Ph.D. sebagai Dosen Pembahas yang telah memberikan masukan yang bermanfaat dalam perbaikan skripsi ini dan mendukung peningkatan akademik penulis.
7. Bapak Dwi Sakethi, S.Si., M.Kom. selaku ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
9. Ibu Anie Rose Irawati S.T., M.Cs. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
10. Ibu Ade Nora Maela, Bang Zainuddin dan Mas Nofal yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
11. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
12. Keluarga Besar Ilmu Komputer 2020 yang tidak bisa penulis sebutkan satu persatu.
13. Seluruh pihak yang terlibat dalam proses penelitian skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Bandar Lampung, 20 Agustus 2024

Saddam Surya Mardiansyah
NPM. 2017051014

DAFTAR ISI

	Halaman
DAFTAR ISI	xiv
DAFTAR TABEL	xvii
DAFTAR GAMBAR	xviii
DAFTAR KODE PROGRAM	xxi
I. PENDAHULUAN	1
I.1. Latar Belakang.....	1
I.2. Rumusan Masalah	2
I.3. Batasan Masalah.....	3
I.4. Tujuan Penelitian.....	3
I.5. Manfaat Penelitian.....	3
II. TINJAUAN PUSTAKA	5
II.1. Penelitian Terdahulu.....	5
II.2. Konsep Dasar Sistem.....	12
II.2.1. Sistem Informasi	12
II.2.2. Aplikasi <i>Web</i>	13
II.2.3. Aplikasi <i>Mobile</i>	14
II.2.4. Bahasa Pemrograman.....	15
II.2.5. Basis Data	16
II.2.6. <i>Unified Modeling Language (UML)</i>	19
II.3. Teori Pendukung	22

II.3.1.	<i>Monitoring</i>	22
II.3.2.	KPI	23
II.3.3.	<i>Framework</i>	24
II.3.4.	REST API.....	24
II.3.5.	<i>Gantt Chart</i>	25
II.4.	Metode <i>Personal Extreme Programming (PXP)</i>	25
II.5.	Aplikasi Pendukung.....	29
II.5.1.	Laravel.....	29
II.5.2.	<i>Bootstrap</i>	29
II.5.3.	JSON Web Token (JWT)	30
II.5.4.	Postman	30
II.5.5.	OWASP ZAP	31
II.5.6.	JMeter.....	31
II.5.7.	Visual Studio Code	31
II.5.8.	Android Studio	32
II.5.9.	XAMPP	32
II.6.	<i>System Testing</i>	33
II.6.1.	<i>Security Testing</i>	33
II.6.2.	<i>Performance Testing</i>	33
II.6.3.	Metode <i>Black Box Testing</i>	33
II.6.4.	<i>User Acceptance Testing (UAT)</i>	34
III.	METODOLOGI PENELITIAN.....	35
III.1.	Waktu dan Tempat Penelitian.....	35
III.2.	Alat Penelitian	36
III.2.1.	Perangkat Lunak.....	36
III.2.2.	Perangkat Keras	36
III.3.	Tahapan Penelitian	37
III.3.1.	<i>Requirement</i>	38

III.3.2. <i>Planning</i>	42
III.3.3. <i>Iteration Initialization</i>	43
III.3.4. <i>Design</i>	43
III.3.5. <i>Implementation</i>	51
III.3.6. <i>System Testing</i>	51
III.3.7. <i>Retrospective</i>	54
IV. HASIL DAN PEMBAHASAN	55
IV.1. <i>Iteration Initialization</i>	55
IV.2. <i>Design</i>	58
IV.2.1. <i>Wireframe</i>	58
IV.2.2. <i>UI UX</i>	61
IV.3. <i>Implementation</i>	66
IV.3.1. <i>Pembangunan Backend Server</i>	67
IV.3.2. <i>Pengembangan Aplikasi Mobile</i>	70
IV.4. <i>System Testing</i>	78
IV.4.1. <i>Security Testing</i>	78
IV.4.2. <i>Performance Testing</i>	91
IV.4.3. <i>Black Box Testing</i>	93
IV.4.4. <i>User Acceptance Testing (UAT)</i>	97
IV.5. <i>Retrospective</i>	99
V. SIMPULAN DAN SARAN	100
V.1. <i>Simpulan</i>	100
V.2. <i>Saran</i>	101
DAFTAR PUSTAKA	102
LAMPIRAN	106

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu	4
2. Simbol – Simbol <i>Use Case Diagram</i>	19
3. Simbol – Simbol <i>Activity Diagram</i>	20
4. Simbol – Simbol <i>Entity Relationship Diagram</i>	21
5. Waktu Penelitian Menggunakan <i>Gantt Chart</i>	33
6. Tabel Wawancara.....	36
7. Kebutuhan Fungsional Sistem.....	38
8. Kebutuhan Non-Fungsional Sistem	39
9. Tabel <i>Planning</i> Pengembangan Sistem.....	40
10. <i>Skor Likert</i>	52
11. Indeks Kepuasan Pengguna.....	52
12. <i>Endpoint Login</i>	66
13. <i>Endpoint Super Admin</i>	67
14. <i>Endpoint Program Kerja</i>	68
15. <i>Endpoint Realisasi</i>	69
16. Hasil <i>Vulnerability Assessment</i>	89
17. Hasil <i>Black Box Testing</i>	93
18. Pertanyaan <i>User Acceptance Testing (UAT)</i>	97
19. Hasil Jawaban <i>Responden</i>	98

DAFTAR GAMBAR

Gambar	Halaman
1. <i>Gantt Chart</i>	25
2. Metode PXP.	27
3. Tahapan Penelitian Menggunakan Metode PXP.....	35
4. <i>Entity Relationship Diagram (ERD) Sistem Monitoring KPI</i>	44
5. <i>Use Case Diagram Sistem Monitoring KPI</i>	45
6. <i>Activity Diagram Monitoring Progress Sistem Monitoring KPI</i>	46
7. <i>Activity Diagram Input Perencanaan Program Kerja Sistem Monitoring KPI</i>	47
8. <i>Activity Diagram Input Pelaporan Realisasi Program Kerja Sistem Monitoring KPI</i>	48
9. <i>Activity Diagram Approve Pelaporan Realisasi Program Kerja Sistem Monitoring KPI</i>	49
10. <i>Activity Diagram Reject Pelaporan Realisasi Program Kerja Sistem Monitoring KPI</i>	50
11. <i>Wireframe Menu Sign In</i>	58
12. <i>Wireframe Menu Dashboard</i>	59
13. <i>Wireframe Menu Program Kerja</i>	59
14. <i>Wireframe Menu Detail Program Kerja</i>	60
15. <i>Wireframe Menu Sidebar</i>	61
16. <i>UI Splash Screen</i>	62
17. <i>UI Menu Sign In</i>	62

18. UI Menu <i>Dashboard</i>	63
19. UI Program Kerja.	64
20. UI Detail Program Kerja.	65
21. UI <i>Sidebar</i>	66
22. Tampilan <i>Splash Screen</i>	72
23. Tampilan <i>Sign In</i>	73
24. Tampilan <i>Dashboard</i>	74
25. Tampilan Program Kerja Direksi.	74
26. Tampilan Program Kerja Pegawai.	75
27. Tampilan Detail Program Kerja.	76
28. Tampilan <i>Task</i> per Bulan.	77
29. Tampilan <i>Sidebar</i>	77
30. Postman <i>Routes Login</i>	79
31. Postman <i>Routes Check Token</i>	79
32. Postman <i>Routes Refresh Token</i>	80
33. Postman <i>Routes Logout</i>	80
34. Postman <i>Routes Index Super Admin</i>	81
35. Postman <i>Routes Show Super Admin</i>	82
36. Postman <i>Routes Store Super Admin</i>	82
37. Postman <i>Routes Update Super Admin</i>	83
38. Postman <i>Routes Delete Super Admin</i>	83
39. Postman <i>Routes Index Program Kerja</i>	84
40. Postman <i>Routes Store Program Kerja</i>	85
41. Postman <i>Routes Update Program Kerja</i>	85
42. Postman <i>Routes Show Realisasi</i>	86
43. Postman <i>Routes Store Realisasi</i>	87
44. Postman <i>Routes Update Realisasi</i>	87
45. <i>Prosess Automated Scanner</i>	88

46. Hasil Kerentanan.....	89
47. <i>Thread Group Properties</i>	91
48. <i>Response Time Graph Result</i>	92

DAFTAR KODE PROGRAM

Kode Program	Halaman
1. Potongan Kode Konfigurasi Koneksi Internet.	71

I. PENDAHULUAN

I.1. Latar Belakang

Dalam era bisnis yang kompetitif dan dinamis, perusahaan-perusahaan modern semakin menyadari pentingnya memahami dan mengukur *Key Performance Indicators* (KPI). KPI merupakan alat bantu atau instrumen manajemen yang digunakan untuk memantau, mengendalikan, dan memastikan bahwa suatu kegiatan atau proses sesuai dengan kinerja yang diinginkan. KPI memungkinkan perbandingan antara hasil yang telah dicapai dengan standar yang telah ditetapkan sebelumnya. Keberhasilan implementasi KPI sangat tergantung pada pelaksanaan strategi pemeliharaan yang efektif, sesuai dengan standar yang telah ditetapkan (Kusumanto et al., 2018).

Saat ini, sebagian besar perusahaan telah mengambil langkah untuk mengimplementasikan KPI sebagai bagian integral dari strategi bisnis perusahaan. Salah satu contoh perusahaan yang menghadapi tantangan dalam mengelola dan memantau sejumlah besar KPI adalah PT Industri Kereta Api (Persero), sebagai perusahaan manufaktur kereta api dengan skala operasional besar. Dalam konteks ini, perusahaan merasa perlunya merancang dan mengimplementasikan sistem *monitoring* KPI yang lebih canggih untuk menangani volume data yang signifikan dan memastikan informasi yang diperoleh benar-benar relevan dan dapat diandalkan.

Pentingnya inovasi dalam sistem dan teknologi terbaru di PT Industri Kereta Api (Persero) menjadi pilar utama dalam kemajuan perusahaan,

khususnya melalui partisipasi tim IT yang memiliki tanggung jawab terhadap penelitian dan pengembangan teknologi. Penggunaan REST API dianggap sebagai langkah proaktif untuk mengatasi kompleksitas data dan memastikan operasional efisien dari sistem pemantauan KPI, memberikan manfaat optimal bagi kebijakan manajemen perusahaan. Dengan menggunakan REST API, sistem pemantauan KPI dapat beroperasi di berbagai *platform* seperti *web* dan *mobile* sesuai dengan kebutuhan, meningkatkan fleksibilitas dan aksesibilitas informasi bagi pengguna.

Perusahaan yang cukup besar harus memperhatikan faktor-faktor pendukung untuk tetap bertahan dan berkembang, terutama peran sumber daya manusia dalam mencapai produktivitas yang efektif dan efisien (Bakhtiar et al., 2016). Untuk mewujudkan visi dan misi, perusahaan juga perlu mengembangkan strategi manajemen, melibatkan karyawan dengan penetapan target kinerja mingguan, bulanan, dan tahunan (Iveta, 2012).

Dengan menerapkan REST API pada sistem *monitoring* KPI di PT Industri Kereta Api (Persero) dapat memperoleh data secara otomatis dan *real-time*, memungkinkan manajemen memiliki visibilitas yang lebih baik terhadap kinerja operasional mereka. Hal ini diharapkan dapat membantu dalam pengambilan keputusan yang lebih cepat dan tepat guna, mendukung pertumbuhan dan daya saing perusahaan di industri perkeretaapian secara keseluruhan. Oleh karena itu, penelitian ini memiliki tujuan untuk mengembangkan sistem *monitoring* KPI dengan menggunakan REST API untuk mendukung efisiensi operasional dalam memantau KPI di PT Industri Kereta Api (Persero).

I.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana mengembangkan REST API pada sistem *monitoring* KPI di PT Industri Kereta Api (Persero).

I.3. Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut.

- a. Terbatas pada konteks PT Industri Kereta Api (Persero) sebagai objek penelitian.
- b. Fokus pada pengembangan REST API pada sistem *monitoring* KPI dengan menggunakan *framework* Laravel.
- c. Aplikasi *mobile* yang dikembangkan hanya untuk platform Android.

I.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut.

- a. Mengembangkan sistem *monitoring* KPI yang dapat mengintegrasikan data dari berbagai unit kerja di PT Industri Kereta Api (Persero).
- b. Menerapkan REST API untuk menyediakan akses data yang efisien dan *real-time*.
- c. Mengembangkan dan menguji aplikasi *mobile* dengan desain UI/UX yang baik untuk memastikan pengalaman pengguna yang optimal dalam penggunaan aplikasi *monitoring* KPI.
- d. Melakukan pengujian menyeluruh yang mencakup *security testing*, *performance testing*, *blackbox testing*, dan *user acceptance testing* (UAT) untuk memastikan bahwa sistem yang dikembangkan aman, dapat diandalkan, dan mampu menangani beban tinggi.

I.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut.

- a. Manfaat untuk PT Industri Kereta Api, mendukung efisiensi operasional melalui sistem *monitoring* KPI yang diperbarui dalam memberikan informasi yang akurat dan terkini.

- b. Manfaat untuk Divisi Riset dan Pengembangan, memantau dan mengevaluasi hasil pengembangan dengan lebih efisien, memungkinkan penyesuaian yang diperlukan secara cepat serta mendorong inovasi dalam pengembangan produk dan proses produksi berdasarkan data kinerja yang terukur.
- c. Manfaat untuk penelitian, menambah pemahaman tentang penggunaan REST API dalam pengembangan sistem *monitoring* KPI dan memberikan panduan praktis bagi perusahaan lain yang ingin mendukung efisiensi operasional melalui sistem *monitoring* KPI yang diperbarui dengan menggunakan REST API.

II. TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Dalam menyusun penelitian ini, beberapa penelitian terdahulu dijadikan sebagai acuan dalam mengadopsi konsep, teori, dan pembangunan kerangka berpikir. Ringkasan dari penelitian sebelumnya dapat dilihat pada Tabel 1.

Tabel 1. Penelitian Terdahulu.

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Sukma, et al., 2019)	Pengembangan Sistem Informasi <i>Monitoring</i> Tugas Akhir (Monita) Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.	Peneliti berhasil mengembangkan Sistem Informasi MONITA menggunakan metode RUP dan melakukan pengujian menggunakan <i>black box testing</i> dan <i>equivalence partitioning</i> .	- Penelitian ini berfokus pada pengembangan sistem <i>monitoring</i> untuk memantau suatu aktivitas atau kinerja. - Penggunaan metode <i>black box testing</i> untuk menguji aplikasi.	- Penelitian ini tidak terintegrasi dengan API sementara penelitian ini menggunakan REST API ke dalam sistem. - Penelitian ini berfokus pada penerapan sistem yang mendukung akademik untuk <i>monitoring</i> Tugas Akhir sementara penelitian ini berfokus kepada sistem yang mendukung kinerja bisnis suatu perusahaan.

Tabel 1. (lanjutan)

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Septiawan, et al., 2021)	Penerapan Metode <i>Waterfall</i> Pengembangan Sistem Informasi <i>Monitoring</i> Proses Bimbingan Skripsi di Jurusan Ilmu Komputer Universitas Lampung.	Peneliti berhasil mengembangkan Sistem Informasi <i>Monitoring</i> Proses Bimbingan Skripsi menggunakan metode <i>Waterfall</i> dan melakukan pengujian menggunakan <i>black box testing</i> .	<ul style="list-style-type: none"> - Penelitian ini berfokus pada pengembangan sistem <i>monitoring</i> untuk memantau aktivitas atau kinerja. - Penggunaan metode <i>black box testing</i> untuk menguji aplikasi. 	<ul style="list-style-type: none"> - Penelitian ini tidak terintegrasi dengan API sementara penelitian ini menggunakan REST API ke dalam sistem. - Penelitian ini berfokus pada penerapan sistem yang mendukung akademik untuk <i>monitoring</i> tugas akhir sementara penelitian ini berfokus kepada sistem yang mendukung kinerja bisnis suatu perusahaan. - Sistem yang dikembangkan peneliti berbasis <i>web native</i> dengan bahasa pemrograman PHP sementara penelitian ini menggunakan <i>framework Laravel 7</i> bahasa pemrograman PHP. - Metode pengembangan perangkat lunak penelitian ini menggunakan <i>Waterfall</i> sementara penelitian ini menggunakan metode PXP.

Tabel 1. (lanjutan)

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Wijaya dan Astuti, 2021)	Pengujian <i>Black Box Sistem Informasi</i> Penilaian Kinerja Karyawan PT INKA (Persero) Berbasis <i>Equivalence Partitions</i> .	Peneliti berhasil dalam melakukan pengujian sistem informasi penilaian kinerja karyawan PT INKA (Persero) menggunakan metode <i>Black Box</i> berbasis <i>Equivalence Partitions</i> yang dapat menemukan kesalahan fungsionalitas pada setiap fitur sistem informasi penilaian kinerja karyawan PT INKA (Persero).	- Penelitian ini berfokus pada sistem untuk meningkatkan efektivitas pengukuran kinerja karyawan di PT Industri Kereta Api (Persero) dan PT INKA (Persero). - Penelitian ini menggunakan metode <i>black box testing</i> untuk menguji sistem tersebut.	- Penelitian ini berfokus pada pengujian fungsionalitas sistem sedangkan penelitian yang akan dilakukan berfokus pada pengembangan sistem.
(Kurniawan , 2020)	Rancang Bangun Sistem Informasi Penilaian Kinerja Karyawan PT INKA (Persero) Menggunakan Metode <i>Extreme Programming</i> .	Peneliti berhasil merancang sistem informasi penilaian kinerja karyawan di PT INKA (Persero) menggunakan metode <i>Extreme Programming</i> yang bertujuan untuk meningkatkan efektivitas kinerja karyawan dengan menilai kinerja dalam beberapa aspek.	- Penelitian ini berfokus pada sistem untuk meningkatkan efektivitas pengukuran kinerja karyawan di PT Industri Kereta Api (Persero) dan PT INKA (Persero). - Penelitian ini menggunakan metode <i>Extreme Programming</i> untuk merancang sistem tersebut.	- Penelitian ini tidak terintegrasi dengan API sementara penelitian ini menggunakan REST API ke dalam sistem.

Tabel 1. (lanjutan)

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Rosyidah dan Saputra, 2019)	Penentuan <i>Key Performance Indicators</i> (KPI) Dengan Metode <i>Performance Prism</i> (Studi Kasus Program Studi Teknik Industri Universitas Muhammadiyah Palembang).	Penelitian ini menunjukkan bahwa uji validitas dan <i>reliability</i> menggunakan Uji Cochran dan SPSS 16 menunjukkan bahwa semua aspek penilaian dari berbagai <i>stakeholder</i> (<i>employee</i> , mahasiswa, alumni, dan masyarakat) memiliki validitas yang baik. Selain itu, juga dilakukan pembobotan terhadap <i>Key Performance Indicator</i> (KPI) Program Studi Teknik Industri yang telah terbentuk dari proses dan kapabilitas yang telah diidentifikasi.	- Penelitian ini berfokus pada pada pengukuran kinerja dengan menggunakan <i>Key Performance Indicators</i> (KPI).	- penelitian ini berfokus pada penentuan KPI sedangkan penelitian ini berfokus pada pengembangan REST API pada sistem <i>monitoring</i> KPI. - Lokasi penelitian yang berbeda, penelitian terdahulu di Program Studi Teknik Industri Universitas Muhammadiyah Palembang sementara penelitian dilaksanakan di PT. Industri Kereta Api (Persero).

Tabel 1. (lanjutan)

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Pranata, 2018)	Perancangan <i>Application Programming Interface</i> (API) Berbasis <i>Web</i> Menggunakan Gaya Arsitektur <i>Representational State Transfer</i> (REST) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit.	Penerapan gaya arsitektur REST telah berhasil dilaksanakan melalui pemanfaatan <i>path, query</i> , dan metode permintaan (<i>request method</i>).	- Penelitian ini menerapkan gaya arsitektur <i>Representational State Transfer</i> (REST) API. - Penggunaan <i>software</i> pengujian API yaitu Postman untuk menguji beberapa <i>endpoint</i> .	- Metode pengembangan perangkat lunak penelitian terdahulu menggunakan <i>Waterfall</i> sementara penelitian ini menggunakan metode PXP.
(Filiana, et al., 2022)	Pengembangan REST API untuk Informasi Pasar Tradisional di Kota Yogyakarta dengan Metode <i>Incremental</i> .	Peneliti berhasil mengembangkan REST API untuk informasi pasar tradisional di Kota Yogyakarta. Namun, API yang telah dikembangkan terbatas hanya pada penggunaan metode permintaan GET.	- Penelitian ini menerapkan gaya arsitektur <i>Representational State Transfer</i> (REST) API. - Penggunaan <i>software</i> pengujian API yaitu Postman untuk menguji beberapa <i>endpoint</i> .	- Metode pengembangan perangkat lunak penelitian terdahulu menggunakan <i>Incremental</i> sementara penelitian ini menggunakan metode PXP.

Tabel 1. (lanjutan)

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Evanandy, 2022)	Implementasi <i>Personal Extreme Programming</i> Dalam Pengembangan Integrasi Sistem Informasi Akademik Dan <i>E-Learning</i> .	Penelitian ini berhasil menggunakan metode <i>Extreme Programming</i> (XP) dalam pengembangan Integrasi Sistem Informasi Akademik dan <i>E-Learning</i> . Penelitian ini berhasil mencapai tingkat efisiensi pengembangan mencapai 97% berdasarkan perbandingan antara estimasi waktu dengan waktu sebenarnya. Penerapan metode REST API dalam pengintegrasian modul jurnal perkuliahan, absensi, dan nilai meningkatkan kecepatan serta mengurangi jumlah tahapan yang diperlukan secara signifikan dibandingkan dengan sistem sebelumnya.	<ul style="list-style-type: none"> - Menggunakan metode pengembangan, yakni <i>Extreme Programming</i> (XP) dan <i>Personal Extreme Programming</i> (PXP). - Bertujuan meningkatkan efisiensi dan efektivitas dalam pengembangan sistem, dengan fokus pada integrasi data dan peningkatan kinerja. - Penelitian ini menerapkan gaya arsitektur <i>Representational State Transfer</i> (REST) API. 	- Penelitian ini berfokus pada integrasi sistem akademik dan <i>e-learning</i> di lingkungan pendidikan sementara penelitian ini berfokus kepada sistem yang mendukung kinerja bisnis suatu perusahaan.

Tabel 1. (lanjutan)

Peneliti	Judul Penelitian	Hasil Penelitian	Persamaan	Perbedaan
(Anjelina, 2023)	Rancang Bangun Sistem Peminjaman <i>Coworking Space</i> Berbasis <i>Website</i> Menggunakan <i>Mern Stack Technology</i> Pada Gedung Lembaga Kemahasiswaan Fisip Universitas Lampung.	Penelitian ini menggunakan metode pengembangan PXP dengan 3 iterasi, mencapai efisiensi 93%. Sistem Peminjaman <i>Coworking Space</i> (Sipenting) berhasil dibangun menggunakan <i>MERN stack</i> , memungkinkan pengelolaan data oleh <i>admin</i> dan <i>booking</i> ruangan oleh <i>customer</i> . Pengujian menunjukkan fitur berjalan sesuai kebutuhan yang menunjukkan keberhasilan sistem dalam memenuhi kebutuhan pengguna dan memberikan pengalaman positif.	- Menggunakan metode PXP dalam pengembangan sistem, menekankan kebutuhan klien dan hasil produk. - Menerapkan pengujian <i>black-box testing</i> untuk memastikan fitur-fitur sistem dapat berjalan sesuai kebutuhan pengguna.	- Bidang aplikasi penelitian berbeda, dengan penelitian ini berkaitan dengan manajemen ruang meeting, sementara penelitian ini berkaitan dengan <i>monitoring</i> kinerja di industri perkeretaapian. - Tujuan pengembangan sistem berbeda, di mana penelitian ini bertujuan memberikan solusi untuk manajemen ruang, sementara penelitian ini bertujuan untuk meningkatkan efektivitas pengukuran kinerja di perusahaan kereta api. - Penelitian ini menggunakan <i>Mern Stack Technology</i> dan MongoDB sedangkan penelitian ini menggunakan MySQL.

II.2. Konsep Dasar Sistem

II.2.1. Sistem Informasi

Sistem dapat didefinisikan sebagai suatu rangkaian elemen yang saling berhubungan atau terintegrasi dengan tujuan mencapai suatu target. Sistem merupakan suatu kesatuan yang terbentuk oleh elemen atau komponen yang saling terhubung untuk mempermudah aliran materi, energi, atau informasi. Keutuhan suatu sistem tergantung pada keterkaitan antar bagian, dan jika salah satu bagian mengalami kerusakan atau tidak dapat menjalankan fungsi, maka tujuan yang ingin dicapai tidak dapat terwujud atau sistem yang sudah terbentuk akan mengalami gangguan (Kadir, 2014).

Informasi adalah data yang telah diproses sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Sebaliknya data merupakan sekumpulan baris fakta yang mewakili peristiwa yang terjadi pada organisasi atau lingkungan fisik sebelum diolah dalam suatu format yang dapat dipahami dan digunakan orang (Kadir, 2014).

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2012).

Menurut Kadir (2014) komponen-komponen sistem informasi antara lain :

1. Orang, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
2. Teknologi informasi, terdiri dari perangkat keras (*hardware*) yang mencakup piranti-piranti fisik seperti komputer dan printer dan

- perangkat lunak (*software*) yang merupakan sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.
3. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersamaan atau diakses oleh sejumlah pemakai.
 4. *Prosedur*, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

II.2.2. Aplikasi Web

Aplikasi berbasis *web* adalah program yang dibuat menggunakan bahasa pemrograman HTML, PHP, CSS, dan JavaScript. Untuk menjalankannya, diperlukan *web* server dan *browser* seperti Chrome, Firefox, Opera, Internet Explorer, Microsoft Edge, dan lainnya. Aplikasi ini dapat diakses baik melalui jaringan lokal (LAN) maupun internet. Keistimewaan aplikasi *web* terletak pada pusatnya data dan kemudahan akses, membuatnya menjadi pilihan yang diminati dan mudah diimplementasikan dalam berbagai konteks kehidupan (Simarmata et al., 2021).

A. Web Browser

Web Browser merupakan suatu program atau perangkat lunak yang digunakan untuk mengakses internet atau mencari informasi dari suatu situs *web* yang disimpan di dalam komputer. Menurut Sibero (2013), *web browser* adalah perangkat lunak aplikasi yang dipergunakan untuk mengunduh dan menyajikan berbagai informasi dari internet, mencakup halaman *web*, video, gambar, dan konten lainnya.

B. Web Server

Web Server adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP pada klien yang biasanya dikenal sebagai *browser* (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya

dalam bentuk beberapa halaman *web* dan pada umumnya akan berbentuk dokumen HTML. Menurut Sibero (2013), *web server* merupakan suatu komputer yang terdiri dari perangkat keras dan perangkat lunak yang mengelola *browser*, dan *output*-nya dikirimkan kembali ke *browser*.

C. HTTP (*Hypertext Transfer Protocol*)

HTTP (*Hypertext Transfer Protocol*) adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif dan menggunakan hipermedia. Menurut Hidayatullah dan Kawistara (2015), *Hypertext Transfer Protocol* (HTTP) merupakan sebuah protokol yang memungkinkan komunikasi antara *client* dan *server* melalui gaya permintaan-respons. HTTP menetapkan standar mengenai format pesan, metode pengiriman, dan interaksi *web browser* terhadap berbagai perintah.

II.2.3. Aplikasi *Mobile*

Aplikasi *mobile* adalah aplikasi yang telah dirancang khusus untuk *platform mobile* (misalnya iOS, Android, atau Windows *Mobile*). Dalam banyak kasus, aplikasi *Mobile* memiliki *user interface* dengan mekanisme interaksi unik yang disediakan oleh *platform mobile*, interoperabilitas dengan sumber daya berbasis *web* yang menyediakan akses ke beragam informasi yang relevan dengan aplikasi, dan kemampuan pemrosesan lokal untuk pengumpulan, analisis, dan format informasi dengan cara yang paling cocok untuk *platform mobile*. Selain itu aplikasi *mobile* menyediakan kemampuan penyimpanan persisten dalam *platform* (Pressman & Maxim, 2014).

II.2.4. Bahasa Pemrograman

A. HTML

Menurut Anhar (2010), HTML digambarkan sebagai kumpulan simbol atau *tag* yang tertulis dalam sebuah *file*, digunakan untuk merender halaman pada *web browser*. *Tag* HTML selalu diawali dengan `<x>` dan diakhiri dengan `</x>`, dengan x sebagai *tag* HTML seperti b, i, u, dan sebagainya. Sementara menurut Solichin (2016) menjelaskan bahwa HTML adalah bahasa pemrograman *web* yang memberikan instruksi kepada peramban *web* (*web browser*) tentang cara menyusun dan menyajikan konten di halaman *web*.

B. PHP

Menurut Anhar (2010), PHP merupakan singkatan dari *Hypertext Preprocessor*, yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. Di sisi lain, Sibero (2013) menggambarkan PHP sebagai pemrograman interpreter, suatu proses penerjemahan baris kode sumber menjadi kode mesin yang langsung dimengerti oleh komputer saat baris kode tersebut dijalankan.

C. CSS

CSS (*Cascading Style Sheets*) adalah sebuah metode yang digunakan untuk mempersingkat penulisan *tag* HTML, seperti *font*, *color*, *text*, dan tabel menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan (Lewenusa, 2020). Menurut Listiyah (2022), CSS juga dapat digunakan untuk membantu *programmer* dalam merancang tampilan *website* dan menciptakan efek animasi yang menarik.

D. Javascript

Menurut Solichin (2016), JavaScript awalnya dikembangkan oleh Netscape dengan nama LiveScript. Fokus utamanya adalah pada proses pengolahan data di sisi *client* dan penyediaan komponen *web* yang lebih interaktif. Sementara itu, Sibero (2013) mendeskripsikan JavaScript sebagai bahasa skrip (*Scripting Language*), yakni

kumpulan instruksi perintah yang digunakan untuk mengendalikan beberapa bagian dari sistem operasi.

E. Kotlin

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh JetBrains. Bahasa pemrograman ini bersifat pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dengan bahasa pemrograman Java pada suatu *project* yang sama. Kotlin juga dapat digunakan untuk mengembangkan aplikasi berbasis *desktop*, *web* dan bahkan untuk *backend* (Bose et al., 2018).

II.2.5. Basis Data

A. Basis Data

Database (basis data) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pendefinisian *database* meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. Menurut Solichin (2016), basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari *database*. *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

B. MySQL

MySQL (*My Structure Query Language*) adalah suatu RDBMS (*Relational Database Management System*) yaitu Aplikasi yang menjalankan fungsi pengolahan data (Sibero, 2013). Menurut Anhar (2010), MySQL (*My Structure Query Language*) adalah salah satu

Database Management System (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postgre SQL, dan lainnya.

MySQL merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). Menurut (Sukamto & Shalahuddin, 2014), SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola *Relational Database Management System* (RDBMS) dan awalnya dikembangkan berdasarkan teori relasional dan kalkulus.

Menurut (Nugroho, 2008), SQL adalah bahasa permintaan yang terkait dengan *Database Management System* (DBMS), termasuk MySQL. Dalam pemakaian perintah SQL, perintah tersebut dapat dibagi menjadi tiga sub-perintah sebagai berikut:

1. *Data Definition Language* (DDL) adalah merupakan sub bahasa SQL yang digunakan untuk membangun kerangka *database*. Ada tiga perintah yang termasuk dalam DDL yaitu:
 - a. *Create*, perintah ini digunakan untuk membuat, termasuk diantaranya membuat *database* baru, tabel baru, *view* baru dan kolom.
 - b. *Alter*, perintah ini digunakan untuk mengubah struktur tabel yang telah dibuat.
 - c. *Drop*, perintah ini digunakan untuk menghapus *database* dan tabel.
2. *Data Manipulation Language* (DML) merupakan sub bahasa SQL yang digunakan untuk memanipulasi data dalam *database* yang terbuat. Perintah yang digunakan, di antaranya:
 - a. *Insert*, perintah ini digunakan untuk menyisipkan atau memasukkan data baru ke dalam tabel.
 - b. *Select*, perintah ini digunakan untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel dalam relasi.

- c. *Update*, perintah ini digunakan untuk menghapus data dari tabel.
 - d. *Delete*, perintah ini digunakan untuk menghapus data dari tabel.
3. *Data Control Language* (DCL) merupakan sub bahasa SQL yang digunakan untuk melakukan pengontrolan data dan *server database*-nya. Perintah DCL, diantaranya:
- a. *Grant*, perintah ini digunakan untuk memberikan hak/izin akses oleh *administrator* (pemilik utama) *server* kepada *user* (pengguna biasa).
 - b. *Revoke*, perintah ini memiliki kegunaan terbalik dengan *Grant* yaitu untuk menghilangkan atau mencabut hak akses yang telah diberikan kepada *user* oleh *administrator*.

C. **Phpmyadmin**

Phpmyadmin merupakan sebuah perangkat lunak untuk mengelola database MySQL dengan antarmuka grafis (GUI), penggunaan antarmuka grafis lebih memudahkan dari antarmuka pengelolaan asli MySQL yang berbasis teks. Menurut Rahman (2013), Phpmyadmin adalah sebuah *software* berbasis pemrograman PHP, berfungsi sebagai administrator MySQL melalui *web browser*, dan digunakan untuk *management database*.

Menurut Sibero (2013) mengemukakan bahwa Phpmyadmin adalah aplikasi *web* yang dibuat oleh Phpmyadmin.net. Phpmyadmin digunakan untuk administrasi *database* pada Phpmyadmin seperti fitur pembuatan *database*, mengubah *database*, pembuatan tabel, menghapus tabel, menambah data, menampilkan data, mengubah data, menghapus data, membuat *view*, menghapus *view*, membuat *index* kolom dan menghapus indeks kolom.

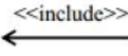
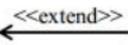
II.2.6. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah salah standar bahasa yang banyak digunakan dalam dunia industri untuk mendefinisikan *requirement* membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek, sarana untuk merancang atau membuat *software* berorientasi objek. Elemen dan diagram pada UML (*Unified Modeling Language*) berbasis pada paradigma *object oriented*. UML sendiri memiliki standar penulisan sebuah sistem, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. UML (*Unified Modeling Language*) dapat juga disebut sebagai bahasa standar untuk pengembangan *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software* (Sukamto & Shalahuddin, 2014). Jenis-jenis diagram UML (*Unified Modeling Language*) adalah sebagai berikut.

A. *Use Case Diagram*

Use case diagram adalah jenis *diagram* dalam bahasa pemodelan UML yang digunakan untuk menggambarkan interaksi antara aktor (*external user* atau sistem lain) dengan sistem yang sedang dianalisis atau dirancang. *Use case diagram* memberikan gambaran visual tentang bagaimana *user* atau aktor berinteraksi dengan sistem serta fungsi dan fungsionalitas sistem tersebut (Rosenberg & Stephens, 2007). Adapun penjelasan simbol – simbol *use case diagram* dapat dilihat pada Tabel 2.

Tabel 2. Simbol – Simbol *Use Case Diagram* (Rosenberg & Stephens, 2007).

Simbol	Nama	Keterangan
	<i>Aktor</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
	<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor.
	<i>Association</i>	Abstraksi dari penghubung antara aktor dan <i>use case</i> .
	<i>Generalization</i>	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .
	<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
	<i>Exclude</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

B. *Activity Diagram*

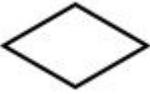
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, aktivitas yang dapat dilakukan oleh sistem (Sukamto & Shalahuddin, 2014). Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari sistem atau *user interface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Simbol-simbol yang digunakan dalam *Activity Diagram* ditunjukkan pada Tabel 3.

Tabel 3. Simbol – Simbol *Activity Diagram* (Sukamto & Shalahuddin, 2014).

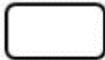
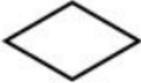
Simbol	Nama	Keterangan
	<i>Initial Node</i>	Sebuah diagram aktivitas memiliki sebuah status awal.
	<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu.
	<i>Join</i>	Asosiasi penggabungan di mana lebih dari satu aktivitas di gabungkan menjadi satu
	<i>Final Node</i>	Status akhir yang dilakukan sistem, sebuah diagram memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

C. ERD (*Entity Relationship Diagram*)

Entity relationship diagram merupakan pendekatan *top-down* terhadap rancangan basis data yang dimulai dengan mengidentifikasi data-data penting yang disebut entitas dan relasi

antara data-data yang akan direpresentasikan ke dalam model. Kemudian menambahkan detail seperti informasi yang diinginkan tentang entitas dan *relationship* yang disebut atribut dan *constraints* yang ada pada *entity*, *relationship*, dan *attribute* (Connolly & Begg, 2015). Adapun penjelasan simbol – simbol *entity relationship diagram* dapat dilihat pada Tabel 4.

Tabel 4. Simbol – Simbol *Entity Relationship Diagram* (Connolly & Begg, 2015).

Simbol	Nama	Keterangan
	Entitas	Kumpulan dari objek yang ada dalam sistem dan memiliki atribut yang mendefinisikan karakteristiknya.
	Relasi	Keterkaitan antara entitas-entitas dalam sistem. <i>Relationship</i> memiliki <i>direction</i> and <i>cardinality</i> yang menggambarkan banyaknya <i>entity</i> yang terlibat dalam hubungan tersebut.
	Atribut	Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

II.3. Teori Pendukung

II.3.1. *Monitoring*

Monitoring didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diimplementasikan (MercyCorps, 2005). Dalam konsep *system monitoring*, *monitoring* adalah proses pengumpulan data dan analisis terhadap penggunaan sumber daya komputer yang terbatas, seperti memori penyimpanan, *central*

processing unit (CPU) , *random access memory (RAM)*, *graphic card*, Virtual RAM, dan berbagai sumber daya komputer lainnya. Tujuan dari proses pemantauan ini adalah untuk mengevaluasi apakah sumber daya komputer masih memadai untuk digunakan atau perlu peningkatan kapasitas (Sulasno & Saleh, 2020).

II.3.2. KPI

Menurut Iveta, (2012), *key performance indicator* adalah ukuran yang bersifat kuantitatif dan bertahap bagi perusahaan serta memiliki berbagai perspektif dan berbasiskan data konkret, dan menjadi titik awal penentuan tujuan dan penyusunan strategi organisasi. Menurut Marr (2016) *key performance indicator* adalah alat navigasi penting yang digunakan oleh para manajer untuk memahami apakah perusahaan mereka sedang mengarah pada kesuksesan atau sedang menjauhi jalur menuju kesuksesan.

Menurut Moeheriono (2012), dalam menyusun *key performance indicator* terdapat beberapa persyaratan indikator kinerja yang harus dipenuhi, yaitu sebagai berikut:

1. *Specific*, yaitu target pencapaian *key performance indicator* harus dirumuskan dengan jelas dan spesifik, sehingga dapat dipahami dengan mudah oleh seluruh anggota organisasi.
2. *Measurable*, yaitu setiap *key performance indicator* (baik ukuran kuantitatif maupun kualitatif) telah ditentukan informasi tentang jenis data-data yang akan digali, sumber data, dan cara mendapatkan data tersebut.
3. *Attitutable*, yaitu setiap *key performance indicator* yang dibuat harus bermanfaat dalam pengambilan keputusan.
4. *Relevant*, yaitu indikator kinerja tersebut harus sesuai dengan ruang lingkup program dan dapat menggambarkan hubungan sebab dan akibat diantara indikator lainnya.

5. *Timely*, yaitu indikator kinerja yang sudah ditetapkan harus dikumpulkan datanya dan dilaporkan tepat pada waktunya.

II.3.3. Framework

Framework adalah suatu himpunan instruksi yang dikelompokkan dalam *class* dan *function* dengan tujuan masing-masing. *Framework* bertujuan untuk mempermudah pengembang dalam memanggilnya tanpa perlu menulis ulang *syntax* program yang sama, sehingga dapat menghemat waktu (Sallaby & Kanedi, 2020).

II.3.4. REST API

Application Programming Interface (API) dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya. Secara umum API merupakan dokumentasi dari fungsi, struktur, kelas, dan lain sebagainya yang dapat diakses oleh yang membutuhkan dengan cara tertentu sesuai yang ditentukan layanan. API memudahkan pengembang lain mengambil data untuk kemudian diintegrasikan dengan perangkat lunak lain (Santoso & Rais, 2015).

Representational State Transfer (REST) adalah arsitektur perangkat lunak yang digunakan untuk pendistribusian sistem layanan *web*, sedangkan REST API adalah aplikasi yang berfungsi untuk melakukan komunikasi dengan layanan *web* menggunakan HTTP untuk melakukan perintah *GET*, *PUT*, *POST*, dan *DELETE* data. Layanan *web* (*web services*) sendiri adalah *server web* yang dibuat khusus untuk mendukung kebutuhan situs atau aplikasi lainnya sehingga bisa secara langsung menerima dan menanggapi permintaan klien menggunakan API yang bisa mengekspos satu set data dan memfasilitasi antar program komputer yang memungkinkan untuk bertukar informasi (Massé, 2011). Teknologi REST lebih banyak dipakai orang karena menggunakan lebih sedikit *bandwidth*, sehingga lebih cocok untuk

digunakan di internet. *Application Program Interface* (API) adalah kode yang bisa menghubungkan dua perangkat lunak untuk bisa saling berkomunikasi dan mengekspos layanan *web*. REST sangat cocok dengan API untuk berkomunikasi dengan layanan cloud, API RESTful sudah digunakan oleh beberapa situs besar seperti Amazon, Google, LinkedIn, dan Twitter.

II.3.5. *Gantt Chart*

Gantt Chart adalah diagram perencanaan yang menerangkan secara visual hubungan antara alokasi penggunaan sumber daya dengan waktu dalam bentuk diagram batang (Heizer & Render, 2012). *Gantt Chart* adalah contoh teknik non matematis yang banyak digunakan dan sangat populer di kalangan para manajer karena sederhana dan mudah dibaca. Adapun contoh *gantt chart* dapat dilihat pada Gambar 1.

Gantt Chart



Gambar 1. *Gantt Chart* (Heizer & Render, 2012).

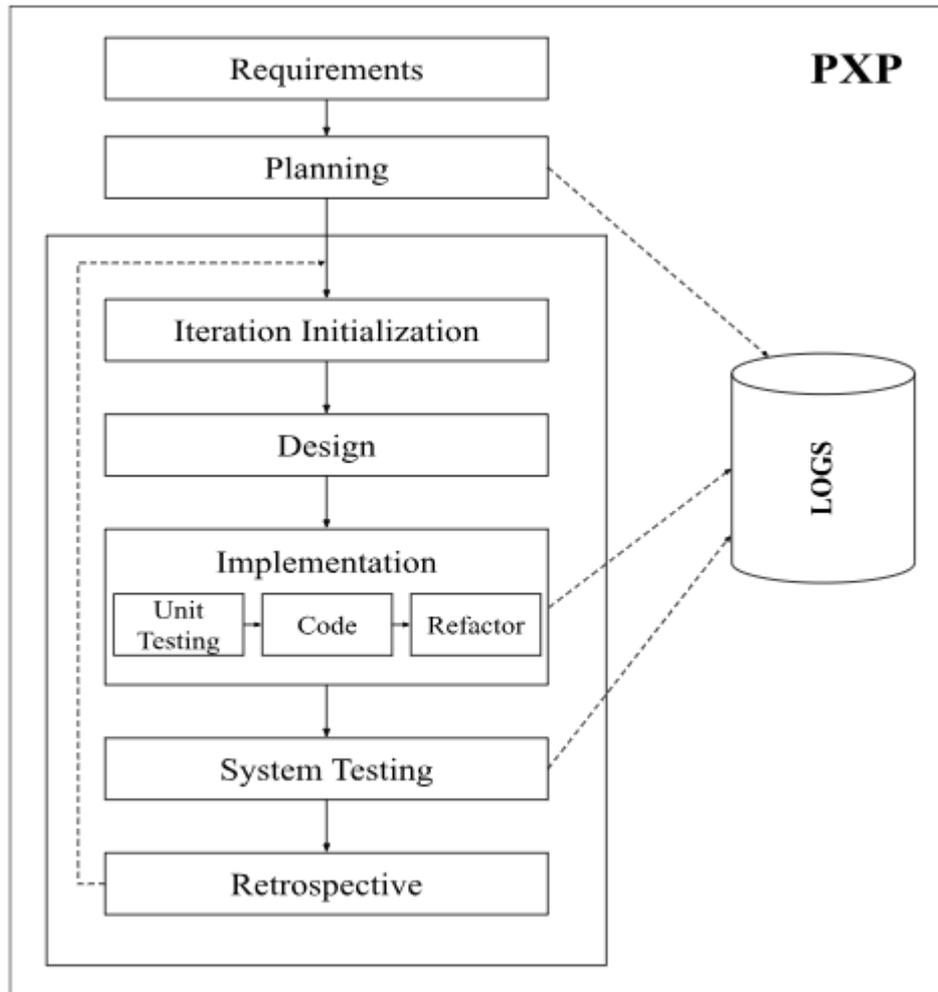
II.4. Metode *Personal Extreme Programming* (PXP)

Menurut panduan yang dikeluarkan oleh Office of The Government Chief Information Officer (2015), metode *Agile* adalah *framework* konseptual dengan pendekatan pengembangan yang *iteratif* dan *increment*. Metode ini membagi *Software Development Life Cycle* (SDLC) menjadi beberapa iterasi *timebox*, dengan kontribusi anggota tim yang sangat penting. *Agile* ditekankan sebagai pendekatan

pengembangan sistem yang fokus pada kecepatan pengiriman dan fleksibilitas untuk perubahan kapan saja.

Agile development memiliki beberapa model pengembangan yakni *Extreme Programming (XP)*, *Relational Unified Proses (RUP)* dan *scrum*. *Relational Unified Process (RUP)* yang memiliki kelebihan menggunakan *Unified Modeling Language (UML)* dengan modul yang telah tersedia untuk setiap anggota tim. Namun, RUP terlalu terfokus pada UML yang ada dan mengakibatkan waktu pengerjaan menjadi lebih lambat. Di sisi lain, keunggulan *Scrum* terletak pada kemampuan tim kecil untuk menyelesaikan pekerjaan lebih cepat karena pembagian tugas kepada semua anggota tim. Meskipun demikian, *Scrum* kurang efektif dalam fase perancangan sistem, yang dapat memperpanjang waktu perancangan (Pressman & Maxim, 2010).

Metode *Extreme Programming (XP)* memiliki kelebihan yakni penggunaannya yang lebih fleksibel. Selain itu, XP memiliki model proses turunan yang cocok untuk situasi *single-person programming* yaitu *Personal Extreme Programming (PXP)*. *Personal Extreme Programming (PXP)* adalah proses pengembangan perangkat lunak yang dirancang untuk diterapkan oleh para insinyur perangkat lunak individual. Pengembangan dalam PXP bersifat iteratif dan memberikan kebebasan *programmer* dalam menangani setiap perubahan *requirement*. Dengan PXP setiap *user story* dibagi ke dalam *task-task* yang lebih kecil. PXP menuntut seorang *programmer* atau *developer* untuk mampu bertanggung jawab atas tugas yang telah dibebankan, mengukur dan menganalisis hasil pekerjaannya (Dzhurov et al., 2009). Berikut adalah tahapan dari PXP yang dapat dilihat pada Gambar 2.



Gambar 2. Metode PXP (Dzhurov et al., 2009).

Metode *Personal Extreme Programming* (PXP) melibatkan tujuh tahap dalam proses pengembangannya:

1. Kebutuhan (*Requirement*) berfungsi untuk menggali informasi kebutuhan dari calon pengguna dengan menggunakan cerita pengguna (*user story*) dan menuliskan kebutuhan berdasarkan pernyataan yang disampaikan oleh pengguna.
2. Perencanaan (*Planning*) bertujuan untuk menganalisis kebutuhan dari *user story* dan mengelompokkannya menjadi satu cerita (*story*). Estimasi waktu pengerjaan tiap *sprint* dilakukan berdasarkan tingkat prioritas, dengan mempertimbangkan data dari *sprint* sebelumnya jika ada.

3. Inisialisasi Iterasi (*Iteration Initialization*) menandakan dimulainya suatu *sprint*. *Sprint* ini dimulai dengan pemilihan *story* berdasarkan nilai prioritas yang telah ditetapkan. Durasi *sprint* dapat berkisar antara satu hingga tiga minggu, bergantung pada kompleksitas *story* yang dikerjakan. Jika terjadi perubahan kebutuhan, kebutuhan tersebut akan dipecah menjadi beberapa *task* dan dimasukkan ke kelompok *task* yang serupa, sambil memperbarui prioritas setiap iterasi.
4. Desain (*Design*) berfokus pada penentuan pola desain yang akan digunakan dalam implementasi program, antarmuka pengguna, dan basis data. Prinsip desain yang sederhana diterapkan di sini, dengan mengacu pada kebutuhan yang telah diajukan oleh pengguna tanpa membuat asumsi terhadap kebutuhan jangka panjang.
5. Implementasi (*Implementation*) melibatkan pengkodean sistem berdasarkan desain yang telah ditetapkan dan pengujian terhadap kode program. Tahap ini terdiri dari tiga tugas utama yaitu pengujian unit, pembuatan kode, dan perbaikan kode (*refactoring*).
6. Uji Sistem (*System Testing*) bertujuan untuk memverifikasi bahwa kode program sesuai dengan kebutuhan yang telah ditetapkan. Setiap kesalahan pada kode atau ketidakcocokan dengan kebutuhan harus didokumentasikan dan diperbaiki. Selesaiannya suatu *sprint* ditandai dengan ketiadaan kesalahan, dan dokumentasi kesalahan menjadi bahan evaluasi pada tahap *Retrospective*.
7. *Retrospective* digunakan untuk menganalisis hasil dari setiap tahap yang telah dilalui. Dilakukan verifikasi terhadap keteraturan waktu pengerjaan dengan rencana awal. Jika terjadi keterlambatan, faktor yang menyebabkannya harus diidentifikasi untuk dijadikan referensi pada *sprint* berikutnya dan proyek mendatang (Dzhurov et al., 2009).

II.5. Aplikasi Pendukung

II.5.1. Laravel

Menurut Awaludin (2014), seorang *Senior Web Developer*, telah menguraikan setiap fungsi dari *framework* Laravel dalam bukunya "Menyelami *Framework* Laravel". Menurutnya, Laravel adalah sebuah *framework* PHP yang pertama kali dikembangkan oleh Taylor Otwell dan dirilis dengan lisensi MIT. *Framework* ini dibangun dengan konsep *Model View Controller* (MVC) dan ditujukan untuk pengembangan *website* berbasis MVP (*Model View Presenter*) dengan penekanan pada peningkatan kualitas perangkat lunak, pengurangan biaya pengembangan awal, dan pemeliharaan yang efisien.

Dalam bukunya, Awaludin juga menyatakan manfaat menggunakan Laravel, termasuk abstraksi yang memungkinkan pengembang lebih fokus pada logika bisnis aplikasi tanpa harus terlalu memikirkan hal-hal dasar di PHP. Awaludin menyoroti tujuan penggunaan Laravel dalam sistem saat ini, yaitu untuk meningkatkan keamanan *website* dan mempercepat proses pekerjaan setiap karyawan. Alasan memilih Laravel juga dikemukakan, di antaranya karena kemampuan *templating* yang lebih mudah, dengan Laravel bekerja di bawah level suatu *library*. Sebagai contoh, jika suatu proyek membutuhkan fungsi mengubah teks menjadi *file* Excel, Laravel telah menyediakan berbagai macam *library* sesuai kebutuhan sistem, menghilangkan kebutuhan untuk membuat *script* fungsi secara berulang-ulang seperti yang mungkin diperlukan dalam PHP *native*.

II.5.2. Bootstrap

Menurut Enterprise (2016), *Bootstrap* adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun mendesain halaman *website*. Sesuai namanya *website* yang dibuat dengan alat bantu ini memiliki tampilan halaman yang sama / mirip dengan tampilan

halaman Twitter atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan.

II.5.3. JSON Web Token (JWT)

JSON Web Token (JWT) adalah standar terbuka (RFC 7519) yang mendefinisikan cara untuk mentransmisikan informasi secara aman antara dua pihak sebagai objek JSON. Informasi ini dapat diverifikasi dan dipercaya karena ditandatangani secara digital (Jones et al., 2015). JWT digunakan secara luas dalam *authentication* and *authorization*. Berikut adalah beberapa kegunaan utama dari JWT:

1. *Authentication*, JWT sering digunakan dalam autentikasi untuk memastikan bahwa pengguna telah terverifikasi. Setelah pengguna berhasil *login* dengan kredensialnya, server mengeluarkan JWT yang menyimpan informasi tentang pengguna tersebut. Token ini kemudian dikirim bersama dengan setiap permintaan yang membutuhkan otentikasi (Jones et al., 2015).
2. *Authorization*, JWT dapat digunakan untuk mengakses rute, layanan, dan sumber daya yang diizinkan. Server yang menerima token dapat memverifikasi tanda tangan dan memeriksa klaim untuk memutuskan apakah pengguna memiliki izin yang diperlukan (Jones et al., 2015).
3. *Information exchange*, JWT adalah cara aman untuk mentransmisikan informasi antar pihak karena token ditandatangani, penerima dapat memastikan bahwa pengirim adalah siapa yang dia klaim. Selain itu, sifat tidak dapat diubah dari token memastikan bahwa informasi di dalamnya tetap utuh (Jones et al., 2015).

II.5.4. Postman

Postman adalah *platform* kolaborasi untuk pengembangan API. Dibuat oleh Abhinav Asthana, seorang *programmer* dan *designer* yang berbasis di Bangalore, India, Postman memudahkan dalam menguji, mengembangkan, dan mendokumentasikan API. Fitur Postman yang sederhana membuat pengujian API dapat dilakukan dengan baik dan

cepat. Cara kerja Postman dengan mengklasifikasi *request* berdasarkan *request method*, URL dan parameter *request* (Postman, 2024).

II.5.5. OWASP ZAP

Zed Attack Proxy (ZAP) adalah alat uji penetrasi *open-source* gratis yang dikelola oleh *Open Web Application Security Project* (OWASP). ZAP dirancang khusus untuk menguji aplikasi *web* dan memiliki fleksibilitas serta kemampuan untuk diperluas. Pada dasarnya, ZAP berfungsi sebagai "*proxy man-in-the-middle*" yang bertindak sebagai perantara antara browser pengujian dan aplikasi *web* untuk mencegat, memeriksa, dan memodifikasi pesan yang dikirim antara keduanya sebelum meneruskannya ke tujuan akhir. Alat ini dapat digunakan sebagai aplikasi mandiri atau sebagai proses daemon. OWASP ZAP digunakan untuk pengujian penetrasi guna menemukan kerentanan pada aplikasi *web* dan menyediakan pemindai otomatis (OWASP ZAP, 2024).

II.5.6. JMeter

Aplikasi JMeter adalah alat pengujian kinerja yang sangat fleksibel, memungkinkan simulasi banyak pengguna aplikasi secara bersamaan untuk mengevaluasi kinerja aplikasi (The Apache Software Foundation, 2014). Pengujian dengan jumlah pengguna yang dapat diatur sesuai kebutuhan ini sangat bermanfaat bagi organisasi karena dapat merefleksikan kondisi yang sebenarnya (Wang & Wu, 2019). Dalam pengujiannya, alat seperti JMeter dirancang untuk mengukur kinerja baik pada level antarmuka pengguna maupun sistem itu sendiri. Menggunakan alat seperti ini dapat membantu organisasi menghemat waktu dan biaya (Bhalla et al., 2017).

II.5.7. Visual Studio Code

Visual Studio Code adalah *editor source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, GIT Control yang disematkan, penyorotan *sintaks*, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga

dapat disesuaikan, sehingga pengguna dapat mengubah tema *editor*, *shortcut keyboard*, dan preferensi. Visual Studio Code gratis dan *open-source*, meskipun unduhan resmi berada di bawah lisensi *proprietary*. Visual Studio Code didasarkan pada Elektron, kerangka kerja yang digunakan untuk menyebarkan aplikasi Node.js untuk *desktop* yang berjalan pada *Blink layout*. Meskipun menggunakan kerangka Elektron, Visual Studio Code tidak menggunakan Atom dan menggunakan komponen *editor* yang sama (diberi kode nama "Monaco") yang digunakan dalam *Visual Studio Team Services* yang sebelumnya disebut *Visual Studio Online* (Lardinois, 2015).

II.5.8. Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada event "Google I/O Conference" untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android (Android Developer, 2024).

II.5.9. XAMPP

XAMPP merupakan singkatan dari X (sistem operasi pada komputer), A (*Apache*), M (MySQL), P (PHP), P (*Perl*). XAMPP adalah *software* yang bersifat *open source* dan mendukung dari beberapa sistem operasi dan gabungan dari beberapa program. Program yang terkandung dalam XAMPP mendukung dari beberapa bahasa pemrograman seperti HTML, Javascript, CSS, PHP, SQL, dan lain-lain. Dalam XAMPP, sudah terkandung *Apache*, yaitu *localhost* atau *web server* yang dapat digunakan dalam proses pembuatan *website*. Dalam pelaksanaannya, penggunaan *localhost* dan *database* pada XAMPP perlu diaktifkan dahulu *Apache* dan MySQL pada *software*-nya lalu mengakses <https://localhost> dan <https://localhost/phpmyadmin> pada *web browser* (Novendri et al., 2019).

II.6. System Testing

Pengujian merupakan elemen dari pengembangan perangkat lunak yang disebut dengan *verification and validation testing* (V&V). Verifikasi mengacu pada serangkaian kegiatan yang memastikan perangkat lunak dapat menjalankan fungsi yang telah ditentukan, sedangkan validasi mengacu pada satu set aktivitas yang memastikan bahwa perangkat lunak yang dikembangkan sudah sesuai dengan kebutuhan pengguna (Pressman & Maxim, 2014).

II.6.1. Security Testing

Security testing dilakukan untuk menegaskan bahwa sistem telah dilengkapi dengan mekanisme keamanan yang efektif, sehingga dapat mencegah akses yang tidak sah atau penetrasi yang tidak diinginkan. Selain memeriksa perlindungan umum, pengujian keamanan juga mengevaluasi mekanisme *API Key* dan *authentication* dalam memastikan bahwa hanya pengguna yang sah yang memiliki akses ke sistem. Dengan menguji token API, sistem dapat memverifikasi bahwa permintaan akses berasal dari sumber yang sah, sementara pengujian autentikasi memastikan bahwa hanya pengguna yang terotorisasi yang dapat mengakses sumber daya sistem (Pressman & Maxim, 2014).

II.6.2. Performance Testing

Performance testing sering digabungkan dengan pengujian tegangan dan biasanya membutuhkan instrumentasi *hardware* dan *software*. Dengan menginstruksikan suatu sistem, *tester* dapat mengungkap situasi yang menyebabkan degradasi dan kemungkinan kegagalan sistem (Pressman & Maxim, 2014).

II.6.3. Metode Black Box Testing

Metode *black box testing* yaitu pendekatan pengujian perangkat lunak yang melihat perangkat lunak sebagai suatu entitas yang berfungsi, tetapi tidak memperhatikan bagaimana entitas tersebut diimplementasikan. Pengujian dilakukan berdasarkan spesifikasi

eksternal perangkat lunak dan tidak memeriksa struktur atau logika internal (Pressman & Maxim, 2014).

II.6.4. *User Acceptance Testing (UAT)*

User Acceptance Testing (UAT) merupakan pengujian yang dilakukan oleh *end-user*, yang umumnya adalah *staff* karyawan perusahaan yang langsung berinteraksi dengan sistem, melakukan evaluasi terhadap kinerja sistem untuk memastikan bahwa fungsi-fungsi yang disediakan sesuai dengan kebutuhan dan tujuan yang diinginkan (Perry, 2006).

Setelah tahap *system testing*, *acceptance testing* menyimpulkan bahwa sistem memenuhi persyaratan yang telah ditetapkan. *Acceptance testing* adalah proses pengujian yang dilakukan oleh pengguna akhir dengan menggunakan metode *black box testing* untuk mengevaluasi sistem sesuai dengan spesifikasi yang telah ditetapkan. Pengguna akhir memiliki tanggung jawab untuk memastikan bahwa semua fungsionalitas yang relevan telah diuji.

User Acceptance Testing (UAT) merupakan tahap akhir dalam proses pengujian perangkat lunak. Selama UAT, perangkat lunak diuji untuk memverifikasi apakah fungsi-fungsi telah sesuai dengan spesifikasi yang telah ditetapkan. UAT adalah salah satu langkah terakhir dan sangat penting dalam siklus proyek perangkat lunak sebelum perangkat lunak tersebut dirilis ke pasar. UAT juga sering disebut sebagai pengujian *beta*, pengujian aplikasi, atau pengujian pengguna akhir (Cimperman, 2006).

III.2. Alat Penelitian

Penelitian ini menggunakan dua jenis alat, yaitu perangkat lunak dan perangkat keras. Berikut merupakan spesifikasi alat yang digunakan selama penelitian.

III.2.1. Perangkat Lunak

Dalam penelitian ini perangkat lunak yang digunakan adalah:

1. Sistem Operasi Windows 11 Home 64-bit.
2. PHP versi 7.4.33.
3. *Framework* Laravel versi 7.30.6.
4. JSON Web Token (JWT) versi 3.3.3.
5. *Library*: maatwebsite/excel versi 3.1 untuk membaca *file* Excel.
6. Aplikasi:
 - a. Visual Studio Code versi 1.85.1.
 - b. Android Studio Dolphin versi 2021.3.1 Patch 1.
 - c. Postman versi 10.15.2.
 - d. XAMPP versi 3.3.0.
 - e. *Browser* Google Chrome versi 121.0.6167.85.

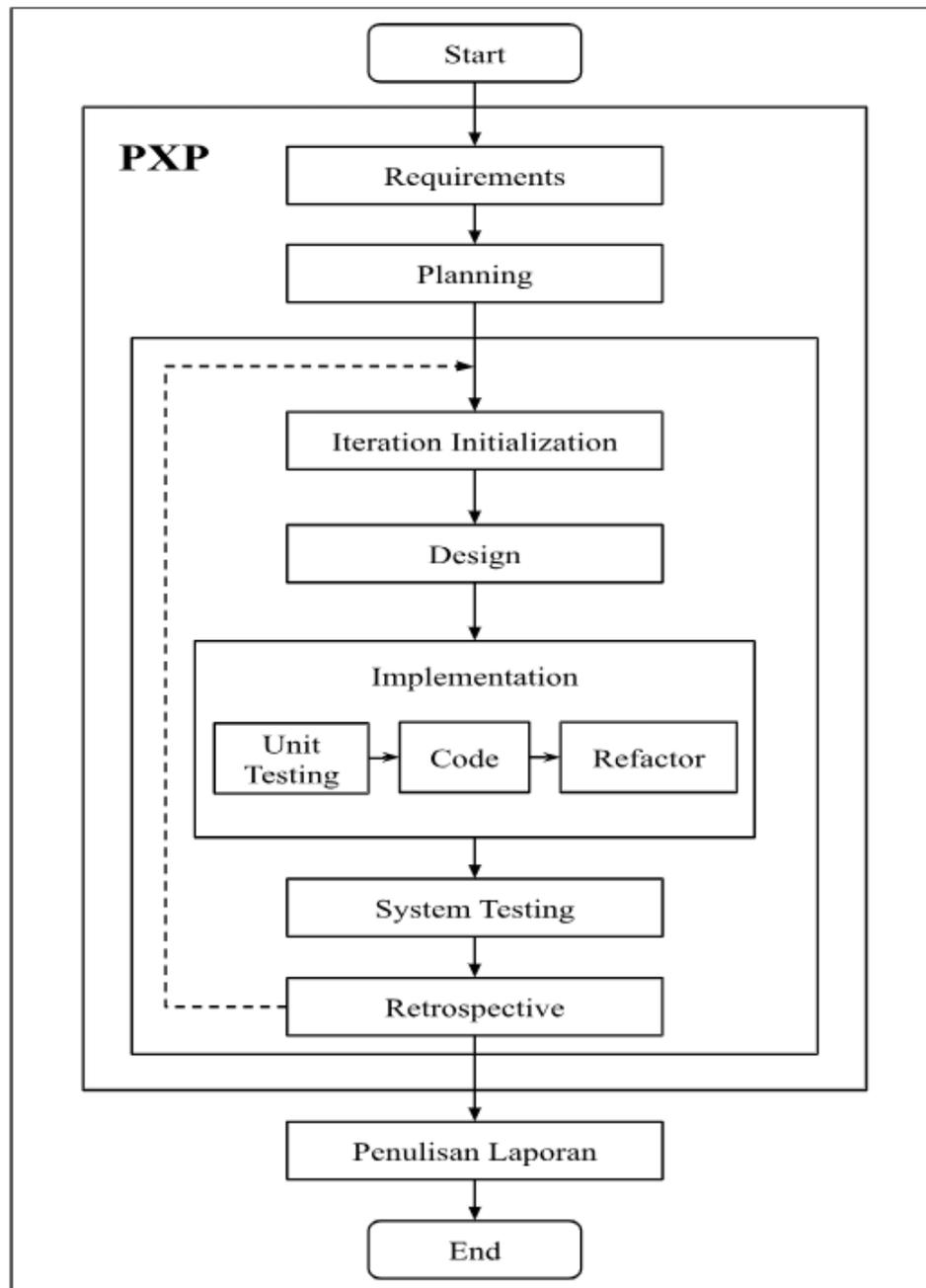
III.2.2. Perangkat Keras

Dalam penelitian ini perangkat keras yang digunakan adalah sebuah laptop dan *smartphone* dengan spesifikasi:

1. Laptop
 - a. *Processor* 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz (16 CPUs), ~2.3GHz.
 - b. Intel(R) UHD *Graphics* dan NVIDIA GeForce RTX 3050 TI Laptop GPU.
 - c. Memori 16 GB RAM.
 - d. Penyimpanan SSD 512GB dan HDD 1 TB.
2. *Smartphone*
 - a. CPU Octa-core Max 2.2 GHz.
 - b. Android versi 13.0 (Minimum versi 7.0).
 - c. Memori 8 GB RAM.
 - d. Penyimpanan 128 GB.

III.3. Tahapan Penelitian

Penelitian ini memanfaatkan pendekatan metodologi PXP (*Personal Extreme Programming*) yang terstruktur dengan serangkaian tahapan, dimulai dari *Requirement*, *Planning*, *Iteration Initialization*, *Design*, *Implementation*, *System Testing*, hingga *Retrospective* sebagaimana diilustrasikan pada Gambar 3.



Gambar 3. Tahapan Penelitian Menggunakan Metode PXP.

III.3.1. Requirement

Pada tahap ini terdapat beberapa metode pengumpulan data yang digunakan dalam pengembangan sistem *monitoring* di PT Industri Kereta Api (Persero), yaitu:

1. Wawancara

Metode wawancara dilakukan dengan cara menanyakan secara langsung kepada beberapa pegawai instansi yang sedang bertugas terkait permasalahan yang ada, perkiraan kebutuhan sistem, serta alur proses bisnis yang sedang berjalan pada PT Industri Kereta Api (Persero). Daftar pertanyaan wawancara dapat dilihat pada Tabel 6.

Tabel 6. Tabel Wawancara.

Pertanyaan	Jawaban
Bagaimana alur dari pelaporan dan <i>monitoring</i> program kerja saat ini?	Proses pelaporan dan <i>monitoring</i> program kerja saat ini dimulai dengan pengumpulan data kinerja dari berbagai departemen atau unit terkait. Setelah data terkumpul, langkah berikutnya melibatkan susunan dan analisis manual untuk menghasilkan informasi yang relevan. Pada setiap awal bulan, rapat bulanan diadakan di mana setiap tim melaporkan kemajuan program kerja mereka. Hasil dari rapat ini, bersama dengan data lainnya, dikumpulkan dan dianalisis oleh staf pengolahan data sebelum disajikan dalam rapat evaluasi manajemen. Proses ini bertujuan untuk memberikan laporan yang komprehensif kepada pihak yang berwenang guna memantau dan mengevaluasi pelaksanaan program kerja secara efektif.

Tabel 6. (lanjutan).

Pertanyaan	Jawaban
Apa saja kendala yang ditemukan ketika melakukan pelaporan program kerja?	Salah satu kendala utamanya adalah ketidakjelasan dalam format pelaporan. Setiap tim sering menggunakan format berbeda, sehingga mempersulit proses analisis yang efisien.
Jika membuat sistem informasi apakah nantinya dapat mempermudah proses pelaporan dan pemantauan program kerja?	Dengan pengembangan sistem informasi terintegrasi, setiap pegawai memiliki kemampuan untuk mengotomatisasi sebagian besar proses pelaporan. Melalui sistem ini, data dapat diakses secara langsung oleh tim analisis dan manajemen, menghasilkan efisiensi waktu dan mengurangi risiko kesalahan. Sistem informasi terintegrasi memungkinkan akses data secara real-time dan menyajikan laporan yang lebih akurat. Keuntungan ini tidak hanya mempermudah pengambilan keputusan, tetapi juga meningkatkan pemantauan kemajuan program kerja secara keseluruhan. Dengan adopsi teknologi ini, organisasi dapat merespons lebih cepat terhadap perubahan dan meningkatkan produktivitas serta akurasi dalam proses pelaporan dan evaluasi.

Setelah melakukan wawancara kepada beberapa pegawai, selanjutnya dilakukan pengumpulan kebutuhan dengan menganalisis kebutuhan sistem. Analisis kebutuhan sistem merupakan proses identifikasi, dokumentasi, dan validasi kebutuhan bisnis yang harus dipenuhi oleh suatu sistem informasi. Tujuan utama analisis kebutuhan sistem adalah memastikan bahwa sistem yang dikembangkan akan memenuhi kebutuhan pengguna dan

organisasi dengan tepat. Analisis kebutuhan fungsional dan non-fungsional dari sistem *monitoring* KPI meliputi:

a. Kebutuhan Fungsional

Kebutuhan fungsional adalah proses atau layanan apa saja yang nantinya disediakan oleh sistem, berikut merupakan kebutuhan fungsional pada Tabel 7.

Tabel 7. Kebutuhan Fungsional Sistem.

<i>Role</i>	Kebutuhan Fungsional
<i>Super Admin</i>	Melakukan <i>login</i> .
	<i>Monitoring</i> program kerja dan proyek.
	Mengelola data akun pegawai.
	Mengelola program kerja.
	Mengelola proyek.
<i>Admin</i>	Mengelola realisasi.
	Melakukan <i>login</i> .
	<i>Monitoring</i> program kerja dan proyek.
	Mengelola program kerja.
	Mengelola proyek.
<i>Direksi</i>	Mengelola realisasi.
	Melakukan <i>approve</i> dan <i>reject</i> laporan realisasi.
	Melakukan <i>login</i> .
<i>Pegawai</i>	<i>Monitoring</i> program kerja dan proyek.
	Mengelola program kerja.
	Mengelola proyek.
	Melakukan <i>login</i> .

b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah batasan layanan yang ditawarkan oleh sistem, berikut merupakan kebutuhan non-fungsional pada Tabel 8.

Tabel 8. Kebutuhan Non-Fungsional Sistem.

Kebutuhan Non-Fungsional	Deskripsi
<i>Availability</i>	Sistem dijalankan menggunakan <i>local server</i> perusahaan sehingga akses ke dalam sistem hanya dapat dilakukan oleh pihak internal perusahaan.
<i>Portability</i>	Sistem ini dirancang untuk memberikan fleksibilitas dan dapat beroperasi pada berbagai sistem operasi yang mendukung penggunaan <i>web browser</i> .
<i>Security</i>	Pada sistem ini setiap aktor memiliki peran (<i>role</i>) tertentu dengan proses autentikasi <i>login</i> yang memvalidasi identitas pengguna sebelum memberikan akses ke fungsionalitas sistem.
<i>Build</i>	Sistem dibangun menggunakan <i>framework</i> Laravel dengan bahasa pemrograman PHP dan Javascript.

2. Observasi

Penelitian ini menggunakan metode observasi dengan mengamati secara langsung jalannya operasional bisnis di instansi tersebut, fokus pada proses pelaporan program kerja yang dilakukan oleh para pegawai. Proses tersebut masih mengandalkan penggunaan perangkat lunak *spreadsheet*, khususnya *Excel*, sebagai alat utama dalam pelaporan program kerja, sehingga dapat ditarik beberapa kesimpulan mengenai permasalahan yang ada dan melakukan diskusi bersama mentor mengenai pengembangan sistem yang sekiranya bisa dikembangkan berdasarkan permasalahan yang sudah dapat disimpulkan sebelumnya.

III.3.2. *Planning*

Pada tahap ini menentukan daftar tugas dengan estimasi waktu pengerjaannya berdasarkan *requirements* yang telah didapatkan. Adapun daftar tugas pada pengembangan sistem *monitoring* KPI dapat dilihat pada Tabel 9.

Tabel 9. Tabel *Planning* Pengembangan Sistem.

No	Perencanaan Tugas	Deskripsi	Perkiraan Waktu
1	<i>Backend Server</i>	Membuat autentikasi <i>user</i> dengan menggunakan REST API pada operasi <i>Create, Read, Update, Delete</i> (CRUD).	4 minggu
2	Halaman Super Admin	Membuat halaman <i>login, dashboard</i> , kelola data pegawai, kelola program kerja dan proyek.	1 minggu
3	Halaman Admin	Membuat halaman <i>login, dashboard</i> , kelola program kerja dan proyek, membuat <i>approve</i> dan <i>reject</i> pelaporan realisasi.	1 minggu
4	Halaman Direksi	Membuat halaman <i>login, dashboard</i> , kelola program kerja dan proyek.	1 minggu
5	Halaman Pegawai	Membuat halaman <i>login, dashboard</i> , kelola program kerja dan proyek.	1 minggu

III.3.3. *Iteration Initialization*

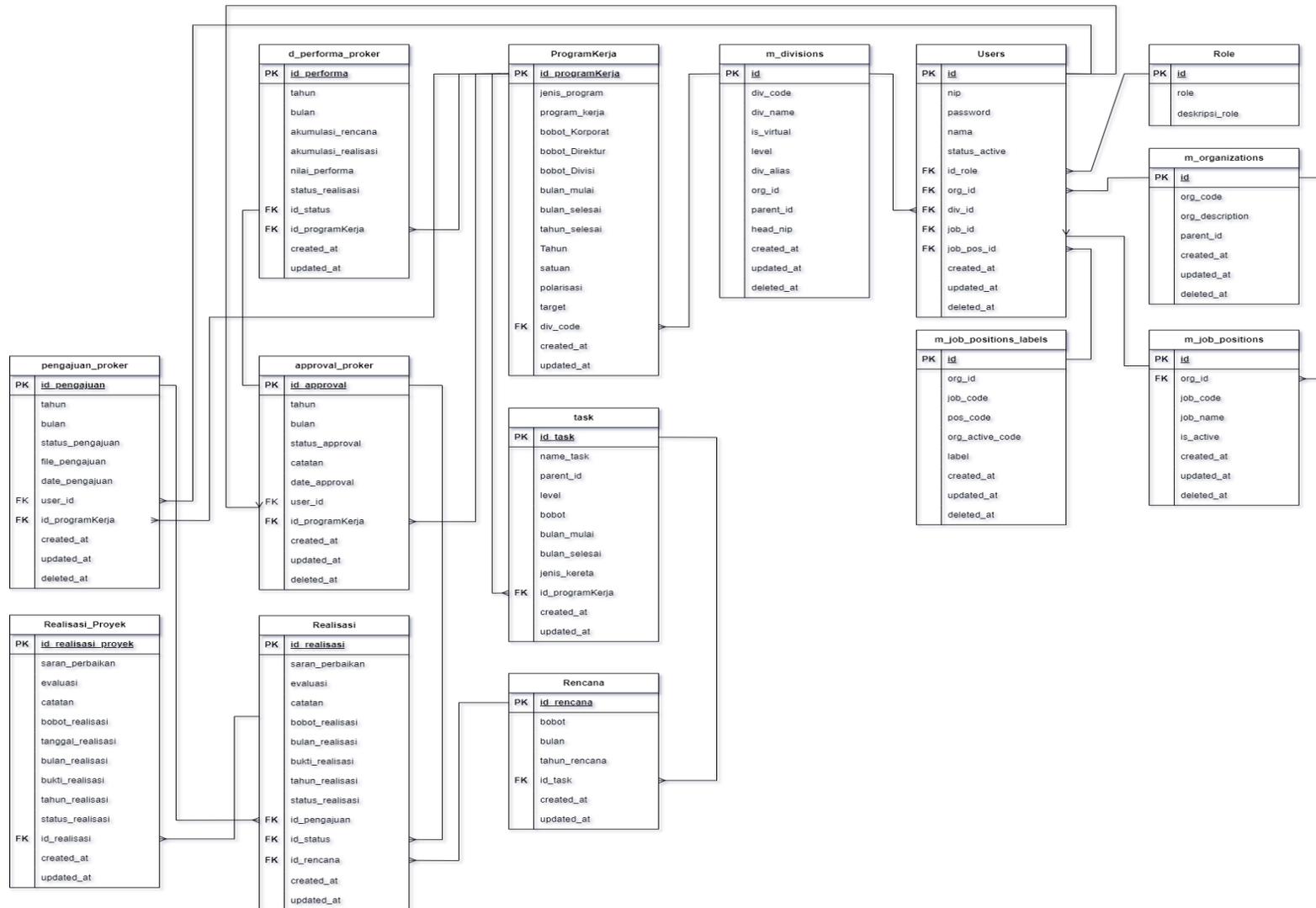
Pada tahap ini merupakan tahap awal dari setiap iterasi dengan melakukan pemilihan fokus tugas dan estimasi waktu yang ditentukan pada iterasi tersebut. Iterasi dilakukan sebanyak 2 kali dengan durasi waktu yang bervariasi. Pada iterasi awal, dilakukan pembangunan *backend server* dengan fokus pada REST API CRUD data dan *authentication*. Sementara pada iterasi berikutnya, dilakukan implementasikan kode program terhadap desain UI yang telah dibuat dan integrasi API pada halaman *super admin*, admin, direksi, dan pegawai.

III.3.4. *Design*

Pada tahap ini melakukan perancangan *diagram* ERD, UML, dan perancangan antarmuka dalam Sistem *Monitoring KPI* menggunakan *tools* Figma untuk membuat tampilan *prototype* sistem yang diimplementasikan ke dalam sistem. Rancangan antarmuka ini dibuat berdasarkan pada diagram UML dan ERD yang telah dibuat. Berikut perancangan ERD dan diagram UML yang terdiri dari *use case diagram* dan *activity diagram*:

1. *Entity Relationship Diagram (ERD)*

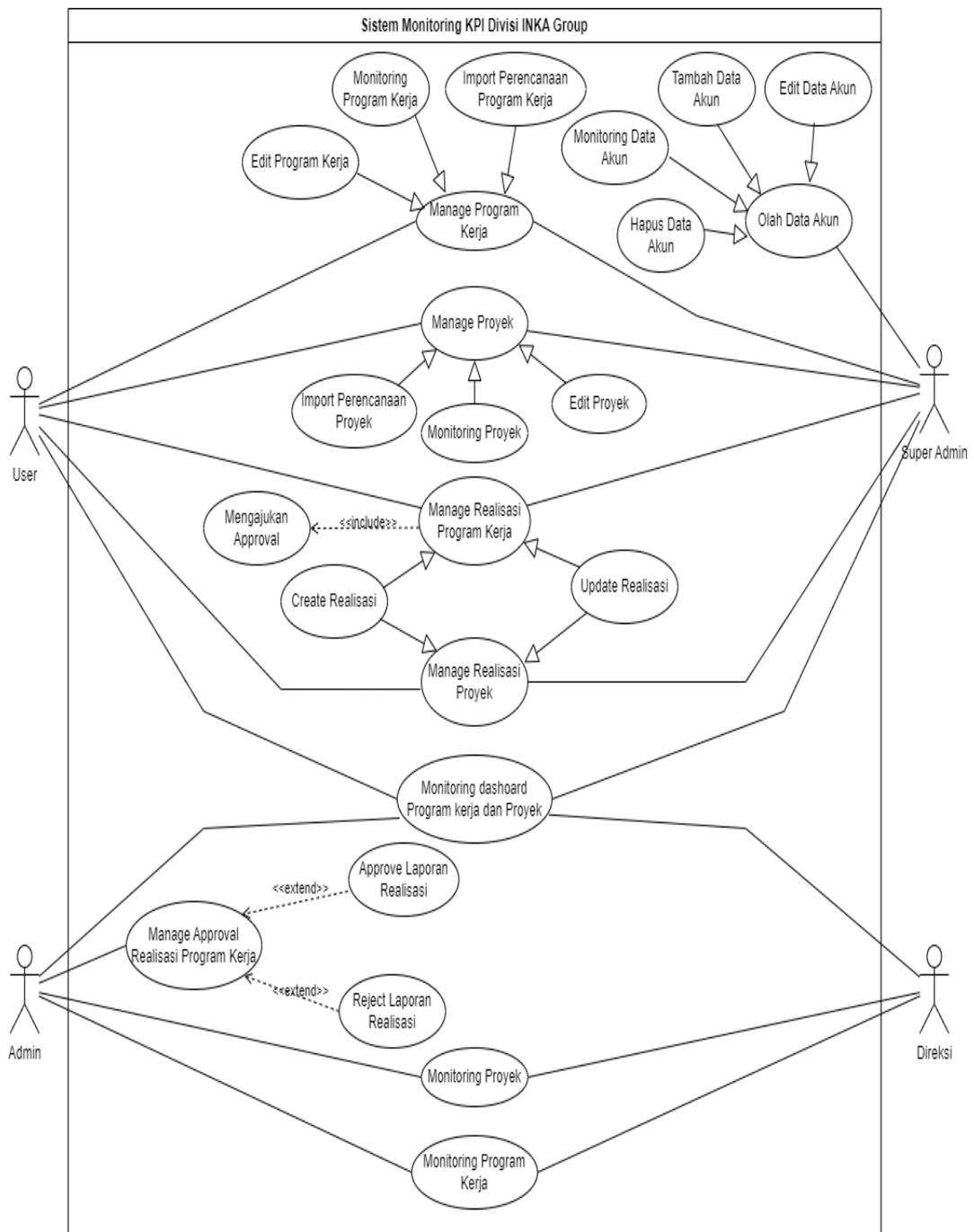
Diagram ini digunakan untuk memodelkan tabel-tabel apa saja beserta data apa saja yang dibutuhkan dalam pengembangan sistem. *Entity Relationship Diagram (ERD)* dapat dilihat pada Gambar 4.



Gambar 4. Entity Relationship Diagram (ERD) Sistem Monitoring KPI.

2. Use Case Diagram

Penelitian ini menggunakan diagram ini untuk memodelkan fitur-fitur yang dibutuhkan oleh pihak instansi pada rancangan pengembangan sistem. *Use Case Diagram* dapat dilihat pada Gambar 5.

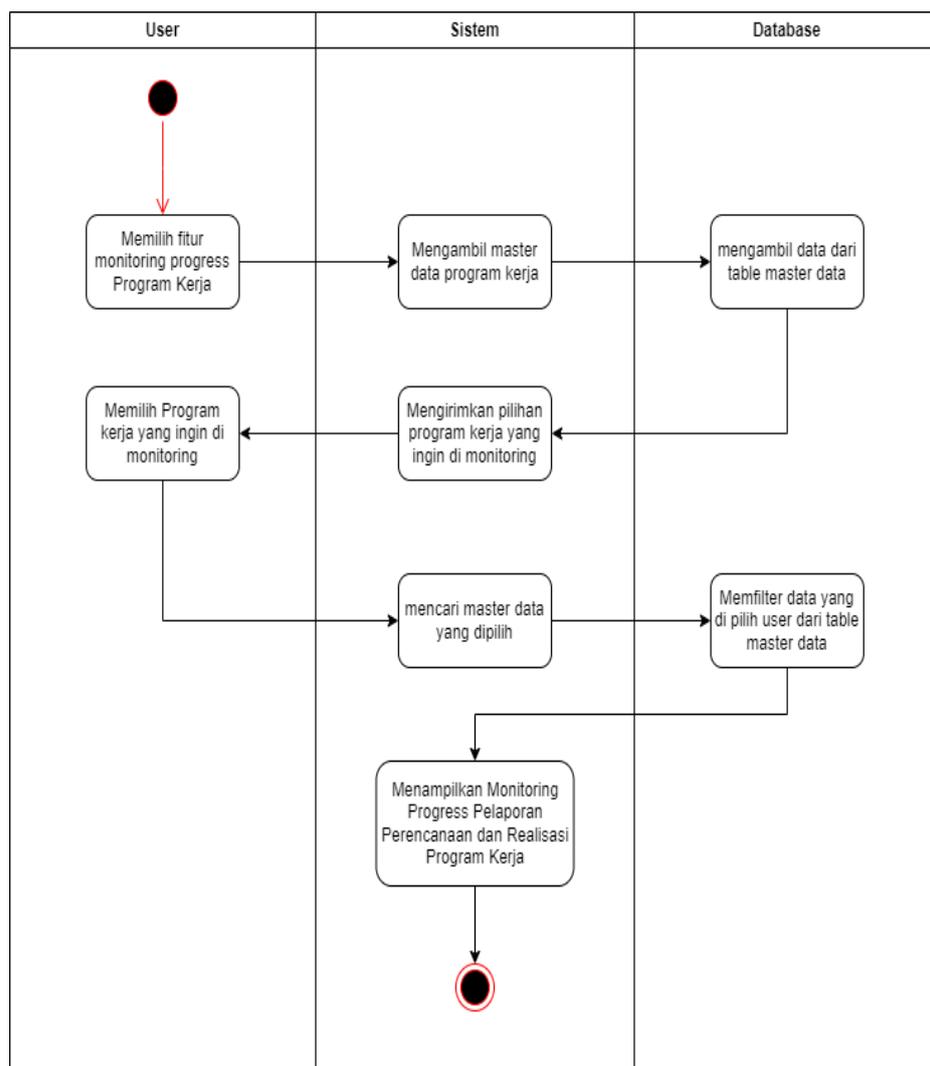


Gambar 5. Use Case Diagram Sistem Monitoring KPI.

3. Activity Diagram

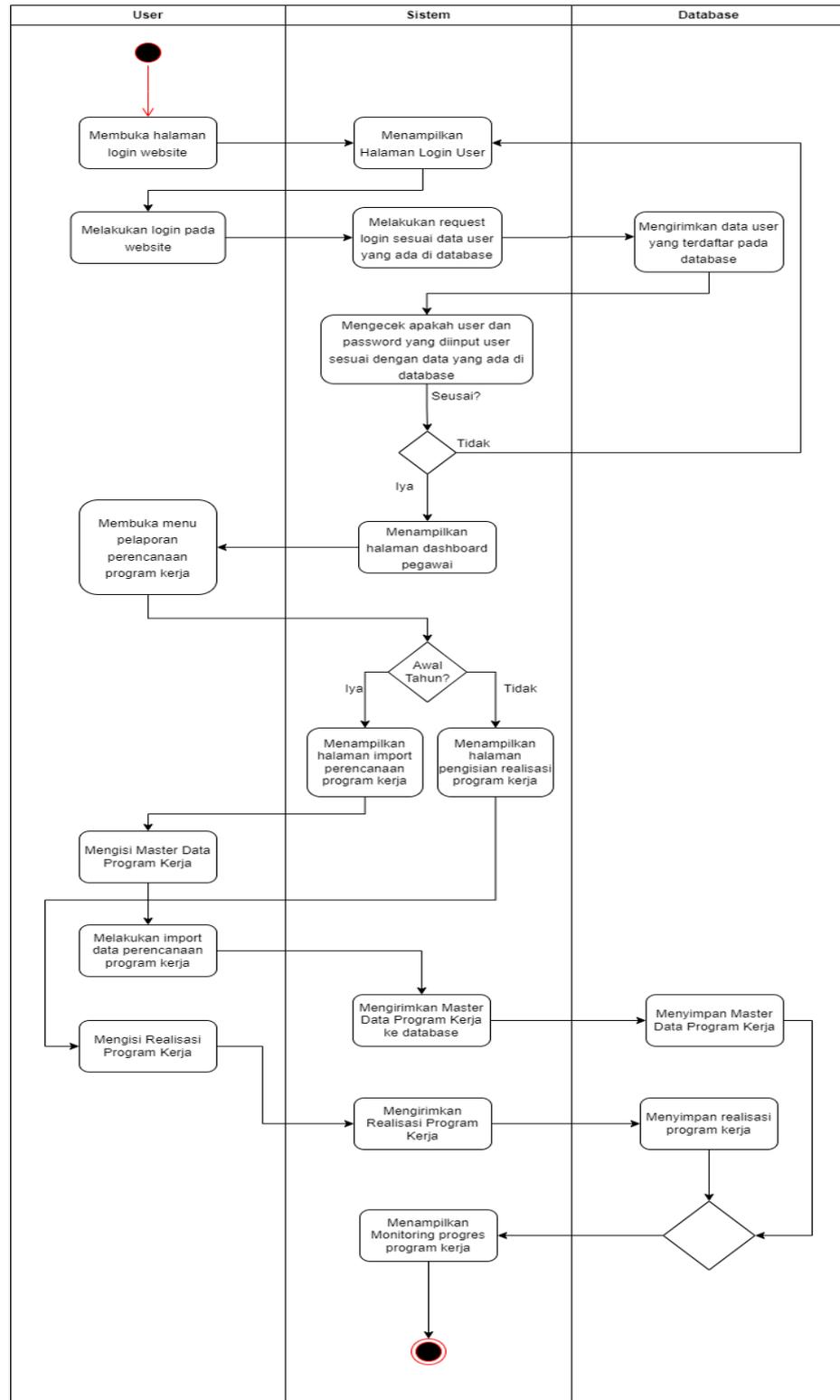
Penelitian ini menggunakan *activity diagram* untuk menggambarkan alur kerja dan proses yang terjadi dalam sistem *monitoring KPI*. Gambar 6 sampai dengan Gambar 10 menampilkan *Activity Diagram* yang mencakup langkah-langkah input perencanaan, input pelaporan realisasi, *monitoring progress*, *approve* dan *reject* pelaporan realisasi sebagai berikut.

a. Activity Diagram Monitoring Progress



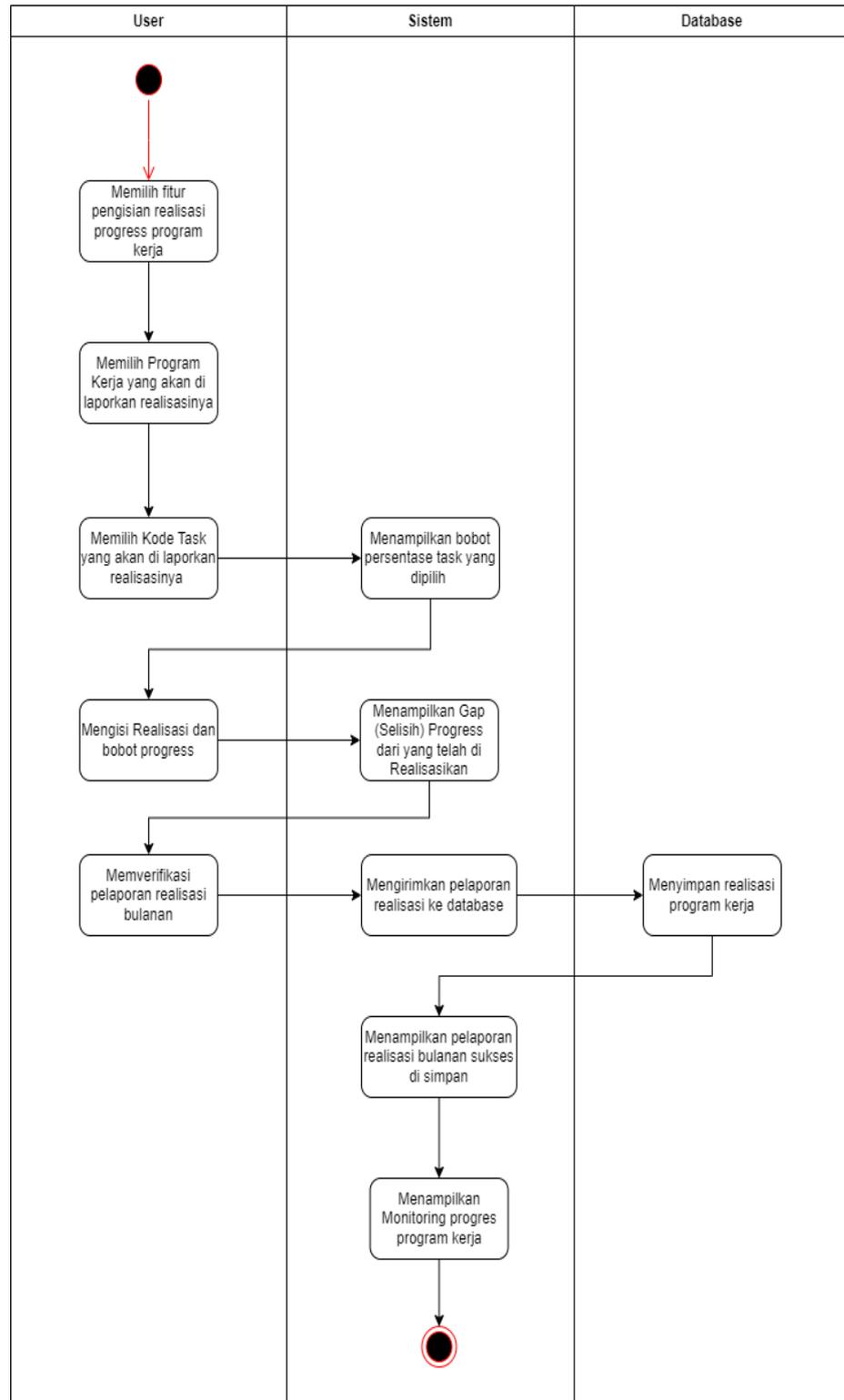
Gambar 6. Activity Diagram Monitoring Progress Sistem Monitoring KPI.

b. Activity Diagram Input Perencanaan Program Kerja



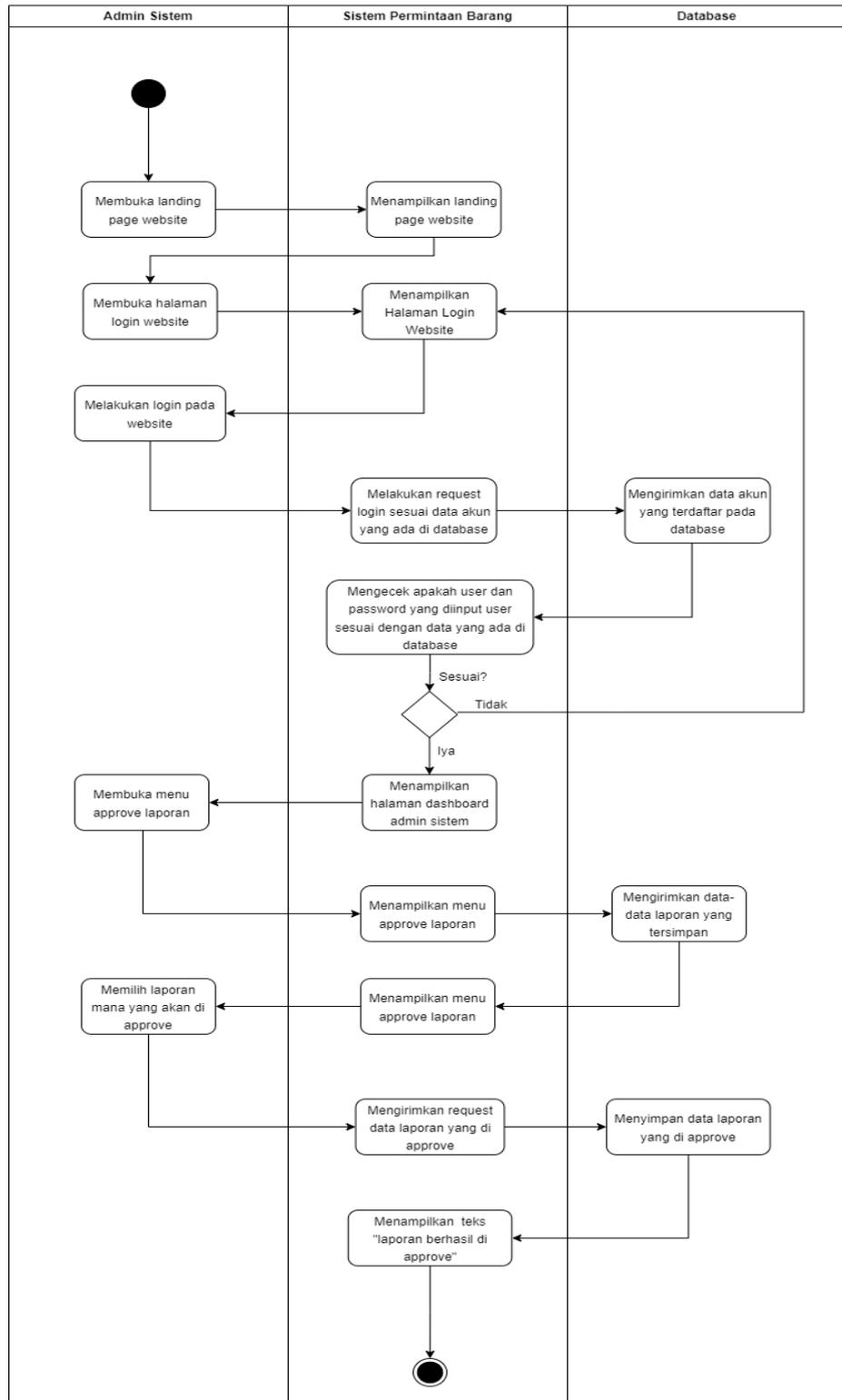
Gambar 7. Activity Diagram Input Perencanaan Program Kerja Sistem Monitoring KPI.

c. *Activity Diagram Input Pelaporan Realisasi Program Kerja*



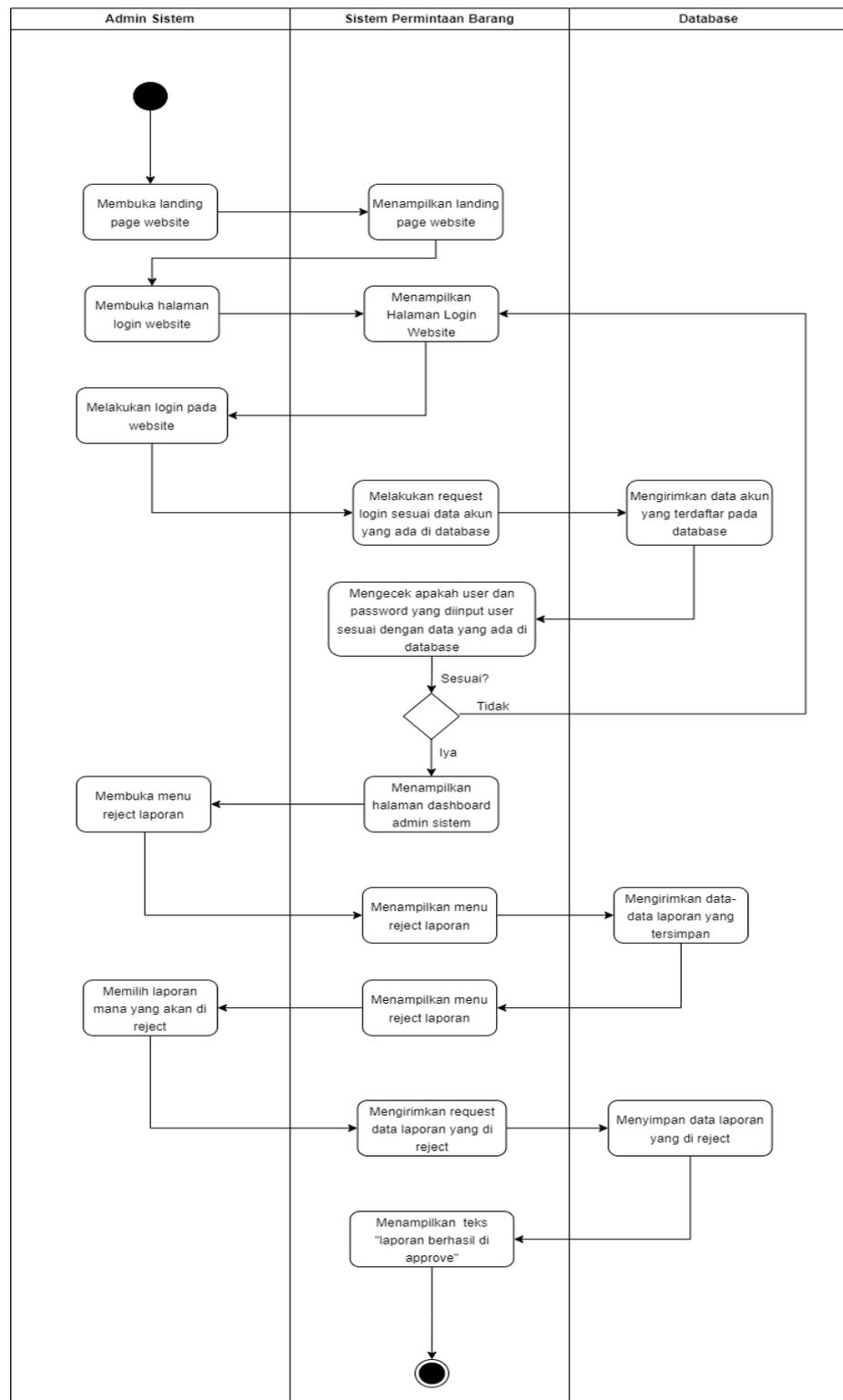
Gambar 8. *Activity Diagram Input Pelaporan Realisasi Program Kerja Sistem Monitoring KPI.*

d. Activity Diagram Approve Pelaporan Realisasi Program Kerja



Gambar 9. Activity Diagram Approve Pelaporan Realisasi Program Kerja Sistem Monitoring KPI.

e. *Activity Diagram* Reject Pelaporan Realisasi Program Kerja



Gambar 10. *Activity Diagram* Reject Pelaporan Realisasi Program Kerja Sistem *Monitoring* KPI.

III.3.5. *Implementation*

Tahapan ini terdiri dari *unit testing*, *code*, dan *refactoring*. Pada tahap pertama yaitu dilakukan *unit testing* untuk memastikan setiap komponen beroperasi dengan baik dan sesuai dengan standar yang telah ditetapkan. Penulisan kode program menggunakan *framework* Laravel, serta integrasi keseluruhan API sesuai dengan rancangan sistem yang telah disusun. Proses *refactoring* kemudian dilakukan untuk membersihkan dan meningkatkan kualitas keseluruhan kode program. Selama proses *implementation*, jika terjadi perubahan pada sistem, Gitlab digunakan sebagai repositori untuk menyimpan setiap perubahan pada *source code*. Setiap kali ada kode program yang di-*push* ke Gitlab, hal ini memudahkan identifikasi *bug* atau perbaikan yang mungkin dibutuhkan.

III.3.6. *System Testing*

Pada tahap pengujian sistem ini, terdapat 4 hal yang dilakukan:

1. *Security Testing*

Security testing dilakukan untuk memastikan bahwa sistem memiliki tingkat keamanan yang memadai terhadap potensi ancaman keamanan. Dalam penelitian ini, *security testing* menggunakan alat Postman dan OWASP ZAP (*Zed Attack Proxy*). OWASP ZAP membantu dalam mengidentifikasi berbagai risiko keamanan. Hasil pengujian dengan OWASP ZAP mengidentifikasi risiko keamanan dan memberikan alert berdasarkan tingkat keparahan diantaranya:

- a. *Informational*, memberikan informasi tambahan yang mungkin berguna tetapi tidak berbahaya.
- b. *Low*, risiko rendah yang sebaiknya diperbaiki tetapi tidak mendesak.
- c. *Medium*, risiko sedang yang perlu diperbaiki untuk menghindari potensi eksploitasi.
- d. *High*, risiko tinggi yang harus segera diperbaiki untuk mencegah serangan serius (OWASP ZAP, 2023)..

2. *Performance Testing*

Performance testing dilakukan untuk mengevaluasi kinerja sistem dalam hal responsivitas, kecepatan, dan kinerja secara umum. Alat yang digunakan untuk *performance testing* dalam penelitian ini adalah *Apache JMeter*. *JMeter* digunakan untuk mensimulasikan beban pengguna pada sistem dan mengukur bagaimana sistem merespons di bawah kondisi beban yang berbeda. Pengujian ini mencakup pengukuran waktu *response*, *throughput*, dan penggunaan sumber daya sistem, memastikan bahwa sistem mampu menangani beban tinggi dengan tetap menjaga kinerja yang optimal.

3. *Black-box testing* dilakukan dalam dua tahapan. Pada tahapan pertama, sistem diuji langsung pada *website* untuk memastikan fitur sesuai dengan analisis kebutuhan. Pada tahapan kedua, sistem diuji menggunakan *tools* Postman untuk memeriksa kesesuaian API dengan *request* dan *response* data yang diharapkan. *Black-box testing* dilakukan tanpa pengetahuan internal tentang desain dan implementasi sistem, sehingga lebih menekankan pada fungsionalitas sistem dari luar.

4. *User acceptance testing*, pengujian ini dilakukan untuk memastikan bahwa sistem yang dikembangkan telah memenuhi persyaratan dan harapan pengguna akhir dari perusahaan. Keterlibatan pengguna akhir, yaitu pegawai di PT Industri Kereta Api, sangat penting dalam tahap ini untuk mengevaluasi sistem dan memastikan bahwa sistem berfungsi sesuai dengan kebutuhan bisnis dan pengguna di perusahaan tersebut. UAT dilaksanakan dengan menyiapkan terlebih dahulu kuesioner dengan pertanyaan terkait sistem yang sudah dikembangkan dengan lima pilihan jawaban dari setiap pertanyaan, yaitu sangat setuju, setuju, cukup, tidak setuju, dan sangat tidak setuju.

Penggunaan *Skor Likert* memungkinkan pengukuran yang lebih terstruktur dan mudah dianalisis, sehingga memberikan gambaran yang lebih akurat tentang kepuasan pengguna terhadap sistem yang diuji, seperti yang dapat dilihat pada Tabel 10.

Tabel 10. *Skor Likert*.

Jawaban	Skor
Sangat Tidak Setuju (STS).	1
Tidak Setuju (ST).	2
Cukup (C).	3
Setuju (S).	4
Sangat Setuju (SS).	5

Setelah mendapatkan total skor, selanjutnya dicari indeks kepuasan dengan menggunakan Persamaan 1 untuk mengukur sejauh mana persyaratan sistem telah diuji dan diverifikasi (Riyadi, 2019).

$$\text{Indeks (\%)} = \left(\frac{\text{Total Skor}}{\text{Total Skor Maksimum}} \right) \times 100\% \quad (1)$$

Setelah didapatkan hasil dari indeks kepuasan data akan dicocokkan dengan interval indeks yang dapat dilihat pada Tabel 11 (Pranatawijaya et al., 2019).

Tabel 11. Indeks Kepuasan Pengguna.

Jawaban	Skor
0% - 19.99%	Sangat Tidak Memuaskan
20% - 39.99%	Tidak Memuaskan
40% - 59.99%	Cukup
60% - 79.99%	Memuaskan
80% - 100%	Sangat Memuaskan

Dengan demikian, UAT tidak hanya memastikan bahwa sistem berfungsi dengan baik, tetapi juga memastikan bahwa pengguna akhir merasa puas dengan kinerja dan fungsionalitas sistem yang telah dikembangkan.

III.3.7. *Retrospective*

Pada tahap akhir dari setiap iterasi, dilakukan introspeksi terhadap pengerjaan iterasi yang telah berlangsung. Hal ini mencakup evaluasi kesesuaian estimasi waktu serta identifikasi penyebab keterlambatan pengerjaan sistem. Tujuan dari introspeksi ini adalah untuk meningkatkan proses iterasi sehingga dapat berjalan lebih efektif pada iterasi berikutnya.

V. SIMPULAN DAN SARAN

V.1. Simpulan

Berdasarkan pembahasan sebelumnya mengenai pengembangan Rest API pada Sistem *Monitoring* KPI menggunakan JWT dan *framework* Laravel, serta pengembangan aplikasi *mobile* di PT Industri Kereta Api (Persero), dapat disimpulkan bahwa:

1. Penelitian ini berhasil mengembangkan sistem *monitoring* KPI yang terintegrasi dengan baik antara *backend server* API dan aplikasi *mobile* yang menyediakan layanan yang efisien dan dapat diandalkan.
2. Penelitian ini berhasil mengimplementasikan dan menguji semua *endpoint* API dengan baik. Penggunaan JWT untuk autentikasi memberikan keamanan tambahan dalam komunikasi antara aplikasi *mobile* dan *server backend*.
3. Desain UI/UX yang baik dan implementasi yang efektif memastikan pengalaman pengguna yang optimal. Pengguna dapat dengan mudah mengakses dan menggunakan aplikasi untuk *monitoring* KPI.
4. Pengujian yang dilakukan mencakup *security testing*, *performance testing*, *black box testing*, dan *user acceptance testing* (UAT). Hasil pengujian menunjukkan bahwa aplikasi ini aman, dapat diandalkan, dan mampu menangani beban tinggi. Selain itu, hasil dari User Acceptance Testing (UAT) menunjukkan bahwa tingkat kepuasan pengguna sangat tinggi, dengan persentase 82.08%.

V.2. Saran

Berdasarkan pembahasan mengenai implementasi sistem yang telah dibahas, terdapat beberapa hal yang mungkin dapat ditingkatkan antara lain sebagai berikut:

1. Optimasi *Endpoint* API, melakukan optimasi lebih lanjut pada *endpoint* yang menunjukkan waktu respons lebih tinggi, seperti *POST Login*, untuk memastikan aplikasi dapat menangani beban yang lebih tinggi di masa mendatang.
2. Pengujian Berkelanjutan, melakukan pengujian secara berkelanjutan dengan skenario pengguna yang lebih kompleks dapat membantu dalam mengidentifikasi dan memperbaiki potensi masalah sebelum aplikasi digunakan oleh pengguna akhir.
3. Pengembangan Fitur Lanjutan, menambahkan fitur-fitur baru yang dapat meningkatkan pengalaman pengguna dan meningkatkan fungsionalitas aplikasi, seperti notifikasi *real-time* dan analisis data yang lebih mendalam.

DAFTAR PUSTAKA

- Android Developer. (2024). *Android Studio*. <http://developer.android.com/sdk/>.
- Anhar. (2010). *Panduan Menguasai PHP Dan MySQL Secara Otodidak*. Media Kita.
- Anjelina, Y. (2023). *Rancang Bangun Sistem Peminjaman Coworking Space Berbasis Website Menggunakan Mern Stack Technology Pada Gedung Lembaga Kemahasiswaan Fisip Universitas Lampung*.
- Awaludin, R. (2014). *Menyelami Framework Laravel 5.2*.
- Bakhtiar, A., Hartanto, A., & Suliantoro, H. (2016). Perbandingan Metode-Metode Pengukuran Kinerja. *Prosiding Seminar Nasional Manajemen Teknologi XXV*.
- Bhalla, S., Bhateja, V., Chandavale, A. A., Suresh, A. S. H., & Satapathy, C. (2017). *Intelligent Computing and Information and Communication: Proceedings of 2nd International Conference, ICICC 2017*. <http://www.springer.com/series/11156>.
- Bose, S., Kundu, A., Mukherjee, M., & Banerjee, M. (2018). A Comparative Study: Java vs Kotlin Programming In Android Application Development. *International Journal of Advanced Research in Computer Science*, 9(3). <https://doi.org/10.26483/ijarcs.v9i3.5978>.
- Cimperman, Rob. (2006). *UAT Defined : A Guide to Practical User Acceptance Testing*. Pearson Education.
- Connolly, T. M. ., & Begg, C. E. . (2015). *Database systems : a practical approach to design, implementation and management* (Sixth Edition). Pearson Education Limited.
- Dzhurov, Y., Krasteva, I., & Ilieva, S. (2009). *Personal Extreme Programming-An Agile Process for Autonomous Developers*.
- Enterprise, J. (2016). *Pemrograman Bootstrap Untuk Pemula*. PT Elex Media Komputindo.

- Evanandy, A. (2022). *Implementasi Personal Extreme Programming Dalam Pengembangan Integrasi Sistem Informasi Akademik dan E-Learning*.
- Filiana, A., Rini, M. N. A., Prabawati, A. G., & Samat, R. A. (2022). *Pengembangan Rest Api Untuk Informasi Pasar Tradisional di Kota Yogyakarta dengan Metode Incremental*. <https://doi.org/10.31598>.
- Heizer, J. H., & Render, B. (2012). *Operations Management* (Tenth). Prentice Hall.
- Hidayatullah, P., & Kawistara, J. K. (2015). *Pemrograman Web* (Second). Informatika Bandung.
- Iveta, G. (2012). Human Resources Key Performance Indicators. *Journal of Competitiveness*, 4(1), 117–128. <https://doi.org/10.7441/joc.2012.01.09>.
- Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)*. RFC 7519. <http://www.rfc-editor.org/info/rfc7519>.
- Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi* (Revisi Edition). Andi Yogyakarta. <https://doi.org/10.13140/2.1.2637.6328>.
- Kurniawan, I. B. (2020). Rancang Bangun Sistem Informasi Penilaian Kinerja Karyawan PT INKA(Persero) Menggunakan Metode Extreme Programming. *Jurnal Sistem Informasi dan Teknologi*. <http://www.jurnal.umk.ac.id/sitech>.
- Kusumanto, I., Permata, E. G., Harpito, Anwardi, & Iglina, P. (2018). *Penilaian Kinerja Menggunakan Metode Key Performance Indicators Pada Bunda Bakery*.
- Lardinois, F. (2015). *Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows*.
- Lewenusa, I. (2020). *Dasar Penggunaan CSS pada Pengembangan Web*.
- Listiyah, A. (2022). *Uji Usability Pada Institutional Repository Perpustakaan Uin Maulana Malik Ibrahim Malang dengan Metode System Usability Scale (SUS) dan Discovery Prototyping*.
- Marr, B. (2016). *Key Performance Indicators yang Harus Diketahui*. Jakarta: Elex Media Komputindo.
- Massé, M. (2011). *REST API design rulebook: designing consistent RESTful web service interfaces*. O'Reilly Media, Inc.
- MercyCorps. (2005). *Design, Monitoring, and Evaluation Guidebook*.
- Moheriono. (2012). *Pengukuran kinerja berbasis kompetensi* (Revisi Edition). Raja Grafindo Persada.

- Novendri, S. M., Saputra, A., & Firman, C. E. (2019). Aplikasi Inventaris Barang Pada Mts Nurul Islam Dumai Menggunakan PHP dan MySQL. *Jurnal Manajemen Dan Teknologi Informasi*, 10(2).
- Nugroho, B. (2008). *Panduan Lengkap Menguasai Perintah SQL*. Mediakita.
- Office of The Government Chief Information Officer. (2015). *Practice Guide For Agile Software Development*. The Government of The Hong Kong Special Administrative Region.
- OWASP ZAP. (2024). *OWASP ZAP Documentation*. <https://www.zaproxy.org/docs/>.
- Perry, W. E. (2006). *Effective Methods for Software Testing Third Edition*.
- Postman. (2024). *Postman Documentation*. <https://www.getPostman.com>.
- Pranata, B. A. (2018). *Perancangan Application Programming Interface (API) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (REST) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit*.
- Pranatawijaya, V. H., Widiatry, Priskila, R., & Putra, P. B. A. A. (2019). Penerapan Skala Likert dan Skala Dikotomi Pada Kuesioner Online. *Jurnal Sains Dan Informatika*, 5(2), 128–137. <https://doi.org/10.34128/jsi.v5i2.185>.
- Pressman, R. S., & Maxim, B. R. (2010). *Software engineering: A practitioner's approach, Seven Edition*. New York: The McGraw–Hill Companies.
- Pressman, R. S., & Maxim, B. R. (2014). *Software engineering: A practitioner's approach, Eight Edition*. New York: The McGraw–Hill Companies.
- Rahman, S. (2013). *Cara gampang bikin cms php tanpa ngoding*. Mediakita.
- Riyadi, N. R. (2019). *Pengujian Usability Untuk Meningkatkan Antarmuka Aplikasi Mobile myUMM Students*.
- Rosenberg, Doug., & Stephens, Matt. (2007). *Use case driven object modeling with UML : theory and practice*. Apress.
- Rosyidah, M., & Saputra, D. (2019). *Penentuan Key Performance Indicators (KPI) Dengan Metode Performance Prism (Studi Kasus Program Studi Teknik Industri Universitas Muhammadiyah Palembang)*.
- Sallaby, A. F., & Kanedi, I. (2020). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter. *Jurnal Media Infotama*.