

**PENERAPAN *REQUIREMENT CHANGE MANAGEMENT PROCESS* DAN
FUNCTION POINT ANALYSIS PADA *BACKEND* SISTEM MANAJEMEN
PROYEK (STUDI KASUS GINK TECHNOLOGY)**

SKRIPSI

Oleh

AHMAD RAFALY

2215061011



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

**PENERAPAN *REQUIREMENT CHANGE MANAGEMENT PROCESS* DAN
FUNCTION POINT ANALYSIS PADA *BACKEND* SISTEM MANAJEMEN
PROYEK (STUDI KASUS GINK TECHNOLOGY)**

Oleh

AHMAD RAFALY

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2026

ABSTRAK

PENERAPAN *REQUIREMENT CHANGE MANAGEMENT PROCESS* DAN *FUNCTION POINT ANALYSIS* PADA *BACKEND* SISTEM MANAJEMEN PROYEK (STUDI KASUS GINK TECHNOLOGY)

Oleh

Ahmad Rafaly

Perubahan kebutuhan perangkat lunak yang tidak dikelola dengan baik seringkali menyebabkan keterlambatan dan pembengkakan biaya proyek. Untuk mengatasi masalah tersebut di Gink Technology, penelitian ini mengembangkan *backend* Sistem Manajemen Proyek berbasis Laravel dengan menerapkan metode *Requirement Change Management* (RCM) dan *Function Point Analysis* (FPA). Pendekatan RCM digunakan untuk mendokumentasikan dan mengendalikan usulan perubahan fungsionalitas, sementara FPA dimanfaatkan untuk mengukur kompleksitas serta memberikan estimasi waktu dan biaya di awal proyek. Sistem dikembangkan menggunakan metode Scrum dengan 7 kali *sprint* dan dievaluasi melalui *Black Box Testing* serta *Unit Testing*. Hasil pengujian fungsional menunjukkan bahwa seluruh fungsi *backend* berjalan sempurna tanpa kesalahan (100% *Passed*). Selain itu, validasi algoritma pada dua proyek riil (Sistem X dan Y) membuktikan bahwa implementasi FPA pada sistem mampu menghasilkan tingkat akurasi estimasi rata-rata sebesar 89% untuk biaya dan 84% untuk waktu. Kesimpulannya, sistem ini terbukti handal dalam menangani perubahan kebutuhan secara terstruktur dan sangat membantu dalam menyusun perencanaan proyek yang lebih realistis serta terukur.

Kata Kunci : Sistem Manajemen Proyek, *Requirement change management*, *Function Point Analysis*, *Software Cost Estimation*.

ABSTRACT

APPLICATION REQUIREMENT CHANGE MANAGEMENT PROCESS AND FUNCTION POINT ANALYSIS ON THE BACKEND PROJECT MANAGEMENT SYSTEM (CASE STUDY OF GINK TECHNOLOGY)

By

Ahmad Rafaly

Poorly managed software requirements changes often lead to project delays and cost overruns. To address these issues at Gink Technology, this study developed a Laravel-based Project Management System backend by implementing the Requirement Change Management (RCM) and Function Point Analysis (FPA) methods. The RCM approach was used to document and control proposed changes, while FPA was utilized to measure complexity and provide time and cost estimates at the beginning of the project. The system was developed using the Scrum method with 7 times sprint processes and evaluated through Black Box Testing and Unit Testing. The functional testing results showed that all backend functions ran perfectly without errors (100% Passed). In addition, estimation validation on two real projects (System X and Y) proved that the implementation of FPA in the system was able to produce an average estimation accuracy rate of 89% for cost and 84% for time.

Keywords: *Sistem Manajemen Proyek, Requirement change management, Function Point Analysis, Software Cost Estimation.*

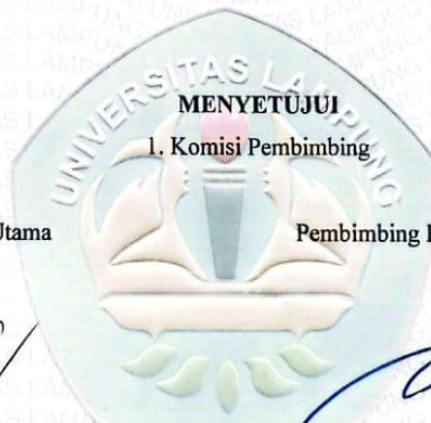
Judul Skripsi : **PENERAPAN *REQUIREMENT CHANGE MANAGEMENT PROCESS* DAN *FUNCTION POINT ANALYSIS* PADA *BACKEND* SISTEM MANAJEMEN PROYEK (STUDI KASUS GINK TECHNOLOGY)**

Nama Mahasiswa : **Ahmad Rafaly**

Nomor Pokok Mahasiswa : 2215061011

Program Studi : S1 Teknik Informatika


Fakultas : Teknik



1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping



Wahyu Eko Sulistiono, S.T., M.Sc.
NIP. 197412012001121001


Ir. Resty Annisa, S.S.T., M.Kom.
NIP. 199008302019032019

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika


Herlinawati, S.T., M.T.
NIP 197103141999032001


Yessi Mulyani, S.T., M.T.
NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua : Wahyu Eko Sulistiono, S.T., M.Sc.

Sekretaris : Ir. Resty Annisa, S.S.T., M.Kom.

Penguji : Ir. Gigih Forda Nama, S.T., M.T.I.



2. Dekan Fakultas Teknik



Dr. Ahmad Herison, S.T., M.T.

NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : 03 Februari 2026

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan dengan sebenarnya bahwa skripsi saya yang berjudul "*Penerapan Requirement Change Management Process dan Function Point Analysis pada Backend Sistem Manajemen Proyek (Studi Kasus Gink Technology)*" merupakan hasil karya saya sendiri, berdasarkan ide dan inovasi saya. Apabila pernyataan ini tidak benar dan terbukti bahwa skripsi ini merupakan salinan, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 6 April 2026

Pembuat pernyataan,



Ahmad Rafaly

NPM 2215061011

RIWAYAT HIDUP



Penulis lahir di Jakarta, tanggal 21 April 2004 dari orang tua Deden Supriatna Pramana dan Nining Fatwana. Penulis telah menyelesaikan pendidikan formal di SDN 1 Nagrak pada tahun 2016, SMPN 29 Bandar Lampung pada tahun 2019, SMAN 15 Bandar Lampung Peminatan IPA pada tahun 2022. Pada tahun 2022, penulis diterima sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa, penulis aktif dalam berbagai kegiatan. Penulis aktif terlibat sebagai penerima beasiswa Karya Salemba Empat dari tahun 2023 hingga 2026 dan beasiswa Tunas Bangsa. Selain kegiatan kampus, penulis juga aktif mengikuti kegiatan Volunteer yaitu Senyum Anak Nusantara selama 2 tahun sebagai anggota Divisi Pendidikan di tahun 2023 dan Ketua PSDA di tahun 2024. Pengembangan diri penulis tidak hanya ada dalam lingkup kampus, pada 2024 penulis ikut dalam program Magang dan Studi Independen Bersertifikat Kampus Merdeka di Kominfo Pesawaran dengan fokus bidang Website Development sebagai Front End Developer. Selain itu penulis juga berpengalaman sebagai Software Engineer di DJPB Kementerian Keuangan Lampung yang membantu instansi untuk mengembangkan website DJPB agar dapat bermanfaat bagi khalayak ramai. Penulis aktif mengikuti berbagai pelatihan, meliputi sertifikasi UI/UX Development dari camp404, Web Developer dari Dicoding Indonesia, serta pelatihan dari Microsoft mengenai cybersecurity. Penulis juga aktif dalam mengikuti kompetisi yang salah satunya adalah turut serta dalam ajang Technology For Indonesia yang diadakan oleh KSE dan berhasil lolos hingga pendanaan. Selain itu, penulis juga menerapkan ilmu yang didapat selama perkuliahan pada lingkup masyarakat dengan ikut serta dalam pelaksanaan Kuliah Kerja Nyata (KKN) di Desa Sidomulyo, Kecamatan Bukit Kemuning, Lampung Utara.

MOTTO

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan”

Q.S Al-Insyirah : 5

“Life is like riding a bicycle. To keep your balance, you must keep moving”

Albert Einstein

“Jika kamu tidak dapat menahan pahitnya belajar, maka kamu akan menanggung perihnya kebodohan”

Imam Syafi'i

“Kalo mimpi kamu ga bikin kamu takut, berarti mimpinya gak cukup gede”

Najwa Shihab

“Tidak Mencoba sudah pasti kalah. Menang, sudah pasti mencoba.”

Ahmad Rafaly

PERSEMBAHAN

Alhamdulillahillobbilamin

Segala Puji bagi Allah Subhanahu wa ta'ala yang telah memberikan rahmat dan kesehatan kepada penulis sehingga skripsi ini bisa diselesaikan dengan tanpa hambatan. Shalawat serta salam juga senantiasa penulis haturkan kepada Nabi Muhammad Shalallahu alaihi wa salam.

Kupersembahkan Skripsi ini kepada

Kedua orang tua, Deden S Pramana & Nining Fatwana yang telah memberikan segala keringat dan upayanya agar penulis bisa menyelesaikan pendidikan dengan baik tanpa terkendala apapun. Semoga Allah senantiasa memberikan kesehatan dan kelimpahan rezekiNya, Aamiin.

Seluruh rekan – rekan seperjuangan Teknik Informatika Universitas Lampung, Departemen Ninja, Paguyuban Beasiswa Karya Salemba Empat, Kementerian Komunikasi dan Digital, Kementerian Keuangan DJPB Lampung serta Beasiswa Tunas Honda atas bantuan, motivasi, dan ilmu yang telah diberikan kepada penulis.

Seluruh dosen dan civitas akademik Jurusan Teknik Elektro Universitas Lampung yang telah membimbing dengan ilmu, kesabaran, dan dedikasi tinggi.

SANWACANA

Puji Syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa, berkat rahmat dan kesehatan yang diberikan kepada penulis sehingga skripsi ini dapat diselesaikan. Skripsi dengan judul “Penerapan *Requirement Change Management Process* dan *Function Point Analysis* pada *Backend* Sistem Manajemen Proyek (Studi Kasus Gink Technology)” yang merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik di Universitas Lampung.

Dalam Kesempatan ini, penulis mengucapkan terima kasih kepada :

1. Tuhan Yang Maha Esa atas segala karunia, rahmat serta hidayah-Nya yang diberikan.
2. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Ibu Yessi Mulyani, S.T., M.T., selaku Ketua Program Studi Teknik Informatika Universitas Lampung.
4. Bapak Wahyu Eko Sulistiono, S.T., M.Sc., selaku dosen pembimbing utama yang senantiasa bersedia membimbing dengan memberikan kritik dan saran membangun selama proses pengerjaan skripsi.
5. Ibu Ir. Resty Annisa, S.S.T., M.T., selaku dosen pembimbing pendamping yang telah memberikan arahan dan saran selama proses pengerjaan skripsi.
6. Bapak Ir. Gigih Forda Nama, S.T., M.T.I. selaku dosen penguji yang telah membantu penulis menyempurnakan skripsi.
7. Bapak Raden Arum Setia P, S.Si., M.T. selaku dosen pembimbing akademik
8. Kedua orang tua dan keluarga yang telah mendukung secara moral dan material.

9. Nisya Ananda Shaliha (2214131024) dan keluarga yang telah menjadi bagian dari perjalanan penulis dari SMA hingga skripsi ini ditulis, atas dukungan semangat dan motivasi yang diberikan.
10. Beasiswa Karya Salemba Empat yang telah memberikan dukungan material kepada penulis selama masa perkuliahan.
11. Grup Departemen Ninja & TI C yang telah menjadi bagian penulis dari semester satu hingga selesai.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan dan kekeliruan. Oleh karena itu, penulis terbuka terhadap kritik dan saran yang membangun untuk meningkatkan kualitas skripsi di masa mendatang. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan banyak wawasan dan manfaat bagi para pembaca.

Bandar Lampung, 6 April 2026

Penulis,

Ahmad Rafaly

DAFTAR ISI

Halaman

DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Profil Perusahaan	5
2.1.1 Gink Technology	5
2.2 Dasar Teori	6
2.2.1 Sistem Manajemen Proyek.....	6
2.2.2 <i>Requirement Change Management (RCM)</i>	7
2.2.3 <i>Function Point Analysis</i>	8
2.2.4 <i>Software Cost Estimation (SCE)</i>	12
2.2.5 <i>Change Control Board (CCB)</i>	12
2.2.6 <i>Scrum</i>	13
2.2.7 Diagram UML	14
2.2.8 Basis Data.....	15
2.2.9 PostgreSQL	16

2.2.10	PHP	16
2.2.11	Laravel	17
2.2.12	<i>Unit Testing</i> (UT).....	17
2.3	Penelitian Terdahulu	18
BAB III METODOLOGI PENELITIAN.....		21
3.1	Waktu dan Tempat Penelitian.....	21
3.1.1	Waktu Penelitian	21
3.1.2	Tempat Penelitian.....	21
3.2	Perangkat Penelitian	22
3.2.1	Perangkat Keras.....	22
3.2.2	Perangkat Lunak.....	22
3.3	Metode Pengembangan Sistem.....	24
3.4	Tahapan Penelitian.....	25
3.4.1	Analisis Kebutuhan Perusahaan	25
3.4.2	Perancangan Sistem.....	30
3.4.3	Implementasi Sistem	42
3.4.4	Pengujian Sistem	43
3.4.5	Pengumpulan Data Proyek Manajemen Perusahaan	44
3.5	Pengukuran Estimasi Biaya dan Waktu Proyek dengan <i>Function Point</i> 45	
3.6	Perhitungan <i>Function Point</i> pada Perubahan Fitur	47
BAB IV HASIL & PEMBAHASAN		49
4.1	Proses Pengembangan Sistem dengan <i>Scrum</i>	49
4.2	<i>Product Backlog</i>	49
4.3	<i>Sprint</i>	51
4.4	Fase Pengembangan.....	52
4.4.1	<i>Sprint 1</i>	52
4.4.1.1	<i>Sprint Planning</i>	52
4.4.1.2	<i>Sprint Task</i>	52
4.4.1.3	<i>Sprint Review</i>	52
4.4.1.4	<i>Sprint Retrospective</i>	55
4.4.2	<i>Sprint 2</i>	56
4.4.2.1	<i>Sprint Planning</i>	56
4.4.2.2	<i>Sprint Task</i>	56

4.4.2.3	<i>Sprint Review</i>	56
4.4.2.4	<i>Sprint Retrospective</i>	58
4.4.3	<i>Sprint 3</i>	58
4.4.3.1	<i>Sprint Planning</i>	58
4.4.3.2	<i>Sprint Task</i>	58
4.4.3.3	<i>Sprint Review</i>	59
4.4.3.4	<i>Sprint Retrospective</i>	65
4.4.4	<i>Sprint 4</i>	65
4.4.4.1	<i>Sprint Planning</i>	65
4.4.4.2	<i>Sprint Task</i>	65
4.4.4.3	<i>Sprint Review</i>	65
4.4.4.4	<i>Sprint Retrospective</i>	68
4.4.5	<i>Sprint 5</i>	68
4.4.5.1	<i>Sprint Planning</i>	68
4.4.5.2	<i>Sprint Task</i>	68
4.4.5.3	<i>Sprint Review</i>	68
4.4.5.4	<i>Sprint Retrospective</i>	73
4.4.6	<i>Sprint 6</i>	73
4.4.6.1	<i>Sprint Planning</i>	73
4.4.6.2	<i>Sprint Task</i>	73
4.4.6.3	<i>Sprint Review</i>	73
4.4.6.4	<i>Sprint Retrospective</i>	78
4.4.7	<i>Sprint 7</i>	78
4.4.7.1	<i>Sprint Planning</i>	78
4.4.7.2	<i>Sprint Task</i>	78
4.4.7.3	<i>Sprint Review</i>	78
4.4.7.4	<i>Sprint Retrospective</i>	81
4.5	Pengujian Sistem.....	82
4.5.1	<i>Blackbox Testing</i>	82
4.5.2	<i>Unit Testing</i>	85
4.5.3	Pengujian Akurasi Estimasi Awal Biaya dan Waktu <i>Function Point</i> pada Proyek	87
4.5.4	Pengujian Akurasi Estimasi Dampak Perubahan Kebutuhan dari Sisi Biaya dan Waktu pada Sistem X & Y.....	92

4.6	Analisis Hasil	95
4.6.1	Alur Tahapan Sistem	95
4.6.2	Implementasi <i>RCM Process</i> pada Sistem.....	98
4.6.3	Analisis Hasil Estimasi Awal Total Waktu dan Biaya FP terhadap Proyek Aktual	98
4.6.4	Analisis Hasil Estimasi Dampak Perubahan Kebutuhan dari Sisi Waktu dan Biaya FP terhadap Proyek Aktual.....	99
BAB V SIMPULAN DAN SARAN		103
5.1	Simpulan	103
5.2	Saran	104
DAFTAR PUSTAKA		105

DAFTAR GAMBAR

	Halaman
Gambar 1 Struktur Organisasi di Gink Technology [4].....	6
Gambar 2 <i>Requirement Change Management Process</i> [1].....	7
Gambar 3 Alur Kerja <i>Scrum</i> [13].....	14
Gambar 4 Alur Tahapan Penelitian.....	25
Gambar 5 Alur Bisnis dari Sistem Manajemen Proyek yang dibuat	26
Gambar 6 Diagram arsitektur sistem keseluruhan	27
Gambar 7 <i>Use Case Diagram</i> Sistem	30
Gambar 8 <i>Activity Diagram Login</i>	31
Gambar 9 <i>Activity Diagram</i> Manajemen data proyek.....	32
Gambar 10 <i>Activity Diagram</i> Pantau <i>progress</i> proyek	33
Gambar 11 <i>Activity Diagram</i> Membuat permintaan perubahan	34
Gambar 12 <i>Activity Diagram</i> Melakukan <i>Approval</i> Perubahan Fitur.....	35
Gambar 13 <i>Activity Diagram</i> Manajemen <i>project board</i>	36
Gambar 14 <i>Activity Diagram</i> Mengajukan <i>meeting</i> & membuat notulensi.....	37
Gambar 15 <i>Activity Diagram</i> Menerima notifikasi.....	38
Gambar 16 <i>Activity Diagram</i> Membuat & menghapus dokumen kontrak	39
Gambar 17 <i>Activity Diagram</i> Manajemen akun & <i>Role</i>	40
Gambar 18 <i>Activity Diagram</i> Manajemen Repositori Fitur.....	41
Gambar 19 <i>Activity Diagram Logout</i>	42
Gambar 20 ERD dari Sistem yang dibuat.....	53
Gambar 21 Konfigurasi file <i>.env</i>	57
Gambar 22 Sintaks Model <i>User</i>	59
Gambar 23 Sintaks Middleware Otorisasi <i>User</i>	60
Gambar 24 Sintaks <i>UserController</i>	60
Gambar 25 Sintaks Model <i>Project</i>	62
Gambar 26 Sintaks <i>ProjectController</i>	63
Gambar 27 Kode Perhitungan estimasi biaya dan waktu proyek	64
Gambar 28 Sintaks Model <i>Feature</i>	66
Gambar 29 Sintaks <i>FeatureController</i>	67
Gambar 30 Sintaks Model <i>Ticket</i>	69
Gambar 31 Sintaks <i>TicketController</i>	70
Gambar 32 Sintaks Model <i>Feature Project</i>	71
Gambar 33 Sintaks <i>ProjectBoardController</i>	72

Gambar 34 Sintaks Model Notifikasi.....	74
Gambar 35 Sintaks <i>NotificationController</i>	75
Gambar 36 Sintaks Model <i>Meeting</i>	76
Gambar 37 Sintaks <i>MeetingController</i>	77
Gambar 38 Sintaks Model <i>Contract</i>	79
Gambar 39 Sintaks <i>ContractController</i>	80
Gambar 40 Pengujian bersama <i>stakeholder</i>	81
Gambar 41 Alur Tahapan Sistem.....	95
Gambar 42 Tampilan Halaman <i>Project Management.jsx</i>	97

DAFTAR TABEL

Halaman

Tabel 1 Bobot <i>Unadjusted Function Point</i> [7].....	9
Tabel 2 Kategori Kompleksitas Implementasi UFP [9].....	9
Tabel 3 Karakteristik GSC [7]	10
Tabel 4 Keterangan Skala GSC [7].....	11
Tabel 5 <i>Factor Productivity</i> Bahasa Pemrograman [10]	11
Tabel 6 Simbol pada <i>use case diagram</i> [14].....	15
Tabel 7 Simbol pada <i>activity diagram</i> [14].....	15
Tabel 8 Waktu Penelitian	21
Tabel 9 Perangkat Keras yang Digunakan Selama Penelitian	22
Tabel 10 Perangkat Lunak yang Digunakan Selama Penelitian	22
Tabel 11 <i>Scrum Team</i>	24
Tabel 12 Kebutuhan Fungsional Sistem	28
Tabel 13 Kebutuhan Non Fungsional Sistem	29
Tabel 14 Perhitungan UFP [7]	45
Tabel 15 Perhitungan TDI [7].....	46
Tabel 16 <i>Product Backlog</i>	49
Tabel 17 Jadwal <i>Sprint</i>	51
Tabel 18 <i>Route data user</i>	61
Tabel 19 <i>Route data proyek</i>	63
Tabel 20 <i>Route data repositori fitur</i>	67
Tabel 21 <i>Route data ticket</i>	70
Tabel 22 <i>Route data fitur pada proyek</i>	72
Tabel 23 Hasil Pengujian <i>Black Box Testing</i>	82
Tabel 24 Hasil Pengujian <i>Unit Testing</i>	85
Tabel 25 Hasil klasifikasi FP per fitur sistem X	87
Tabel 26 Hasil perhitungan total UFP sistem X.....	88
Tabel 27 Hasil klasifikasi TDI sistem X	88
Tabel 28 Hasil klasifikasi FP per fitur sistem Y	89
Tabel 29 Hasil perhitungan total UFP sistem Y.....	90
Tabel 30 Hasil klasifikasi TDI sistem Y	91
Tabel 31 Hasil estimasi dampak perubahan kebutuhan pada sistem X & Y	92
Tabel 32 Penyesuaian khusus komponen UFP pada perusahaan.....	102

I. PENDAHULUAN

1.1 Latar Belakang

Perubahan kebutuhan merupakan hal yang tidak dapat dihindari dalam pengembangan perangkat lunak, khususnya pada perusahaan teknologi seperti Gink Technology [1]. Namun dalam praktiknya, kebutuhan dari *stakeholder* terhadap teknologi tersebut sering berubah karena faktor regulasi atau kebutuhan layanan digital yang berkembang. Ketika perubahan ini tidak dikelola dengan baik, maka bisa terjadi keterlambatan implementasi *software*, pemborosan anggaran, hingga hasil pengembangan yang tidak sesuai ekspektasi [1]. Belum adanya sistem manajemen proyek di Gink Technology, menjadi urgensi utama dalam penelitian ini untuk membangun Sistem Manajemen Proyek yang menerapkan *Requirement Change Management Process* [1] dalam mengelola suatu perubahan kebutuhan dan memberikan estimasi dari sisi biaya dan waktu menggunakan *Function Point Analysis* [2] menggunakan Laravel, yang mencakup seluruh siklus suatu sistem manajemen proyek yang telah ditetapkan pada ISO 21502:2020 [3], yaitu sistem manajemen proyek harus menggabungkan praktek pengarahannya, perencanaan, pemantauan, pengendalian, dan penutupan proyek.

Penelitian yang dilakukan hanya berfokus pada pengembangan sisi *backend* yang mengatur logika dan alur data pada sistem. Dengan menerapkan sistem manajemen proyek ini pada perusahaan Gink Technology, diharapkan proses manajemen proyek dapat dilakukan secara kolaboratif dan terukur, terutama dari sisi waktu pengerjaan dan biaya.

1.2 Rumusan Masalah

Adapun Rumusan Masalah pada penelitian ini adalah sebagai berikut :

1. Bagaimana sistem yang dibangun dapat sesuai dengan *Requirement Change Management Process* dalam menangani setiap perubahan kebutuhan yang terjadi di dalam *software*?
2. Bagaimana sistem yang dibangun dapat memberikan estimasi awal total waktu dan biaya yang diperlukan untuk membangun suatu sistem di perusahaan Gink Technology?
3. Bagaimana sistem yang dibangun dapat menjadi alat yang dapat melakukan pengukuran dampak perubahan kebutuhan fitur dari sisi biaya dan waktu secara jelas di perusahaan Gink Technology?

1.3 Tujuan Penelitian

Adapun Tujuan dari Penelitian ini adalah sebagai berikut :

1. Membangun sistem manajemen proyek yang sesuai dengan *Requirement Change Management Process* dengan Laravel sebagai sistem yang dapat menangani setiap perubahan kebutuhan yang terjadi di dalam *software*.
2. Membangun sistem yang dapat mengukur estimasi awal total waktu dan biaya yang diperlukan menggunakan *Function Point Analysis* yang diimplementasikan menggunakan Laravel.
3. Membangun sistem yang dapat mengukur dampak perubahan kebutuhan fitur dari sisi biaya dan waktu menggunakan *Function Point Analysis* yang diimplementasikan menggunakan Laravel.

1.4 Manfaat Penelitian

Adapun Manfaat dari Penelitian ini adalah sebagai berikut :

1. Menjadi referensi akademik bagi penelitian selanjutnya terkait penerapan sistem manajemen proyek.
2. Menyediakan sistem yang dapat mengelola setiap permintaan perubahan yang diterima oleh perusahaan Gink Technology.

1.5 Batasan Masalah

Adapun Batasan masalah dalam penelitian ini adalah :

1. Penelitian ini hanya berfokus pada pengembangan *backend*, karena sistem dibuat melalui kolaborasi antar tiga anggota tim, di mana Den Muhammad Wyzdan (2215061003) mengembangkan bagian *Frontend* pada *website*, dan Arnora Mardiansyah (2215061015) menangani pengembangan aspek *Quality Assurance*.
2. Sistem yang dirancang merupakan sistem yang hanya dapat menangani proses manajemen proyek pembuatan *software website* dan aplikasi, serta sistem ini bersifat tertutup karena menyimpan data milik perusahaan, maka dari itu *domain website* tidak dapat dicantumkan serta tidak diperlukan pembuatan API pada sistem ini.
3. Penulis mengerjakan bagian perhitungan *function point* di sisi *frontend* karena perusahaan merubah proses kalkulasi *function point* yang sebelumnya dilakukan di sisi *backend* menjadi dilakukan di sisi *frontend* agar perusahaan langsung dapat melihat hasil kalkulasi tanpa perlu menyimpan data terlebih dahulu.

1.6 Sistematika Penulisan

Adapun struktur penulisan yang diterapkan pada penelitian ini sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan laporan.

BAB II: TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian-penelitian sebelumnya dan teori-teori yang mendukung sebagai referensi dalam penelitian ini.

BAB III: METODOLOGI PENELITIAN

Bab ini berisi tentang waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, dan prosedur serta analisis kebutuhan perangkat lunak.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan mengenai penelitian yang dilakukan.

BAB V: KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan berdasarkan hasil penelitian dan saran yang diharapkan untuk pengembangan penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

II. TINJAUAN PUSTAKA

2.1 Profil Perusahaan

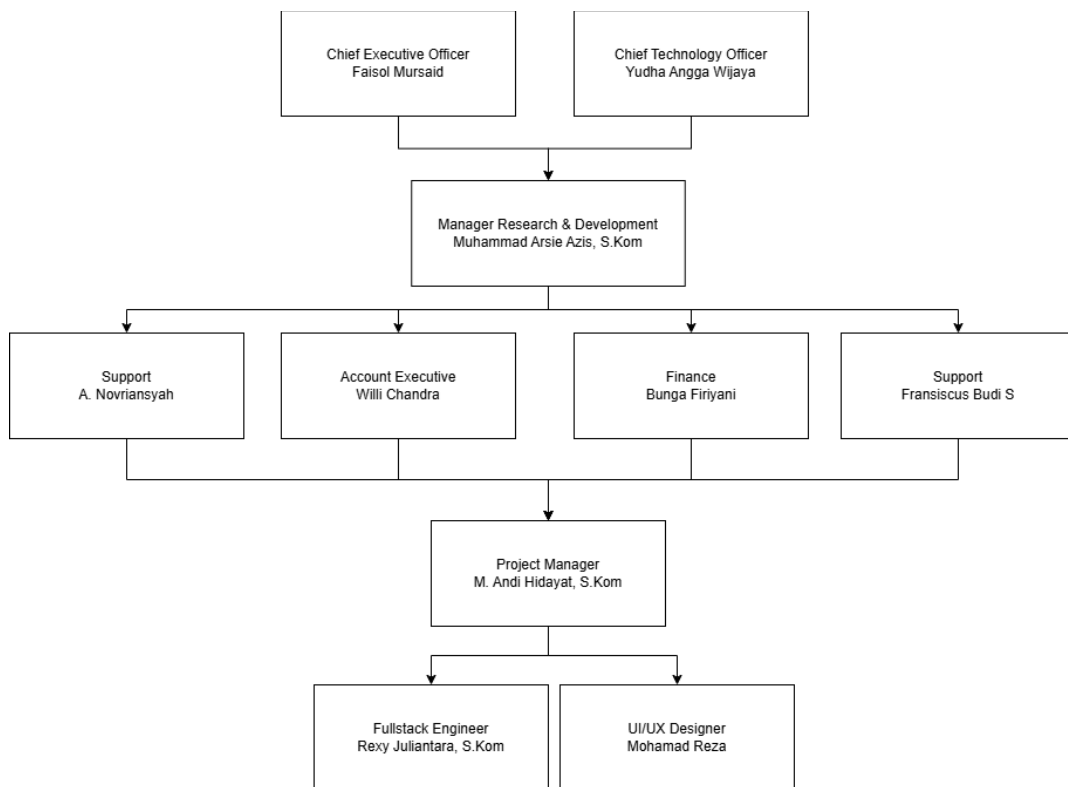
2.1.1 Gink Technology

Gink Technology adalah perusahaan yang bergerak di bidang layanan teknologi informasi, berkantor di Jalan Way Pengubuan No. 16, Pahoman, Enggal, Bandar Lampung, Lampung. Gink Technology berdiri pada tahun 2012 dan melayani berbagai klien baik institusi pemerintahan, perusahaan komersial, sekolah, hingga LSM, dari dalam dan luar Lampung. Beberapa peran yang dimiliki oleh perusahaan ini adalah *Full Stack Web Developer, Business Analyst, UI/UX Designer, Project Manager* yang menunjukkan bahwa perusahaan aktif dalam mengembangkan kapabilitas internal dan menangani proyek pengembangan perangkat lunak yang cukup kompleks [4].

Perusahaan ini menyediakan beragam layanan teknologi digital, di antaranya:

- a) Pengembangan *website* (*company profile*, portal, *E-Commerce*, dll.)
- b) Pengembangan aplikasi berbasis *web* (*web-based applications*)
- c) Pengembangan aplikasi *mobile* untuk Android, iOS, dan *cross-platform* dalam beberapa proyek.
- d) Produksi video profil perusahaan serta *branding digital* dan *video marketing*.
- e) Layanan digital *marketing* seperti konten, promosi *online*, serta optimalisasi keberadaan digital klien.

Struktur Organisasi di perusahaan Gink Technology adalah sebagai berikut :



Gambar 1 Struktur Organisasi di Gink Technology [4]

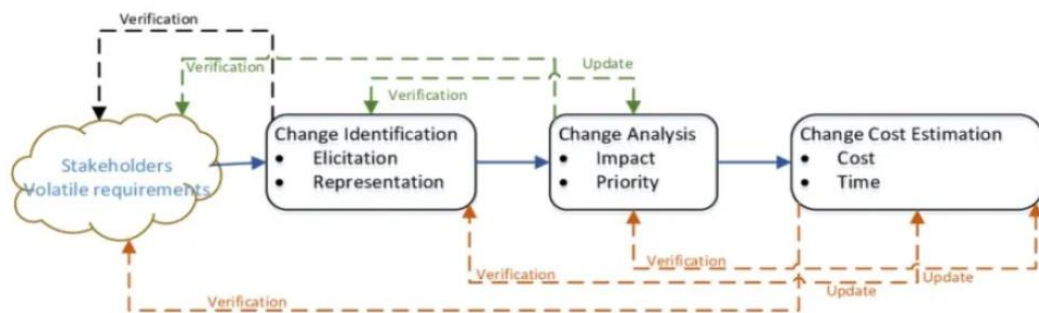
2.2 Dasar Teori

2.2.1 Sistem Manajemen Proyek

Manajemen Proyek adalah bidang yang melibatkan proses perencanaan, pengorganisasian, pengendalian untuk mencapai tujuan yang telah ditetapkan di dalam suatu proyek [5]. Sementara itu, sistem manajemen proyek adalah sebuah sistem yang dibuat dengan tujuan merencanakan dan mengendalikan proses dari awal hingga akhir proyek meliputi teknologi, dokumentasi, dan penyediaan informasi yang sesuai dengan kebutuhan yang diperlukan. Sistem manajemen proyek dapat memudahkan perusahaan dalam memantau seluruh pekerjaan yang berlangsung di berbagai daerah, memberikan informasi tentang kemajuan proyek, serta waktu dan biaya yang efisien [5]. Berdasarkan standar yang telah ditetapkan pada ISO 21502:2020, sistem manajemen proyek harus menggabungkan praktek pengarahan, perencanaan, pemantauan, pengendalian, dan penutupan proyek [3].

2.2.2 Requirement Change Management (RCM)

Requirement Change Management adalah kegiatan pengelolaan perubahan kebutuhan selama siklus hidup pengembangan perangkat lunak. RCM bertujuan untuk memastikan bahwa setiap perubahan yang diajukan dapat dianalisis secara menyeluruh, dilacak, terdokumentasi, dan dikendalikan agar tidak mengganggu tujuan awal proyek [6]. Perubahan kebutuhan yang tidak dikelola dengan baik dapat menyebabkan gangguan pada jadwal proyek, anggaran, dan kualitas perangkat lunak [6]. Sistem yang dibangun untuk RCM harus mendukung proses yang iteratif dan dinamis untuk umpan balik, serta komunikasi yang berkelanjutan untuk seluruh pemangku kepentingan [1]. Proses RCM pada dasarnya melibatkan tiga aktivitas utama, yaitu identifikasi perubahan, analisis dampak perubahan, dan estimasi biaya perubahan [1].



Gambar 2 *Requirement Change Management Process* [1]

Berdasarkan gambar di atas, identifikasi perubahan merupakan tahap awal dan paling krusial dalam proses manajemen perubahan kebutuhan karena seluruh tahapan selanjutnya sangat bergantung pada ketepatan dalam mengenali masalah dan kebutuhan perubahan [1]. Kebutuhan perubahan harus digali dan dipahami secara detail agar perubahan yang diusulkan benar-benar sesuai dengan pemangku kepentingan. Tahap kedua yaitu analisa dampak perubahan, analisa penting karena perubahan yang tampak kecil dapat menimbulkan efek berantai yang signifikan pada bagian sistem lainnya [1]. Tahap ketiga yaitu estimasi sumber daya berupa biaya dan waktu, estimasi biaya umumnya dilakukan di awal proyek namun jika terjadi perubahan kebutuhan di dalam masa pengembangan, maka diperlukan sumber daya tambahan untuk mengimplementasikan perubahan tersebut [1].

2.2.3 *Function Point Analysis*

Metode *Function Point Analysis* merupakan bagian dari metode *Functional Size Measurement* yang pertama kali diperkenalkan oleh Albrecht sebagai metode untuk mengukur tingkat kompleksitas dan fungsionalitas dalam suatu proyek perangkat lunak. Dalam prosedur FPA terdapat berbagai proses, yang meliputi data masuk dan data keluar yang akan diproses dalam sistem [7]. Metode ini digunakan untuk memperkirakan waktu dan biaya pengembangan perangkat lunak, bahkan sebelum sistem diimplementasikan. Pendekatan ini digunakan untuk menghitung dampak perubahan kebutuhan terhadap durasi dan anggaran proyek [8]. Tahap pertama untuk menghitung nilai *Function Point* yaitu menentukan nilai dari *Adjusted Function Point* (AFP). Adapun rumus untuk menghitung AFP yang telah diperbarui IFPUG adalah sebagai berikut :

$$AFP = UFP \times VAF [7]$$

AFP = *Adjusted Function Point*

UFP = *Unadjusted Function Point*

VAF = *Value Adjustment Factor*

Unadjusted Function Point menghitung ukuran sistem perangkat lunak berdasarkan lima komponen utama. Setiap komponen dinilai dengan bobot tertentu sesuai tingkat kompleksitasnya, yang dibagi menjadi tiga kategori: *Simple*, *Average*, dan *Complex* [2]. Lima komponen tersebut meliputi:

- 1) *External Input* (EI) yaitu data masukan yang diterima dari luar sistem.
- 2) *External Output* (EO) yaitu hasil keluaran yang dihasilkan oleh sistem, misalnya notifikasi atau *file*.
- 3) *External Inquiry* (EQ) berupa permintaan data dari pengguna terhadap sistem tanpa melakukan perubahan apapun pada data.
- 4) *Internal Logical File* (ILF) mencakup jumlah berkas atau basis data internal yang digunakan oleh sistem.
- 5) *External Interface File* (EIF) yaitu jumlah berkas atau basis data eksternal yang terhubung dan berinteraksi dengan sistem.

Masing-masing komponen ini dikalikan dengan bobot kompleksitas implementasinya yang dikategorikan sebagai berikut :

Tabel 1 Bobot *Unadjusted Function Point* [7]

Parameter	<i>Simple</i>	<i>Average</i>	<i>Complex</i>
<i>External Input</i>	3	4	6
<i>External Output</i>	4	5	7
<i>Logical Internal files</i>	7	10	15
<i>External Interface files</i>	5	7	10
<i>External Inquiry</i>	3	4	6

Pengkategorian level dari kompleksitas implementasi UFP dapat dijabarkan sebagai berikut :

Tabel 2 Kategori Kompleksitas Implementasi UFP [9]

Komponen	Mudah	Sedang	Kompleks
<i>External Input (EI)</i>	Tidak memerlukan banyak data masukan.	Membutuhkan <i>input</i> dari pengguna dengan cara yang cukup sederhana.	Membutuhkan sejumlah besar data masukan.
<i>External Output (EO)</i>	Hanya menampilkan teks sederhana.	Menghasilkan keluaran berupa laporan seperti <i>file</i> CSV, PDF, dan sejenisnya.	Melibatkan banyak proses <i>query</i> untuk menghasilkan <i>output</i> .
<i>External Inquiry (EQ)</i>	Menampilkan sedikit data (misalnya <i>query</i> dengan panjang kurang dari tiga baris).	Menampilkan data dengan rentang waktu tertentu.	Menampilkan data yang membutuhkan proses identifikasi dan pengecualian.

Internal Logical File (ILF)	Menggunakan sejumlah tabel.	Menggunakan beberapa tabel untuk menyimpan data dalam periode tertentu.	Menggunakan tabel yang saling berelasi secara kompleks.
External Interface File (EIF)	Terhubung dengan sistem atau aplikasi lain yang dirujuk oleh sistem.	Berhubungan dengan sistem eksternal dan koordinasi dengan pemilik sistem tersebut.	Terintegrasi dengan sistem eksternal yang sulit diakses atau memiliki batasan tertentu.

General System Characteristics digunakan untuk menghitung kompleksitas umum sistem yang memiliki pengaruh dari skala 0 hingga 5 yang selanjutnya diakumulasi menjadi sebuah nilai bernama *Total Degree Influence*. Adapun kompleksitas GSC dan keterangan skala dijelaskan pada tabel 3 dan 4 di bawah :

Tabel 3 Karakteristik GSC [7]

No	Karakteristik
1	Tingkat kompleksitas komunikasi data
2	Tingkat kompleksitas pemrosesan terdistribusi
3	Tingkat kompleksitas performa
4	Tingkat kompleksitas konfigurasi
5	Tingkat frekuensi penggunaan perangkat lunak
6	Tingkat frekuensi <i>input</i> data
7	Tingkat kemudahan penggunaan bagi pengguna
8	Tingkat frekuensi perbarui data
9	Tingkat kompleksitas pemrosesan data
10	Tingkat <i>reusable code</i> program
11	Tingkat kemudahan dalam instalasi
12	Tingkat kemudahan operasional perangkat lunak
13	Tingkat perangkat lunak dibuat untuk <i>multi</i> organisasi/perusahaan/ <i>client</i>
14	Tingkat kompleksitas dalam mengadaptasi perubahan

Tabel 4 Keterangan Skala GSC [7]

Nilai Skala	Keterangan
0	Tidak memiliki pengaruh
1	Pengaruh tidak signifikan
2	Pengaruh Moderat
3	Pengaruh rata – rata
4	Berpengaruh Signifikan
5	Berpengaruh Kuat

Value Adjustment Factor (VAF) mencerminkan nilai dari fungsionalitas umum dari software yang dibuat. Nilai VAF dihitung menggunakan nilai TDI yang telah diperoleh, adapun rumus untuk menghitung nilai VAF adalah sebagai berikut :

$$VAF = (TDI \times 0.01) + 0.65 [7] (2)$$

Rumus Function Point yang digunakan pada penelitian ini dapat digabungkan menjadi persamaan berikut :

$$FP = UFP \times ((TDI \times 0.01) + 0.65) (3)$$

Adapun nilai *Factor Productivity* yang untuk setiap bahasa pemrograman, menggunakan acuan yang dibuat oleh Capers Jones pada tabel di bawah ini :

Tabel 5 *Factor Productivity* Bahasa Pemrograman [10]

Bahasa Pemrograman	<i>Factor Productivity</i>
Python	53
HTML	160
PHP	53
Javascript	71
Java	53

2.2.4 *Software Cost Estimation (SCE)*

Tahap perencanaan merupakan tahap awal yang akan menentukan keberhasilan suatu proyek perangkat lunak. Proyek tidak dapat dikesampingkan jika proyek tersebut tidak mempunyai estimasi dan perencanaan [9]. Estimasi dan perencanaan dalam proyek perangkat lunak dapat diawali dengan pengukuran perangkat lunak yang akan dibuat atau dikembangkan, maka dari itu dalam membangun perangkat lunak sebaiknya dimulai dengan perencanaan proyek yang matang [9]. Perhitungan estimasi waktu pada penelitian ini, menggunakan rumus yang diberikan oleh Ali [11] yaitu sebagai berikut :

$$E = \left\{ Nw \times \left(\frac{factor \times FP}{n} \right) \right\} [11]$$

E = Perkiraan waktu pengembangan (jam)

Nw = Waktu pengerjaan dalam sehari

Factor = Baris kode pada perangkat lunak per *Function Point*

FP = *Function Point*

n = Jumlah baris kode per hari

Persamaan untuk menghitung estimasi biaya yaitu sebagai berikut :

$$\Delta C_D = E \times W_P [11]$$

ΔC_D = Estimasi biaya

E = Perkiraan waktu pengembangan (jam)

W_P = Biaya pengembangan per jam

2.2.5 *Change Control Board (CCB)*

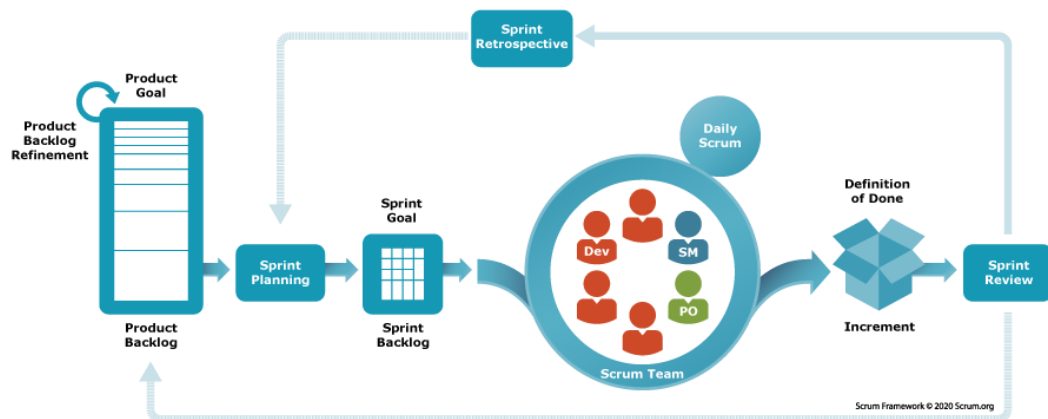
CCB (*Change Control Board*) adalah sebuah tim pengambil keputusan dalam mengelola perubahan kebutuhan dalam suatu proyek atau sistem. Tim ini terdiri dari berbagai anggota yang memiliki keahlian dan pengetahuan yang relevan, sehingga dapat membuat keputusan yang tepat dan berdasarkan analisis yang mendalam [12].

Anggota CCB biasanya terdiri dari analis sistem yang memahami kebutuhan dan teknis sistem, *project manager* yang mengelola proyek dan sumber daya, developer senior yang memiliki pengetahuan teknis yang mendalam, perwakilan *stakeholder* yang memahami kebutuhan dan prioritas *stakeholder*, serta *quality assurance* yang memastikan kualitas dan kepatuhan terhadap standar [12].

Fungsi utama CCB adalah untuk menilai dampak dari perubahan kebutuhan, baik dari sisi waktu, biaya, maupun teknis. Berdasarkan hasil analisis, CCB kemudian dapat menyetujui atau menolak perubahan tersebut. Selain itu, CCB juga bertanggung jawab untuk menjaga agar perubahan kebutuhan tetap sejalan dengan tujuan proyek, sehingga setiap perubahan yang dilakukan memiliki arah yang jelas dan terfokus [12].

2.2.6 Scrum

Scrum adalah salah satu metodologi dalam kerangka *Agile* yang paling banyak digunakan dan terbukti efektif dalam pengembangan perangkat lunak. Pendekatan ini menekankan pada fleksibilitas, kerja sama tim, serta pengiriman produk secara bertahap. Dengan *Scrum*, tim mampu merespons perubahan kebutuhan maupun prioritas proyek dengan cepat, sehingga cocok diterapkan pada lingkungan pengembangan yang serba dinamis. Prinsip *Scrum* sejalan dengan *Agile*, mencakup tahap kebutuhan, analisis, desain, pengembangan, hingga penyampaian produk. Seluruh aktivitas tersebut dilaksanakan dalam siklus yang disebut *sprint*, yakni periode tetap (umumnya 1–4 minggu) di mana tim berfokus menyelesaikan fitur atau kebutuhan yang diambil dari *backlog*. *Backlog* sendiri merupakan daftar prioritas fitur yang dapat diperbarui oleh *product owner* sesuai kebutuhan, meskipun tidak boleh diubah selama *sprint* berlangsung agar tim bekerja secara stabil [13].



Gambar 3 Alur Kerja *Scrum* [13]




Dalam praktiknya, *Scrum* mengenal *scrum meeting* harian berdurasi singkat, sekitar 15 menit, di mana anggota tim membahas apa yang sudah dikerjakan, hambatan yang dihadapi, dan rencana pekerjaan berikutnya. Pertemuan ini dipandu oleh *scrum master* yang berperan membantu mengatasi masalah sekaligus menjaga keteraturan tim. Setelah *sprint* berakhir, diadakan sesi demo untuk menampilkan fungsionalitas baru yang telah selesai. Berbeda dengan model tradisional *plan-driven* seperti *waterfall*, yang menuntut perencanaan *detail* sejak awal dan berjalan secara berurutan, *Scrum* lebih sesuai menghadapi ketidakpastian. Metode *waterfall* efektif untuk masalah yang jelas terdefinisi, tetapi kurang fleksibel terhadap perubahan. Umpan balik dari pengguna di tiap iterasi membuat tim lebih cepat beradaptasi, memperbaiki kekurangan, dan menyesuaikan prioritas tanpa merombak keseluruhan rencana. Karena itulah, *Scrum* dianggap solusi ideal dalam menghadapi tuntutan yang dinamis serta membantu bisnis terus relevan dengan kebutuhan pengguna yang selalu berkembang [13].

2.2.7 Diagram UML



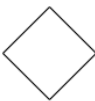

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk memvisualisasikan dan mendokumentasikan hasil analisis dan desain sistem perangkat lunak. Tujuan utama penggunaan UML adalah menyediakan bahasa pemodelan visual yang dapat dipahami bagi pengembang, mengetahui alur sebuah sistem, dan menjadikannya dasar formal untuk pemahaman

bahasa pemodelan. Ada beberapa jenis UML seperti *use case diagram*, *activity diagram*, dan *activity diagram* [14]. Simbol pada *use case diagram* dan *activity diagram* yang digunakan pada penelitian ini tercantum pada tabel di bawah :

Tabel 6 Simbol pada *use case diagram* [14]

Simbol	Nama Simbol	Keterangan
	Aktor	Entitas yang berinteraksi dengan sistem
	<i>Use Case</i>	Interaksi antara sistem dan aktor
	<i>Association</i>	Penghubung antara aktor dengan <i>use case</i>

Tabel 7 Simbol pada *activity diagram* [14]

Simbol	Nama Simbol	Keterangan
	Status awal	Simbol yang menandakan awal dari suatu <i>activity diagram</i>
	Aktivitas	Aktivitas yang dilakukan sistem
	<i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu
	Status Akhir	Simbol yang menandakan akhir dari suatu <i>activity diagram</i>

2.2.8 Basis Data

Basis data adalah sistem digital yang berfungsi untuk menyimpan dan mengelola data. Proses pengolahan data ini dilakukan melalui *query* menggunakan perangkat lunak khusus yang disebut *Database Management Systems (DBMS)*. Ada banyak DBMS populer yang digunakan, seperti MySQL, Oracle, PostgreSQL, dan MongoDB [15]. Secara umum, basis data dibagi menjadi dua kategori utama yaitu *Relational Databases* dan *Non-Relational Databases*. *Relational Databases*, misalnya, merepresentasikan data dalam bentuk tabel. Setiap baris dalam tabel ini

memiliki pengenal unik yang disebut *identifier key* atau ID, dan setiap kolomnya disebut atribut. Tipe basis data ini mengandalkan bahasa SQL (*Structured Query Language*) untuk memanipulasi data [15]. Saat mengembangkan dan mengelola *database*, konsep *entity relationship* menjadi sangat penting. Konsep ini digunakan untuk mengelompokkan data berdasarkan entitas (hal-hal yang ingin disimpan), atribut (data spesifik dari entitas), dan relasi (hubungan antar entitas). Setiap entitas akan menjadi sebuah tabel, dengan atribut sebagai kolom-kolomnya. Relasi antar entitas dapat berupa *one-to-one*, *one-to-many*, atau *many-to-many*, yang menunjukkan bagaimana data dalam satu tabel terhubung dengan data di tabel lain [15].

2.2.9 PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional yang dapat mendukung tipe data relasional maupun non-relasional dengan standar teknis yang kuat. Sejak dikembangkan pada tahun 1986 oleh Michael Stonebraker, teknologi ini terus berevolusi menjadi solusi yang sangat skalabel, memiliki kinerja tinggi, dan efisien dalam mengelola data melalui sistem *Multiversion Concurrency Control* (MVCC) di mana proses baca-tulis tidak saling menghalangi. Keunggulan utamanya terletak pada fleksibilitas dukungan berbagai bahasa pemrograman, fitur pemulihan data *Point-In-Time Recovery* (PITR), serta kemampuannya untuk diperluas dengan fitur seperti PostGIS untuk data geospasial. Karena sifatnya yang sepenuhnya bebas biaya lisensi dan tanpa risiko *vendor lock-in*, PostgreSQL banyak digunakan oleh perusahaan untuk mengelola transaksi online (OLTP), melakukan analisis data mendalam, hingga mengonsolidasi basis data lama guna menekan total biaya operasional. *open-source* [16].

2.2.10 PHP

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman yang pertama kali diciptakan oleh Rasmus Lerdorf pada tahun 1994. Seiring perkembangannya, PHP memiliki karakteristik unik dalam penulisannya. Salah satu ciri khas utamanya adalah penggunaan penanda blok kode, yaitu `<?php` untuk memulai dan `?>` untuk mengakhirinya. Setiap perintah atau instruksi dalam kode PHP harus diakhiri

dengan titik koma, yang berfungsi sebagai penanda akhir dari satu perintah sebelum beralih ke perintah berikutnya [17].

Selain itu, PHP sangat peka terhadap huruf besar dan kecil, atau yang dikenal sebagai *case-sensitive*, terutama untuk nama variabel yang dibuat oleh pengguna. Setiap variabel dalam PHP diawali dengan simbol \$, dan nama variabel seperti \$nama, \$Nama, dan \$NAMA akan diperlakukan sebagai tiga variabel yang berbeda oleh sistem. Mengenai tipe data, PHP memiliki keunggulan karena dapat secara otomatis mengenali dan menyesuaikan tipe data sesuai dengan nilai yang diberikan, sebuah fitur yang dikenal sebagai pengetikan dinamis. Meskipun demikian, PHP tetap memiliki beberapa tipe data dasar yang umum, seperti *string*, *integer*, *float*, *boolean*, dan *array* yang menjadi fondasi dalam pengembangan aplikasi *web* [17].

2.2.11 Laravel

Laravel adalah salah satu *framework web* berbasis PHP yang bersifat *open-source*. Framework ini dikembangkan untuk mempermudah pembuatan aplikasi *web* dengan menerapkan pola arsitektur MVC (*Model-View-Controller*). Struktur MVC yang digunakan Laravel memiliki perbedaan dibandingkan dengan implementasi MVC pada umumnya. Salah satu fitur utama dalam Laravel adalah *routing*, yang berfungsi sebagai perantara antara permintaan pengguna (*request*) dan *controller* yang menanganinya. Dengan sistem ini, *controller* tidak dapat langsung menerima permintaan tanpa melewati proses *routing* terlebih dahulu [18]. Selain *routing*, Laravel juga dilengkapi dengan berbagai fitur unggulan seperti *Eloquent ORM* untuk pengelolaan basis data yang lebih efisien, *Blade Template Engine* untuk pembuatan tampilan yang dinamis, serta sistem *middleware* yang berperan dalam pengelolaan akses pengguna. Selain itu, Laravel memiliki fitur *artisan command-line interface* yang mempermudah pengembang dalam menjalankan berbagai tugas otomatisasi, seperti migrasi database dan pembuatan *controller* [18].

2.2.12 Unit Testing (UT)

Unit testing adalah pengujian yang berfokus pada bagian terkecil dari *software* atau modul dan dilakukan untuk memastikan modul berfungsi sesuai dengan tujuan yang

ditetapkan [3]. Aliran data diuji terlebih dahulu karena kegagalan pada tahap ini akan memengaruhi pengujian lainnya. Selain itu, struktur data lokal maupun pengaruhnya terhadap data global juga harus diperiksa [13].

Unit Testing sangat penting agar kesalahan perhitungan, perbandingan, maupun alur kontrol dapat terdeteksi. Pengujian batas menjadi aspek penting karena kegagalan perangkat lunak sering terjadi pada kondisi ini, misalnya saat memproses bagian terakhir, melampaui batas pengulangan, atau mencapai nilai maksimum maupun minimum [13].

2.3 Penelitian Terdahulu

No	Peneliti	Hasil	Gap Penelitian
1	Minhas, dkk [6]	Mengusulkan sebuah kerangka kerja untuk mengelola perubahan kebutuhan dalam proses pengembangan perangkat lunak yang berfokus pada aktivitas inti dalam RCM dan menawarkan perbaikan dibandingkan model sebelumnya.	Penelitian masih terbatas pada <i>level</i> konseptual, yaitu hanya berupa kerangka abstrak tanpa adanya demonstrasi implementasi ke dalam sebuah sistem nyata.
2	Bayu, dkk [8]	Telah mengimplementasikan sistem yang menerapkan <i>RCM Process</i> sehingga <i>RCM Process</i> tidak lagi sekadar konsep melainkan sudah diterapkan dalam aplikasi nyata.	Implementasi sistem yang dilakukan belum disertai dengan pengujian estimasi biaya dan waktu yang dihasilkan dari FP dengan data biaya dan waktu proyek aktual.
3	Mendez, dkk [19]	Telah mengimplementasikan sistem manajemen proyek di github dengan beberapa fitur seperti manajemen penugasan proyek, status fitur, <i>login</i> , <i>logout</i> ,	Belum adanya fitur yang menghitung biaya suatu proyek serta sistem belum dapat mengelola perubahan suatu fitur jika

		<i>progress</i> proyek, prioritas fitur menggunakan Django & Bulma.io	terjadi permintaan tambahan.
4	Lakshay [20]	Telah mengimplementasikan sistem manajemen proyek di github dengan beberapa fitur seperti deskripsi proyek, durasi proyek, prioritas & capaian fitur, serta manajemen akun menggunakan Javascript & PHP	Belum adanya fitur yang menghitung biaya suatu proyek serta sistem belum dapat mengelola perubahan suatu fitur jika terjadi permintaan tambahan.
5	Aplikasi Jira [21]	Aplikasi manajemen proyek yang memiliki fitur manajemen tiket, akun dan hak otoritas, status fitur, notifikasi, serta kolom diskusi dan komentar	Belum adanya fitur yang menghitung estimasi biaya dan waktu suatu proyek serta tidak terdapat <i>workflow approval</i>
6	Aplikasi Trello [22]	Aplikasi manajemen proyek yang memiliki fitur label <i>deadline feature</i> , akun, status fitur, notifikasi, serta kolom diskusi dan komentar	Belum adanya fitur yang menghitung estimasi biaya dan waktu suatu proyek serta tidak terdapat <i>workflow approval</i>
7	Yhurinda, dkk [23]	Penelitian berisi perbandingan biaya aktual dengan biaya estimasi dari <i>function point</i> 4 proyek pemerintahan dengan biaya total Rp229.680.000. Hasil dari perhitungan estimasi biaya yang menggunakan <i>Function Point</i> yaitu Rp216.956.881 yang berarti estimasi tersebut memiliki akurasi sebesar 94,46%.	Perhitungan masih dilakukan secara manual dan belum diimplementasikan menjadi sebuah sistem.
8	Rachmat, dkk [24]	Penelitian menghasilkan estimasi jumlah baris kode (LOC) dari	Perhitungan masih dilakukan secara manual

		suatu sistem pembelajaran berbasis <i>web</i> sebesar 29194 <i>lines</i> . Setelah proyek selesai, didapatkan hasil LOC dari proyek sebenarnya yaitu sebesar 29647 <i>lines</i> yang menandakan bahwa hasil estimasi yang didapat melalui perhitungan <i>Function Point</i> memiliki akurasi sebesar 98,5%.	dan belum diimplementasikan menjadi sebuah sistem.
9	Satrio, dkk [25]	Penelitian menghasilkan pengukuran kualitas perangkat lunak SIMRS Khanza menggunakan <i>Function Point</i> dengan nilai 6011,28 yang menunjukkan bahwa sistem memiliki kompleksitas tinggi	Perhitungan <i>Function Point</i> masih dilakukan dengan cara <i>manual</i> dan belum diimplementasikan menjadi sebuah sistem.
10	Sandhea, dkk [26]	Penelitian menggunakan <i>Function Point</i> untuk mengukur estimasi biaya pengembangan SIBIMA di Universitas XYZ yaitu sebesar Rp22.858.705.	Penelitian menggunakan metode <i>Function Point</i> untuk mengukur estimasi biaya <i>software</i> namun belum dibandingkan dengan biaya aktual.
11	Gigih, dkk [27]	Penelitian menghasilkan aplikasi <i>game</i> edukasi aksara lampung yang menggunakan metode pengembangan <i>scrum</i> dengan 3 kali proses <i>sprint</i>	Telah dibangun aplikasi <i>game</i> edukasi aksara lampung menggunakan metode <i>scrum</i>
12	Khori, dkk [28]	Penelitian menghasilkan <i>website</i> jual beli produk kuliner UMKM yang menggunakan metode pengembangan <i>scrum</i> dengan 10 kali proses <i>sprint</i> .	Telah dibangun <i>website</i> jual beli produk kuliner UMKM menggunakan metode <i>scrum</i>

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

3.1.1 Waktu Penelitian

Pembagian waktu penelitian dapat dilihat pada tabel berikut :

Tabel 8 Waktu Penelitian

No	Nama Kegiatan	2025				2026	
		Sep	Okt	Nov	Des	Jan	Feb
1	Studi Literatur	■					
2	Perencanaan Sistem	■	■				
3	Pembuatan Sistem		■	■	■	■	
4	Pengujian Sistem				■	■	
5	Perbaikan Sistem				■	■	
6	Penulisan Laporan Ilmiah	■	■	■	■	■	■

3.1.2 Tempat Penelitian

Penelitian ini dilakukan di Kantor Gink Technology yang beralamat di Jalan Way Pengubuan Nomor 16, Kecamatan Enggal, Kelurahan Pahoman, Kota Bandar Lampung, Provinsi Lampung.

3.2 Perangkat Penelitian

Perangkat yang digunakan selama penelitian terdapat 2 jenis, yaitu perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

3.2.1 Perangkat Keras

Perangkat keras pada penelitian ini dapat dilihat di tabel berikut :

Tabel 9 Perangkat Keras yang Digunakan Selama Penelitian

No	Perangkat Keras	Spesifikasi	Kegunaan
1	<i>Laptop</i>	Merk Lenovo ideapad D330, Intel(R) Celeron(R) N4020 CPU @ 1.10GHz 1.10 GHz. 8 GB, Windows 10 Pro	Membuat dan mengembangkan Sistem
2	<i>Smartphone</i>	Redmi 12 C 128 GB, Android 13	Menguji Responsivitas Sistem yang telah dikembangkan

3.2.2 Perangkat Lunak

Perangkat lunak pada penelitian ini dapat dilihat di tabel berikut :

Tabel 10 Perangkat Lunak yang Digunakan Selama Penelitian

No	Perangkat Lunak	Versi	Kegunaan
1	Google Chrome	131.0.6778.204/.205	Digunakan sebagai peramban <i>web</i> untuk menguji dan mengakses sistem berbasis <i>website</i> yang dikembangkan.
2	Visual Studio Code	2.15.0	Digunakan sebagai <i>editor</i> kode untuk menulis dan mengelola seluruh kode sistem.
3	PHP	8.2.23	Bahasa pemrograman utama yang digunakan untuk membangun logika sisi <i>server (backend)</i> dari sistem.

4	Laravel	12	<i>Framework</i> PHP yang digunakan untuk mempercepat pengembangan sistem berbasis <i>website</i> dan mengimplementasikan arsitektur <i>Model-View-Controller</i> (MVC).
5	Laragon	6	Lingkungan pengembangan lokal yang digunakan untuk menjalankan <i>web server</i> , PHP, dan basis data PostgreSQL selama proses pengembangan sistem.
6	Github	1.224.0	Digunakan sebagai platform untuk kontrol versi kode (<i>version control</i>) dan kolaborasi dalam tim pengembangan skripsi.
7	Draw.io	28.1.2	Digunakan untuk membuat diagram-diagram yang dibutuhkan dalam dokumentasi skripsi, seperti diagram alir proses dan <i>use case</i> .
8	PostgreSQL	16	Sistem manajemen basis data relasional yang digunakan untuk menyimpan dan mengelola data sistem.

3.3 Metode Pengembangan Sistem

Metode yang digunakan pada penelitian ini menggunakan kerangka kerja *Scrum*. *Scrum* berfungsi sebagai pedoman dalam mengatur kolaborasi tim agar dapat mencapai tujuan proyek melalui siklus kerja repetitif yang disebut *sprint*. *Scrum* memiliki beberapa peran utama, antara lain:

- 1) *Product Owner*, yaitu orang yang bertanggung jawab dalam memaksimalkan produk yang dikembangkan oleh tim. Tugasnya ialah memastikan produk yang dibuat sesuai dengan kebutuhan yang diperlukan.
- 2) *Scrum Master*, yaitu orang yang berperan dalam memastikan tim menjalankan prinsip dan praktik *Scrum* dengan baik, serta membantu mengatasi berbagai hambatan yang dapat menurunkan efektivitas kerja tim.
- 3) *Development Team*, yaitu sekelompok orang yang memiliki tanggung jawab untuk merancang, mengembangkan, dan menguji produk yang sedang dibuat.

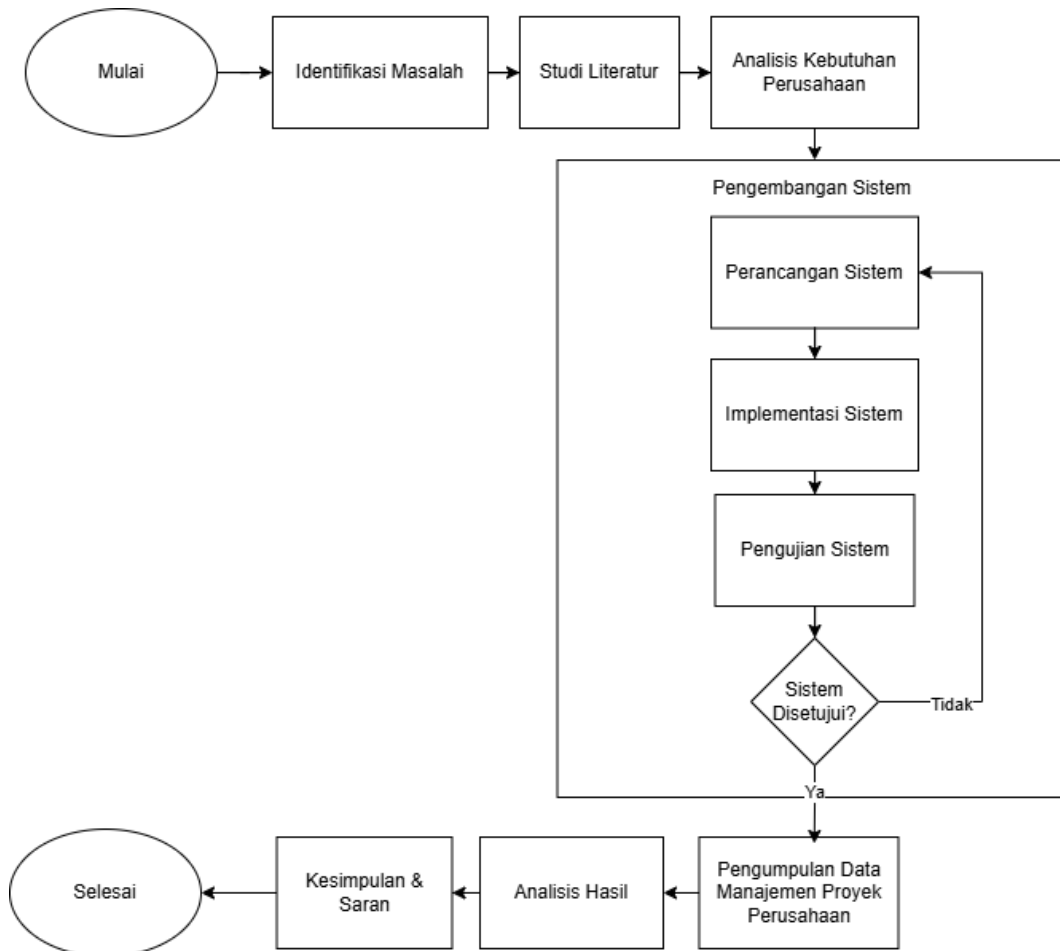
Anggota tim *scrum* pada penelitian ini dapat dilihat pada tabel berikut :

Tabel 11 *Scrum Team*

Peran	Nama
<i>Product Owner</i>	Muhammad Arsie Aziz, S.Kom
<i>Scrum Master</i>	Ahmad Rafaly
<i>Development Team</i>	Ahmad Rafaly (<i>Back End Developer</i>)
	Den Muhammad Wyzdan (<i>Front End Developer</i>)
	Arnora Mardiansyah (<i>Quality Assurance</i>)

3.4 Tahapan Penelitian

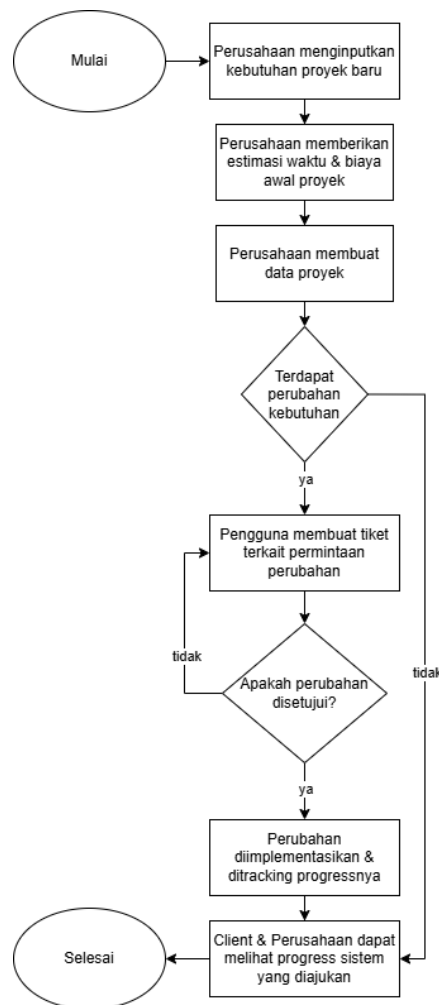
Alur mengenai Tahapan Penelitian yang dilakukan tercantum pada gambar di bawah :



Gambar 4 Alur Tahapan Penelitian

3.4.1 Analisis Kebutuhan Perusahaan

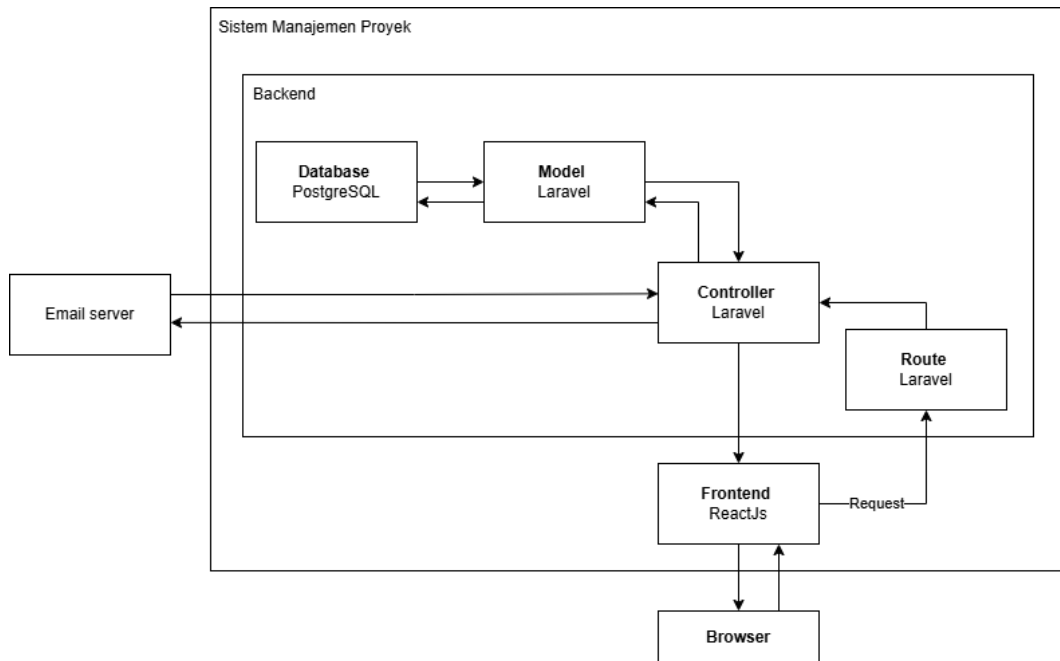
Pada tahap ini, dilakukan analisis terkait kebutuhan perusahaan terhadap sistem yang dikembangkan yaitu sistem manajemen proyek. Tahap identifikasi kebutuhan diawali dengan melakukan wawancara kepada pihak perusahaan Gink Technology untuk mengetahui alur bisnis mulai dari awal proyek dibuat, terjadi perubahan kebutuhan *client*, hingga proyek diterima oleh *client*.



Gambar 5 Alur Bisnis dari Sistem Manajemen Proyek yang dibuat

Pada Gambar di atas ditunjukkan alur bisnis dari sistem manajemen proyek yang telah disetujui oleh perusahaan Gink Technology. Proses dimulai ketika perusahaan mulai meng-*inputkan* kebutuhan yang diminta oleh klien, selanjutnya setelah kebutuhan di-*inputkan* pada sistem akan muncul estimasi waktu & biaya awal dari proyek tersebut dan data proyek dibuat. Jika terdapat perubahan kebutuhan, maka pengguna dapat membuat tiket terkait permintaan perubahan. Hasil estimasi kemudian menjadi dasar dalam proses diskusi antara Gink Technology dan klien untuk mencapai kesepakatan terkait perubahan yang diajukan. Apabila perubahan disetujui, maka tahap implementasi dilaksanakan. Namun ketika perubahan tidak disetujui, maka tidak terdapat perubahan apapun. Selama proses implementasi berlangsung, klien dapat memantau perkembangan secara berkala. Proses ini

berakhir ketika implementasi selesai dilaksanakan dan hasilnya disetujui oleh kedua belah pihak. Berdasarkan alur bisnis yang telah ditentukan, berikut adalah gambar diagram arsitektur sistem secara keseluruhan :



Gambar 6 Diagram arsitektur sistem keseluruhan

Diagram arsitektur sistem tersebut menggambarkan alur peran Quality Assurance, *backend* menggunakan Laravel, dan *frontend* yang menggunakan ReactJS. Di dalam area *Backend*, data disimpan dan dikelola pada Database PostgreSQL, kemudian diakses oleh model sebagai representasi data dan logika bisnis. Model menyediakan data kepada *controller*, yang bertugas memproses logika dan menyiapkan respon untuk permintaan yang masuk. Seluruh permintaan dari luar diarahkan terlebih dahulu ke *Route* yang berfungsi untuk menentukan *controller* mana yang harus menangani permintaan tersebut. *Frontend* yang menggunakan ReactJS berperan sebagai antarmuka pengguna yang mengirimkan *request* ke *Route* pada *backend*.

Setelah menganalisis alur bisnis dan arsitektur sistem, selanjutnya dilakukan analisis kebutuhan fungsional dan non fungsional pada sistem. Adapun kebutuhan fungsional dan non fungsional sistem, dijabarkan pada tabel di bawah ini:

Tabel 12 Kebutuhan Fungsional Sistem

Kode	Kebutuhan Fungsional	Aktor
KF-01	Melakukan <i>login</i> sistem sebagai <i>Project Manager, Developer, atau Client</i>	<i>Project Manager, Developer, Client</i>
KF-02	Membuat, merubah, dan menghapus <i>Project & Repositori</i> fitur	<i>Project Manager</i>
KF-03	Melihat <i>Dashboard Project Timeline & Progress</i>	<i>Project Manager, Developer, Client</i>
KF-04	Mengajukan perubahan fitur melalui <i>Ticket</i>	<i>Project Manager, Developer, Client</i>
KF-05	Melihat daftar <i>ticket</i> dalam proyek	<i>Project Manager, Client</i>
KF-06	Menyetujui <i>ticket</i> dan menginputkan perubahan	<i>Project Manager</i>
KF-07	Mengelola <i>Project Board (To Do, In Progress, Done)</i>	<i>Project Manager, Developer</i>
KF-08	Melihat status <i>Project Board (To Do, In Progress, Done)</i> beserta <i>Timeline</i> dan <i>Biaya</i>	<i>Project Manager, Developer, Client</i>
KF-09	Mengelola agenda <i>meeting</i> dan notulensi	<i>Project Manager, Developer, Client</i>
KF-10	Melihat notifikasi ketika ada tiket baru dan pengajuan <i>meeting</i>	<i>Project Manager, Developer, Client</i>
KF-11	Membuat & menghapus dokumen kontrak	<i>Project Manager</i>
KF-12	Mengakses halaman kontrak	<i>Project Manager, Developer, Client</i>
KF-13	Mengelola Manajemen Akun dan <i>Assign Project</i> ke masing masing akun	<i>Project Manager</i>

KF-14	Logout pengguna untuk mengakhiri sesi.	<i>Project Manager, Developer, Client</i>
-------	--	---

Tabel 13 Kebutuhan Non Fungsional Sistem

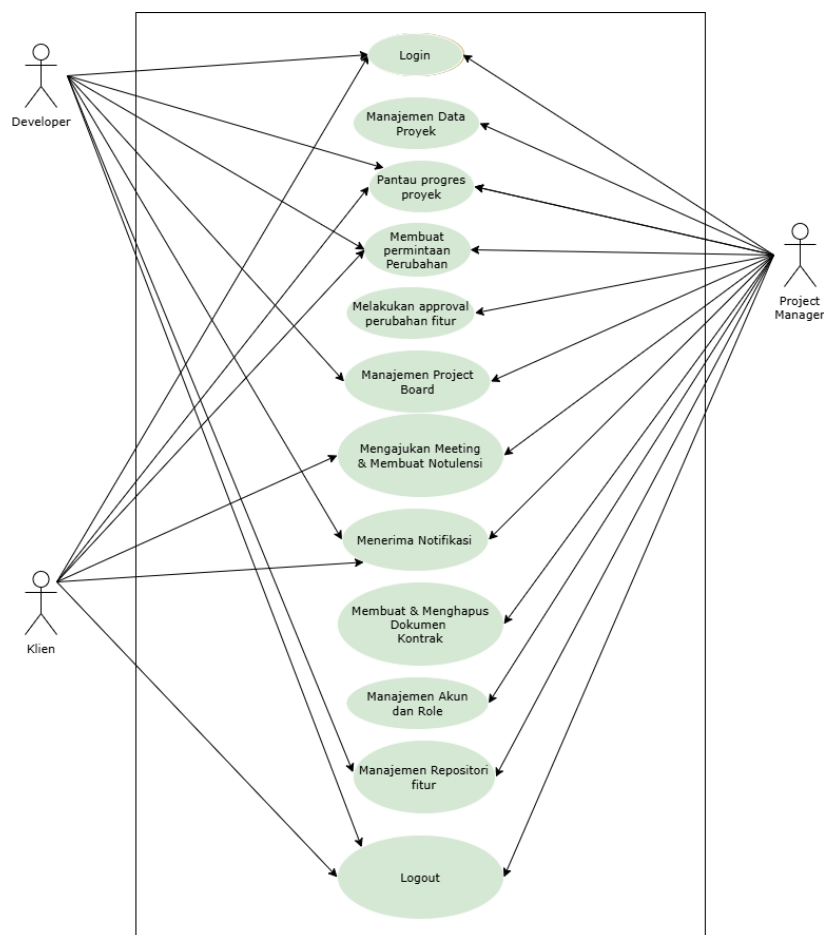
ID	Parameter	Deskripsi
KNF.01	<i>Availability</i>	Sistem harus dapat selalu diakses pada jam kerja.
KNF.02	<i>Usability</i>	Sistem harus mudah dipahami dan digunakan oleh semua peran pengguna (<i>Project Manager, Developer, Client</i>).
KNF.03	<i>Response Time</i>	Sistem harus merespon dalam waktu maksimal 3 detik pada koneksi standar saat membuka halaman.
KNF.04	<i>Portability</i>	Sistem harus dapat diakses melalui perangkat <i>desktop</i> maupun <i>mobile</i> dengan antarmuka responsif.
KNF.05	<i>Security</i>	Sistem harus menyimpan kata sandi pengguna dalam bentuk terenkripsi di <i>database</i> .
KNF.06	<i>Auditability</i>	Sistem harus menyimpan jejak audit dari semua perubahan.
KNF.07	<i>Scalability</i>	Sistem harus mampu menampung pertambahan jumlah proyek, kebutuhan, dan pengguna tanpa penurunan performa signifikan.
KNF.08	<i>Maintainability</i>	Sistem harus mudah dipelihara dan dikembangkan lebih lanjut.

3.4.2 Perancangan Sistem

Pada tahap perancangan sistem, dilakukan pemodelan untuk mendeskripsikan kebutuhan fungsional sistem. Pemodelan yang dibuat yakni UML *Use Case Diagram* dan *Activity Diagram*.

1) *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan fungsionalitas pada Sistem Manajemen Proyek yang dikembangkan. Pada sistem ini terdapat 3 aktor utama yaitu *Project Manager*, *Developer*, dan *Klien* di mana masing-masing aktor memiliki fungsionalitas berbeda yang tercantum pada gambar di bawah.



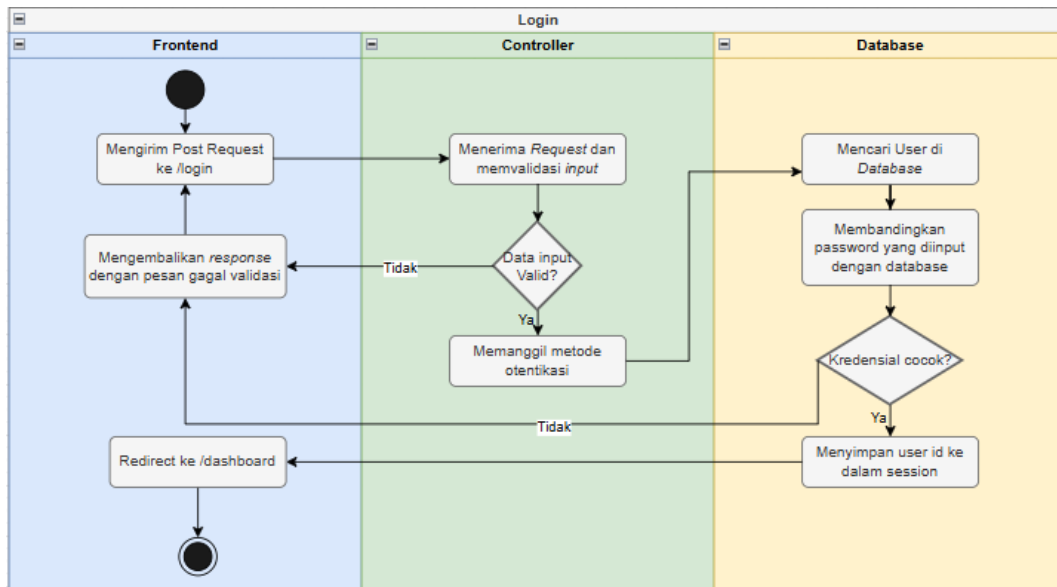
Gambar 7 *Use Case Diagram* Sistem

Project Manager memiliki hak akses untuk melakukan *login*, manajemen data proyek, memantau *progress* proyek, membuat permintaan perubahan, melakukan *approval* perubahan fitur, manajemen *Project Board*, mengajukan *meeting* dan

membuat notulensi, menerima notifikasi, membuat & menghapus dokumen kontrak, manajemen akun dan *Role*, Manajemen repositori fitur serta melakukan *logout* dari sistem. *Developer* memiliki hak akses untuk melakukan *login*, memantau *progress* proyek, manajemen *Project Board*, menerima notifikasi, serta melakukan *logout* dari sistem. Klien memiliki hak akses untuk melakukan *login*, memantau *progress* proyek, membuat permintaan perubahan, mengajukan *meeting* dan membuat notulensi, menerima notifikasi, serta melakukan *logout* dari sistem.

2) Activity Diagram

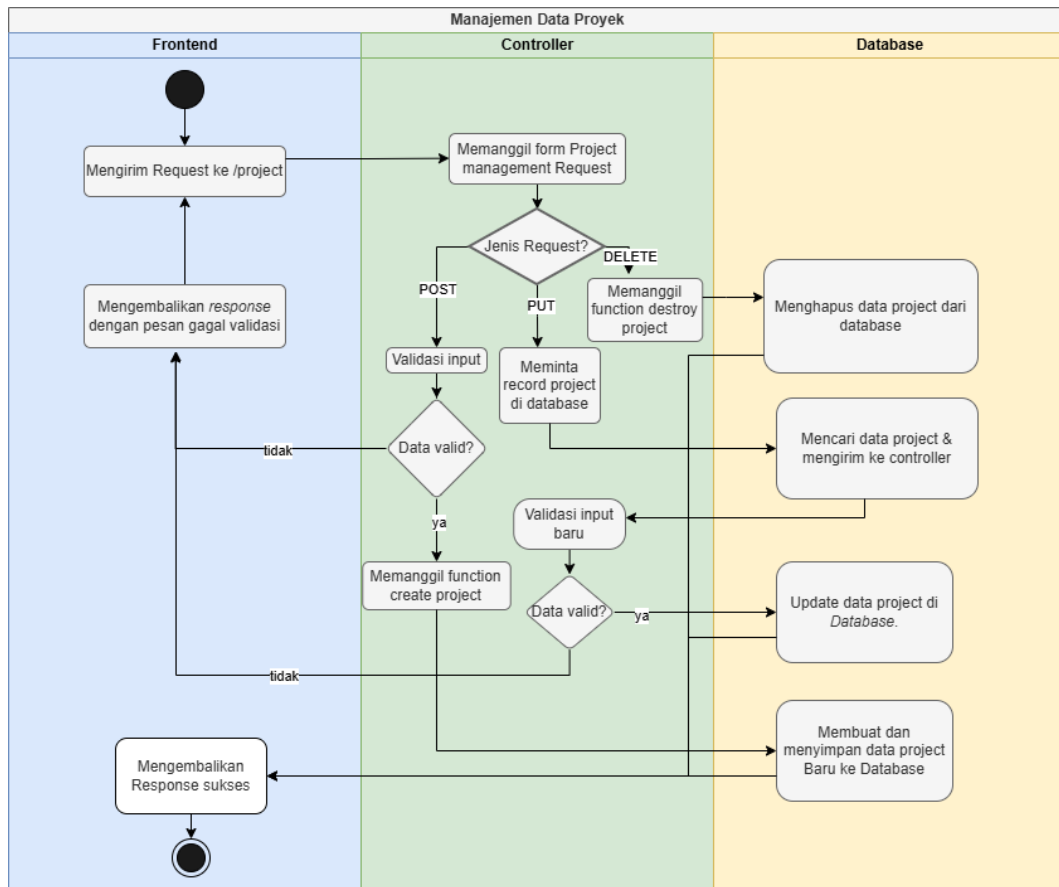
Activity diagram pada Sistem Manajemen Proyek menggambarkan alur aktivitas yang dilakukan oleh para aktor, mulai dari awal mereka masuk ke dalam sistem hingga mengakhiri sesi penggunaan.



Gambar 8 Activity Diagram Login

Berdasarkan gambar di atas, Proses dimulai saat Pengguna (Klien, *Developer*, *Project Manager*) mengirimkan *Post Request* ke */login* yang segera diterima oleh *Controller* untuk divalidasi *input*-nya. Jika *input* tidak *valid*, sistem mengembalikan pesan gagal validasi ke *Frontend*, tetapi jika *valid*, *Controller* akan memanggil metode otentikasi dan meneruskan eksekusi ke *Database* untuk mencari *User* dan membandingkan *password* yang dimasukan. Hasil perbandingan kredensial ini menentukan langkah selanjutnya yaitu jika kredensial tidak cocok, sistem kembali ke *Frontend* dengan pesan gagal validasi, namun jika kredensial cocok maka

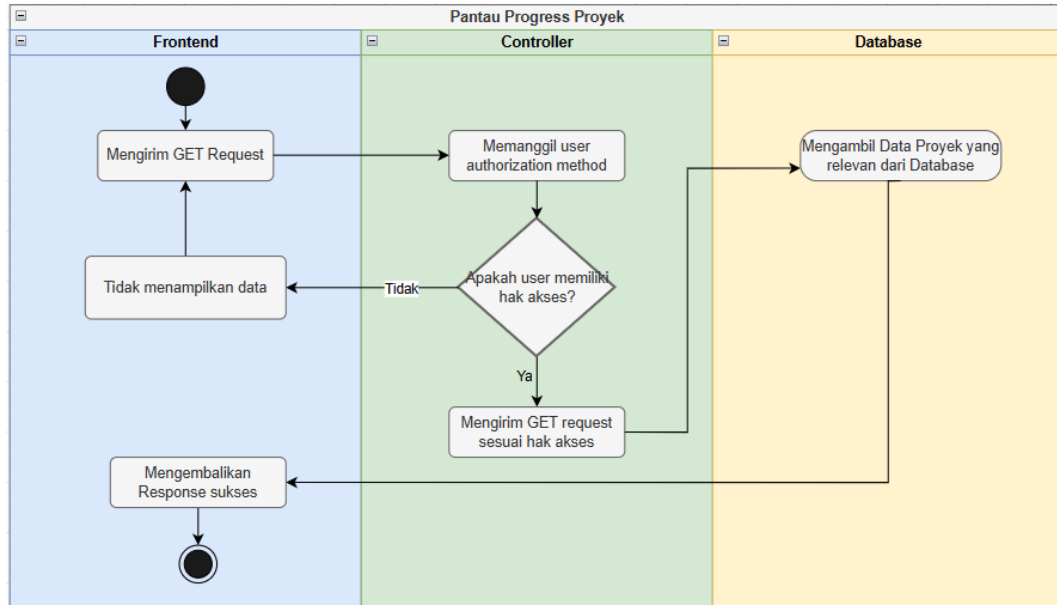
Database akan menyimpan *User id* ke dalam *session* yang kemudian memicu *Frontend* untuk *Redirect* ke */dashboard* serta menandai selesainya proses login.



Gambar 9 Activity Diagram Manajemen data proyek

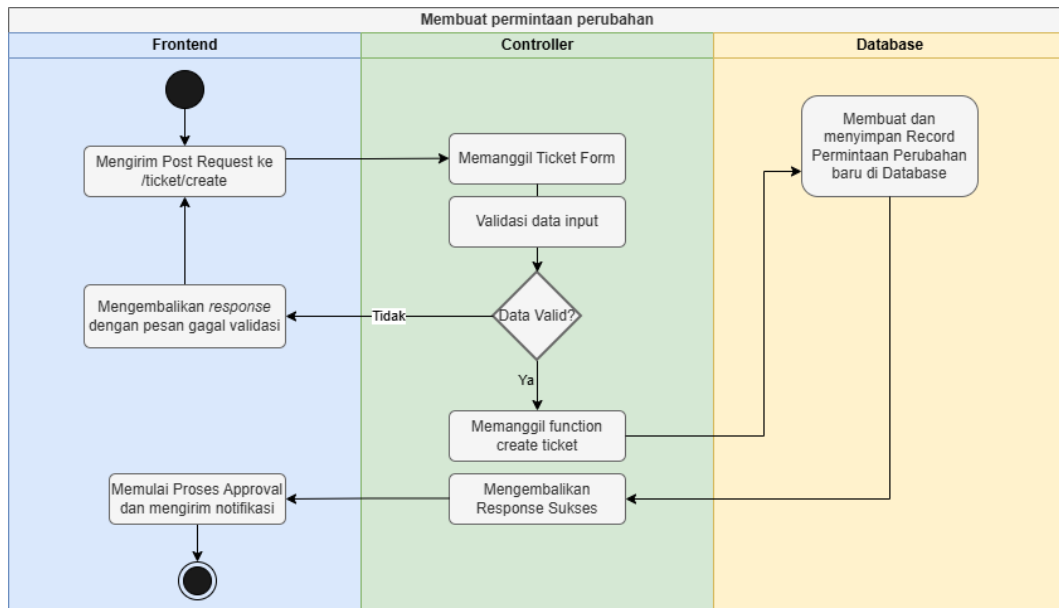
Berdasarkan gambar di atas, hanya *Project Manager* yang dapat memulai proses ini. Proses dimulai dari *Request* ke */project* yang diterima oleh *Controller* untuk memanggil *form Project management Request* kemudian mengidentifikasi Jenis Request yang masuk. Jika *Request* bertipe *POST*, *Controller* akan memvalidasi *input* jika data tidak valid, akan mengembalikan respon dengan pesan gagal validasi, namun jika valid maka *Controller* memanggil *function create project*, mengirim eksekusi ke *Database* untuk membuat dan menyimpan data *project* baru ke *Database* lalu memberikan respon sukses ke *Frontend*. Untuk *Request* bertipe *PUT*, *Controller* meminta *record project* di *database* yang dijawab oleh *Database* dengan mencari data *project* & mengirim ke *controller*. *Controller* kemudian memvalidasi *input* baru dan jika valid, *Database* akan *Update* data *project* di *Database* diikuti oleh respon sukses. Terakhir untuk *Request* bertipe *DELETE*,

Controller langsung memanggil *function destroy project* yang menginstruksikan *Database* untuk menghapus data *project* dari *database* dan *Controller* memberikan Respon sukses ke *Frontend*.



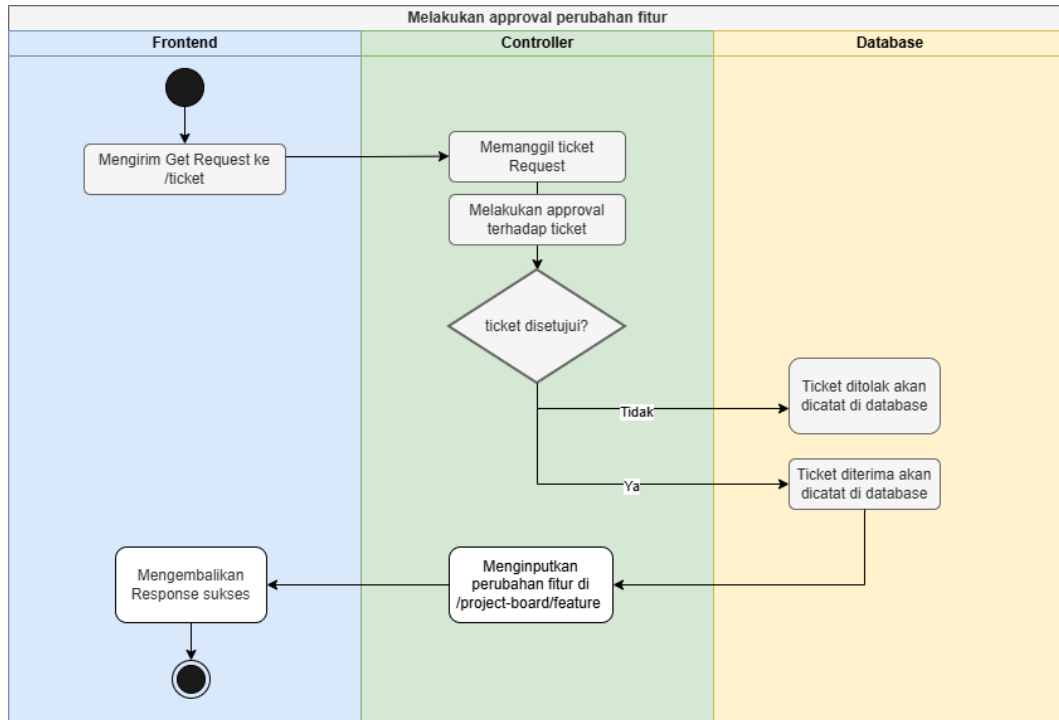
Gambar 10 Activity Diagram Pantau progress proyek

Berdasarkan gambar di atas, Semua Pengguna yaitu *Project Manager*, *Developer*, dan *Klien* dapat memantau perkembangan proyek. Proses dimulai di *Frontend* dengan pengguna mengirim *GET Request* ke sistem yang kemudian diterima oleh *Controller* yang segera memanggil *User authorization method* untuk memeriksa izin akses pengguna. Pemeriksaan ini menjadi titik keputusan yaitu jika *User* tidak memiliki hak akses, *Controller* menginstruksikan *Frontend* untuk tidak menampilkan data, namun jika *User* memiliki hak akses maka *Controller* akan Mengirim *GET request* sesuai hak akses pengguna dan meneruskan eksekusi ke *Database*. *Database* kemudian bertugas mengambil data proyek yang relevan dari *Database* berdasarkan hak akses tersebut. Setelah data berhasil diambil, sistem mengirimkannya kembali ke *Frontend* yang kemudian mengembalikan respon sukses untuk menampilkan data progress proyek kepada pengguna.



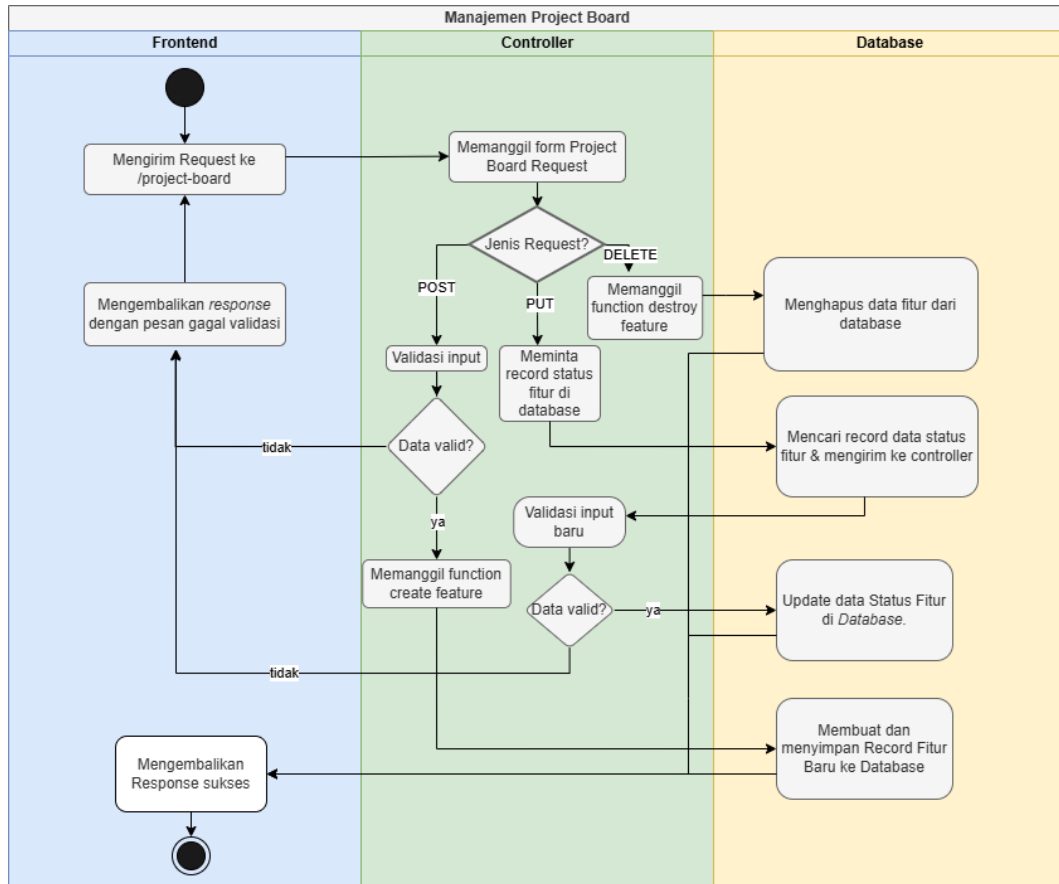
Gambar 11 Activity Diagram Membuat permintaan perubahan

Berdasarkan gambar di atas, Proses ini dapat dilakukan oleh *Project Manager*, *Developer* dan *Client*. Proses diawali di *Frontend* dengan pengguna mengirim *Post Request* ke */ticket/create* yang diterima oleh *Controller* dan segera memanggil *ticket form* serta melakukan validasi data *input*. Validasi ini menjadi titik keputusan yaitu jika data tidak *valid*, *Controller* menginstruksikan *Frontend* untuk mengembalikan respon dengan pesan gagal validasi, namun jika data *valid* maka *Controller* akan memanggil *function create ticket* dan meneruskan eksekusi ke *Database* untuk membuat dan menyimpan *record* permintaan perubahan baru di *Database*. Setelah proses penyimpanan berhasil, *Controller* mengembalikan respon sukses ke *Frontend* yang kemudian melanjutkan proses ke *approval* dan mengirim notifikasi.



Gambar 12 Activity Diagram Melakukan Approval Perubahan Fitur

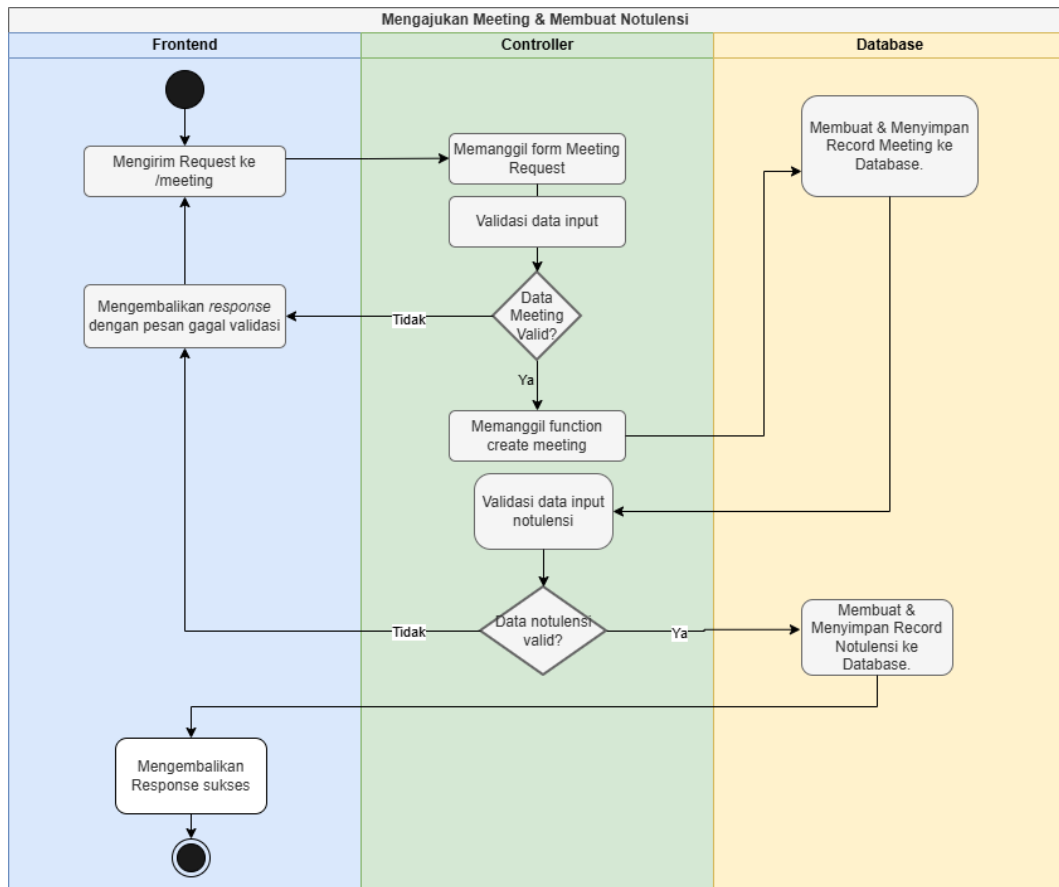
Berdasarkan gambar di atas, Pengguna dengan hak akses *Project Manager* mengirim *Post Request* ke */ticket* yang langsung diterima oleh *Controller* dan bertugas memanggil *ticket request* dan melakukan *approval* terhadap *ticket*. Jika *ticket* disetujui, tiket akan dicatat di *database* dan selanjutnya menginputkan perubahan fitur di */project-board/feature* serta mengembalikan respon sukses, namun jika tiket tidak disetujui, hasilnya juga dicatat di *database* tetapi tidak ditambahkan ke */project-board/feature*.



Gambar 13 Activity Diagram Manajemen *project board*

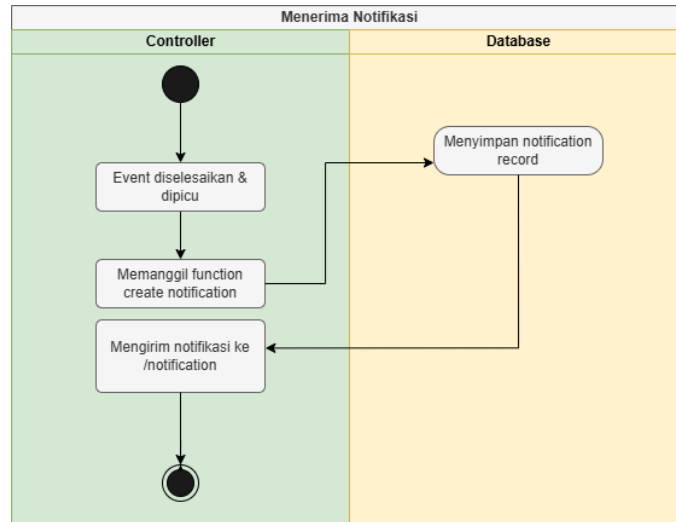
Berdasarkan gambar di atas, proses manajemen *project board* dimulai di *Frontend* dengan pengguna mengirim *request* ke */project-board* yang diterima oleh *Controller* untuk memanggil *form Project Board Request*. Selanjutnya *Controller* akan menentukan *Jenis Request* yang masuk di mana terdapat tiga alur utama yaitu untuk *Request* bertipe *POST*, *Controller* melakukan validasi *input*, jika data tidak valid, sistem kembali ke *Frontend* untuk mengembalikan respon dengan pesan gagal validasi, namun jika data valid maka *Controller* akan memanggil *function create feature* dan mengirim eksekusi ke *Database* untuk membuat dan menyimpan *record* fitur baru ke *Database*, lalu kembali memberikan respon sukses ke *Frontend*. Untuk *Request* bertipe *PUT*, *Controller* akan meminta *record* status fitur di *database* yang akan dijawab oleh *Database* dengan Mencari *record* data status fitur & mengirim ke *controller*. *Controller* kemudian melakukan *Validasi input*, dan jika valid, *Database* akan *Update* data Status Fitur di *Database*. Sementara untuk *Request* bertipe *DELETE*, *Controller* langsung memanggil *function destroy*

feature yang menginstruksikan *Database* untuk menghapus data fitur dari *database*, dan memberikan respon sukses ke *Frontend*.



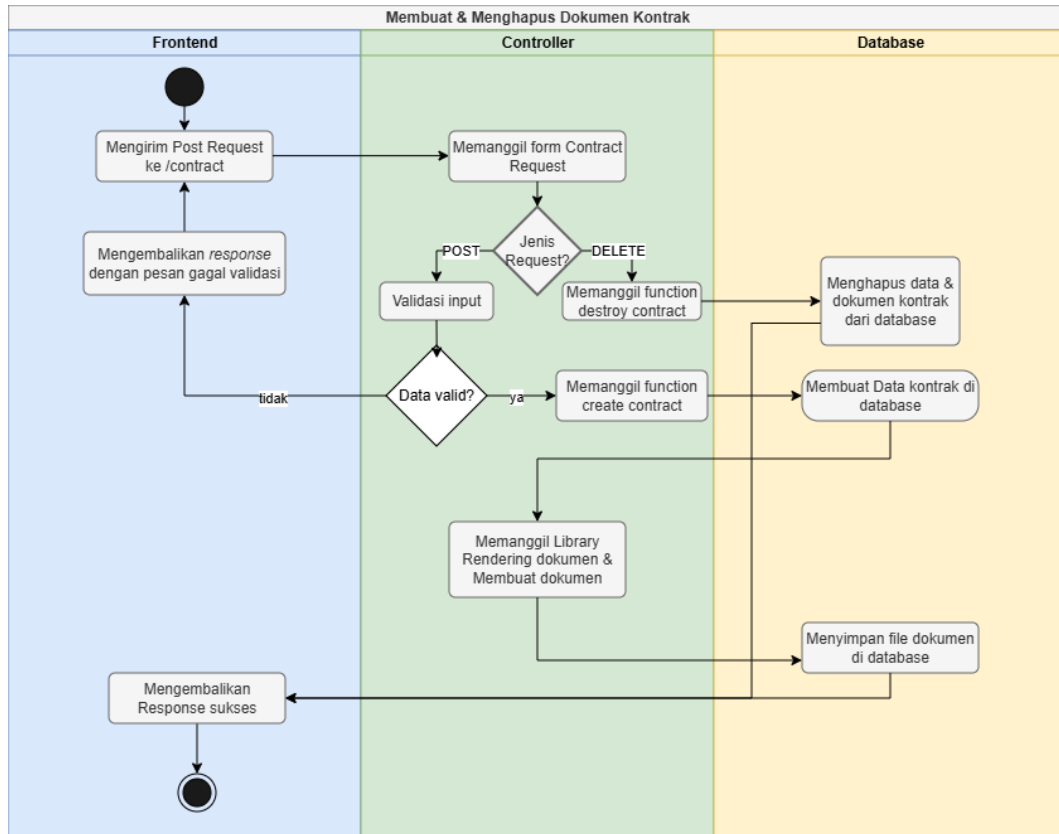
Gambar 14 Activity Diagram Mengajukan meeting & membuat notulensi

Berdasarkan gambar di atas, Pengguna yaitu *Project Manager, Developer, Client* dapat mengirim *request* ke */meeting* yang diterima oleh *Controller* untuk memanggil *form meeting request* dan melakukan validasi data *input meeting*. Jika data *meeting* tidak valid, *Controller* mengembalikan respon dengan pesan gagal validasi, namun jika data *meeting* valid, *Controller* memanggil *function create meeting* yang mengirim eksekusi ke *Database* untuk membuat & menyimpan *record meeting* ke *Database*. Setelah data *meeting* tersimpan, *Controller* melakukan Validasi data *input*. Pada validasi kedua ini, jika data notulensi tidak valid, sistem mengembalikan pesan gagal validasi, namun jika data notulensi valid, *Controller* menginstruksikan *Database* untuk membuat & menyimpan *record notulensi*. Setelah proses penyimpanan berhasil, *Controller* memberikan respon sukses ke *Frontend*.



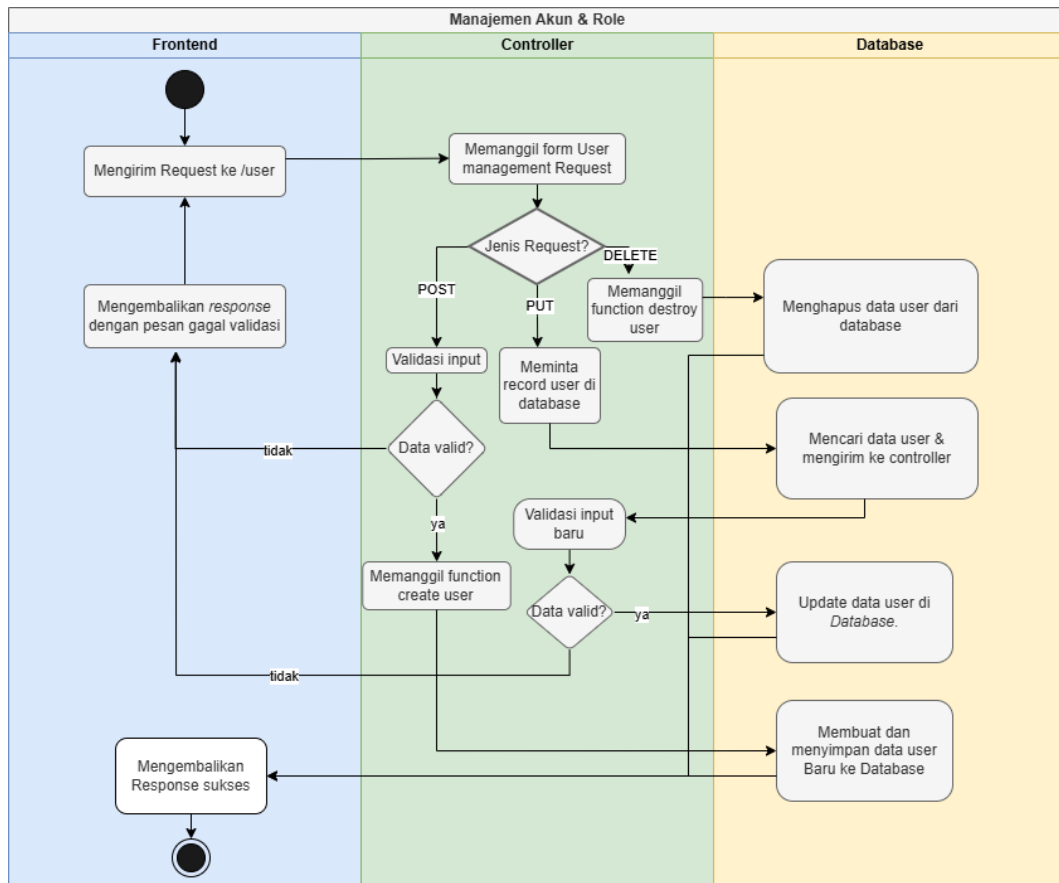
Gambar 15 *Activity Diagram* Menerima notifikasi

Berdasarkan gambar di atas, Semua Pengguna dapat menerima notifikasi dimulai pada lapisan *Controller* ketika suatu *Event* diselesaikan & dipicu. Setelah *event* dipicu, *Controller* akan Memanggil *function create notification* yang kemudian meneruskan eksekusi ke lapisan *Database* untuk menyimpan *notification record*. Setelah *record* notifikasi berhasil disimpan di *Database*, proses kembali ke *Controller* untuk mengirim notifikasi ke */notification*, yang menandai selesainya alur proses penerimaan notifikasi.



Gambar 16 Activity Diagram Membuat & menghapus dokumen kontrak

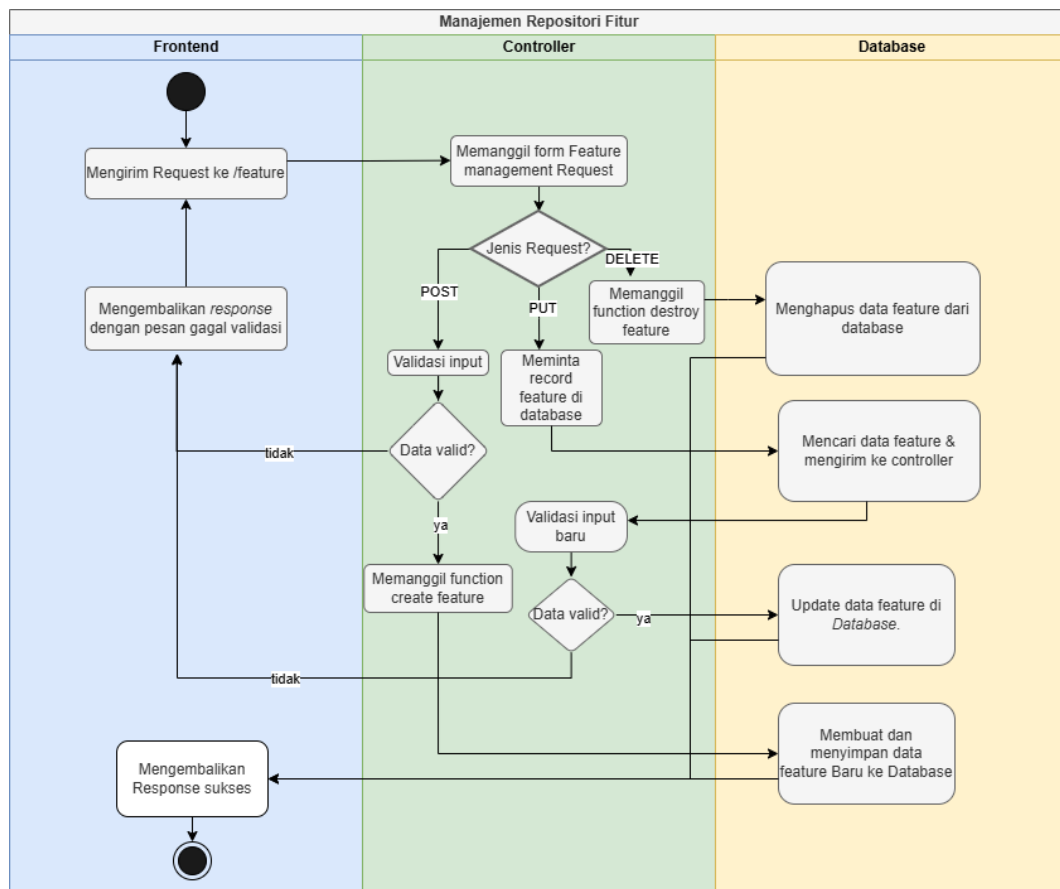
Berdasarkan gambar di atas, proses ini dijalankan oleh *Project Manager*. Proses dimulai di *Frontend* dengan pengguna mengirim *Post Request* ke */contract* yang diterima oleh *Controller* untuk memanggil *form Contract Request*. Langkah selanjutnya menentukan jenis *request* yang diterima. Jika menerima *request POST* maka akan memvalidasi *input* dari pembuatan kontrak, dan mengecek validitas data yaitu jika data tidak valid, *Controller* menginstruksikan *Frontend* untuk mengembalikan *response* dengan pesan gagal validasi, namun jika data valid maka *Controller* akan Memanggil *function create contract* yang segera mengirim eksekusi ke *Database* untuk membuat data kontrak di *database*. Setelah data kontrak disimpan, *Controller* memanggil *Library Rendering dokumen & Membuat dokumen* dan selanjutnya meminta *Database* untuk Menyimpan *file* dokumen di *database*. Setelah dokumen berhasil dibuat dan disimpan, *Controller* memberikan response sukses ke *Frontend*. Jika yang diterima adalah *request DELETE* maka *Controller* akan memanggil *function destroy contract* dan menghapus data serta dokumen kontrak dari *database*.



Gambar 17 Activity Diagram Manajemen akun & Role

Berdasarkan gambar di atas, proses ini khusus dilakukan oleh Pengguna dengan peran *Project Manager*. Proses diawali di *Frontend* dengan pengguna mengirim *Request* ke */User* yang diterima oleh *Controller* untuk memanggil *form User management Request* kemudian mengidentifikasi Jenis Request yang masuk. Jika *Request* bertipe *POST*, *Controller* akan memvalidasi *input* jika data tidak valid, akan mengembalikan respon dengan pesan gagal validasi, namun jika valid maka *Controller* memanggil *function create User*, mengirim eksekusi ke *Database* untuk membuat dan menyimpan data *User* baru ke *Database* lalu memberikan respon sukses ke *Frontend*. Untuk *Request* bertipe *PUT*, *Controller* meminta *record User* di *database* yang dijawab oleh *Database* dengan mencari data *User* & mengirim ke *controller*. *Controller* kemudian memvalidasi *input* baru dan jika valid, *Database* akan *Update* data *User* di *Database* diikuti oleh respon sukses. Terakhir untuk *Request* bertipe *DELETE*, *Controller* langsung memanggil *function destroy User*

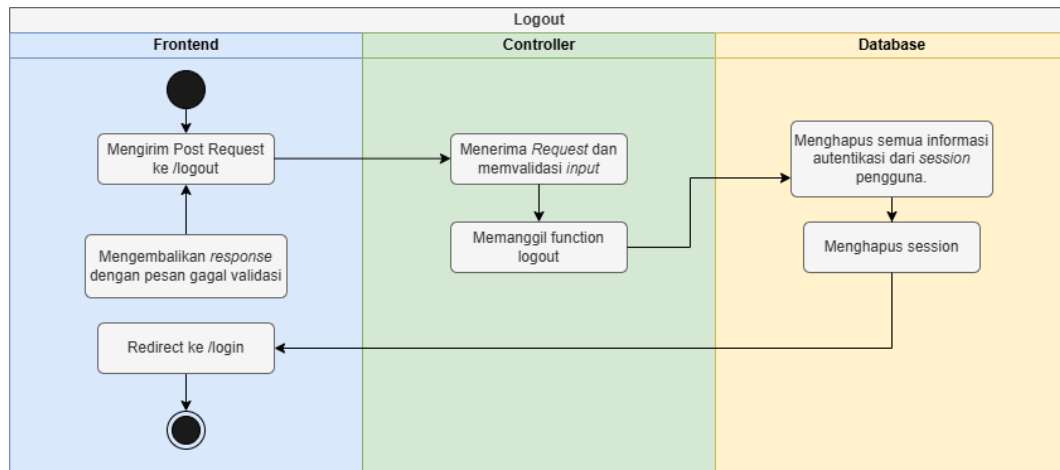
yang menginstruksikan *Database* untuk menghapus data *User* dari *database* dan *Controller* memberikan Respon sukses ke *Frontend*.



Gambar 18 Activity Diagram Manajemen Repositori Fitur

Berdasarkan gambar di atas, proses ini dilakukan oleh Pengguna dengan peran *Project Manager & Developer*. Proses diawali di *Frontend* dengan pengguna mengirim *Request* ke */Feature* yang diterima oleh *Controller* untuk memanggil *form Feature management Request* kemudian mengidentifikasi Jenis Request yang masuk. Jika *Request* bertipe *POST*, *Controller* akan memvalidasi *input* jika data tidak valid, akan mengembalikan respon dengan pesan gagal validasi, namun jika valid maka *Controller* memanggil *function create Feature*, mengirim eksekusi ke *Database* untuk membuat dan menyimpan data *Feature* baru ke *Database* lalu memberikan respon sukses ke *Frontend*. Untuk *Request* bertipe *PUT*, *Controller* meminta *record Feature* di *database* yang dijawab oleh *Database* dengan mencari data *Feature &* mengirim ke *controller*. *Controller* kemudian memvalidasi *input* baru dan jika valid, *Database* akan *Update* data *Feature* di *Database* diikuti oleh

respon sukses. Terakhir untuk *Request* bertipe *DELETE*, *Controller* langsung memanggil *function destroy Feature* yang menginstruksikan *Database* untuk menghapus data *Feature* dari *database* dan *Controller* memberikan Respon sukses ke *Frontend*.



Gambar 19 Activity Diagram Logout

Berdasarkan gambar di atas, Semua Pengguna dapat melakukan proses ini. Proses dimulai di *Frontend* ketika pengguna mengirim *Post Request* ke */logout*, yang segera diterima oleh *Controller* untuk Menerima *Request* dan memvalidasi *input*. Jika *input* tidak valid, sistem akan Mengembalikan *response* dengan pesan gagal validasi ke *Frontend*, namun jika *input* valid, *Controller* akan memanggil *function logout* yang mengirim eksekusi ke lapisan *Database* untuk menghapus informasi autentikasi dari *session* pengguna. Setelah *session* dihapus, proses kembali ke *Frontend* yang kemudian melakukan *redirect* ke */login*.

3.4.3 Implementasi Sistem

Pada tahap implementasi sistem, dilakukan pengembangan sistem berbasis *website* menggunakan *Laravel* dan *database PostgreSQL* dengan fokus *backend* menggunakan metode pengembangan *Scrum* pada sistem manajemen proyek. Pemilihan *Laravel* dan *PostgreSQL* sebagai *tools* yang digunakan pada sistem ini, karena didasarkan pada kebutuhan perusahaan *Gink Technology*. *Laravel* dan *PostgreSQL* adalah *tools* yang paling sering digunakan pada proses pengembangan perangkat lunak dalam perusahaan tersebut sehingga jika suatu saat terjadi *bug* atau

error pada sistem, perusahaan tersebut mampu dengan mudah memperbaiki sistem itu kembali. Implementasi sistem menggunakan beberapa *tools* pendukung seperti Visual Studio Code berfungsi sebagai *editor* kode utama dalam proses penulisan dan pengelolaan kode program. Laragon dimanfaatkan sebagai lingkungan pengembangan lokal yang menyediakan layanan *web server*, PHP, serta basis data PostgreSQL. Bahasa pemrograman yang digunakan adalah PHP versi 8.2.23, dengan *framework* Laravel versi 12 yang menerapkan arsitektur *Model-View-Controller* (MVC) sehingga mempercepat pengembangan sistem. Proses ini juga dilengkapi dengan penggunaan GitHub sebagai *platform version control* untuk mengelola kode secara terstruktur. Sementara itu, Draw.io dimanfaatkan dalam pembuatan *business flow*, diagram UML, ERD, serta dokumentasi yang diperlukan dalam skripsi.

3.4.4 Pengujian Sistem

Pada tahap pengujian sistem, dilakukan proses *blackbox testing* untuk mengevaluasi fungsionalitas sistem dengan cara membandingkan kesesuaian antara *input* dan *output* dari sistem. Metode yang digunakan adalah pendekatan *use case*, sehingga penyusunan *test case* didasarkan pada skenario *use case* yang telah dirancang sebelumnya. Melalui penerapan pengujian *blackbox* ini, dapat dipastikan bahwa sistem yang dikembangkan berfungsi dengan baik serta sesuai dengan kebutuhan pengguna.

Selain pengujian terhadap fungsional sistem, terdapat juga pengujian *Unit Testing* untuk memastikan setiap fungsi pada sistem dapat berjalan sesuai dengan yang diharapkan. *Unit Testing* dipilih agar sistem dapat diverifikasi bahwa perhitungan biaya dan waktu yang dihasilkan oleh sistem sesuai dengan rumus dan ketentuan yang telah ditentukan. Pengujian dilakukan menggunakan *feature* bawaan Laravel yaitu PHPUnit, jika hasil *output* sesuai dengan yang diharapkan, maka fungsi tersebut telah memenuhi kriteria fungsionalitas yang benar.

3.4.5 Pengumpulan Data Proyek Manajemen Perusahaan

Penelitian ini menggunakan dua jenis sumber data, yaitu:

1) Data Primer, berupa data yang diperoleh secara langsung dari perusahaan Gink melalui wawancara dengan *Project Manager* dan *developer*, serta melalui penggunaan Sistem Manajemen Proyek yang dibangun dalam penelitian ini. Data primer meliputi hasil perhitungan waktu pengembangan, biaya, dokumentasi perubahan kebutuhan, dan hasil pengujian sistem.

2) Data Sekunder, berupa data historis proyek sebelum sistem diterapkan, meliputi identitas proyek, laporan keuangan proyek, catatan perubahan kebutuhan, serta dokumen teknis proyek terdahulu yang disediakan oleh perusahaan.

Teknik pengumpulan data yang digunakan dalam penelitian ini meliputi:

1) Studi Dokumen

Mengumpulkan data historis proyek sebelum sistem diterapkan, termasuk laporan biaya proyek, durasi penyelesaian, serta data perubahan kebutuhan yang pernah terjadi. Data ini digunakan sebagai landasan untuk dibandingkan dengan data hasil setelah penerapan sistem.

2) Observasi Langsung

Mengamati langsung penggunaan Sistem Manajemen Proyek yang dibangun, khususnya dalam proses sistem dalam melakukan pencatatan perubahan kebutuhan dan perhitungan biaya, serta waktu pengembangan.

2) Menghitung karakteristik sistem menggunakan TDI

Tahap ini digunakan untuk menilai kompleksitas sistem secara keseluruhan berdasarkan 14 faktor umum sistem, Setiap faktor diberi skor 0–5 tergantung tingkat pengaruhnya terhadap sistem.

Tabel 15 Perhitungan TDI [7]

No	Karakteristik	Nilai skala
1	Tingkat kompleksitas komunikasi data	
2	Tingkat kompleksitas pemrosesan terdistribusi	
3	Tingkat kompleksitas performa	
4	Tingkat kompleksitas konfigurasi	
5	Tingkat frekuensi penggunaan perangkat lunak	
6	Tingkat frekuensi <i>input</i> data	
7	Tingkat kemudahan penggunaan bagi pengguna	
8	Tingkat frekuensi perbarui data	
9	Tingkat kompleksitas pemrosesan data	
10	Tingkat <i>reusable code</i> program	
11	Tingkat kemudahan dalam instalasi	
12	Tingkat kemudahan operasional perangkat lunak	
13	Tingkat perangkat lunak dibuat untuk <i>multi</i> organisasi/perusahaan/ <i>client</i>	
14	Tingkat kompleksitas dalam mengadaptasi perubahan	
Total nilai TDI		

3) Menghitung Nilai *Function Point*

Setelah memperoleh nilai UFP dan TDI, langkah selanjutnya yaitu menghitung nilai *Function Point* (FP) akhir menggunakan rumus:

$$FP = UFP \times ((TDI \times 0.01) + 0.65) \quad (3)$$

4) Menghitung Estimasi Biaya dan Waktu menggunakan nilai dari *Function Point*

Nilai FP yang telah diperoleh selanjutnya digunakan untuk dimasukkan ke dalam rumus *Software Cost Estimation* [11] untuk menghitung estimasi biaya pengembangan dan waktu pengerjaan proyek yaitu sebagai berikut :

$$E = \left\{ Nw \times \left(\frac{factor \times FP}{n} \right) \right\} [11]$$

E = Perkiraan waktu pengembangan (jam)

Nw = Waktu pengerjaan dalam sehari

$Factor$ = Baris kode pada perangkat lunak per *Function Point*

FP = *Function Point*

n = Jumlah baris kode per hari

Persamaan untuk menghitung estimasi biaya yaitu sebagai berikut :

$$\Delta C_D = E \times W_p [11]$$

ΔC_D = Estimasi biaya

E = Perkiraan waktu pengembangan (jam)

W_p = Biaya pengembangan per jam

3.6 Perhitungan *Function Point* pada Perubahan Fitur

Setelah melakukan diskusi dengan pihak perusahaan Gink Technology, untuk mengukur dampak perubahan fitur terhadap estimasi biaya dan waktu, dilakukan perhitungan *Function Point* hanya pada UFP bagian yang terpengaruh.

Apabila terdapat penambahan fitur, maka komponen *Function Point* yang baru akan dihitung dan ditambahkan pada total FP sistem. Sebaliknya, apabila terdapat penghapusan fitur, maka nilai FP dari komponen yang dihapus akan dikurangi dari

total FP sistem. Sementara itu, apabila terjadi perubahan pada fitur yang sudah ada, maka dilakukan perhitungan ulang terhadap kompleksitas fungsi tersebut untuk memperoleh selisih nilai FP antara kondisi sebelum dan sesudah perubahan.

Perubahan nilai *Function Point* akibat adanya modifikasi fitur disebut sebagai *Delta Function Point* (ΔFP), yang merepresentasikan besarnya peningkatan atau penurunan ukuran sistem berdasarkan fungsi yang berubah.

Secara matematis, total *Function Point* setelah perubahan dapat dinyatakan dengan rumus berikut:

$$FP_{total} = FP_{awal} + \sum (\Delta FP_{fitur})$$

Nilai Akhir *Function Point* diperoleh setelah memperhitungkan seluruh perubahan fungsi yang terjadi pada sistem. Nilai akhir ini yang menjadi estimasi akhir biaya dan waktu dalam pengerjaan suatu proyek.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa kesimpulan yaitu sebagai berikut :

1. Sistem yang dibangun telah mengimplementasikan *Requirement Change Management Process* sehingga dapat digunakan untuk menangani setiap perubahan kebutuhan yang muncul. Setiap perubahan disimpan dan disertai informasi pengguna dan deskripsi modifikasi yang dilakukan karena fitur ini berperan penting dalam mendukung prinsip *traceability* dalam manajemen proyek.
2. Berdasarkan hasil implementasi *Function Point Analysis* pada sistem, estimasi biaya dan waktu yang dibutuhkan untuk membangun sistem X dan Y terbukti mampu melakukan perhitungan dengan akurasi rata-rata biaya yaitu 89% dan waktu mencapai 84%. Hasil estimasi awal ini dapat membantu pengguna dalam menyusun perencanaan proyek yang lebih realistis dan terukur sejak tahap awal pengembangan.
3. Berdasarkan hasil implementasi *function point analysis* pada sistem, sistem berhasil menghasilkan estimasi dampak perubahan kebutuhan pada sistem X dan Y dalam aspek biaya dengan tingkat akurasi mencapai 85% dan waktu mencapai 86%. Hasil ini menunjukkan bahwa *Function Point* yang diterapkan terbukti dalam memberikan hasil estimasi terhadap kondisi proyek aktual sehingga pengambilan keputusan dapat dilakukan lebih cepat.
4. Pengujian yang dilakukan menggunakan metode *blackbox* dan *unit testing* menunjukkan bahwa seluruh fungsi dalam sistem berjalan dengan baik serta sudah sesuai dengan ekspektasi. Hal ini menandakan bahwa setiap fungsi telah beroperasi sesuai kebutuhan pengguna tanpa ditemukan kesalahan logika maupun kesalahan *input - output*.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang bisa dilakukan untuk penelitian selanjutnya yaitu :

1. Pada penelitian ini, sistem masih dibangun secara monolitik sehingga seluruh proses berjalan dalam satu kesatuan *website*. Untuk penelitian selanjutnya, disarankan agar sistem dikembangkan dengan menambahkan *Application Programming Interface (API)*. Pembuatan API akan memudahkan sistem jika di kemudian hari ingin diintegrasikan dengan sistem lain.
2. Penelitian berikutnya disarankan untuk melakukan perbandingan estimasi biaya dan waktu dengan data proyek aktual yang lebih banyak dan lengkap agar penyesuaian standar pengisian UFP bisa menghasilkan akurasi estimasi yang lebih tinggi.
3. Pihak Gink Technology disarankan untuk melakukan pelatihan penggunaan sistem terutama di bagian analisis *Unadjusted Function Points* pada tiap - tiap fitur agar estimasi biaya dan waktu yang dihasilkan bisa optimal.

DAFTAR PUSTAKA

- [1] S. Jayatilleke dan R. Lai, “A systematic review of requirements change management,” *Journal Information and Software Technology*, vol. 93, hlm. 163–185, 2017, doi: 10.1016/j.infsof.2017.09.004.
- [2] A. J Albrecht dan J. E Gaffney, “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation,” *IEEE Transactions On Software Engineering*, vol. 9, no. 6, hlm. 639–648, 1983.
- [3] ISO 21502, “Project, programme and portfolio management-Guidance on project management Management,” 2020.
- [4] Gink Technology, *Company Profile Gink Technology*. Gink Technology, 2024.
- [5] I. Andika, S. Nevile, R. Satya, dan A. Farisi, “Analisis Sistem Informasi Manajemen Proyek: Systematic Literature Review Information System Management Project Analysis : Systematic Literature Review,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 11, no. 1, hlm. 220–230, 2024, [Daring]. Tersedia pada: <http://jurnal.mdp.ac.id>
- [6] N. Mehmood Minhas, Q. Ul ain, Z. Ul Islam, dan A. Zulfiqar, “An Improved Framework for Requirement Change Management in Global Software Development,” *Journal of Software Engineering and Applications*, vol. 07, no. 09, hlm. 779–790, 2014, doi: 10.4236/jsea.2014.79072.
- [7] IFPUG, “Function point counting practices manual, release 4.3.1,” USA, 2010.
- [8] W. B. Wicaksono, F. Pradana, dan B. Priyambadha, “Pembangunan Sistem Perubahan Kebutuhan Menggunakan Improved Framework,” *Jurnal*

- Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 5, hlm. 1947–1953, 2018, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [9] R. Parlika, A. Maghfur, A. Deka, B. Atingsari, dan S. Marsya, “Panduan Pengukuran Perangkat Lunak Metode Function Point Serta Implementasinya pada Website Pemesanan Tiket Bus,” *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 5, hlm. 1–12, 2023.
- [10] C. Jones, “Software Economics and Function Point Metrics: Thirty years of IFPUG Progress,” Namcook Analytics LLC, United State, 2017. Diakses: 14 Januari 2026. [Daring]. Tersedia pada: <https://www.ifpug.org/wp-content/uploads/2017/04/IYSM.-Thirty-years-of-IFPUG.-Software-Economics-and-Function-Point-Metrics-Capers-Jones.pdf>
- [11] N. Ali dan R. Lai, “A method of requirements change management for global software development,” *Inf. Softw. Technol.*, vol. 70, hlm. 49–67, 2016.
- [12] Boulanger, “Management of a Software Application’s Versions,” dalam *Certifiable Software Applications 2*, ISTE Press - Elsevier, 2017, hlm. 67–86.
- [13] R. Predsman, *Software Engineering A Practitioner’s Approach 7th Edition*. United State: Mc Graw Hill Companies, 2010.
- [14] R. Swastika, *Buku Ajar Pemodelan Sistem Berbasis Objek Beserta Studi Kasus*. Jakarta: Universitas Bina Sarana Informatika, 2024. Diakses: 5 November 2025. [Daring]. Tersedia pada: https://repository.bsi.ac.id/repo/files/443894/download/A5_Rahayu-Swastika_Buku-Ajar-full.pdf
- [15] P. O. Neil, *Database: Principles Programming Performance*. San Francisco: Morgan Kaufmann, 1994.
- [16] IBM, “Apa itu PostgreSQL?,” IBM Think Topic. Diakses: 16 September 2025. [Daring]. Tersedia pada: <https://www.ibm.com/id-id/think/topics/postgresql>

- [17] L. Welling dan L. Thomson, *PHP and MySQL Web development*, Second Edition. Indianapolis: Sams Publishing, 2003.
- [18] T. Bin Tahir, M. Rais, dan M. A. Hs, “Aplikasi Point OF Sales Menggunakan Framework Laravel Point OF Sales Appilaction using Laravel Framework,” *Jurnal Informatika dan Ilmu Komputer (JIKO)*, vol. 2, no. 2, hlm. 55–59, 2019, doi: 10.33387/jiko.
- [19] M. Agustín dan I. Benedetti, “Matorral,” Github. Diakses: 22 November 2025. [Daring]. Tersedia pada: <https://github.com/matorral-project/matorral>
- [20] L. Dhoundiyal, “Project Management System,” Github. Diakses: 22 November 2025. [Daring]. Tersedia pada: <https://github.com/LakshayD02/Project-Management-System>
- [21] Atlassian, “Jira, Best Project Management Tool,” 2024, *Sydney*. Diakses: 16 Januari 2026. [Daring]. Tersedia pada: <https://www.atlassian.com/software/jira>
- [22] Atlassian, “About Trello,” 2024. Diakses: 16 Januari 2026. [Daring]. Tersedia pada: <https://trello.com/about>
- [23] A. Yhurinda, P. Putri, dan A. P. Subriadi, “Software Cost Estimation Using Function Point Analysis,” dalam *International Seminar on Science and Technology*, Surabaya: Postgraduate Program Institut Teknologi Sepuluh Nopember, 2018, hlm. 79–83.
- [24] N. Rachmat, “Estimasi Ukuran Perangkat Lunak Menggunakan Function Point Analysis-Studi Kasus Aplikasi Pengujian dan Pembelajaran Berbasis Web,” dalam *Seminar ilkom unsri*, Universitas Sriwijaya, 2017. [Daring]. Tersedia pada: <https://www.researchgate.net/publication/320825809>
- [25] A. Satrio Nugroho, F. Rakhmat Umbara, dan R. Yuniarti, “Pengukuran Kualitas Perangkat Lunak pada SIMRS KHANZA Menggunakan Function Point dan System Usability Scale,” *Jurnal Tekno Kompak*, vol. 19, no. 1, hlm. 189–201, 2024.

- [26] A. A. Sandhea, S. Maulidiyah, dan R. D. Sari, “Estimasi Biaya Perangkat Lunak Pada Aplikasi SIBIMA Universitas XYZ dengan Menggunakan Metode Function Point,” *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 1, hlm. 41, Feb 2020, doi: 10.30865/jurikom.v7i1.1893.
- [27] G. F. Nama, A. D. Pamungkas, M. Mardiana, dan H. D. Septama, “Rancang Bangun Aplikasi Game Edukasi Koleksi Permainan Aksara Lampung (Koper Apung) Berbasis Android Menggunakan Metode Scrum,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, hlm. 420, Jul 2019, doi: 10.25126/jtiik.2019641096.
- [28] K. S. Alim, N. A. Ekowati, R. Y. Kisworini, dan L. Riyandari, “Design and Development of Web-Based Application Cangkungan Using Scrum Methods,” *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, hlm. 953–965, Agu 2023, doi: 10.52436/1.jutif.2023.4.4.1311.
- [29] S. Atin, T. Harihayati, dan U. D. Widianti, “Utilization of function point method for measuring software project complexity,” dalam *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Sep 2018. doi: 10.1088/1757-899X/407/1/012086.
- [30] R. K. Hapsari, D. M. Jauhari, dan H. Wp, “Estimasi Kualitas Perangkat Lunak Berdasarkan Pengukuran Kompleksitas Menggunakan Metrik Function Oriented,” dalam *Seminar Nasional Sains dan Teknologi Terapan III 2015*, Institut Teknologi Adhi Tama Surabaya, 2015, hlm. 425–434.
- [31] R. R. Mahendra *dkk.*, “Pengukuran Kualitas Perangkat Lunak Website Pendataan Ekskul Siswa Menggunakan Function Point,” *Jurnal Ilmiah Informatika*, vol. 11, no. 1, hlm. 1–14, 2023.
- [32] A. Sidqi, A. Yuswanto, dan M. A. Yaqin, “Analisis Pengukuran Kompleksitas Website SOC Menggunakan Function Point,” *Journal Automation Computer Information System*, vol. 4, no. 2, hlm. 67–75, Jul 2024, doi: 10.47134/jacis.v4i2.82.

- [33] C. Lenny Dwi Rizka, F. Shinta Dewi, dan S. Rizky Wicaksono, “Pengukuran dan Kualitas Perangkat Lunak Website ‘LINKEDIN’ Menggunakan Metode Function Point Analysis,” *JISKa*, vol. 3, no. 2, hlm. 79–83, 2018.
- [34] H. Rohayani, F. Lumban, G. Benfano, S. Harco, dan L. Hendric, “Estimated Measurement Quality Software On Structural Model Academic System With Function Point Analysis,” dalam *International Conference on Applied Computer and Communication Technologies (ComCom)*, IEEE, 2017, hlm. 1–5.
- [35] A. P. Satria, K. Widodo, dan A. Indrati, “Analysis Yummy App Site Functionality on Costs Use Method Function Point,” *International Research Journal of Advanced Engineering and Science*, vol. 6, no. 1, hlm. 164–168, 2021.
- [36] T. Wahyono, B. Soewito, S. W. H. L. Hendric, dan F. L. Gaol, “Software complexity measurement of water poverty mapping application with function point method,” dalam *2017 International Conference on Applied Computer and Communication Technologies (ComCom)*, IEEE, Mei 2017, hlm. 1–5. doi: 10.1109/COMCOM.2017.8167102.
- [37] A. P. Nugroho, S. Y. Muliana, Murtiningrum, dan S. S. Arif, “Economic Analysis Based on Software Cost Estimation Model on the Development of Telemetry Equipment to Support the Irrigation Modernization,” dalam *Advances in Biological Sciences Research*, Atlantic Press, 2022, hlm. 114–121. doi: 10.2991/absr.k.220305.017.
- [38] A. R. Irawati dan K. Mustofa, “Measuring Software Functionality Using Function Point Method Based On Design Documentation,” *International Journal of Computer Science Issues*, vol. 9, no. 3, hlm. 124–130, 2012, [Daring]. Tersedia pada: www.IJCSI.org
- [39] A. P. Subriadi, Sholiq, V. Lukitosari, dan R. Permatasari, “The Comparison Analysis Of Estimation Effort Among Software Development Using Function Point Method,” dalam *2018 International Seminar on Research of*

Information Technology and Intelligent Systems (ISRITI), IEEE, Nov 2018, hlm. 204–209. doi: 10.1109/ISRITI.2018.8864481.

- [40] S. Atin, Z. Priyanto, I. Afrianto, dan V. Pratiwi, “Integration of Function Point and Earned Value Management Methods in Information Systems Project Development Evaluation,” dalam *2023 International Conference on Informatics Engineering, Science & Technology (INCITEST)*, IEEE, Okt 2023, hlm. 1–7. doi: 10.1109/INCITEST59455.2023.10395922.