



**JUDUL**

**RANCANG BANGUN SISTEM PENGECEKAN BUG APLIKASI MENGGUNAKAN  
API BERBASIS WEB**

**(STUDI KASUS: PT INTIMAP)**

RIFANNY SUCI SYAHDA	41818110129
DIYANA ZEVIRA PEBRIANA	41818110050
SULTAN ATHORIQ ISMAIL	41818110134



**UNIVERSITAS  
MERCU BUANA**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MERCU BUANA**

**JAKARTA**

**2021**



**RANCANG BANGUN SISTEM PENGECEKAN BUG APLIKASI MENGGUNAKAN  
API BERBASIS WEB**

**(STUDI KASUS: PT INTIMAP)**

*Laporan Tugas Akhir*

Diajukan Untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

UNIVERSITAS  
MERCU BUANA

Oleh :

RIFANNY SUCI SYAHDA	41818110129
DIYANA ZEVIRA PEBRIANA	41818110050
SULTAN ATHORIQ ISMAIL	41818110134

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MERCU BUANA**

**JAKARTA**

**2021**

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa (1) : RIFANNY SUCI SYAHDA  
NIM (41818110129)  
Judul Tugas Akhir : RANCANG BANGUN SISTEM PENGECEKAN  
BUG APLIKASI MENGGUNAKAN API  
BERBASIS WEB  
(STUDI KASUS : PT INTIMAP)

Menyatakan bahwa Laporan Tugas Akhir ini adalah hasil karya nama yang tercantum diatas dan bukan plagiat. Apabila ternyata ditemukan di dalam laporan Tugas Akhir ini terdapat unsur plagiat, maka nama diatas siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Jakarta, 08 September 2022



Rifanny Suci Syahda

## SURAT PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR

Sebagai mahasiswa Universitas Mercu Buana, saya yang bertanda tangan di bawah ini :

Nama Mahasiswa (1) : RIFANNY SUCI SYAHDA  
NIM (41818110129)  
Judul Tugas Akhir : RANCANG BANGUN SISTEM PENGECEKAN  
BUG APLIKASI MENGGUNAKAN API  
BERBASIS WEB  
(STUDI KASUS : PT INTIMAP)

Dengan ini memberikan izin dan menyetujui untuk memberikan kepada Universitas Mercu Buana **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul diatas beserta perangkat yang ada (jika diperlukan).

Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Mercu Buana berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya.

Selain itu, demi pengembangan ilmu pengetahuan di lingkungan Universitas Mercu Buana, saya memberikan izin kepada Peneliti di Lab Riset Fakultas Ilmu Komputer, Universitas Mercu Buana untuk menggunakan dan mengembangkan hasil riset yang ada dalam tugas akhir untuk kepentingan riset dan publikasi selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, 08 September 2022



Rifanny Suci Syahda

## LEMBAR PENGESAHAN

Nama Mahasiswa (1) : RIFANNY SUCI SYAHDA  
NIM (41818110129)  
Nama Mahasiswa (2) : SULTAN ATHORIQ ISMAIL  
NIM (41818110134)  
Nama Mahasiswa (3) : DIYANA ZEVIRA PEBRIANA  
NIM (41818110050)  
Judul Tugas Akhir : RANCANG BANGUN SISTEM PENGECEKAN BUG APLIKASI  
MENGUNAKAN API BERBASIS WEB (STUDI KASUS: PT  
INTIMAP)

Tugas Akhir ini telah diperiksa dan disidangkan sebagai salah satu persyaratan untuk memperoleh gelar Sarjana pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Mercu Buana.

Jakarta, 26 Agustus 2022

Menyetujui,




(Nur Ari, S.T., M.M.S.I)

Dosen Pembimbing

Mengetahui,



(Yunita Sartika Sari, S.Kom., M.Kom)  
Sek. Prodi Sistem Informasi



(Ratna Mutu Manikam, S.Kom., M.T)  
Ka.Prodi Sistem Informasi

## KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah Subhanahu Wa Ta'ala karena atas berkat dan rahmat – Nya saya dapat menyelesaikan Proyek Pengembangan Sistem Informasi yang merupakan salah satu persyaratan dalam menyelesaikan program studi strata satu (S1) pada jurusan sistem informasi di Universitas Mercubuana. Adapun judul penulisan laporan Proyek Pengembangan Sistem Informasi sebagai berikut :

### **“RANCANG BANGUN SISTEM PENGECEKAN BUG APLIKASI MENGUNAKAN API BERBASIS WEB”**

Dalam penyusunan laporan Proyek Pengembangan Sistem Informasi ini, penulis banyak sekali mendapatkan bantuan dari berbagai pihak, oleh sebab itu penulis ingin mengungkapkan rasa terima kasih kepada :

1. Ibu Ratna Mutu Manikam, S.Kom., M.T selaku Ketua Program Studi Sistem Informasi
2. Ibu Yunita Sartika Sari, S.Kom., M.Kom selaku Sekretariat Program Studi Sistem Informasi
3. Ibu Nur Ani, ST, MMSI selaku Dosen Pembimbing Tugas Akhir yang telah memberikan bimbingan, arahan, saran, serta dukungan kepada saya dalam menyelesaikan laporan ini.
4. Staff, Karyawan, dan Dosen di lingkungan Universitas Mercu Buana.
5. Rekan-rekan Sistem Informasi Universitas Mercu Buana, Khususnya angkatan 2018
6. Beserta semua pihak yang telah membantu dalam penyusunan tugas akhir.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna karena itu kritik dan saran akan senantiasa penulis terima dengan senang hari. Akhir kata, penulisa berharap semoga tugas akhir ini diterima dan bermanfaat bagi penulisan dimasa yang akan dating.

Jakarta, 24Agustus 2022

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN JUDUL .....</b>	<b>ii</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>ABSTRAK .....</b>	<b>xv</b>
<b>ABSTRACT.....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	4
1.3. Batasan Masalah .....	4
1.4. Tujuan dan Manfaat Penelitian .....	4
1.5. Sistematika Penulisan .....	5
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>7</b>
1.1. Teori.....	7
1.1.1. Sistem Informasi Monitoring Error .....	7
1.1.2. Bug Perangkat Lunak.....	7
1.1.3. Website .....	7
1.1.4. Waterfall Model.....	8
1.1.5. Server .....	9
1.1.6. Node js .....	9
1.1.7. PHP .....	10
1.1.8. Vue.js .....	10
1.1.9. Mysql .....	10
1.1.10. Syntax.....	10
1.1.11. Fishbone Diagram (Diagram Ikan) .....	11
1.1.12. BPMN (Bussiness Process Management Notation).....	11
1.1.13. Monolog .....	12
1.1.14. Javascript.....	12
1.1.15. Application Programming Interface (API).....	12

1.1.16.	Express js.....	12
1.2.	Penelitian Terkait.....	13
1.2.1.	Summary.....	20
1.2.2.	Perbandingan Penelitian Terkait.....	20
<b>BAB III</b>	<b>METODE PENELITIAN.....</b>	<b>21</b>
2.1.	Lokasi Penelitian.....	21
2.2.	Sarana Pendukung.....	21
2.3.	Teknik Pengumpulan Data.....	22
2.4.	Diagram Penelitian.....	23
<b>BAB IV</b>	<b>HASIL DAN PEMBAHASAN.....</b>	<b>26</b>
3.1.	Analisa Sistem Berjalan.....	26
3.1.1.	Analisa Proses Bisnis Berjalan Sebelum Dilakukan Percancangan.....	26
3.1.2.	Identifikasi Masalah.....	27
3.2.	Analisa Kebutuhan dan Rancangan Kebutuhan.....	28
3.2.1.	Analisa kebutuhan sistem.....	28
3.2.2.	Proses Bisnis Rancangan Kebutuhan Sistem Informasi.....	28
3.2.3.	Alur proses kerja API ke DB (Backend).....	29
3.3.	Perancangan UML.....	30
3.3.1.	Usecase Diagram.....	30
3.3.2.	Dekripsi Usecase.....	32
3.4.	Activity Diagram.....	42
3.4.1.	Activity Diagram Login.....	42
3.4.2.	Activity Diagram Check My Task (QA).....	43
3.4.3.	Activity Diagram Check My Task (Programmer).....	44
3.4.4.	Activity Diagram Testing.....	45
3.4.5.	Activity Diagram Active Issue.....	46
3.4.6.	Activity Diagram History.....	47
3.4.7.	Activity Diagram Logout.....	48
3.4.8.	Activity Diagram Status Server.....	49
3.4.9.	Activity Diagram Generate Report.....	49
3.4.10.	Activity Diagram Report Programer Activity.....	50
3.4.11.	Activity Diagram Report QA Activity.....	51



3.4.12.	Activity Diagram Report Review Kinerja.....	52
3.4.13.	Activity Diagram Report Total Task Programmer.....	53
3.4.14.	Activity Diagram Report Programmer Activity.....	54
3.5.	Sequence Diagram.....	55
3.5.1.	Sequence Diagram Login.....	55
3.5.2.	Sequence Diagram Assigned.....	55
3.5.3.	Sequence Diagram Programmer Repair.....	56
3.5.4.	Sequence Diagram Testing.....	56
3.5.5.	Sequence Diagram Logout.....	57
3.5.6.	Sequence Diagram Report Programmer Activity.....	57
3.5.7.	Sequence Diagram Report QA Activity.....	58
3.5.8.	Sequence Diagram Report Performance Review.....	58
3.5.9.	Sequence Diagram Report Total Task Programmer.....	59
3.5.10.	Sequence Diagram Report Total Task QA.....	59
3.5.11.	Sequence Diagram Status Server.....	60
3.6.	Class Diagram.....	60
3.7.	Perancangan Basis Data.....	61
3.7.1.	Tb_temp_login.....	61
3.7.2.	Tb_temp_user.....	61
3.7.3.	Tb_temp_status.....	61
3.7.4.	Tb_temp_server.....	61
3.7.5.	Tb_temp_messages.....	62
3.7.6.	Tb_temp_histories.....	63
3.8.	Perancangan Antar Muka.....	63
3.8.1.	Modul Login.....	63
3.8.2.	Modul Dashboard.....	64
3.8.3.	Modul Active Issue QA.....	64
3.8.4.	Detail Active Issue QA.....	65
3.8.5.	Modul My Task QA.....	65

3.8.6.	Modul My Task Programmer.....	66
3.8.7.	Detail My Task Programmer .....	66
3.8.8.	Modul Active Issue Programmer .....	67
3.8.9.	Modul History QA dan Programmer .....	67
3.8.10.	Modul Status Server QA dan Programmer .....	68
3.8.11.	Modul Report Programer Activity .....	68
3.8.12.	Modul Report QA Activity .....	69
3.8.13.	Modul Report Review Kinerja .....	69
3.8.14.	Modul Report Total Task Programer .....	70
3.8.15.	Modul Report Programmer Activity .....	70
<b>BAB V</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>77</b>
4.1.	Kesimpulan .....	77
4.2.	Saran .....	77
<b>DAFTAR PUSTAKA.....</b>		<b>78</b>
<b>LAMPIRAN 1 KARTU BIMBINGAN .....</b>		<b>82</b>
<b>LAMPIRAN 2 KEGIATAN BIMBINGAN <i>online</i> .....</b>		<b>84</b>

## DAFTAR TABEL

Tabel 2.1 Jurnal Penelitian Terkait .....	13
Tabel 4.1 Keterangan Usecase .....	30
Tabel 4.2 Keterangan Actor .....	31
Tabel 4.3 Deskripsi Aktor .....	33
Tabel 4.4 Deskripsi Usecase Login .....	33
Tabel 4.5 Deskripsi Usecase Assigned .....	33
Tabel 4.6 Deskripsi Usecase Testing .....	34
Tabel 4.7 Deskripsi Usecase Check Task Bug .....	34
Tabel 4.8 Deskripsi Usecase Check Task .....	35
Tabel 4.9 Deskripsi Usecase Active Issues .....	36
Tabel 4.10 Deskripsi Usecase Repair Bug .....	36
Tabel 4.11 Deskripsi Usecase Status Server .....	37
Tabel 4.12 Deskripsi Usecase Generate Report .....	37
Tabel 4.13 Deskripsi Usecase QA Activity .....	37
Tabel 4.14 Deskripsi Usecase Programmer Activity .....	38
Tabel 4.15 Deskripsi Usecase Review Kinerja .....	38
Tabel 4.16 Deskripsi Usecase Programmer Activity .....	39
Tabel 4.17 Deskripsi Usecase Total Task Programmer .....	39
Tabel 4.18 Deskripsi Usecase History .....	40
Tabel 4.19 Deskripsi Usecase Logout .....	40
Tabel 4.20 Tb_temp_login .....	61
Tabel 4.21 Tb_temp_user .....	61
Tabel 4.22 Tb_temp_status .....	61
Tabel 4.23 Tb_temp_server .....	61

Tabel 4.24 Tb_temp_messages .....	62
Tabel 4.25 Tb_temp_histories .....	63



## DAFTAR GAMBAR

Gambar 2.1 Tahap Tahap Dalam Model Waterfall .....	8
Gambar 3.1 Alir Diagram .....	23
Gambar 4.1 BPMN Proses Pengecekan Bug Yang Masih Berjalan .....	26
Gambar 4.2 Fishbone Diagram .....	27
Gambar 4.3 BPMN Rancangan Sistem Informasi .....	29
Gambar 4.4 Usecase Diagram.....	32
Gambar 4.5 Activity Diagram Login .....	42
Gambar 4.6 Activity Diagram Check My Task (QA).....	43
Gambar 4.7 Activity Diagram Check My Task (Programmer) .....	44
Gambar 4.8 Activity Diagram Testing.....	45
Gambar 4.9 Activity Diagram Active Issue.....	46
Gambar 4.10 Activity Diagram History.....	47
Gambar 4.11 Activity Diagram Logout.....	48
Gambar 4.12 Activity Diagram Status Server .....	49
Gambar 4.13 Activity Diagram Generate Report .....	49
Gambar 4.14 Activity Diagram Report Programmer Activity.....	50
Gambar 4.15 Activity Diagram Report QA Activity .....	51
Gambar 4.16 Activity Diagram Report Review Kinerja .....	52
Gambar 4.17 Activity Diagram Report Total Task Programmer.....	53
Gambar 4.18 Activity Diagram Programmer Activity .....	54
Gambar 4.19 Sequence Diagram Login.....	55
Gambar 4.20 Sequence Diagram Assigned .....	55
Gambar 4.21 Sequence Diagram Programmer Repair.....	56
Gambar 4.22 Sequence Diagram Testing .....	56

Gambar 4.23 Sequence Diagram Logout.....	57
Gambar 4.24 Sequence Diagram Report Programmer Activity .....	57
Gambar 4.25 Sequence Diagram Report QA Activity.....	58
Gambar 4.26 Sequence Diagram Report Performance Review.....	58
Gambar 4.27 Sequence Diagram Report Total Task Programmer .....	59
Gambar 4.28 Sequence Diagram Report Total Task QA .....	59
Gambar 4.29 Sequence Diagram Status Server .....	60
Gambar 4.30 Class Diagram .....	60
Gambar 4.31 Modul Login .....	63
Gambar 4.32 Modul Dashboard.....	64
Gambar 4.33 Modul Active Issue QA .....	64
Gambar 4.34 Detail Active Issue QA .....	65
Gambar 4.35 Modul My Task QA.....	65
Gambar 4.36 Modul My Task Programmer.....	66
Gambar 4.37 Detail My Task Programmer.....	66
Gambar 4.38 Modul Active Issue Programmer .....	67
Gambar 4.39 Modul History QA dan Programmer .....	67
Gambar 4.40 Modul Status Servers QA dan Programmer.....	68
Gambar 4.41 Modul Report Programmer Activity .....	68
Gambar 4.42 Modul Report QA Activity .....	69
Gambar 4.43 Modul Report Review Kinerja.....	69
Gambar 4.44 Modul Report Total Task Programmer .....	70
Gambar 4.45 Modul Report Programmer Activity .....	70

## ABSTRAK

Nama : Sultan Athariq Ismail  
NIM : 41818110134  
Program Studi : Sistem Informasi  
Judul Laporan Skripsi : RANCANG BANGUN SISTEM PENGECEKAN BUG  
APLIKASI MENGGUNAKAN API BERBASIS WEB (STUDI  
KASUS: PT INTIMAP)  
Pembimbing : Nur Ani, ST, MMSI

PT Intimap memiliki mekanisme pengelolaan bug masih menggunakan cara yang kurang baik dalam pencatatan maupun pelaporan terjadinya bug. Pengelolaan bug dalam pembuatan aplikasi masih menggunakan cara manual yaitu dengan menggunakan excel, SMS, dan email setiap harinya. PT Intimap juga memiliki proses pengembangan perangkat lunak yang kurang baik, karena ketidaktepatan programmer saat pembuatan perangkat lunak atau penulisan script yang berdampak pada tidak jalannya kode secara normal. Tujuan dari dilakukannya penelitian ini yaitu untuk memastikan bug pada aplikasi sudah terminimalisir, memastikan aplikasi sudah melalui pengecekan bug dan tercatat dengan baik, dan meningkatkan kualitas aplikasi yang dibuat oleh PT Intimap. Teknik pengumpulan data dalam penelitian ini yaitu menggunakan observasi dan studi literatur. Pada penelitian ini penulis menggunakan metode Waterfall untuk mengembangkan sistem dan BPMN untuk menganalisa proses bisnis yang masih dilakukan belum menggunakan sistem dan membandingkan proses bisnis existing dengan analisa proses bisnis dengan sistem informasi yang kami ajukan. Hasil penelitian ini adalah sebuah aplikasi yang dapat digunakan untuk memantau bug yang terjadi pada saat pengembangan aplikasi di PT Intimap, menghasilkan suatu data yaitu berapa jumlah bug yang masih aktif, yang sedang dikerjakan dan bug yang sudah selesai dikerjakan.

Kata kunci:

Web Application, API, Bug, BPMN, Waterfall.

## ABSTRACT

Name : Sultan Athariq Ismail  
NIM : 41818110134  
Study Program : Information System  
Title Thesis : RANCANG BANGUN SISTEM PENGECEKAN BUG APLIKASI  
MENGUNAKAN API BERBASIS WEB (STUDI KASUS: PT  
INTIMAP)  
Counsellor : Nur Ani, ST, MMSI

*PT Intimap has a bug management mechanism that still uses a bad way of recording and reporting bugs. Bug management in making applications still uses the manual method, namely by using excel, SMS, and email every day. PT Intimap also has a poor software development process, due to the programmer's inaccuracy when making software or writing scripts which have an impact on the code not running normally. The purpose of this research is to ensure that bugs in the application have been minimized, ensure that the application has gone through bug checks and is well recorded, and improve the quality of applications made by PT Intimap. Data collection techniques in this study are using observation and literature study. In this research, the author uses the Waterfall method to develop the system and BPMN to analyze business processes that are still being carried out without using the system and compare existing business processes with business process analysis with the information system that we propose. The results of this study are an application that can be used to monitor bugs that occur during application development at PT Intimap, producing data, namely how many bugs are still active, currently being worked on and bugs that have been completed.*

*Keywords :  
Web Application, API, Bug, BPMN, Waterfall.*



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Dalam pembuatan sistem terdapat proses pengembangan sistem yang memiliki beberapa tahapan, mulai dari tahap *requirements gathering* yang dilakukan oleh *business analyst* kepada *user* terkait untuk mengumpulkan informasi kebutuhan sistem yang akan dibangun, dilanjutkan dengan menganalisa hasil kebutuhan *user* yang kemudian didokumentasikan menjadi dokumen desain, selanjutnya *programmer* mulai membuat konstruksi teknologi informasi dalam skala yang lebih detail berdasarkan dokumen desain yang telah dibuat oleh *business analyst*. Setelah sistem dibuat, tentunya sistem perlu dipastikan apakah sudah berjalan sesuai dengan bisnis proses yang telah didokumentasikan. Oleh karena itu perlu dilakukan pengecekan apakah ada *bug* dalam sistem yang telah dibuat [1]. *Bug* adalah hal yang biasa ditemukan dalam proses pengembangan perangkat lunak [2], bahkan disebuah perangkat lunak yang sudah besar dan terkenal sekalipun masih terdapat *bug*, yang bisa dilakukan oleh pengembang adalah mengurangi *bug* tersebut dengan melakukan pengecekan *bug*. Hal tersebut adalah tugas dari *quality assurance* dalam fase *testing*. Setelah sistem dinyatakan lolos dari pengecekan *bug* maka selanjutnya sistem dapat di-*deploy* dan bisa digunakan oleh *user* terkait [3]. Proses pengecekan *bug* adalah proses menganalisa suatu *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan untuk menemukan *bugs* dan mengevaluasi fitur-fitur dari *software* [4]. Proses pengecekan *bug* merupakan tahapan yang penting karena untuk memastikan sistem sudah bebas dari *bug* [1], yang nantinya menentukan apakah sistem tersebut sudah layak untuk di-*deploy* di server *production* untuk kemudian digunakan oleh *user* terkait. Dalam proses pengecekan *bug*, *quality assurance* perlu mencatat *bug* apa yang terjadi, kapan terjadinya *bug*, kepada siapa *bug* tersebut diserahkan untuk diperbaiki. Ketika *bug* sudah diserahkan kepada *programmer*, *programmer* akan memperbaiki *bug* tersebut kemudian ketika *bug* sudah diperbaiki, *quality assurance* perlu melakukan pengecekan kembali dan perlu mencatat status *bug* terakhir apakah sudah lolos dari *bug* atau masih perlu dilakukan perbaikan kembali oleh *programmer*. Dalam rangkaian tahap pengecekan *bug* tersebut seringkali ditemukan kendala yaitu pencatatan *bug* yang belum dikelola dengan baik, seperti pelaporan *bug* ke *programmer* yang menggunakan banyak sistem seperti *email*, *chatting*, atau bahkan secara lisan, sehingga menyulitkan *programmer* untuk mengumpulkan *bug* apa saja yang terjadi dan *bug* apa saja yang perlu diperbaiki [5]. Tidak adanya pencatatan yang baik juga

menyebabkan *bug* yang sudah terjadi tidak terlacak [6], sehingga jika ditemukan *bug* yang serupa kemudian hari *programmer* perlu mengeceknya dari awal yang tentu saja memerlukan waktu yang lebih lama untuk memperbaiki *bug* tersebut.

Saat ini PT Intimap memiliki mekanisme pengelolaan *bug* masih menggunakan cara yang kurang efektif dalam pencatatan maupun pelaporan terjadinya *bug* pada proses pembuatan aplikasi maupun aplikasi tersebut sudah digunakan oleh pengguna. Pengelolaan *bug* dalam pembuatan aplikasi masih dikatakan seadanya dengan bermodalkan kebiasaan yang dilakukan setiap harinya seperti hanya dengan lisan, *chatting* pada *grup* maupun pribadi, *email* antara *quality assurance* dan *programmer* dengan diketahui oleh atasan. Satu satunya prosedur yang dapat dikatakan cukup baik adalah dengan melakukan komunikasi melalui *email* dalam pembahasan terkait *bug* tersebut, tetapi dengan begitu hal-hal yang kurang diperhatikan adalah resiko dan dampak dari resiko itu sendiri. Resiko dalam penggunaan *email* disini bisa terjadi atas ketidak sengajaan atau *human error* baik dari sisi *programmer*, *quality assurance*, maupun pihak yang mengetahui *email* tersebut. Contoh *bug* yang terjadi dengan atas ketidak sengajaan seperti terhapusnya *email request*, terhapusnya *progress* pekerjaan perbaikan *bug*, *server* dan *storage* penyimpanan *email* penuh sehingga terjadi hilangnya data, *email* terinfeksi virus, dan resiko lainnya terhadap penggunaan *email*[7]. Dengan begitu tidak adanya pencatatan dari setiap proses pengerjaan *bug* yang konsisten datanya untuk digunakan sebagai dasar pembuatan laporan dari proyek tersebut terkait *bug* yang terjadi. Pencatatan yang dikelola dengan kurang efektif ini akan mengakibatkan sulitnya dalam melakukan *tracking history* terhadap *bug* yang terjadi selama proses pembuatan dan pengembangan aplikasi. *Tracking history* ini berguna untuk melakukan evaluasi dari segi proyek sampai penilaian kinerja dalam penanggung jawaban terhadap tugas masing masing khususnya pada kasus ini *programmer* dan *quality assurance* sebagai peran utama dari penelitian ini. Tidak adanya sistem *tracking history* ini menghambat proses pengelolaan dari *bug* yang ditemukan pada proyek[7].

Selain permasalahan tersebut, pada PT Intimap juga memiliki proses pengembangan perangkat lunak yang kurang baik, karena ketidaktelitian *programmer* saat pembuatan perangkat lunak atau penulisan *script* yang berdampak pada tidak jalannya kode secara normal karena program tidak dapat memahaminya. Ketidaktelitian *programmer* saat penulisan *script* biasa disebut dengan *Syntax Error*. *Syntax error* adalah *bug* yang terjadi saat ada *bug* ejaan pada salah satu kata kunci atau saat ada *bug* dengan struktur kodenya pada program. Penanganan *Syntax error* ini dilakukan karena pada PT Intimap menggunakan bahasa pemrograman *PHP* dan *Javascript* untuk membangun suatu *website* dimana bahasa

pemrograman tersebut tidak ada kompiler seperti pada bahasa pemrograman *Java* dan *Python*. Programmer biasanya harus mengecek *syntax error* secara *manual* dengan membuka *console* pada *website* yang sedang dibangun[8].

Macam-macam *syntax error* yang terjadi yaitu seperti pada *variable*, bug yang berkaitan dengan *variabel* antara lain penggunaan *variabel* yang belum dideklarasikan, contohnya mengisi nilai *variabel* dengan tipe data yang berbeda dari deklarasi sebelumnya seperti menggunakan *String* untuk operasi matematis dan penggunaan operasi matematis yang tidak sesuai dengan tipe data seperti penggunaan *integer* untuk operasi pembagian akan menghasilkan nilai yang kurang akurat, karena *integer* hanya dapat digunakan untuk bilangan bulat[8]. Pada *function*, bug yang berkaitan dengan *function* misalnya ketika suatu fungsi memiliki parameter, namun ketika programmer memanggilnya tanpa parameter, akan berakibat *error*. Pada *undefined*, bug yang berkaitan dengan suatu kondisi ketika *variable* yang dipanggil tidak didefinisikan terlebih dahulu. Pada *null*, bug yang berkaitan dengan *object* khusus yang mengindikasikan ‘tidak adanya nilai’ di dalam *JavaScript*, ketika terdapat *script* yang *error* dan bernilai *null* memiliki arti bahwa balikan data yang dipanggil dari suatu fungsi bernilai kosong. Pada *typo*, bug yang biasanya diakibatkan karena bug penulisan *script* oleh programmer[8].

Mengetahui dampak terjadinya *syntax error* dan mengakibatkan tidak berjalannya program dengan baik, dibutuhkan penanganan yang cepat untuk *syntax error* tersebut oleh programmer. Saat ini programmer mendapatkan laporan dari pengguna mengenai program yang tidak berjalan dengan baik, hal itu disebabkan beberapa *syntax error* yang lolos dalam pembuatan perangkat lunak oleh programmer karena terbatasnya waktu programmer untuk pengecekan *syntax error* yang cukup rumit. *Syntax error* yang lolos juga diakibatkan oleh *quality assurance* yang kurang teliti pada saat proses *testing*.

Berdasarkan masalah-masalah tersebut, dibutuhkan sistem pengecekan *bug* secara *realtime* agar programmer dapat segera memperbaiki *syntax error* yang terjadi yang mengakibatkan program tidak berjalan dengan baik[9]. Selain dibutuhkan sistem pengecekan *bug* secara *realtime*, pada PT Intimap juga membutuhkan sistem yang dapat mencatat *history bug* dengan baik sebagai dokumentasi bug. Mengacu kepada sistem sejenis yang telah dibangun oleh Anggik L. Y. Prita1 , Febriliyan Samopa , Radityo Prasetyanto Wibowo dengan judul jurnal “PENGEMBANGAN APLIKASI MANAJEMEN PROYEK PERANGKAT LUNAK BERBASIS SPRING : MODUL KOMUNIKASI DAN BUG TRACKER” terkait pengembangan perangkat lunak untuk melakukan manajemen *bug* dan

permasalahan *error* pada proyek pengembangan aplikasi atau perangkat lunak[9]. Pada penelitian sebelumnya, pencatatan bug masih dilakukan secara manual oleh *Quality Assurance*. Pencatatan bug secara manual dapat mengakibatkan terjadinya human error seperti tidak lengkapnya informasi bug yang dimasukkan oleh *Quality Assurance* ataupun jika bug yang ditemukan cukup banyak memungkinkan ada bug yang tidak tercatat. Oleh karena itu, peneliti mengusulkan sistem pencatatan bug aplikasi secara otomatis menggunakan API untuk meminimalisir kesalahan yang terjadi pada penelitian terkait.

## 1.2. Rumusan Masalah

Permasalahan pokok pada penelitian ini antara lain :

1. Bagaimana *Quality Assurance* dapat memantau aplikasi yang mengalami bug dan terdokumentasi?
2. Bagaimana memudahkan programmer track bug yang terjadi sebelumnya?
3. Bagaimana meminimalisir bug agar tidak ditemukan oleh customer?

## 1.3. Batasan Masalah

Batasan masalah pada penelitian ini antara lain :

1. Penelitian ini membahas dokumentasi pelacakan bug, peningkatan quality control aplikasi.
2. Pencatatan bug aplikasi secara otomatis dari website yang menggunakan API [https://tms.street-directory.com.au/system\\_reporting](https://tms.street-directory.com.au/system_reporting).
3. Penelitian hanya membahas syntax error pada program.
4. Sistem ini memuat informasi seperti task QA, task programmer, history bug, dan report.

## 1.4. Tujuan dan Manfaat Penelitian

Tujuan pada penelitian ini :

1. Memastikan aplikasi yang digunakan customer sudah melalui pengecekan sehingga meminimalisir bug yang terjadi
2. Pencatatan bug dengan baik sehingga programmer dapat tracking bug sebelumnya
3. Peningkatan kualitas aplikasi yang dibuat oleh PT Intimap.

Manfaat pada penelitian ini :

1. Data real time mengenai bug dalam proses pembuatan aplikasi.
2. Perbaikan pada bug tercatat dan dapat digunakan ketika terjadi bug serupa untuk mempermudah programmer dalam melakukan perbaikan.
3. Mengurangi terjadinya redundansi pada proses pencatatan *bug*
4. Mempermudah monitoring kinerja programmer dan QA
5. Penemuan bug tidak terjadi oleh pengguna aplikasi.

## 1.5. Sistematika Penulisan

Sistematika penulisan laporan kerja praktek ini yang digunakan adalah sebagai berikut :

### **BAB I PENDAHULUAN**

Pada BAB ini berisi tentang latar belakang permasalahan yang dihadapi oleh PT. Intimap dalam melakukan pengecekan *bug* aplikasi yang dibangun sehingga menghasilkan rumusan masalah yang akan dibahas pada penelitian ini. Batasan masalah menjelaskan tentang fokus pembahasan pada penelitian. Tujuan dan manfaat berisikan tentang hal yang akan dicapai.

### **BAB II TINJAUAN PUSTAKA**

BAB ini menguraikan tentang teori-teori yang digunakan pada penelitian ini, mulai dari teori tentang sistem informasi pengecekan *bug*, teknologi yang digunakan untuk merancang sistem pengecekan *bug* aplikasi pada PT. Intimap, serta studi literatur berdasarkan jurnal terkait pada penelitian ini.

### **BAB III METODOLOGI PENELITIAN**

BAB ini menjelaskan tentang lokasi penelitian yang diteliti yaitu di PT. Intimap, sarana pendukung yang dipergunakan, teknik pengumpulan data yang dipakai pada penelitian ini serta diagram alir penelitian yang akan menjelaskan alur penelitian dari awal sampai akhir.

## **BAB IV HASIL DAN PEMBAHASAN**

Dalam bab ini menjelaskan tentang hasil yang dicapai serta pembahasan perancangan sistem yang berisikan analisa proses bisnis, analisa kebutuhan, perancangan *UML*, perancangan basis data, perancangan antar muka dan juga implementasi dari perancangan yang sudah dibuat di penelitian.

## **BAB V KESIMPULAN**

Dalam bab ini menjelaskan kesimpulan penelitian yang telah dilakukan.



## BAB II

### TINJAUAN PUSTAKA

#### 1.1. Teori

##### 1.1.1. Sistem Informasi Monitoring Error

Sistem Informasi *Monitoring Error* berguna untuk memberitahukan akan *bug* yang terjadi di dalam sebuah program agar tercatat dengan baik dan dapat di monitor dalam 1 (satu) platform sehingga memudahkan pengembang untuk bisa *me-maintenance bug* apa saja yang terjadi pada program-program yang sudah dibuatnya dan supaya *bug* itu dapat diperbaiki secepatnya oleh pengembang[6].

Sistem monitoring merupakan hal yang sangat *crucial* dalam hal pemantauan *log* dari sebuah aplikasi terutama untuk mendeksi *errors* maupun ketidak sesuaian baik dari sisi konfigurasi dan rangkaian aplikasi yang sedang dilakukan *development* atau pada aplikasi yang sudah berjalan menjadi *system* yang aktif sebagai *system* informasi. Dalam hal ini sistem pengecekan *bug* memiliki peran dalam pengembangan bisnis pada sistem informasi yang disajikan kepada *customer*[11].

##### 1.1.2. Bug Perangkat Lunak

Bug adalah error atau kerusakan yang terjadi sehingga menyebabkan perangkat lunak tidak berjalan dengan seharusnya dan sering terjadi. Penyebab dari munculnya bug dapat bervariasi mulai dari bug pada tahap konstruksi perangkat lunak maupun bug dalam perancangan. Selain dari bug manusia, bug juga dapat muncul akibat perubahan ruang lingkup perangkat lunak. Semakin banyak jumlah bug pada perangkat lunak akan berakibat mengganggu kinerja dari perangkat lunak tersebut. Perangkat lunak yang mengalami banyak bug akan digolongkan sebagai buggy atau cacat[12].

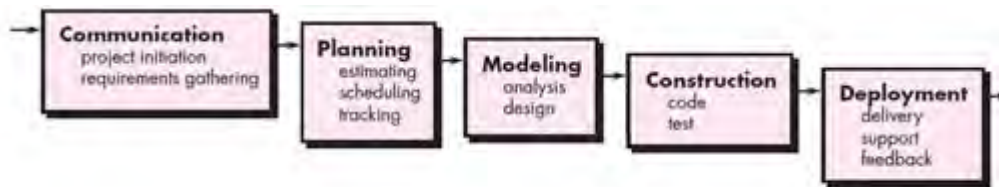
##### 1.1.3. Website

*Website* adalah tempat berkumpulnya halaman-halaman web yang biasanya diringkas dalam domain atau subdomain di *World Wide Web* di *Internet*. *Website* adalah sebuah sistem yang menangani dokumen-dokumen yang digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lain-lain pada jaringan internet. *Website* atau situs dapat diartikan sebagai kumpulan halaman yang digunakan untuk menampilkan teks, gambar diam atau gambar bergerak, animasi, suara, atau kombinasi keduanya statis dan dinamis[13].

#### 1.1.4. Waterfall Model

*Waterfall model* datang dengan banyak keuntungan. *Waterfall model* dapat diprediksi dan memperhatikan perencanaan arsitektur dan struktur sistem perangkat lunak secara rinci yang sangat penting ketika berhadapan dengan sistem besar. Tanpa fokus pada perencanaan arsitektur, ada risiko bahwa keputusan desain didasarkan pada pengetahuan implisit dan tidak didokumentasikan dan ditinjau secara eksplisit[14]. Model *Waterfall* mudah dipahami dan dikelola sepanjang *Systems Development Lifecycle*[15].

Berikut adalah penjelasan dari tahap – tahap yang dilakukan di dalam model ini menurut Pressman[16]:



**Gambar 2.1 Tahap Tahap Dalam Model Waterfall**

1. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi dari aplikasi. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, paper dan internet.

2. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan tracking proses pengerjaan sistem.

3. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan pemodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan interface, dan algoritma



program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

#### 4. *Construction (Code & Test)*

Tahapan ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

#### 5. *Deployment (Delivery, Support, Feedback)*

Tahapan terakhir ini merupakan tahapan implementasi *software* ke *customer*, perbaikan *software*, evaluasi *software*, dan pengembangan *software* berdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya.

### 1.1.5. Server

*Server* adalah sebuah memori yang terbagi secara multi-dimensional untuk komputer yang saling terhubung dan menyediakan layanan untuk pertukaran data. Sederhananya, *server* menunggu klien dan kemudian sistemnya mencakup organisasi teoretis yang dapat diubah oleh tindakan *server*. Melalui sistem *file sharing*, layanan disimpan dan dilindungi dalam sistem rahasia dari klien. Caranya dengan menyimpan format asli dari data/pesan kedalam satu PC. Tetapi harus dipertimbangkan bahwa *server* ini juga berfungsi untuk *server* lain [17].

### 1.1.6. Node js

Pada penelitian ini menggunakan *node.js* untuk *runtime environment* untuk *JavaScript* dalam membangun aplikasi *monitoring error* pada aplikasi karena *nodes* memiliki performa yang sangat baik dan ringan untuk proses *development* dilingkungan *enterprise*. Hal ini sejalan dengan pengertian *Node.js* yang merupakan aplikasi *server-side* yang dirancang pada *runtime JavaScript Chrome* oleh Ryan Dahl pada tahun 2009. Menggunakan teknologi *event-driven* serta *I/O blocking* model *event-driven* sendiri merupakan suatu teknik pemrogram yang mana mempunyai konsep kerja tergantung event atau kejadian tertentu, dengan konsep seperti itu membuat aplikasi menjadi ringan dan efisien [18].

### 1.1.7. PHP

PHP adalah bahasa pemrograman sederhana namun kuat yang dirancang untuk membuat konten HTML. PHP beraksi dengan panduan cepat dari beberapa program PHP yang mengilustrasikan tugas-tugas umum, seperti memproses data, berinteraksi dengan *database*, dan membuat grafik. Salah satu fitur PHP yang paling signifikan adalah dukungannya yang luas untuk database. PHP mendukung semua database utama (termasuk *MySQL*, *PostgreSQL*, *Oracle*, *Sybase* dan lain-lain). Bahkan database yang lebih baru seperti MongoDB juga didukung. Dengan PHP, membuat halaman web dengan konten dinamis dari database menjadi sangat sederhana[19].

### 1.1.8. Vue.js

*Vue.js* merupakan sebuah *Framework JavaScript* pada bagian *front end* yang digunakan untuk membangun tampilan *user interface*. *Vue.js* merupakan proyek *open-source* dengan lisensi MIT yang diciptakan oleh Evan You pada bulan Februari 2014 [20]. *Vue.js* menjadi kerangka utama pengembangan *front-end* terbaru, yang meningkatkan pengembangan efisiensi situs web, logika kode yang lebih jelas, lebih mudah untuk mengatasi perubahan kebutuhan, dan sistem lebih mudah untuk dipelihara dan diintegrasikan sehingga dapat mengurangi biaya pengembangan[20].

### 1.1.9. Mysql

MySQL adalah salah satu sistem manajemen basis data sumber terbuka relasional yang paling populer. Ini menghemat data dalam tabel yang berbeda dan cepat dan fleksibel. Sistem ini menggunakan MySQL untuk mewujudkan fungsi-fungsi dari menyimpan, memodifikasi dan menghapus data, yang memungkinkan lebih sistematis dan ilmiah pengelolaan data[20].

### 1.1.10. Syntax

*Syntax* adalah perintah-perintah dalam bahasa program yang telah ditentukan dalam *coding* mempengaruhi berjalannya normal atau tidak suatu sistem, pentingnya ketelitian dalam proses pengetikan *syntax* karena jika salah memasukkan kata atau huruf dan lain sebagainya, akan berakibat program aplikasi yang dibuat mengalami *debug* atau *error*[21].

### **1.1.11. Fishbone Diagram (Diagram Ikan)**

Fishbone diagram atau diagram ikan dalam penerapannya adalah untuk mengidentifikasi berbagai sebab potensial dari satu efek atau masalah, serta menganalisa masalah tersebut melalui brainstorming[22]. Dalam menganalisa suatu permasalahan menggunakan fishbone, masalah akan dipecah kedalam beberapa kategori potensial yang berkaitan lalu permasalahan tersebut akan diuraikan sehingga mendapatkan hasil dari faktor-faktor penyebab permasalahan.

### **1.1.12. BPMN (Business Process Management Notation)**

Penggunaan BPMN merupakan penggambaran dari model bisnis konsep tual yang menunjukkan aliran data atau pesan serta asosiasi artefak data kegiatan[23]. BPMN dirancang untuk memodelkan proses bisnis, proses-proses sebuah web services. Sehingga dengan adanya BPMN, proses bisnis mudah dipahami oleh semua pengguna bisnis termasuk analis bisnis yang menciptakan draft dari awal proses sampai dengan pengembangan teknis untuk mengimplementasikan teknologi yang menjalankan proses-prosesnya[24].

Kategori dasar yang terdapat pada elemen BPMN :

#### **1. Flow Objects**

Digambarkan melalui event yang berbentuk lingkaran yang merupakan penyebab (trigger) atau hasil (result) dari sebuah proses. Selanjutnya activity yang berbentuk persegi Panjang yang merepresentasikan suatu aktivitas proses. Dan yang terakhir adalah gateway digambarkan dengan bentuk belah ketupat yang merepresentasikan kontrol percabangan untuk suatu keputusan, ataupun penggabungan aliran.

#### **2. Connecting Object**

Merupakan elemen yang menggabungkan flow object dan memiliki tiga jenis elemen yaitu alur sequence untuk menunjukkan urutan kegiatan, alur pesan untuk menunjukkan aliran pesan antara dua entitas, asosiasi untuk menunjukkan asosiasi data, informasi dan artifak dengan aliran benda.

#### **3. Swimlanes**

Digambarkan dengan bentuk garis yang memisahkan dan mengelompokkan aktor, terdapat dua bentuk swimlane objects yaitu pool yang mewakili partisipan dalam sebuah proses sedangkan lanes digunakan untuk mengatur mengkategorikan aktivitas.

#### 4. Artifacts

Merupakan elemen yang digunakan untuk memberikan informasi tambahan dari sebuah proses. Terdapat tiga tipe artifacts yaitu data object untuk menunjukkan bagaimana data dibutuhkan atau diproduksi oleh aktivitas. Group untuk digunakan tujuan dokumentasi atau analisis. Annotation untuk memberikan informasi teks tambahan.

##### 1.1.13. Monolog

*Monolog* adalah *library logging* PHP yang mengirimkan *log* ke *file*, *socket*, *inbox*, *database* dan berbagai layanan web[25]. *Logging* merupakan catatan yang merekam segala aktifitas suatu aplikasi dijalankan, terkadang *Log* juga digunakan untuk menganalisa status suatu aplikasi. *Monolog* mendukung *level* berupa *debug*, *info*, *notice*, *warning*, *error*, *critical*, *alert* dan *emergency*.

##### 1.1.14. Javascript

*JavaScript* adalah bahasa pemrograman *scripting* yang kuat yang telah mendapatkan banyak perhatian selama dekade terakhir ini[26]. Awalnya digunakan secara eksklusif untuk pengembangan web sisi klien, ini telah berkembang menjadi salah satu bahasa pemrograman paling populer, dengan pengembang sekarang menggunakannya untuk pengembangan aplikasi sisi klien dan sisi *server*. *JavaScript* merupakan bahasa pemrograman populer yang juga rawan bug karena sifatnya yang asinkron, dinamis, dan mudah diketik.

##### 1.1.15. Application Programming Interface (API)

*Application Programming Interface* atau API adalah sebuah dokumentasi yang terdiri dari *interface*, *class*, *function*, *struktur* dan sebagainya agar dapat membangun sebuah perangkat lunak[27]. API suatu kode pemrograman penghubung antara aplikasi atau web yang telah dibuat dengan fungsi yang dikerjakan. Dengan menggunakan API, aplikasi dapat diakses tanpa mengubah struktur kode utama maupun *database* sistem[27].

##### 1.1.16. Express js

Express.JS merupakan sebuah Node.JS framework yang menyediakan API yang disederhanakan beberapa dari fungsionalitas inti node. Express JS berguna untuk mempermudah pembuatan aplikasi berbasis Node.JS dengan menggunakan design pattern

yang dapat disesuaikan dan sangat fleksibel. Selain itu Express.JS juga merupakan framework yang sangat ringan dan cocok untuk membuat web aplikasi dan API[28].

## 1.2. Penelitian Terkait

Berikut adalah penelitian terkait yang digunakan sebagai referensi dalam melakukan penelitian ini.

**Tabel 2.1 Jurnal Penelitian Terkait**

No	Sumber	Masalah	Metode	Hasil
1	Implementasi Bug Tracking System dengan Metodologi Scrum dan Algoritma Cosine Similarity [29]	Mengidentifikasi, menyelesaikan, dan mencegah muncul kembalinya suatu bug sangatlah memakan waktu. Pengembang junior membutuhkan waktu lebih lama untuk mengidentifikasi bug yang umum terlihat. Penelitian yang dilakukan difokuskan pada pengembangan Bug Tracking System (BTS) berbasis web.	Menggunakan metodologi pengembangan sistem agile yang dikenal dengan Scrum Framework. Menurut, Scrum adalah model pengembangan produk yang dirancang untuk tim pengembangan agile dengan membanggakan kemampuan beradaptasinya atas metodologi konvensional lainnya.	Hasil dari penelitian ini adalah BTS berbasis web yang dirancang menggunakan metodologi Scrum dan dilengkapi dengan Content-Based Filtering. Pengguna dapat mengelola tiket masalah dengan baik dan efisiensi resolusi bug pada akhirnya dapat ditingkatkan.
2	SISTEM INFORMASI PENCATATAN	Pelaporan temuan bug ini biasanya dilakukan melalui	Dalam perancangan sitsem informasi	Sistem berbasis website dengan alamat URL cdsbugbase.site,

No	Sumber	Masalah	Metode	Hasil
	BUG BERBASIS WEB [30]	message, email, atau catatan biasa. Pelaporan seperti ini biasanya kurang rapi dan kurang efisien karena dapat menyulitkan pengembang dalam proses pencarian data atau riwayat pelaporan bug. Dalam kasus lain, pengembang juga kesulitan memahami maksud pelapor karena kurangnya detail pada pelaporan.	pencatatan bug ini menggunakan UML	di mana semua informasi yang dibutuhkan dapat disimpan dan diakses secara terpusat
3	Analisis Bug Coding Pemrograman Java Pada Matakuliah Algoritma Pemrograman Mahasiswa Tadris Matematika Iain Kediri[8]	Permasalahan sederhana sampai kompleks (numerik) sangat erat kaitannya dengan komputer, sebagai akibatnya kiprah software tadi sangat diharapkan agar dominasi	Metode penelitian kualitatif dengan melakukan analisis dari kasus yang dilakukan penelitian dan menemukan solusi dari kasus permasalahan tersebut.	Solusi yang Dilakukan Untuk Mengatasi Bug Penulisan Sintaks Java dengan memberikan pengarahan dan koreksi dari setiap <i>assignment</i> yang diberikan.

No	Sumber	Masalah	Metode	Hasil
		mahasiswa terhadap teori, konsep dan perangkat lunak matematika untuk menyelesaikan problem dapat dikuasai dengan baik.		
4	Rancang Bangun Aplikasi Pengelolaan Bug Tracker Berbasis Web pada Pengembangan Software [31]	Suatu tim pengembang berkemungkinan besar mendapatkan bug dalam pengembangan perangkat lunak. Untuk mengatasi masalah dibutuhkan aplikasi bug tracker yaitu untuk memudahkan suatu tim proyek software development dalam mengevaluasi, mengoreksi, melacak, dan mengontrol bug dan masalah pada suatu proyek.	Metode yang digunakan adalah Rational Unified Process yang memiliki empat tahap, yaitu Inception, Elaboration, Construction, dan Transition dan menggunakan framework php yaitu laravel. Hasil	Berupa aplikasi bug tracker berbasis web untuk mengelola bug pada suatu project secara online yang terdapat fitur seperti Upload gambar untuk menambah informasi berupa file gambar, dan fitur list step to reproduce

No	Sumber	Masalah	Metode	Hasil
5	Pembangunan Sistem Informasi Monitoring Logistik di PT Dirgantara Indonesia [32]	Dalam proses produksi maupun pengiriman mengalami kendala yaitu keterlambatan dalam kinerja yang tidak dilakukan <i>monitoring</i> secara baik	Sistem informasi <i>monitoring</i> logistik yang dibangun menggunakan metode <i>waterfall</i> .	Kemudahan dalam pengolahan data dan pengawasan atau monitoring terhadap pekerjaan logistik baik dari segi pengiriman maupun produksi dalam dimonitor dengan lebih baik sehingga mengefisienkan kinerja dan kegiatan logistik
6	SISTEM INFORMASI MONITORING PROYEK PADA PT. SURYAPRIMA KARYA ARRYVYA BERBASIS JAVA [33]	Kegagalan maupun keterlambatan proyek yang mengakibatkan banyaknya kerugian yang dikarenakan sistem pemantauan atau <i>monitoring</i> masih dilakukan secara manual dengan <i>visit</i> yang memakan waktu serta kinerja.	Metode penelitian yang digunakan adalah metode <i>grounded (grounded research)</i> adalah suatu metode penelitian yang mendasar dari fakta dan penggunaan analisi perbandingan yang bertujuan untuk mengadakan generalisasi empiris,	Aplikasi <i>monitoring</i> proyek berbasis java ini dapat dikembangkan kembali dalam hal desain atau penambahan <i>database</i> sesuai kebutuhan data.



No	Sumber	Masalah	Metode	Hasil
			menetapkan konsep, membuktikan teori, dimana pengumpulan data dan Analisa data berjalan dalam waktu yang bersamaan.	
7	Rancang Bangun Sistem Informasi <i>Monitoring Dashboard Traffic Work Order</i> Berbasis Web [34]	Dalam rangka meningkatkan efektifitas dan efisiensi kinerja dalam melakukan pelayanan kepada <i>customer</i> guna mengurangi proses yang masih tergolong lambat dalam melakukan pelayanan sehingga Sistem Informasi ini dirancang untuk meningkatkan efektifitas dan efisiensi kinerja terhadap pelayanan kepada customer	Pada penelitian ini digunakan metode penilaian dalam penyampaian <i>Work Order</i> dengan menggunakan metode SDLC ( <i>Waterfall</i> ). Pada pengembangan ini perencanaan, perancangan sistem informasi yang berkaitan dengan penyampaian pada <i>Monitoring</i>	Rancang Bangun Sistem Informasi <i>Monitoring Dashboard Traffic Work Order</i> Berbasis Web
8	Rancang Bangun Aplikasi <i>Monitoring Kegiatan</i>	Pencatatan perbaikan maupun permintaan tambahan modul	Menggunakan UML ( <i>Unified Modelling Language</i> )	<i>Software</i> berbasis web yang memiliki empat antar muka yang berbeda untuk

No	Sumber	Masalah	Metode	Hasil
	Implementasi Proyek Perangkat Lunak Berbasis Web [35]	pada proyek yang diimplementasikan menggunakan Ms.Excel sebagai media pencatatannya dinilai kurang efisien karena pencatatan menjadi berantakan dan tidak terstruktur dengan baik ketika seorang pelaksana mengerjakan lebih dari satu proyek besar dimana proyek tersebut masih banyak perbaikan. Juga untuk <i>programmer</i> / analis sistem.	yaitu : <i>usecase</i> diagram dan <i>class</i> diagram.	<i>implementor</i> , <i>programmer</i> / <i>system analyst</i> dan <i>project manager</i> dalam menyelesaikan permasalahan yang sering terjadi pada kegiatan implementasi proyek.
9	Implementasi Chatbot Untuk Error Reporting Dengan Natural Language Processing Pada CV. Solusi Prima [36]	Sistem yang berjalan yaitu client dan staff dalam hal melaporkan bug, membuat laporan yang dilakukan secara terpisah dan manual (via Whatshapp). Hal ini membuat rancu antara chat dengan melihat laporan bug pada sistem, menjadi kendala tersendiri bagi para	Menggunakan metode penelitian berupa wawancara dan observasi serta penelitian pepustakaan.	Hasil pengujian sistem berjalan baik dengan penggunaan chatbot yang telah memiliki Natural Language Processing, membuat pengguna dapat lebih mudah dalam melakukan laporan tanpa harus berpindah

No	Sumber	Masalah	Metode	Hasil
		staff untuk melaporkan kepada kepala tim.		tangan dari keyboard ke mouse.
10	Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Berbasis Spring : Modul Komunikasi Dan Bug Tracker[9]	Dalam proses pekerjaan proyek banyak masalah terjadi salah satunya bug error pada proses pembuatan perangkat lunak yang tidak terkelola dengan baik sehingga menyebabkan bug berulang dan sulitnya melakukan evaluasi pekerjaan proyek yang sudah di lakukan pekerjaan.	Metode Prototype dengan tujuan untuk menyetarakan dan mengatasi ketidak selarasan pemikiran dari pihak pihak terkait dan juga melakukan uji coba dan analisis dari perangkat lunak yang akan dibuat.	1. Pembuatan modul komunikasi dan bug tracker di software manajemen proyek dapat dilakukan menggunakan mengimplementasikan teknologi Spring MVC, Hibernate dan JSP. 2. dengan menggunakan OSGi, komponen Recent Activity, News, arsip Sharing, Bug Tracker serta Chatting bisa diintegrasikan dan bisa diimplementasikan untuk menghasilkan modul komunikasi serta bug tracker pada perangkat lunak manajemen proyek

### 1.2.1. Summary

Dari literatur review tersebut dapat disimpulkan bahwa permasalahan yang dihadapi sama, yaitu masalah pada aplikasi web baik berupa kesalahan atau gangguan dikirimkan ke pengembang melalui email atau pertemuan tatap muka yang mengakibatkan pengembang sulit untuk berkomunikasi atau melaporkan bug atau kesalahan dalam aplikasi web. 6 dari 10 penelitian tersebut menggunakan metode SDLC yaitu Waterfall untuk mengembangkan sistem yang dibangun karena dinilai lebih baik untuk mengontrol proses pengembangan model tahap demi tahap, sehingga meminimalisir kesalahan yang mungkin akan terjadi.

Penelitian ini juga mendasari kebutuhan Sistem Informasi untuk melakukan pengecekan *bug* dengan acuan dari kasus kasus pada literatur yang digunakan sebagai referensi dalam penelitian ini.

### 1.2.2. Perbandingan Penelitian Terkait

Mengacu pada latar belakang sistem ini dibuat selain dibutuhkan sistem pengecekan *bug* secara *realtime*, pada PT Intimap juga membutuhkan sistem yang dapat mencatat *history bug* dengan baik sebagai dokumentasi bug. Mengacu kepada sistem sejenis yang telah dibangun oleh Anggik L. Y. Prital , Febriliyan Samopa , Radityo Prasetyanto Wibowo dengan judul jurnal “PENGEMBANGAN APLIKASI MANAJEMEN PROYEK PERANGKAT LUNAK BERBASIS SPRING : MODUL KOMUNIKASI DAN BUG TRACKER” terkait pengembangan perangkat lunak untuk melakukan manajemen *bug* dan permasalahan *error* pada proyek pengembangan aplikasi atau perangkat lunak[9]. Pada penelitian sebelumnya, pencatatan bug masih dilakukan secara manual oleh *Quality Assurance*. Pencatatan bug secara manual dapat mengakibatkan terjadinya human error seperti tidak lengkapnya informasi bug yang dimasukkan oleh *Quality Assurance* ataupun jika bug yang ditemukan cukup banyak memungkinkan ada bug yang tidak tercatat. Oleh karena itu, peneliti mengusulkan sistem pencatatan bug aplikasi secara otomatis menggunakan API untuk meminimalisir kesalahan yang terjadi pada penelitian terkait.

## BAB III

### METODE PENELITIAN

#### 2.1. Lokasi Penelitian

Lokasi Penelitian dilakukan di PT Intimap. PT Intimap merupakan perusahaan teknologi internasional dengan merek dagang Intellitrac GPS yang didirikan pada 2003 oleh Virtual Map (Australia) Pty Ltd. PT Intimap hadir untuk memberikan solusi di bidang online map generation dan portal development, digital map imaging, GIS, GPS Tracking, mobile mapping, web development, dan spatial database programming. Layanan yang ditawarkan oleh PT Intimap diantaranya adalah Vehicle Tracking, Transport Management System, IoT Platform, dan Virtual Mapp. Penelitian ini dilakukan untuk mengumpulkan data dalam melakukan perancangan sistem Monitoring.

#### 2.2. Sarana Pendukung

Aplikasi Sistem *Monitoring* Aplikasi pada PT Intimap dibangun dalam bentuk aplikasi berbasis *website*. Pada penelitian kali ini juga dibutuhkan teknologi pendukung baik itu perangkat keras maupun perangkat lunak.

##### 1. Perangkat Keras :

###### a. *Personal Computer*, dengan spesifikasi :

- *Operating System : Windows 10 Professional*
- *Processor : Intel® Core™ i5 3320M*
- *Memory : RAM 8GB DDR3*
- *Graphics : Intel®*
- *Storage : 167 GB*

###### b. *Server Virtual Machine*, dengan spesifikasi :

- *Operating System : Ubuntu 18.04*
- *Machine Type : e2-medium (2 vCPUs, 4 GB memory)*
- *Storage : 30GB SSD*

##### 2. Perangkat Lunak

1. *Windows 10 Professional*
2. *Ubuntu 18.04*
3. *Node.js*
4. *Vue.js*

5. *Mysql*
6. *Express js*
7. *Visual Studio Code*

### 2.3. Teknik Pengumpulan Data

Dalam penelitian ini, peneliti menggunakan teknik pengumpulan data :

#### 1. Observasi

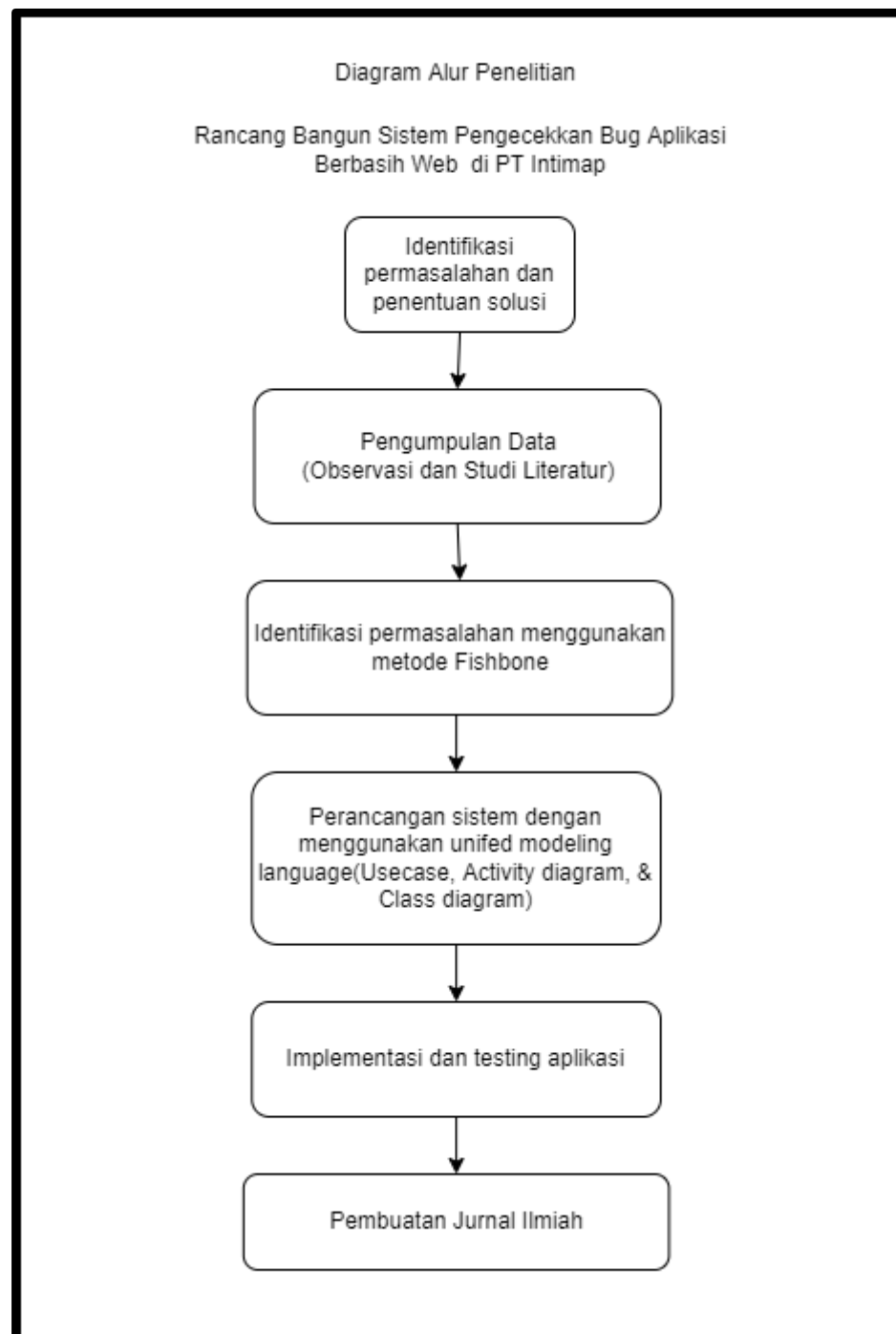
Pada penelitian ini dilakukan observasi dari API dengan url [https://tms.street-directory.com.au/system\\_reporting/](https://tms.street-directory.com.au/system_reporting/), yang memiliki beberapa *method request* yang menghasilkan *return* data bertipe JSON. Data JSON berisikan *object* yang dibutuhkan untuk mengolah data pada penelitian ini. Data itu seperti jenis *bug* yang terdapat pada aplikasi khususnya *syntax error* , server pada beberapa aplikasi yang digunakan untuk mendapatkan data *bug*, beberapa *bug* yang masih terjadi pada aplikasi atau bisa disebut *active issues*, daftar *bug* yang telah dilakukan pengecekan dan diperbaiki akan masuk ke dalam *histories error*, data *users* yang berguna untuk *programmer* dan *quality assurance* melakukan login disistem[10].

#### 2. Studi Literatur

Melakukan studi literatur atau kepustakaan dengan mengumpulkan bahan-bahan Pustaka seperti artikel dan jurnal dari penelitian terdahulu pada kurun waktu tahun 2015 – 2020 yang dapat mendukung penelitian terkait sistem monitoring error pada aplikasi.

## 2.4. Diagram Penelitian

Berikut diagram alir pada penelitian “Rancang Bangun Sistem Informasi *Monitoring Error* Pada Aplikasi di PT Intimap” :



**Gambar 3.1 Alir Diagram**

Penjelasan Alir Diagram :

1. Identifikasi permasalahan dan penentuan solusi

Pada tahap ini dilakukan identifikasi masalah dan penentuan solusi, ditahap ini penulisan melakukan perumusan masalah, Batasan masalah, manfaat dan tujuan penelitian.

2. Pengumpulan data dengan observasi dan Studi literatur

Melakukan studi literatur dari jurnal-jurnal penelitian terkait dan melakukan observasi dari API dengan url [https://tms.street-directory.com.au/system\\_reporting/](https://tms.street-directory.com.au/system_reporting/), yang memiliki beberapa *method request* yang menghasilkan *return* data bertipe JSON. Data JSON berisikan *object* yang dibutuhkan untuk mengolah data pada penelitian ini. Data itu seperti jenis *bug* yang terdapat pada aplikasi khususnya *syntax error*, server pada beberapa aplikasi yang digunakan untuk meny data *bug*, beberapa *bug* yang masih terjadi pada aplikasi atau bisa disebut *active issues*, daftar *bug* yang telah dilakukan pengecekan dan diperbaiki akan masuk ke dalam *histories error*, data *users* yang berguna untuk *programmer* dan *quality assurance* melakukan login disistem[10].

3. Pada tahap ini mengidentifikasi permasalahan yang dihadapi menggunakan Fishbone.

Metode analisis yang ini disebut dengan fishbone dikarenakan bentuk dari diagram tersebut merujuk kepada bentuk tulang ikan yang memiliki ruas ruas tulang yang berperan sebagai titik atau nilai permasalahan yang terjadi serta solusi yang akan diakomodasi untuk menanggulangi permasalahan tersebut. Bentuk moncong ke depan seperti kepala ikan melambangkan sistem yang dirancang untuk permasalahan yang ada.

4. Perancangan

Perancangan aplikasi ini dilakukan dengan metode Waterfall sehingga mengikuti dalam alur metode itu diantaranya: Melakukan analisa kebutuhan untuk sistem informasi yang akan dibangun, selanjutnya melakukan desain UML berdasarkan analisa kebutuhan dari proses sebelumnya, kemudia, sistem dibangun sesuai dengan desain yang sudah dirancang, unit yang dikembangkan pada tahap implementasi diintegrasikan ke dalam sistem setelah pengujian setiap unit, setelah itu seluruh sistem diuji untuk menemukan bug pada sistem, kemudian yang terakhir adalah pemeliharaan sistem, apakah diperlukan peningkatan atau tidak.

5. Implementasi dan Evaluasi.

Pada tahap ini dilakukan implementasi sistem, evaluasi dan pengujian aplikasi.



6. Pembuatan Artikel Ilmiah

Pada tahap ini dilakukan perancangan untuk menghasilkan artikel/jurnal ilmiah yang dapat dipublikasikan.



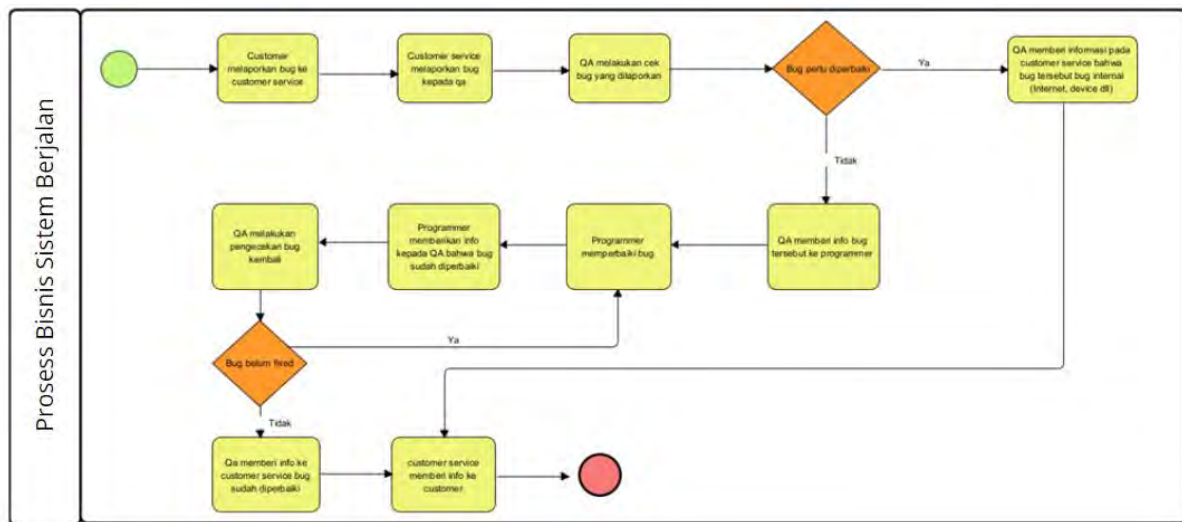
## BAB IV HASIL DAN PEMBAHASAN

### 3.1. Analisa Sistem Berjalan

#### 3.1.1. Analisa Proses Bisnis Berjalan Sebelum Dilakukan Percancangan

Pada penelitian ini penulis menggunakan BPMN untuk menganalisa proses bisnis yang masih dilakukan belum menggunakan sistem dan membandingkan proses bisnis *existing* dengan analisa proses bisnis dengan sistem informasi yang kami ajukan.

1. Berikut proses bisnis yang terjadi pada proses pengecekan bug berjalan sebelum dilakukan percancangan.

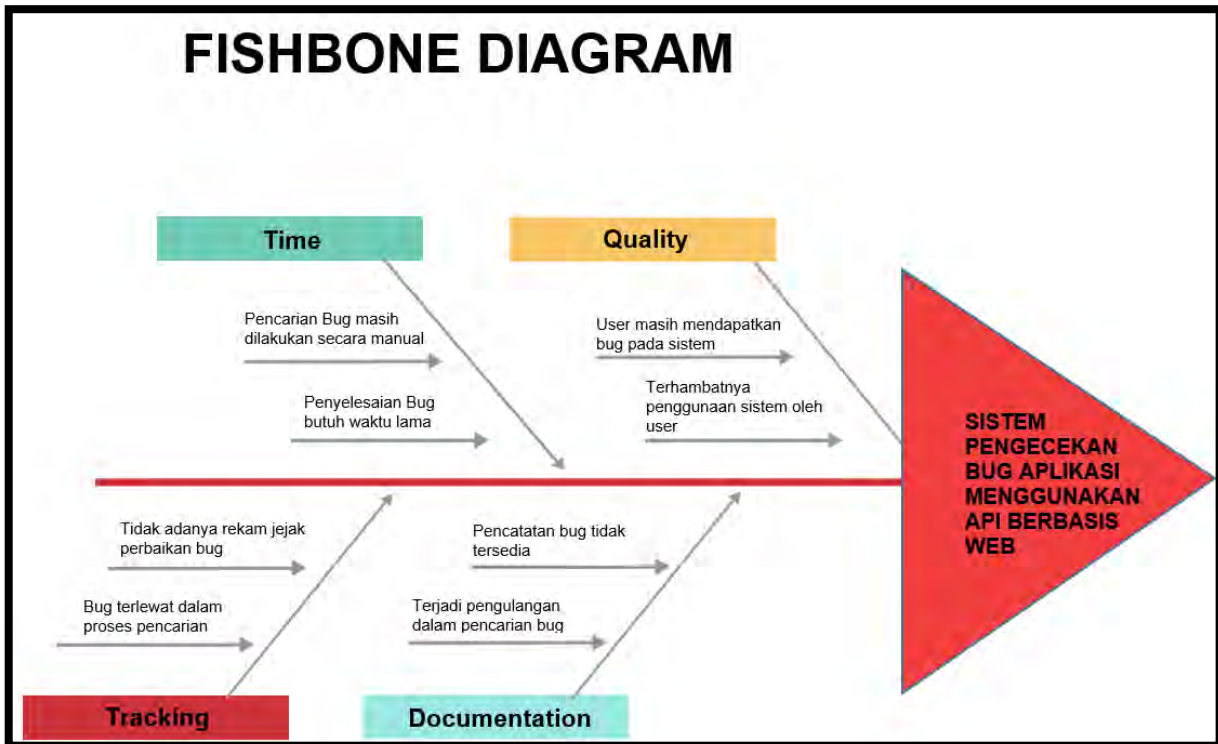


**Gambar 4.1 BPMN Proses Pengecekan Bug Yang Masih Berjalan**

Penggambaran alur proses bisnis existing membutuhkan waktu yang lama dan juga dengan tidak ada dokumentasi sehingga menjadikan sulitnya dalam melakukan *tracking* dari *bug* yang telah dilakukan perbaikan untuk melakukan pencegahan atau penanganan *bug* yang sama. Selain itu, adanya kesalahan komunikasi pada alur proses bisnis existing dapat terjadi dan menyebabkan perbedaan info yang diterima pada sisi Programmer, Quality Assurance, Customer service, serta Customer.

### 3.1.2. Identifikasi Masalah

#### Identifikasi Masalah Dengan Metode Fishbone



Gambar 4.2 Fishbone Diagram

Pada tahap ini menjelaskan terkait permasalahan yang ada proses yang selama ini dilakukan oleh programmer di PT Intimap dengan menggunakan fishbone diagram. Diagram ini disebut dengan fishbone dikarenakan bentuk dari diagram tersebut merujuk kepada bentuk tulang ikan yang memiliki ruas ruas tulang yang berperan sebagai titik atau nilai permasalahan yang terjadi serta solusi yang akan diakomodasi untuk menanggulangi permasalahan tersebut. Bentuk moncong ke depan seperti kepala ikan melambangkan sistem yang dirancang untuk permasalahan yang ada.

Didasari dengan penggunaan diagram fishbone ini telah dilakukan analisa yang menghasilkan empat poin permasalahan yang akan ditelusuri untuk mencari akar permasalahan serta sebagai acuan untuk menciptakan solusi yang akan disediakan. Empat poin permasalahan ini meliputi *Time*, *Quality*, *Tracking*, *Documentation*. Maka dari itu dibuatlah perancangan aplikasi. Dari hasil analisa tersebut maka dibuatkan perancangan aplikasi RANCANG BANGUN SISTEM

PENGECEKAN BUG APLIKASI MENGGUNAKAN API BERBASIS WEB yang akan diimplementasikan untuk permasalahan yang terdapat pada PT Intimap.

### **3.2. Analisa Kebutuhan dan Rancangan Kebutuhan**

#### **3.2.1. Analisa kebutuhan sistem**

tahap awal dimana analisis dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Tahap ini berguna untuk memutuskan sesuai atau tidak sesuainya komponen-komponen kebutuhan sistem yang akan digunakan.

Berdasarkan analisa permasalahan yang telah dijelaskan maka secara kebutuhan fungsionalitas sebagai berikut :

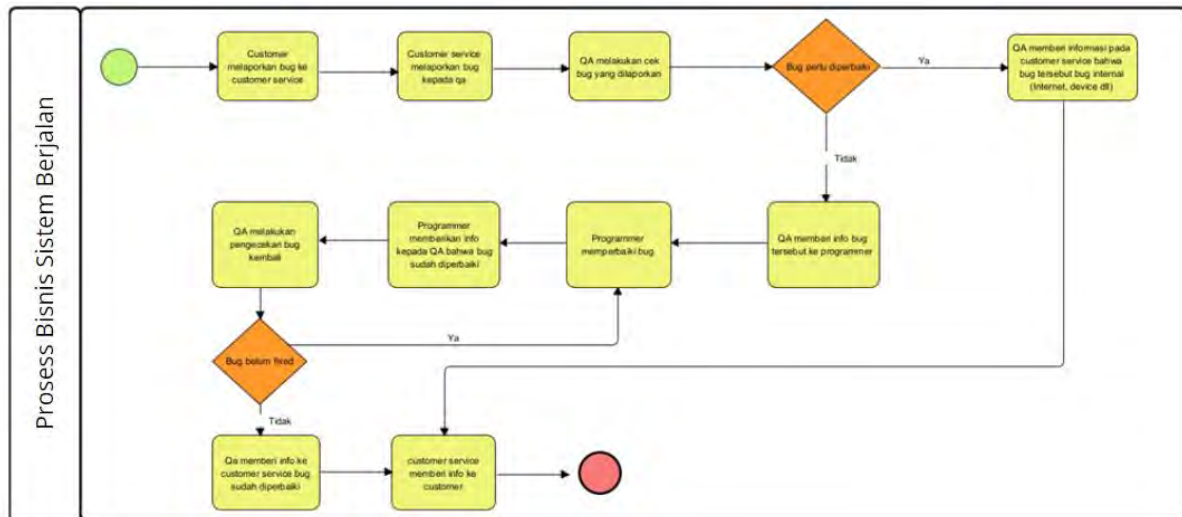
1. QA dapat login dan memberi tugaskan bug kepada programmer, lalu sistem akan menampilkan bug tersebut di halaman My Task Programmer.
2. Programmer dapat login dan mengecek bug yang telah ditugaskan dan dicek oleh QA.
3. QA dan Programmer dapat melihat history bug sebagai rekam jejak atau dokumentasi bug yang terjadi.
4. Dibutuhkannya sistem yang mendukung untuk kepentingan pengolahan informasi baik berupa dashboard, dan rekam jejak bug.
5. Sistem dibuat berbasis web dimana user mengakses sistem tersebut secara global/ melalui internet



#### **3.2.2. Proses Bisnis Rancangan Kebutuhan Sistem Informasi**

Berdasarkan analisa proses bisnis yang sudah berjalan dan analisa kebutuhan yang ingin dicapai, maka terbentuklah rancangan proses bisnis untuk menyesuaikan terhadap sistem yang akan dibuat.

1. Berikut rancangan proses bisnis aplikasi



**Gambar 4.3 BPMN Rancangan Sistem Informasi**

Rancangan analisa proses bisnis pada sistem informasi pengecekan bug ini mengedepankan kebutuhan untuk *manage* bug yang terjadi pada setiap pembuatan aplikasi dengan memberikan kemudahan dalam melakukan *tracking bug* baik dari *bug* yang perlu diperbaiki maupun *bug* yang sudah diperbaiki sebagai dokumentasi perbaikan *bug*. Pada proses analisa ini mempunyai QA untuk melakukan testing dari *bug* yang telah diperbaiki oleh *programmer* untuk memastikan *bug* sudah diperbaiki, hal ini juga berlaku bila *bug* tersebut belum dapat dikatakan *fixed*, QA dapat lebih mudah untuk melakukan *follow up* kepada programmer terkait untuk memperbaiki kembali *bug* yang masih ada. Proses ini memberikan efisiensi waktu dalam pengecekan, pengerjaan, perbaikan, sampai aplikasi dapat dikatakan siap untuk selanjutnya diteruskan kepada user terkait sehingga aplikasi telah dapat digunakan oleh user.

### 3.2.3. Alur proses kerja API ke DB (Backend)

Website yang digunakan untuk mendapat syntax error ini adalah Hawk GPS Intellitrac. Web ini dibuat menggunakan PHP. Untuk mendapatkan bug tersebut, Programmer menggunakan logger sebagai bantuannya. Logger yang digunakan menggunakan framework dari PHP, yaitu Monolog. Semua logger yang dihasilkan dari web tersebut dikumpulkan menjadi satu kedalam file. Semua data log yang ada di file tersebut kemudian di import ke dalam database yang mana akan digunakan dalam pembuatan API. Untuk menggunakan API di Vue.js, Programmer menggunakan Axios, yaitu http client untuk browser, cara kerjanya sama seperti AJAX. Jika Programmer ingin mendapatkan data dari API, Programmer menggunakan metode GET request dari Axios, dan Programmer masukan URL dari API GET data (contoh API GET active issue, [http://localhost:3002/active\\_issues](http://localhost:3002/active_issues)).

### 3.3. Perancangan UML

#### 3.3.1. Usecase Diagram

Sebelum membuat use case digaram, penulis melakukan identifikasi dahulu daftar dari aktor dan usecase dari tiap tiap module yang terdapat pada use case diagram. berikut ini adalah daftar dari usecase dan aktor pada perancangan sistem informasi yang penulis usulkan:

**Tabel 4.1 Keterangan Usecase**

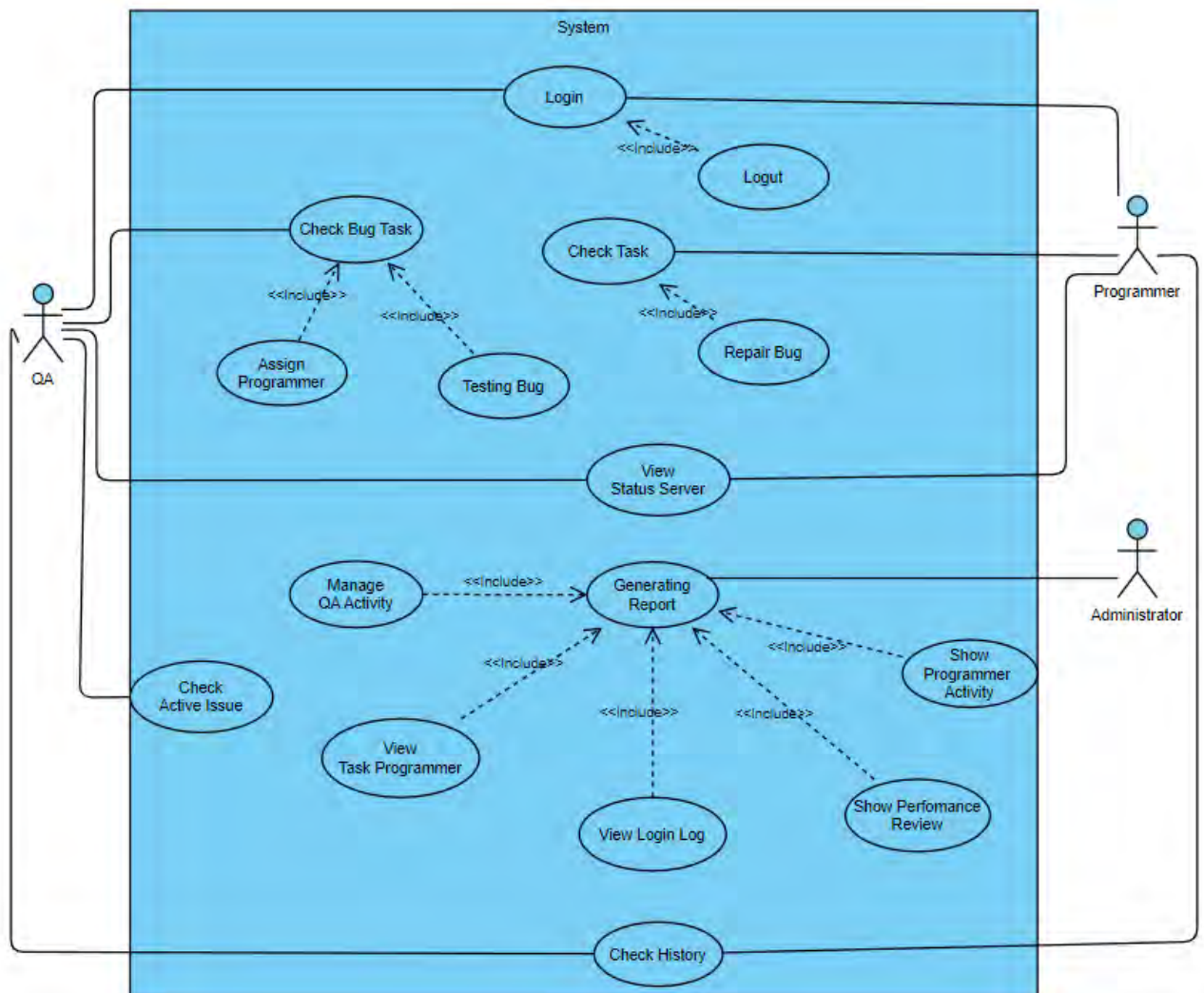
No	Use case	Definisi Use Case	Deskripsi
1	Login	Halaman pertama user masuk kedalam sistem	User akan memasukan credential atau username password untuk memasuki sistem
2	Check Task	Halaman untuk pengecekan task yang di assign pada programmer	Programmer masuk kedalam halaman ini untuk melihat task apa saja yang diassign kepada programmer itu
3	Repair Bug	Halaman untuk update status bug oleh programmer	Programmer akan melakukan perbaikan pada bug yang telah diassign oleh QA. Setelah itu programmer akan melakukan update status dari bug yang telah diperbaiki.
4	Check Task Bug	Halaman untuk QA melihat status bug	QA melakukan pengecekan dari list bug yang dari masing masing status dan QA akan melakukan follow up kepada programmer untuk memperbaiki dan mengubah status bug
5	Assign Bug	Halaman Issue assignment	Qa akan melakukan assignment kepada programmer untuk mekalkukan perbaikan pada bug terdaftar
6	Testing Bug	Halaman testing bug setelah dilakukan perbaikan oleh programmer	Pada halaman ini QA akan melakukan testing terhadap bug yang telah diupdate statusnya oleh progreammer sebagai bug yang telah diperbaiki.
7	Check Active Issue	Halaman Pengecekan Activie Issue yang dilakukan oleh QA	Monitor Active Issue yang masih dalam progress ataupun yang sudah selesai
8	Check History	Halaman History akan menampilkan history dari issue yang telah	QA dan Programmer melakukan pengecekan issue yang sudah pernah diperbaiki sebelumnya yang terkumpul dalam satu halaman

		dilakukan dari sisi QA maupun Programmer	
9	Status Server	Halaman yang menampilkan status server terkini	User melakukan pengecekan atau monitoring kondisi server untuk memastikan server dalam kondisi baik
10	Logout	Halaman untuk keluar dari sistem	User melakukan logout yang keluar dari sistem.

**Tabel 4.2 Keterangan Actor**

No	Actor	Usecase
1	QA	<ol style="list-style-type: none"> <li>1. Login</li> <li>2. Check Task Bug</li> <li>3. Assign Bug</li> <li>4. Testing Bug</li> <li>5. Check Active Issue</li> <li>6. Check History</li> <li>7. Logout</li> </ol>
2	Programmer	<ol style="list-style-type: none"> <li>1. Login</li> <li>2. Check Task</li> <li>3. Repair Bug</li> <li>4. Check History</li> </ol>
3	Administrator	<ol style="list-style-type: none"> <li>1. Generate Report</li> </ol>

Berikut adalah diagram use case untuk masing-masing actor didalam sistem, yaitu QA dan Programmer :



Gambar 4.4 Usecase Diagram

### 3.3.2. Dekripsi Usecase

Pada sub bab ini penulis menjelaskan tentang rincian use case diagram sehingga dapat memudahkan oembaca dalam memahami dan dianalisa oleh pembaca. Berikut ini adalah penjelasan usecase diagram dari perancangan sistem informasi yang penulis usulkan.



### a. Deskripsi Aktor

Tabel 4.3 Deskripsi Aktor

Aktor	Deskripsi
QA	Pengguna aplikasi dengan kepentingan memeriksa bug yang masih terdapat dalam program dan memberi tugas perbaikan kepada programmer
Programmer	Pengguna aplikasi sebagai eksekutor dari tugas perbaikan yang telah diberikan oleh QA
Administrator	Pengguna aplikasi sebagai orang yang dapat melihat dan memantau report bug dalam aplikasi

### b. Deskripsi Usecase Login

Tabel 4.4 Deskripsi Usecase Login

Nama	<i>Use Case Login</i>
Deskripsi Singkat	Modul untuk user masuk kedalam sistem
Pengguna	QA & Programmer
Kondisi Awal	Tampilan <i>Login Username</i> dan <i>Password</i>
Alur Utama	1. <i>User</i> membuka aplikasi 2. Sistem menampilkan halaman <i>login</i> 3. <i>User</i> memasukan <i>username</i> dan <i>password</i>
Alur Alternatif	-
Kondisi Akhir	Sistem menampilkan <i>dashboard</i> terkait sistem yang menampilkan bug terdaftar yang perlu dilakukan perbaikan dan status bug lainnya

### c. Deskripsi Usecase Assigned

Tabel 4.5 Deskripsi Usecase Assigned

Nama	<i>Use Case Assigned</i>
Deskripsi Singkat	<i>User</i> masuk menu Assigned dengan fungsi QA menugaskan sebuah bug ke programmer.
Pengguna	QA

Kondisi Awal	Tampilan bug yang telah terdaftar dengan status <i>Open</i>
Alur Utama	<ol style="list-style-type: none"> <li>1. <i>User</i> klik menu <i>dashboard</i> jumlah bug yang telah terdaftar dengan status <i>Open</i></li> <li>2. QA akan mengklik “Assigned to...” pada bug berstatus open</li> <li>3. Setelahnya QA akan memilih “Assign User”</li> <li>4. sistem akan mengirimkan bug ke Task List Programmer.</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Sistem memperbarui status bug “ <i>Open</i> ” menjadi “Assigned”.

#### d. Deskripsi Usecase Testing

Tabel 4.6 Deskripsi Usecase Testing

Nama	<i>Use Case Testing</i>
Deksripsi Singkat	Menu untuk memastikan bug telah diperbaiki, QA akan melakukan testing terhadap bug
Pengguna	QA
Kondisi Awal	Tampilan untuk memastikan <i>bug</i> telah diperbaiki.
Alur Utama	<ol style="list-style-type: none"> <li>1. <i>User</i> masuk menu <i>check task bug</i></li> <li>2. , QA akan melakukan <i>testing</i> terhadap <i>bug</i>.</li> <li>3. Jika dari test tersebut masih ditemui adanya <i>bug</i>, maka QA akan mengirimkan ulang <i>script</i> ke programmer dengan cara meng-klik “<i>Re-Assign User</i>”</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Setelah test tak ada lagi bug, maka QA akan memilih pilihan “Closed” yang menandakan jika bug ini telah diperbaiki dan diselesaikan. Sistem pun akan memperbarui status bug menjadi “CLOSED”

#### e. Deskripsi Usecase Check Task Bug

Tabel 4.7 Deskripsi Usecase Check Task Bug

Nama	<i>Use Case Check Task Bug</i>
------	--------------------------------

Deksripsi Singkat	Menggambarkan bug yang telah ditugaskan kepada <i>programmer</i> memiliki status yang telah <i>diupdate</i>
Pengguna	QA
Kondisi Awal	Tampilan daftar issue yang telah ditugaskan kepada programmer
Alur Utama	1. <i>User</i> masuk menu <i>Check Task Bug</i> 2. QA akan dialihkan ke halaman <i>Active issues</i> yang menampilkan <i>list bug</i> dengan status <i>terupdate</i> .
Alur Alternatif	-
Kondisi Akhir	QA melakukan follow up pada bug yang belum diperbaiki dan melakukan testing pada bug yang sudah diperbaiki.

#### f. Deskripsi Usecase Check Task

Tabel 4.8 Deskripsi Usecase Check Task

Nama	<i>Use Case Check Task</i>
Deksripsi Singkat	Menggambarkan bug yang telah ditugaskan kepada programmer yang telah di assign oleh QA
Pengguna	Programmer
Kondisi Awal	Tampilan issue yang telah ditugaskan kepada salah satu programmer
Alur Utama	1. <i>User</i> masuk menu <i>Check Task</i> 2. , QA memilih menu " <i>Check Task</i> " dari <i>dashboard web</i> 3. Programmer akan dialihkan ke halaman <i>Active issues</i> yang menampilkan <i>list bug</i> yang harus diperbaiki maupun pending oleh programmer atas penugasan dari QA.
Alur Alternatif	-
Kondisi Akhir	Programmer melakukan perbaikan pada bug yang telah ditugaskan dan mengubah status bug menjadi active

#### g. Deskripsi Usecase Active Issues

**Tabel 4.9 Deskripsi Usecase Active Issues**

Nama	<i>Use Case Active Issues</i>
Dekripsi Singkat	menggambarkan aktifitas saat QA memilih menu “Active Issues” dari <i>dashboard web</i> .
Pengguna	QA
Kondisi Awal	Tampilan daftar issue yang belum diperbaiki ataupun dalam status pending.
Alur Utama	1. <i>User</i> masuk menu <i>Active Issues</i> 2. , QA memilih menu “Active Issues” dari <i>dashboard web</i> 3. QA akan dialihkan ke halaman <i>Active issues</i> yang menampilkan <i>list bug</i> yang harus diperbaiki maupun pending.
Alur Alternatif	-
Kondisi Akhir	QA menugaskan sebuah bug ke programmer. QA akan memilih programmer terkait untuk memperbaiki bug tersebut.

#### **h. Deskripsi Usecase Repair Bug**

**Tabel 4.10 Deskripsi Usecase Repair Bug**

Nama	<i>Use Case Repair Bug</i>
Dekripsi Singkat	Programmer melakukan perbaikan bug yang telah ditugaskan oleh QA
Pengguna	Programmer
Kondisi Awal	Programmer mengakses daftar bug dari halaman “ <i>my task</i> ”.
Alur Utama	1. <i>User</i> mengakses halaman “ <i>my task</i> ” 2. Sistem akan menampilkan semua bug yang ditugaskan pada programmer tersebut 3. Programmer dapat memilih salah satu bug dan memperbaikinya
Alur Alternatif	-
Kondisi Akhir	Programmer memperbaiki dan memilih “Testing” untuk mengirimkan hasil perbaikan bug ke QA untuk di lakukan test.

#### **i. Deskripsi Usecase Status Server**

**Tabel 4.11 Deskripsi Usecase Status Server**

Nama	<i>Use Case Status Server</i>
Deksripsi Singkat	Monitoring status server
Pengguna	QA, Programmer
Kondisi Awal	Tampilan dashboard
Alur Utama	4. <i>User</i> mengakses ke dalam dashboar 5. <i>User</i> memilih menu status server 6. <i>System</i> menampilkan status server terkini
Alur Alternatif	-
Kondisi Akhir	Status server ditampilkan oleh sistem untuk dilakukan pemantauan oleh user

**j. Deskripsi Usecase Generate Report**

**Tabel 4.12 Deskripsi Usecase Generate Report**

Nama	<i>Use Case Generate Report</i>
Deksripsi Singkat	User melakukan generate report untuk melihat report
Pengguna	administrator
Kondisi Awal	Administrator mengakses modul generate report
Alur Utama	1. User masuk kedalam dashboard 2. User melakukan klik menu pada modul generate report 3. Report akan ditampilkan oleh sistem untuk melakukan generate report
Alur Alternatif	-
Kondisi Akhir	Menampilkan pilihan report yang akan dilakukan generate

**k. Deskripsi Usecase QA Activity**

**Tabel 4.13 Deskripsi Usecase QA Activity**

Nama	<i>Use Case QA Activity</i>
Deksripsi Singkat	User melakukan generate report QA Activity

Pengguna	administrator
Kondisi Awal	Administrator melakukan generate report QA Activity
Alur Utama	<ol style="list-style-type: none"> <li>1. User masuk kedalam dashboard</li> <li>2. User melakukan klik menu pada modul generate report</li> <li>3. Report akan ditampilkan oleh sistem untuk melakukan generate report</li> <li>4. User memilih report yang akan dilakukan generate</li> <li>5. Report generated</li> <li>6. Menampilkan report QA Activity</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Menampilkan data QA Activity report

## I. Deskripsi Usecase Programmer Activity

**Tabel 4.14 Deskripsi Usecase Programmer Activity**

Nama	<i>Use Case</i> Programmer Activity
Deksripsi Singkat	User melakukan generate report Programmer Activity
Pengguna	administrator
Kondisi Awal	Administrator melakukan generate report Programmer Activity
Alur Utama	<ol style="list-style-type: none"> <li>1. User masuk kedalam dashboard</li> <li>2. User melakukan klik menu pada modul generate report</li> <li>3. Report akan ditampilkan oleh sistem untuk melakukan generate report</li> <li>4. User memilih report yang akan dilakukan generate</li> <li>5. Report generated</li> <li>6. Menampilkan report Programmer Activity</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Menampilkan data Programmer Activity report

## m. Deskripsi Usecase Review Kinerja

**Tabel 4.15 Deskripsi Usecase Review Kinerja**

Nama	<i>Use Case</i> Review Kinerja
Deksripsi Singkat	User melakukan generate report Review Kinerja

Pengguna	administrator
Kondisi Awal	Administrator melakukan generate report Review Kinerja
Alur Utama	<ol style="list-style-type: none"> <li>1. User masuk kedalam dashboard</li> <li>2. User melakukan klik menu pada modul generate report</li> <li>3. Report akan ditampilkan oleh sistem untuk melakukan generate report</li> <li>4. User memilih report yang akan dilakukan generate</li> <li>5. Report generated</li> <li>6. Menampilkan report Review Kinerja</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Menampilkan data Review Kinerja report

#### n. Deskripsi Usecase Programmer Activity

**Tabel 4.16 Deskripsi Usecase Programmer Activity**

Nama	<i>Use Case Programmer Activity</i>
Deskripsi Singkat	User melakukan generate report Programmer Activity
Pengguna	administrator
Kondisi Awal	Administrator melakukan generate report Programmer Activity
Alur Utama	<ol style="list-style-type: none"> <li>1. User masuk kedalam dashboard</li> <li>2. User melakukan klik menu pada modul generate report</li> <li>3. Report akan ditampilkan oleh sistem untuk melakukan generate report</li> <li>4. User memilih report yang akan dilakukan generate</li> <li>5. Report generated</li> <li>6. Menampilkan report Programmer Activity</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Menampilkan data Programmer Activity report

#### o. Deskripsi Usecase Total Task Programmer

**Tabel 4.17 Deskripsi Usecase Total Task Programmer**

Nama	<i>Use Case Total Task Programmer</i>
------	---------------------------------------

Deksripsi Singkat	User melakukan generate report Total Task Programmer
Pengguna	administrator
Kondisi Awal	Administrator melakukan generate report Total Task Programmer
Alur Utama	<ol style="list-style-type: none"> <li>1. User masuk kedalam dashboard</li> <li>2. User melakukan klik menu pada modul generate report</li> <li>3. Report akan ditampilkan oleh sistem untuk melakukan generate report</li> <li>4. User memilih report yang akan dilakukan generate</li> <li>5. Report generated</li> <li>6. Menampilkan report Total Task Programmer</li> </ol>
Alur Alternatif	-
Kondisi Akhir	Menampilkan data Total Task Programmer report

**p. Deskripsi Usecase History**

**Tabel 4.18 Deskripsi Usecase History**

Nama	<i>Use Case History</i>
Deksripsi Singkat	Penampilkan halaman jika user mengakses halaman "History"
Pengguna	QA & Programmer
Kondisi Awal	<i>User</i> mengakses halaman history bug.
Alur Utama	<ol style="list-style-type: none"> <li>1. <i>User</i> mengakses halaman "history"</li> <li>2. Jika user adalah QA maka akan ditampilkan semua bug yang telah "Closed"</li> <li>3. jika user adalah Programmer, maka akan ditampilkan bug-bug yang pernah ia selesaikan atau perbaiki saja.</li> </ol>
Alur Alternatif	-
Kondisi Akhir	History list dari aktifitas penugasan oleh QA dan History list dari bug yang telah dilakukan oleh programmer.

**q. Deskripsi Usecase Logout**

**Tabel 4.19 Deskripsi Usecase Logout**

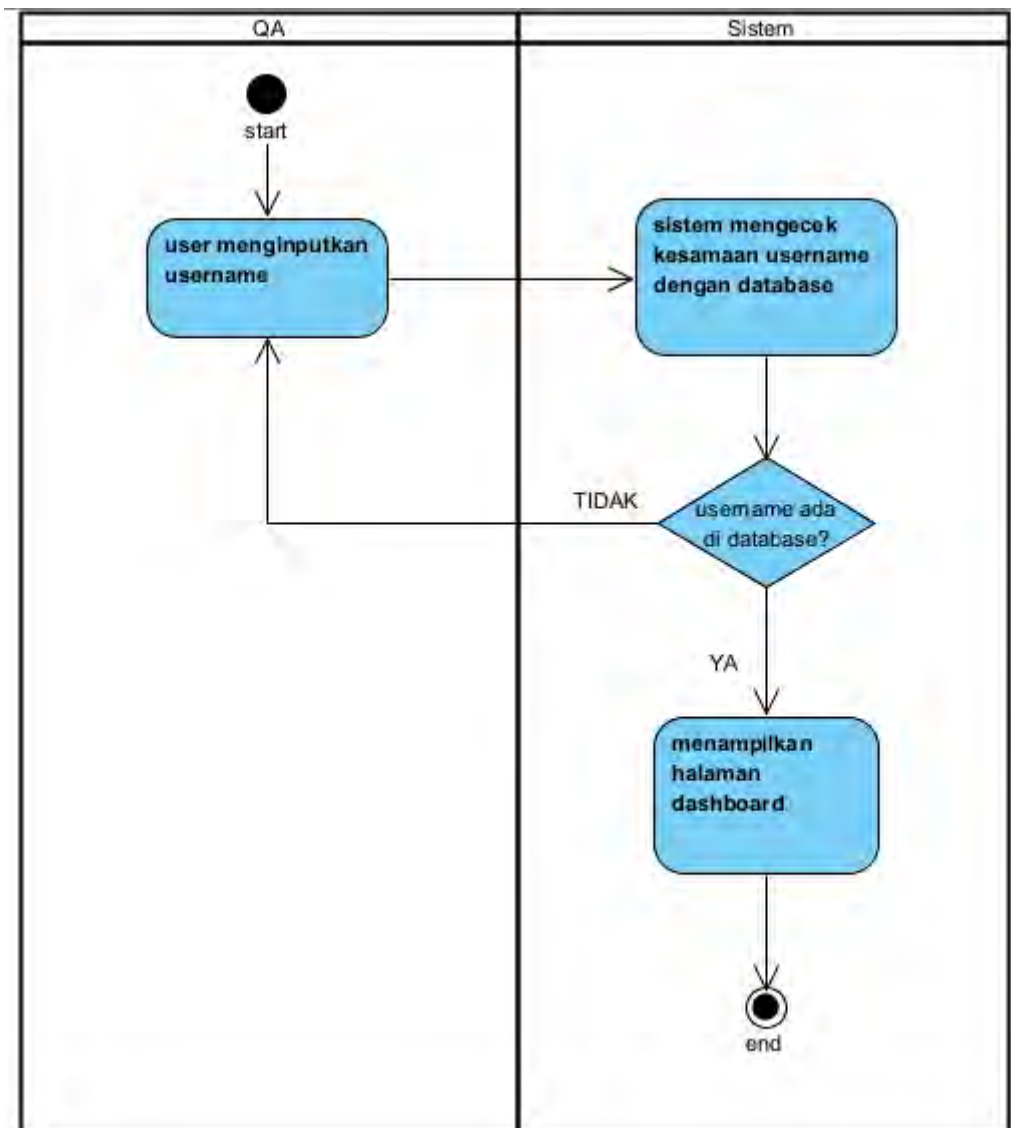


Nama	<i>Use Case Logout</i>
Deksripsi Singkat	Menggambarkan aktifitas user untuk keluar dari sistem
Pengguna	QA & Programmer
Kondisi Awal	Tampilan dashboard sistem sebelum melakukan logout
Alur Utama	<ol style="list-style-type: none"> <li>1. <i>User</i> telah selesai melakukan aktifitas pada system</li> <li>2. <i>User</i> mengklik pilihan logout untuk keluar dari system</li> <li>3. <i>User</i> telah keluar dari sistem</li> </ol>
Alur Alternatif	-
Kondisi Akhir	User keluar dari sistam dan kembali ke halaman login



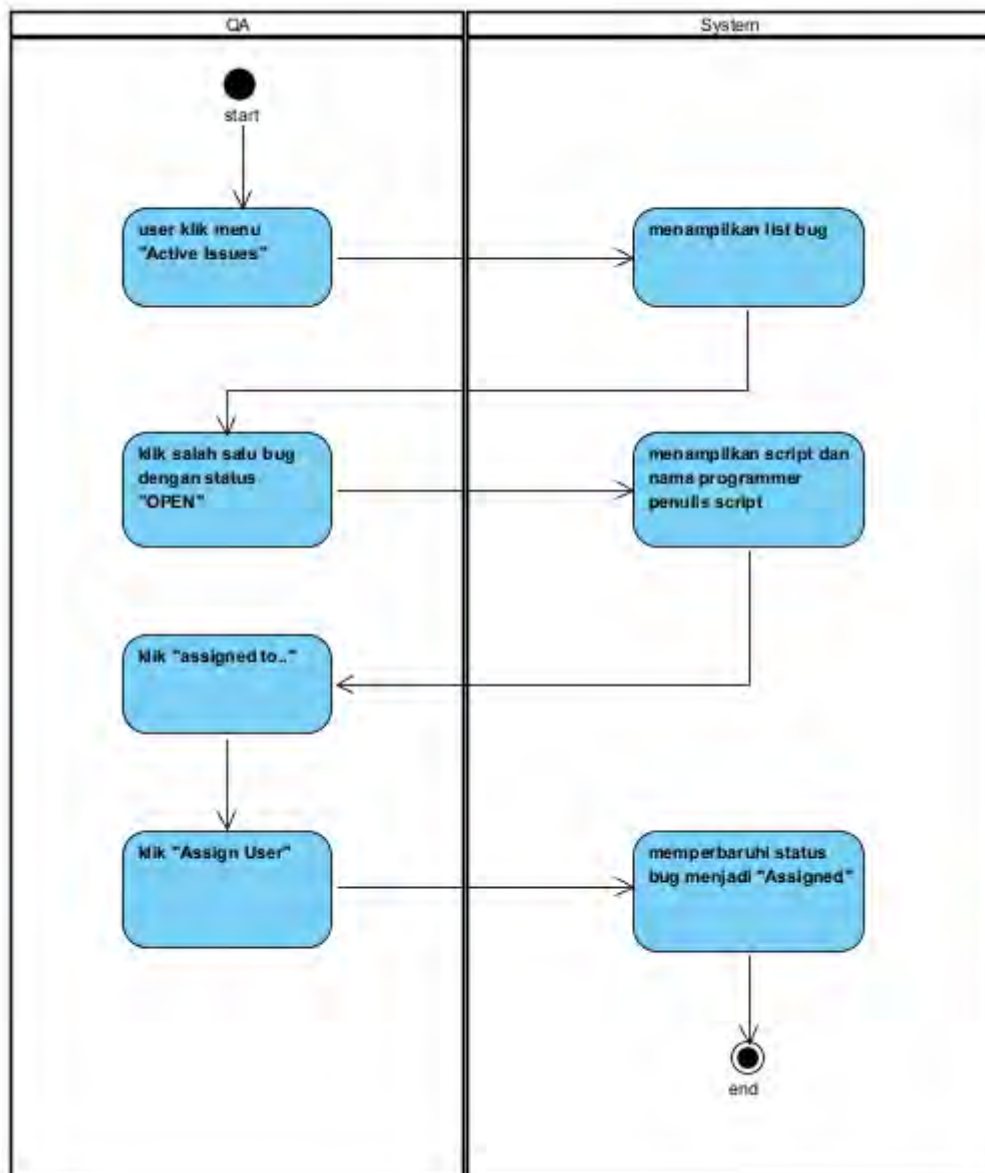
### 3.4. Activity Diagram

#### 3.4.1. Activity Diagram Login



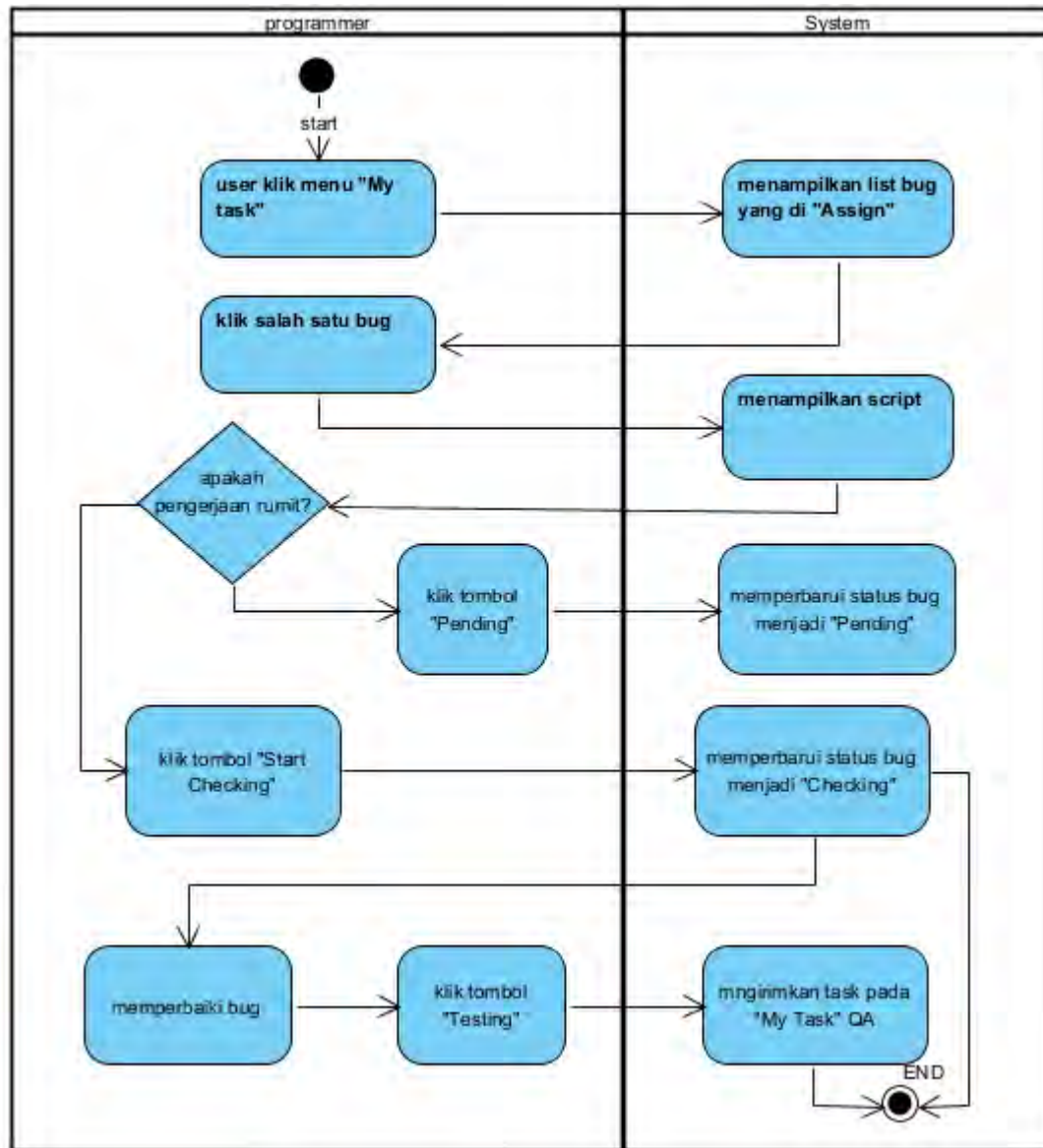
Gambar 4.5 Activity Diagram Login

### 3.4.2. Activity Diagram Check My Task (QA)



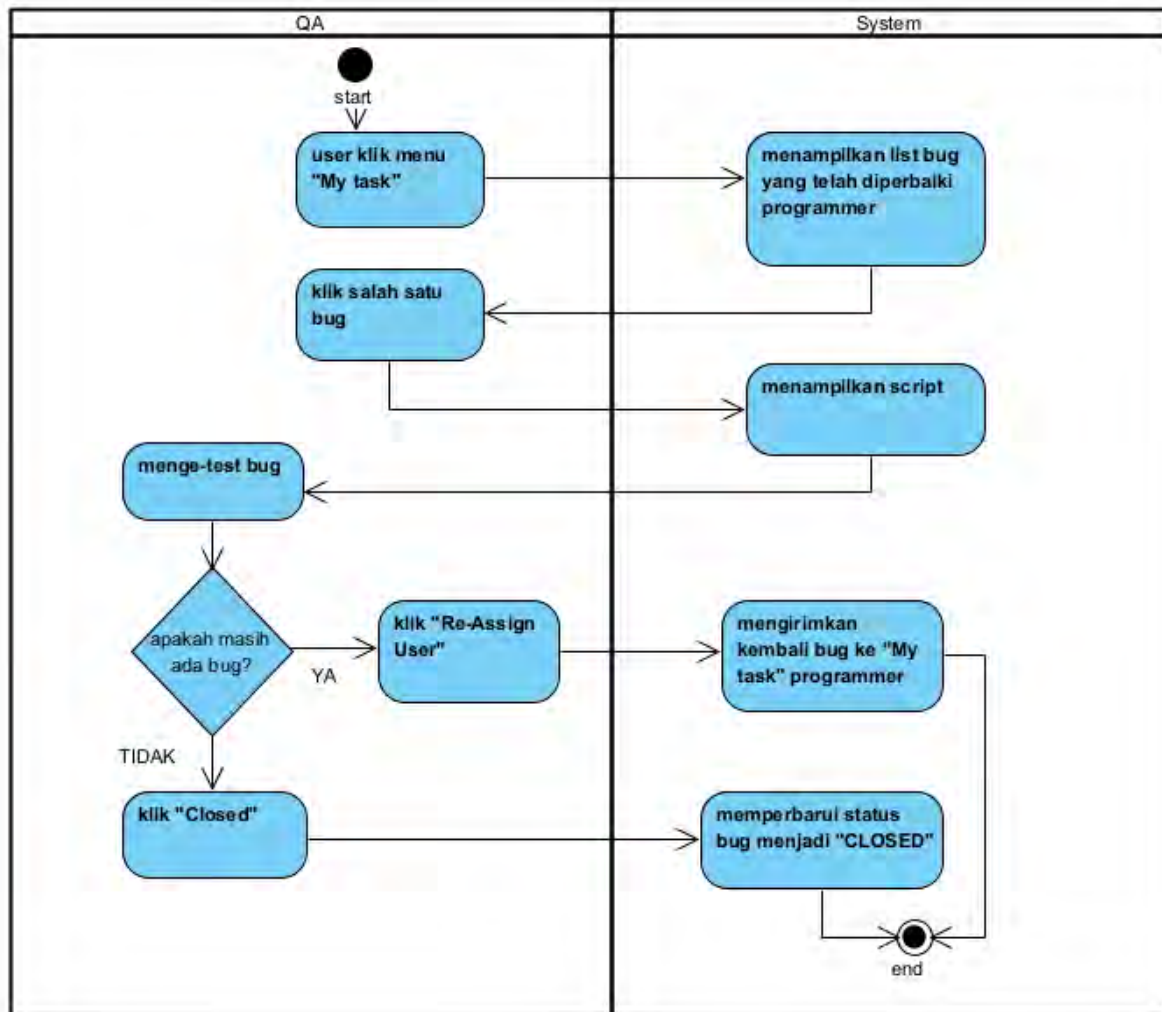
Gambar 4.6 Activity Diagram Check My Task (QA)

### 3.4.3. Activity Diagram Check My Task (Programmer)



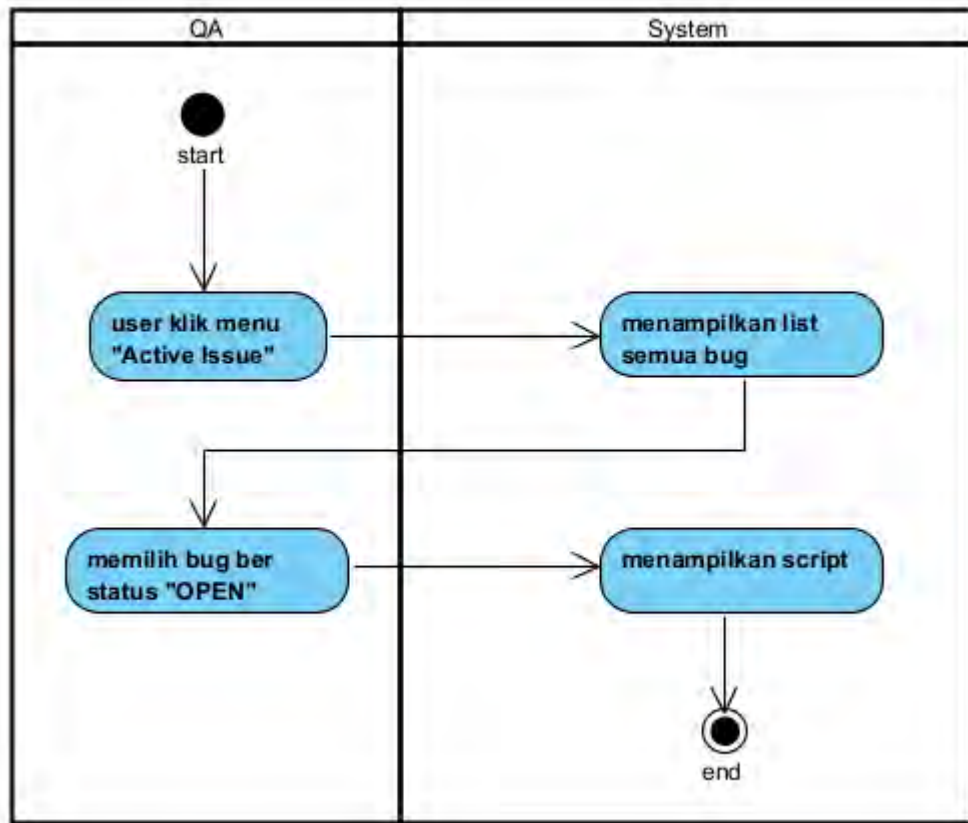
Gambar 4.7 Activity Diagram Check My Task (Programmer)

### 3.4.4. Activity Diagram Testing



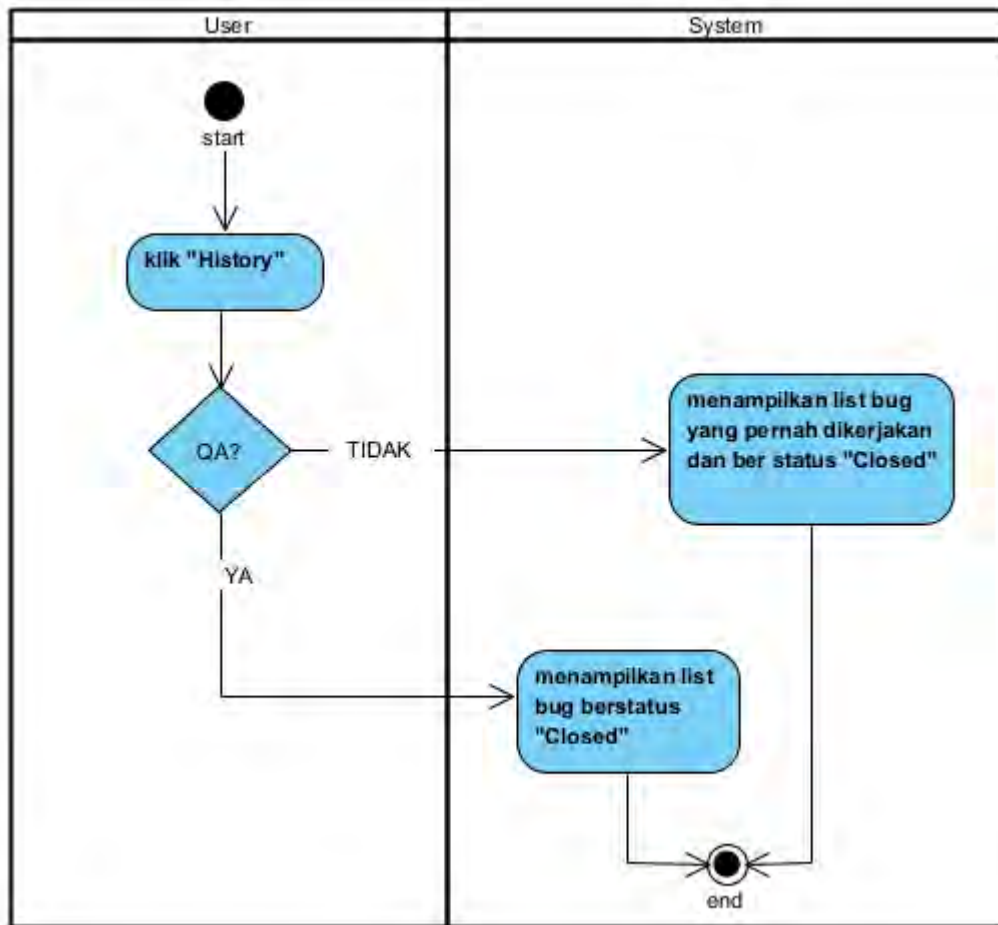
Gambar 4.8 Activity Diagram Testing

### 3.4.5. Activity Diagram Active Issue



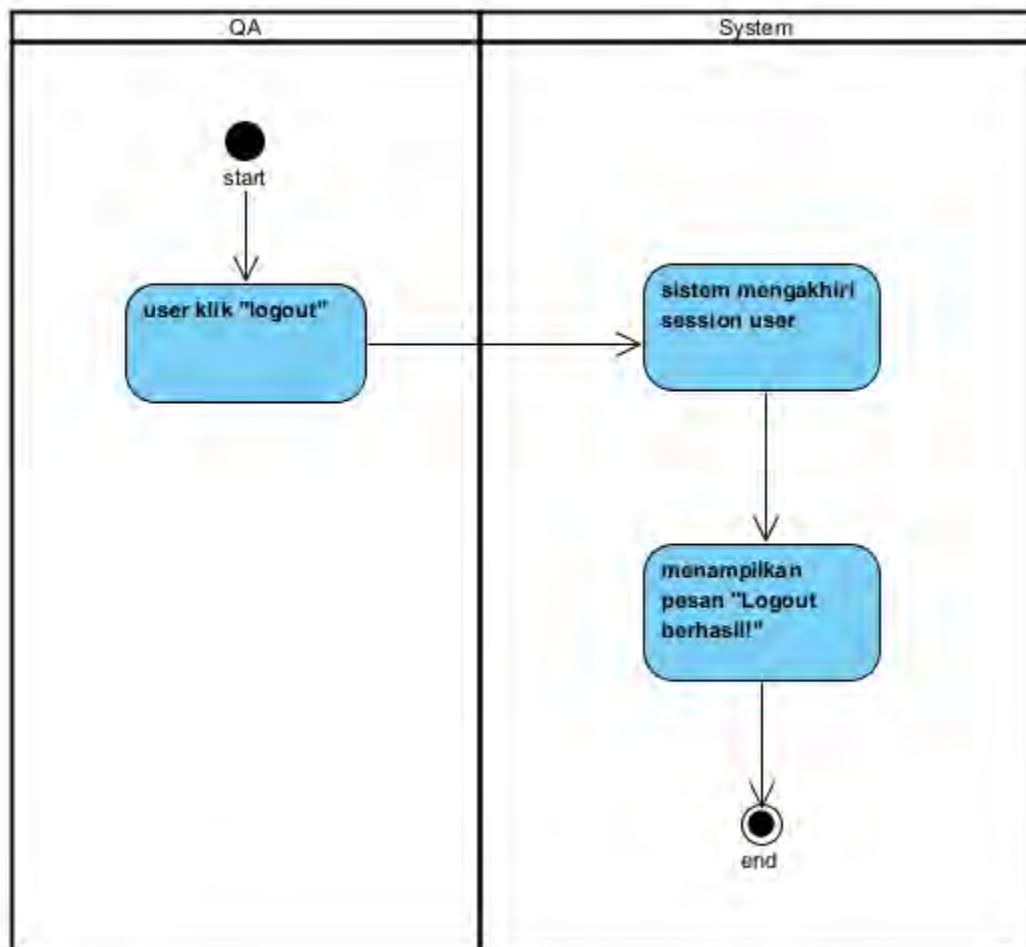
Gambar 4.9 Activity Diagram Active Issue

### 3.4.6. Activity Diagram History



MERCU BUANA  
Gambar 4.10 Activity Diagram History

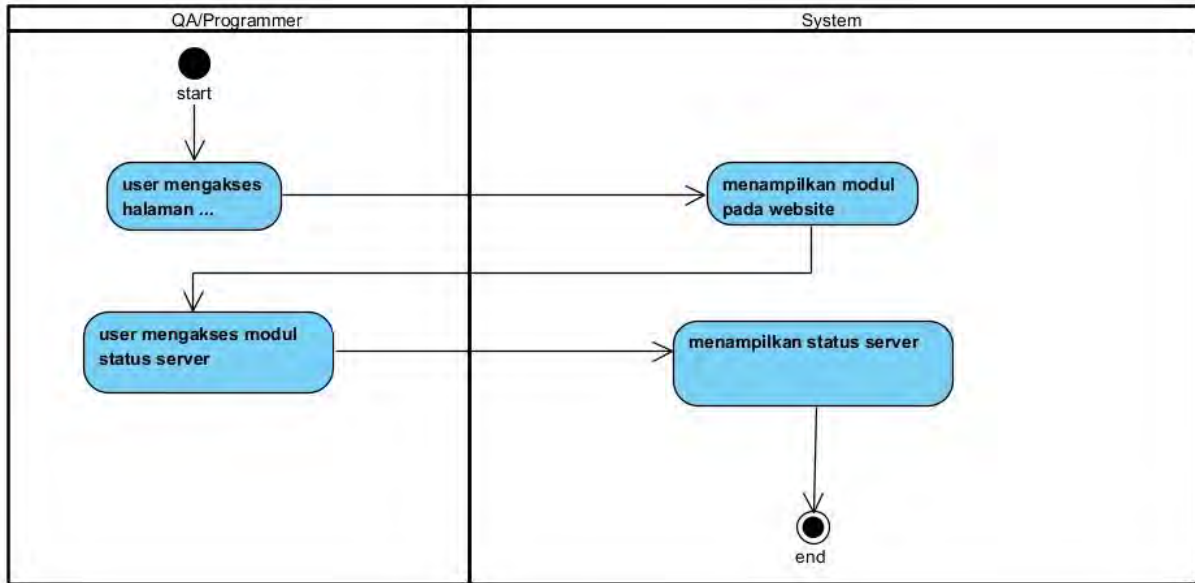
### 3.4.7. Activity Diagram Logout



Gambar 4.11 Activity Diagram Logout

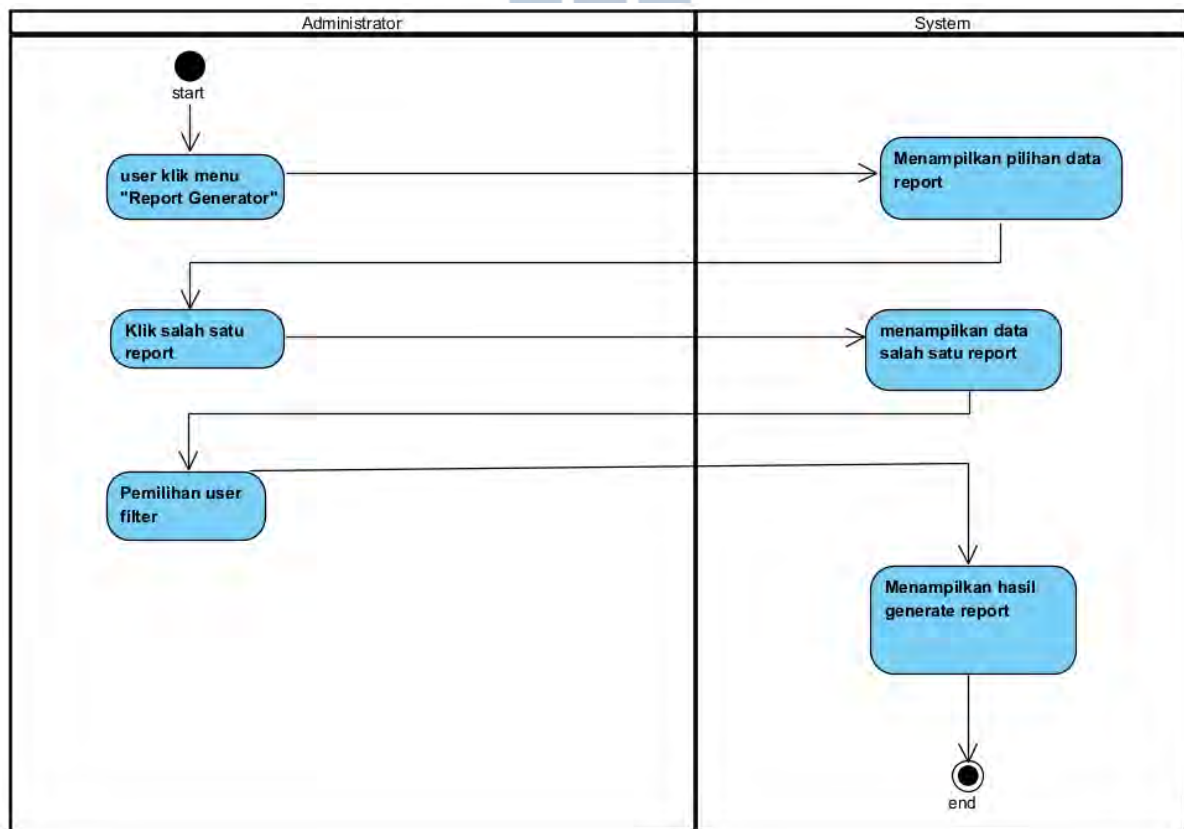


### 3.4.8. Activity Diagram Status Server



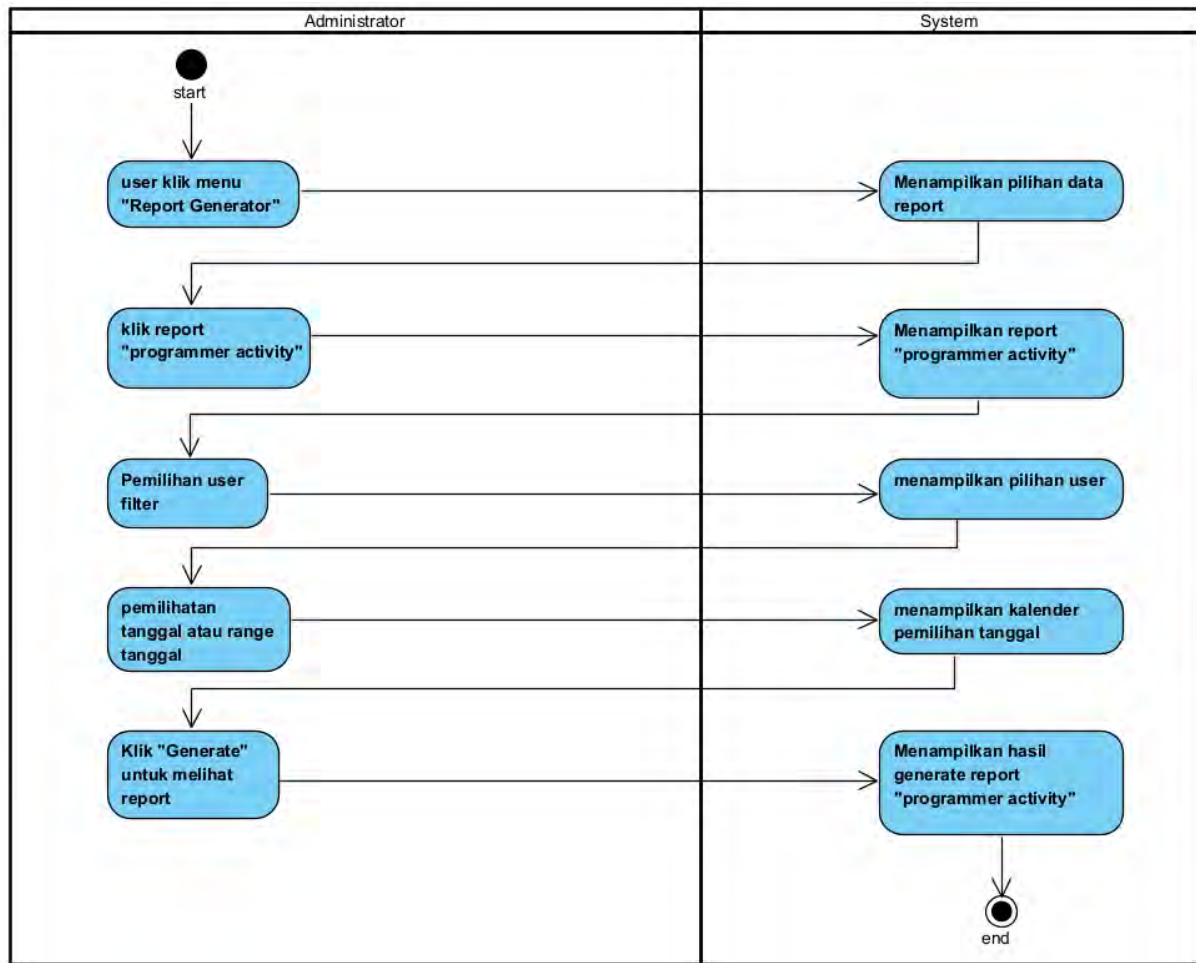
Gambar 4.12 Activity Diagram Status Server

### 3.4.9. Activity Diagram Generate Report



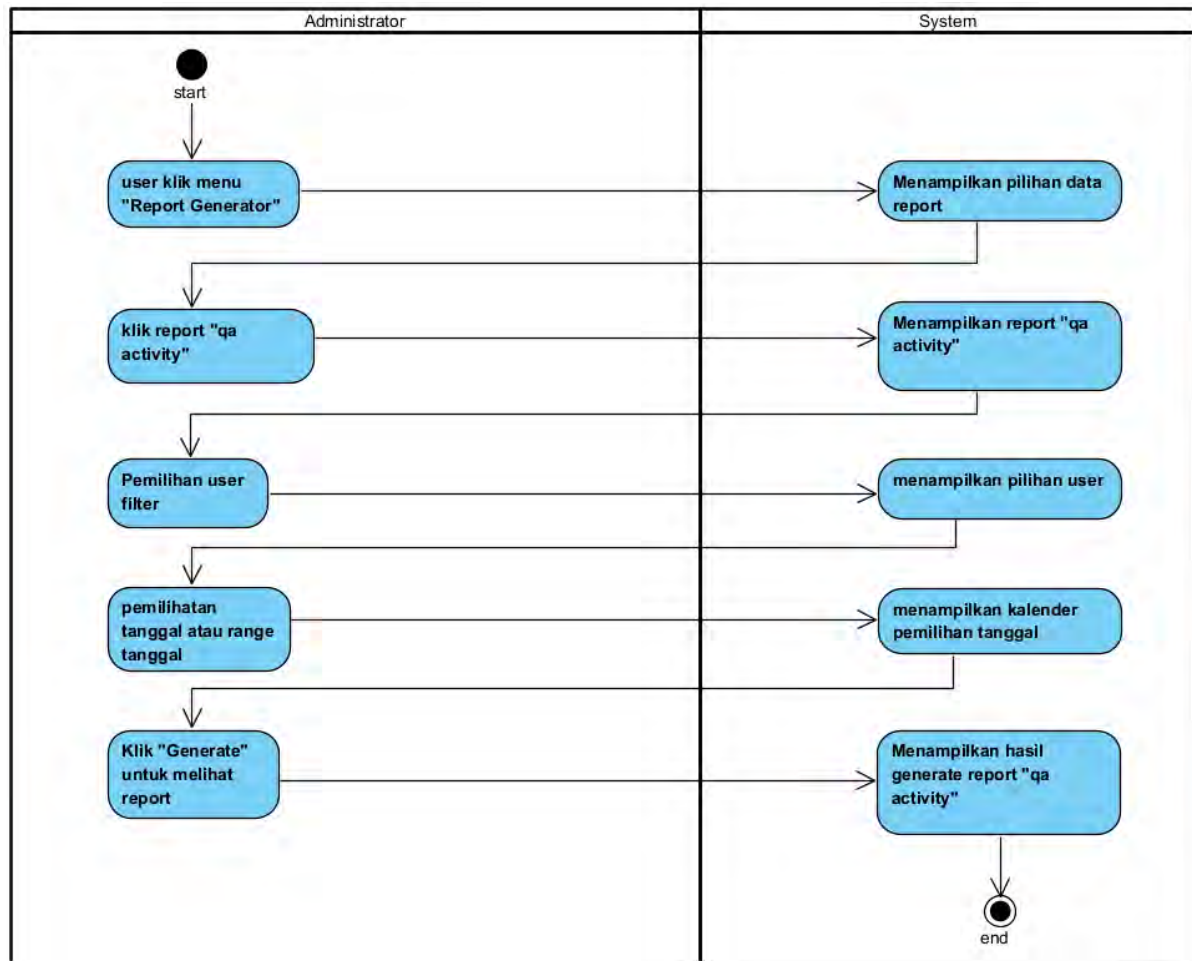
Gambar 4.13 Activity Diagram Generate Report

### 3.4.10. Activity Diagram Report Programmer Activity



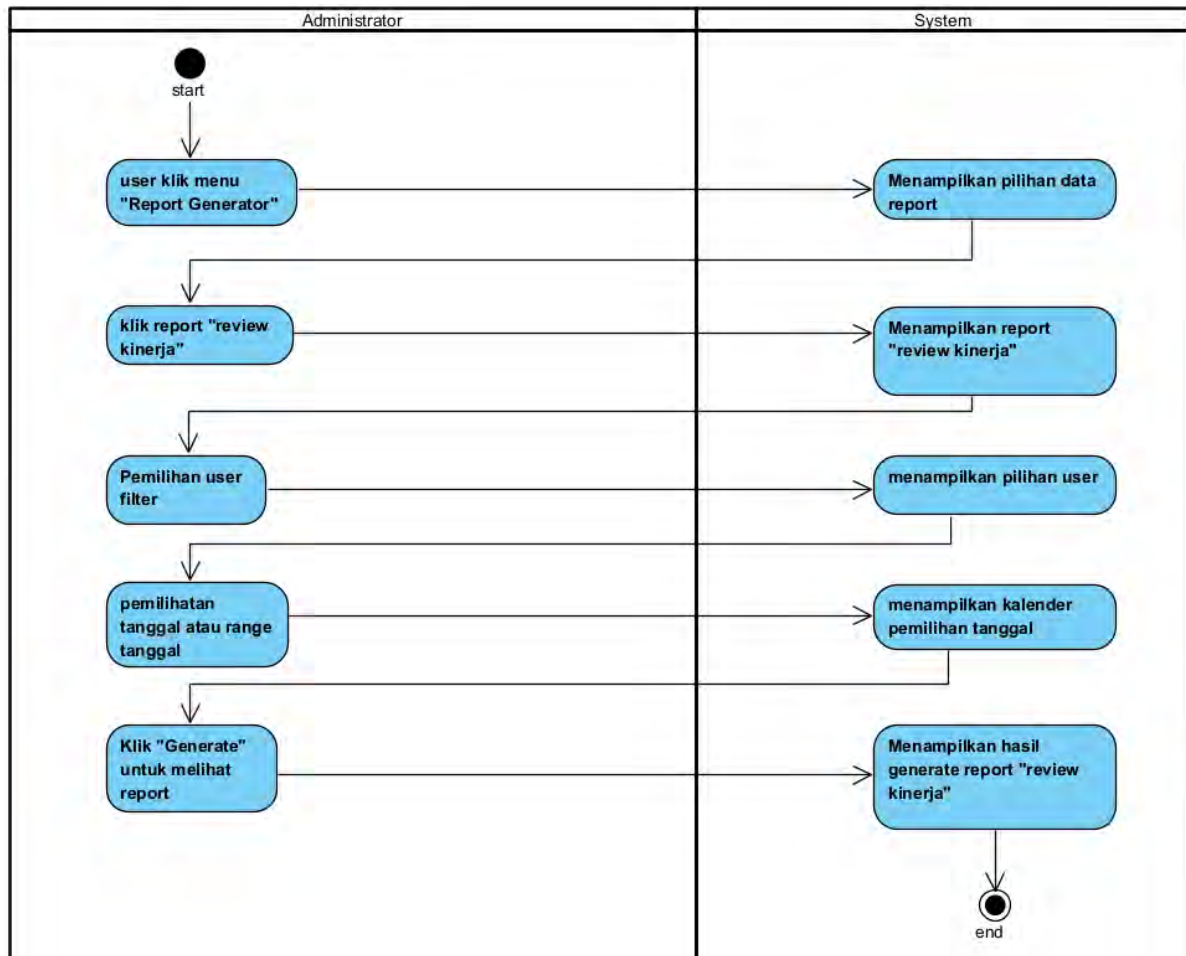
Gambar 4.14 Activity Diagram Report Programmer Activity

### 3.4.11. Activity Diagram Report QA Activity



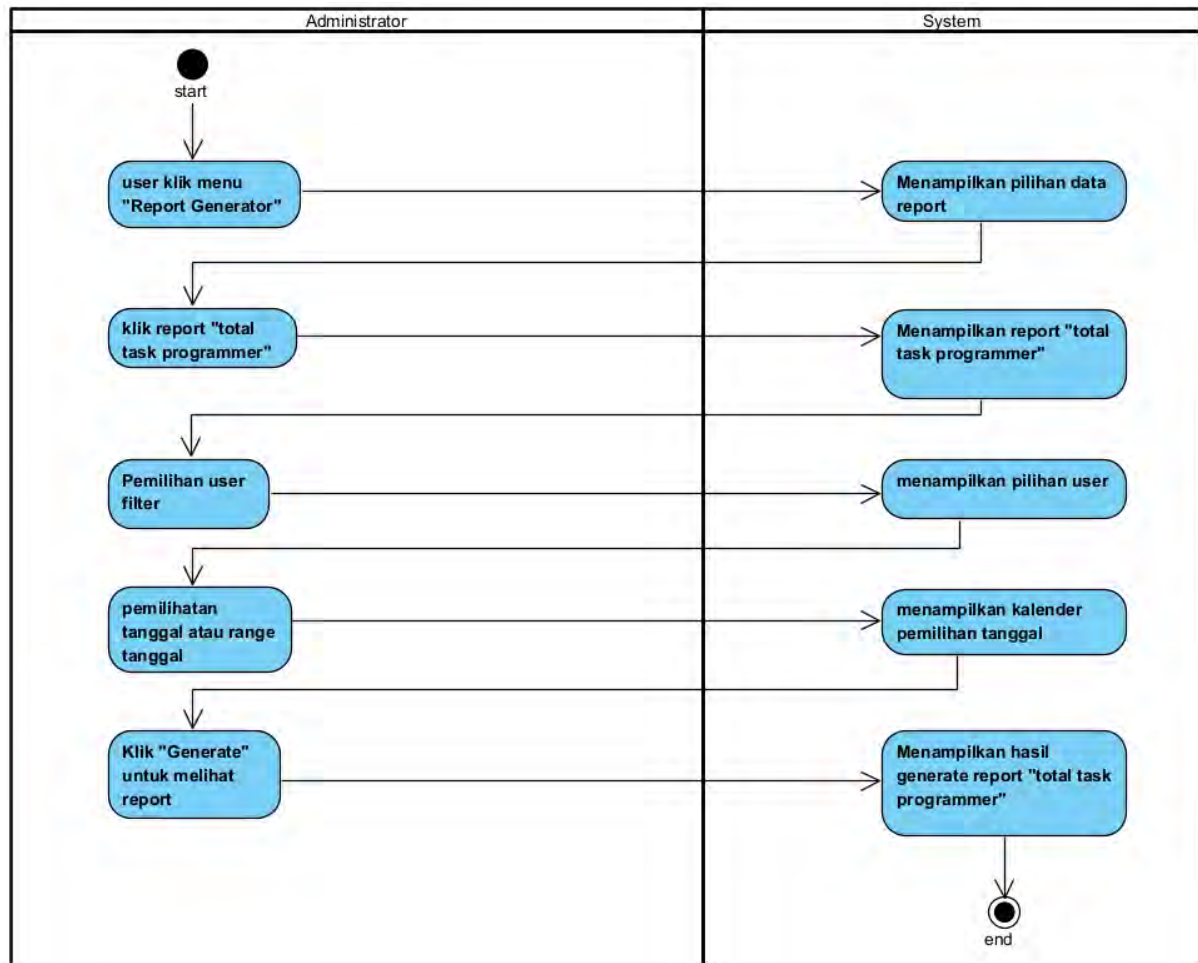
Gambar 4.15 Activity Diagram Report QA Activity

### 3.4.12. Activity Diagram Report Review Kinerja



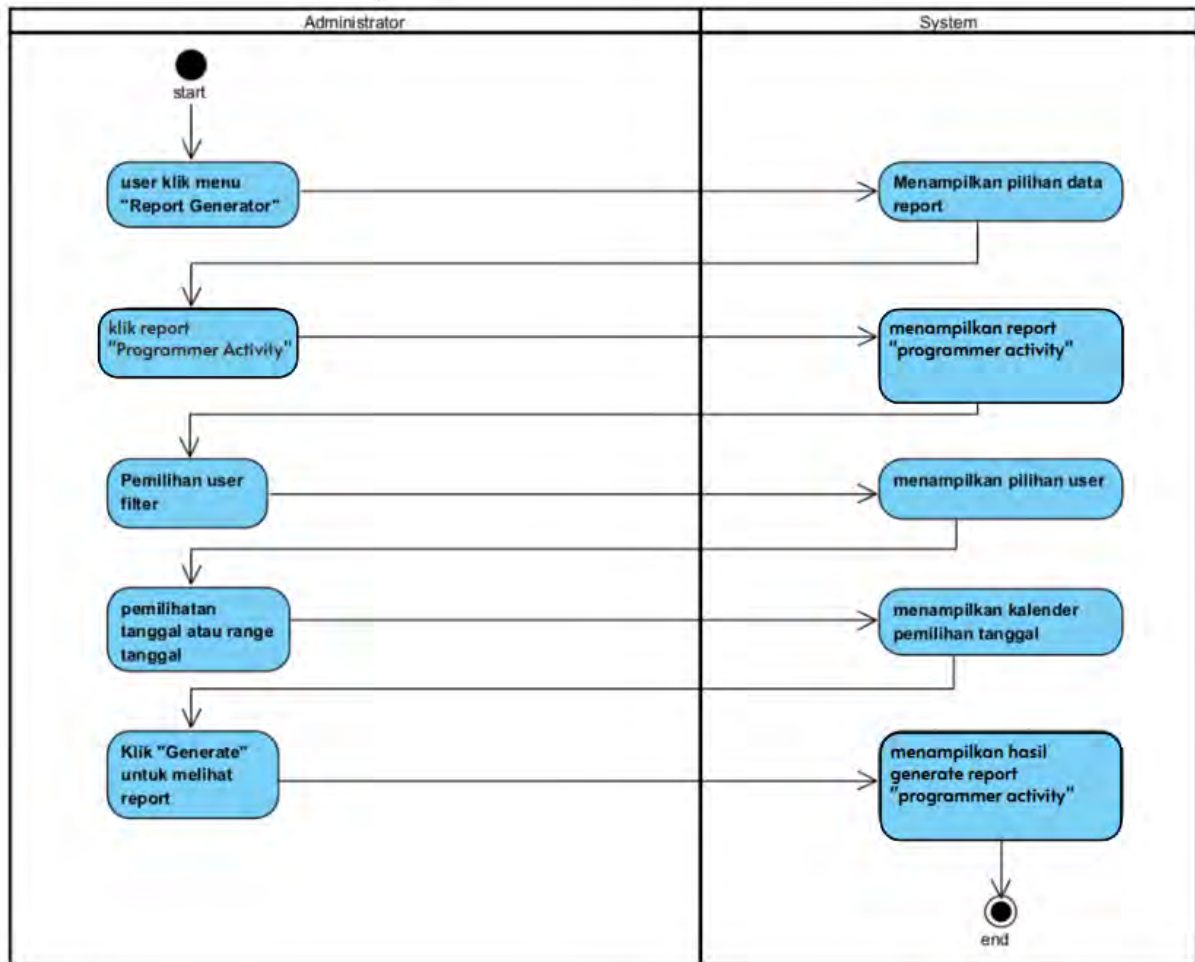
Gambar 4.16 Activity Diagram Report Review Kinerja

### 3.4.13. Activity Diagram Report Total Task Programmer



Gambar 4.17 Activity Diagram Report Total Task Programmer

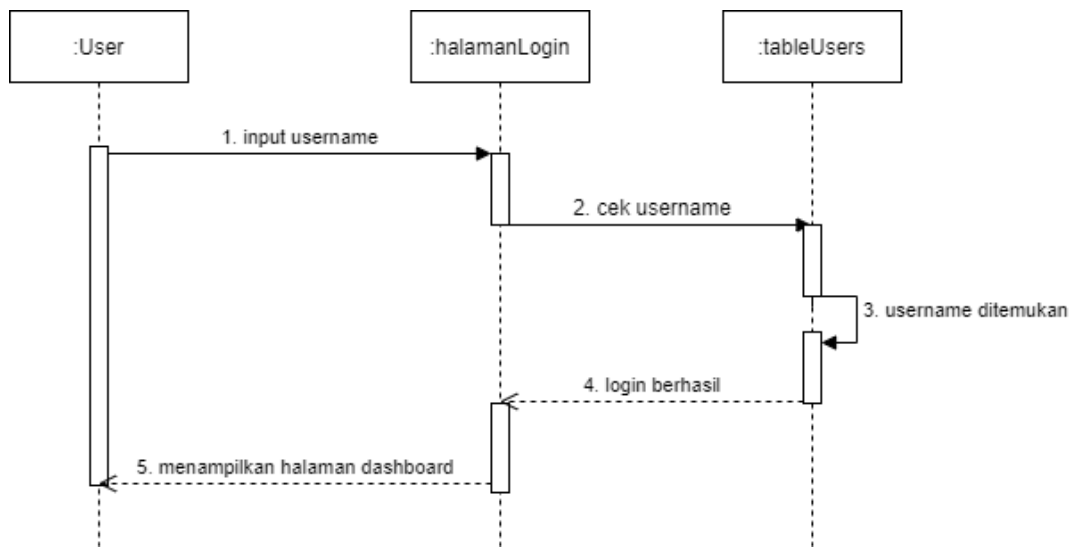
### 3.4.14. Activity Diagram Report Programmer Activity



Gambar 4.18 Activity Diagram Programmer Activity

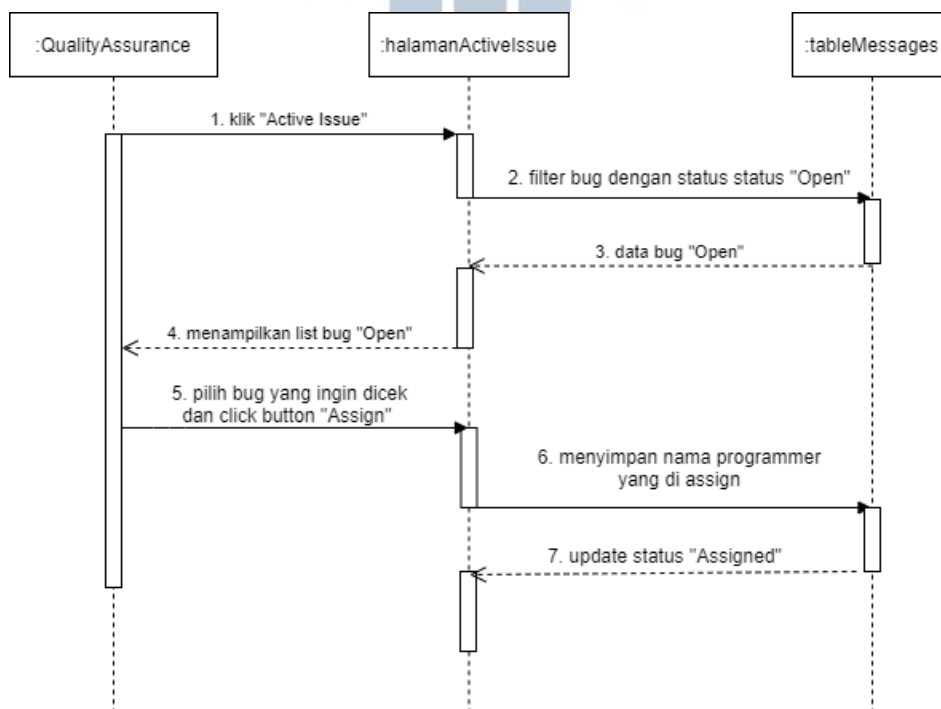
### 3.5. Sequence Diagram

#### 3.5.1. Sequence Diagram Login



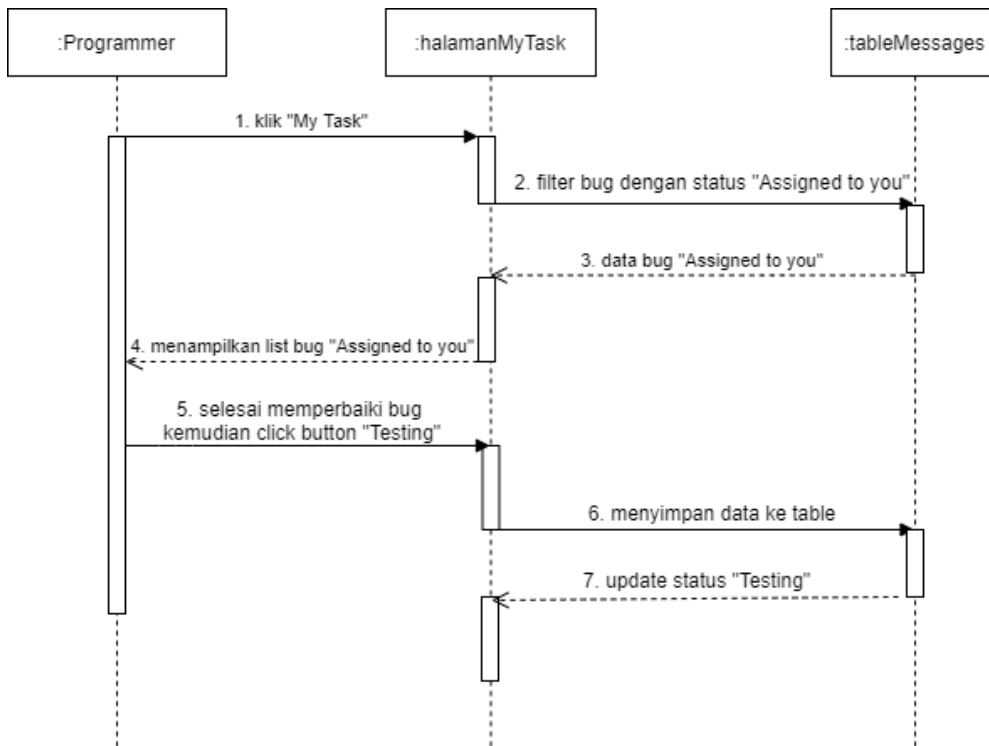
Gambar 4.19 Sequence Diagram Login

#### 3.5.2. Sequence Diagram Assigned



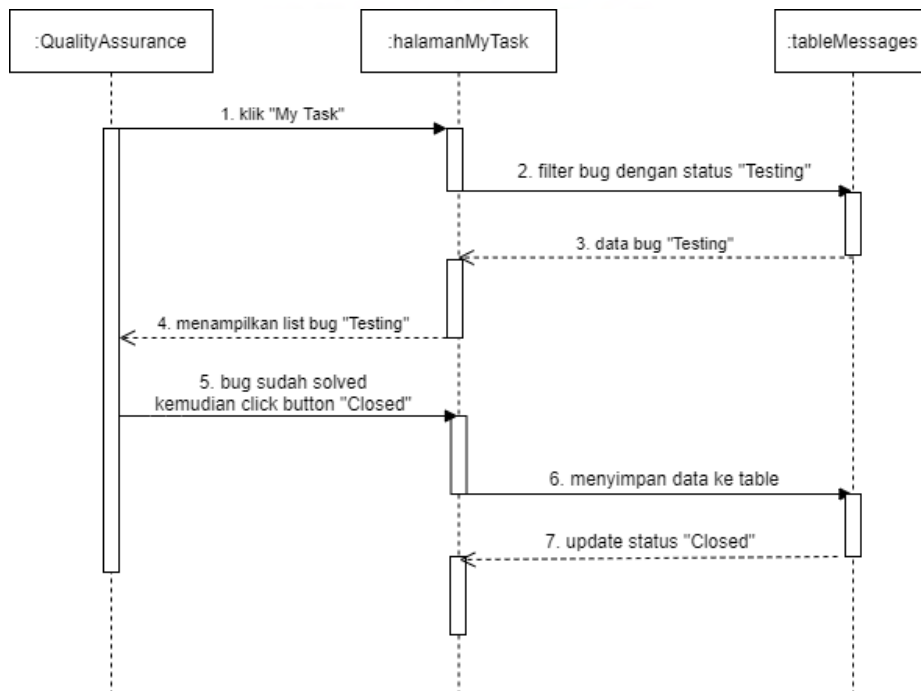
Gambar 4.20 Sequence Diagram Assigned

### 3.5.3. Sequence Diagram Programmer Repair



Gambar 4.21 Sequence Diagram Programmer Repair

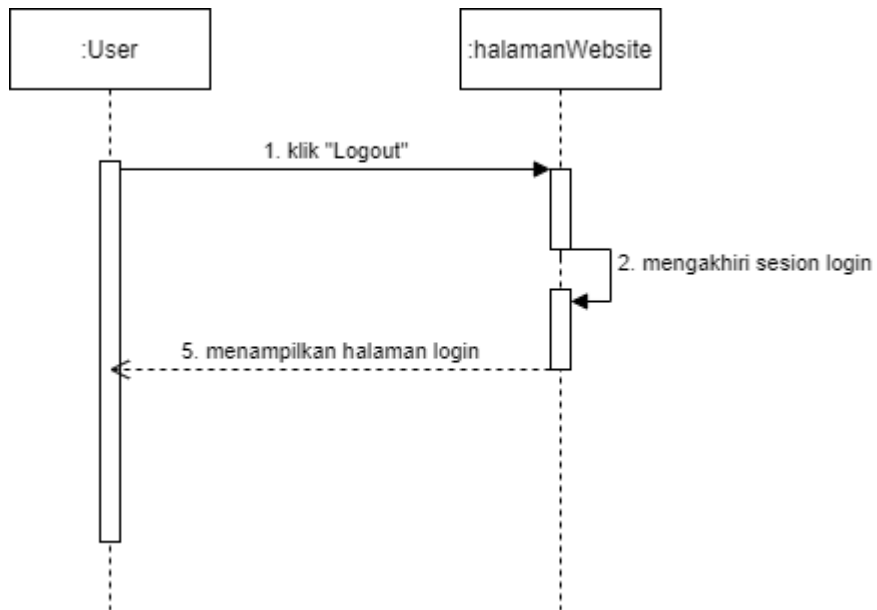
### 3.5.4. Sequence Diagram Testing



Gambar 4.22 Sequence Diagram Testing

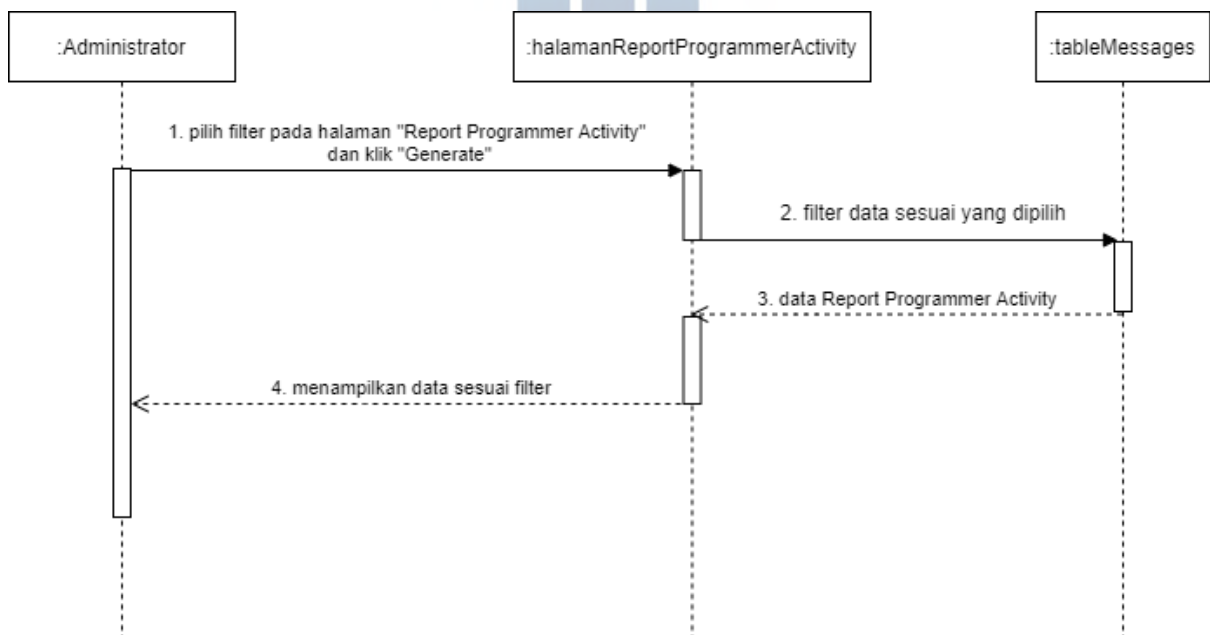


### 3.5.5. Sequence Diagram Logout



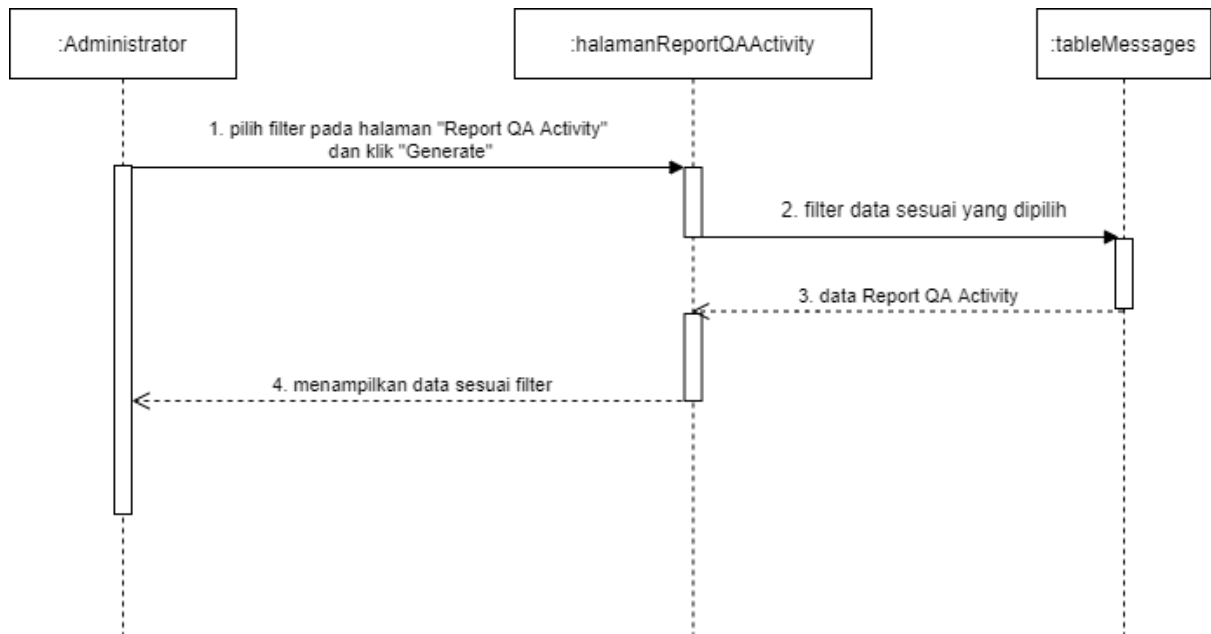
Gambar 4.23 Sequence Diagram Logout

### 3.5.6. Sequence Diagram Report Programmer Activity



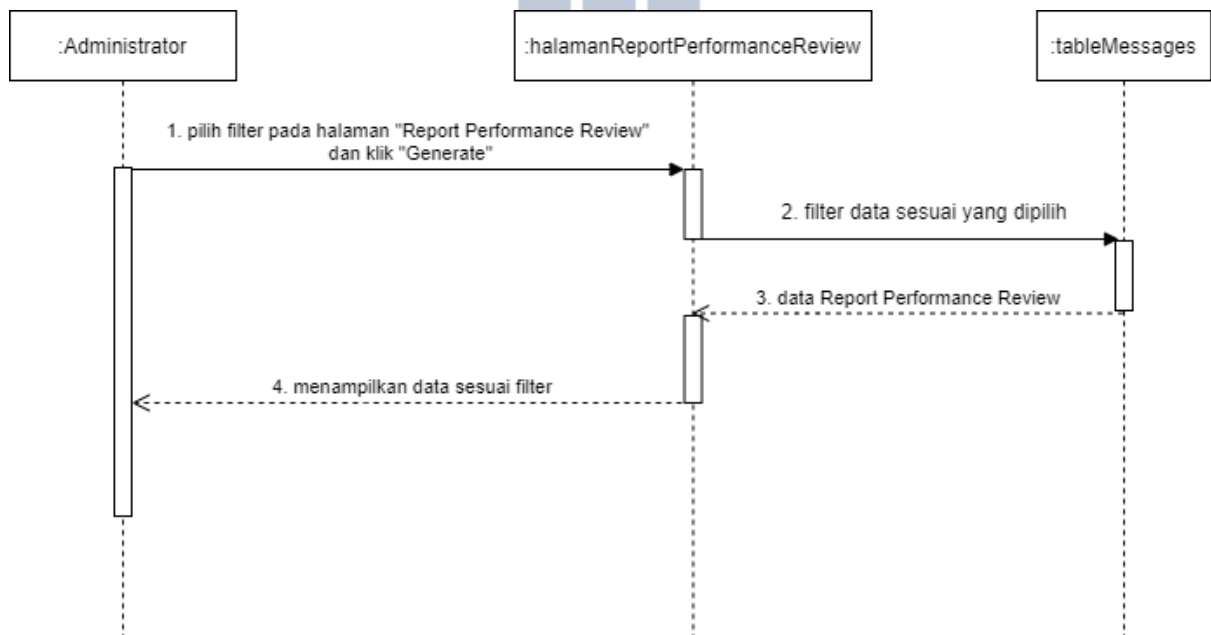
Gambar 4.24 Sequence Diagram Report Programmer Activity

### 3.5.7. Sequence Diagram Report QA Activity



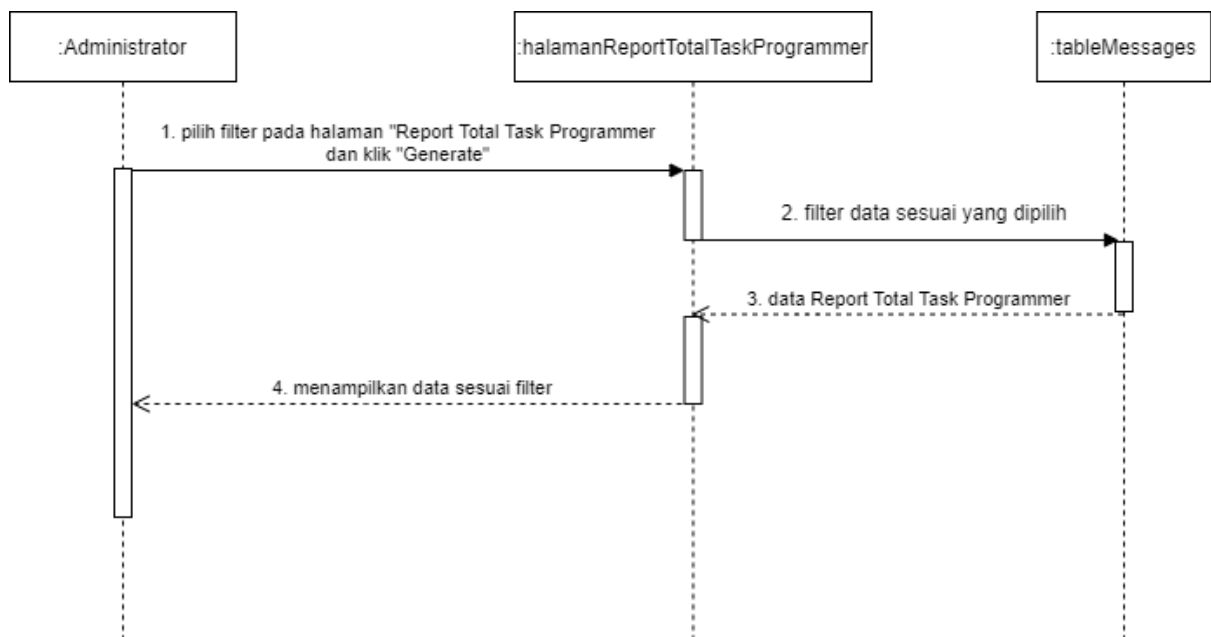
Gambar 4.25 Sequence Diagram Report QA Activity

### 3.5.8. Sequence Diagram Report Performance Review



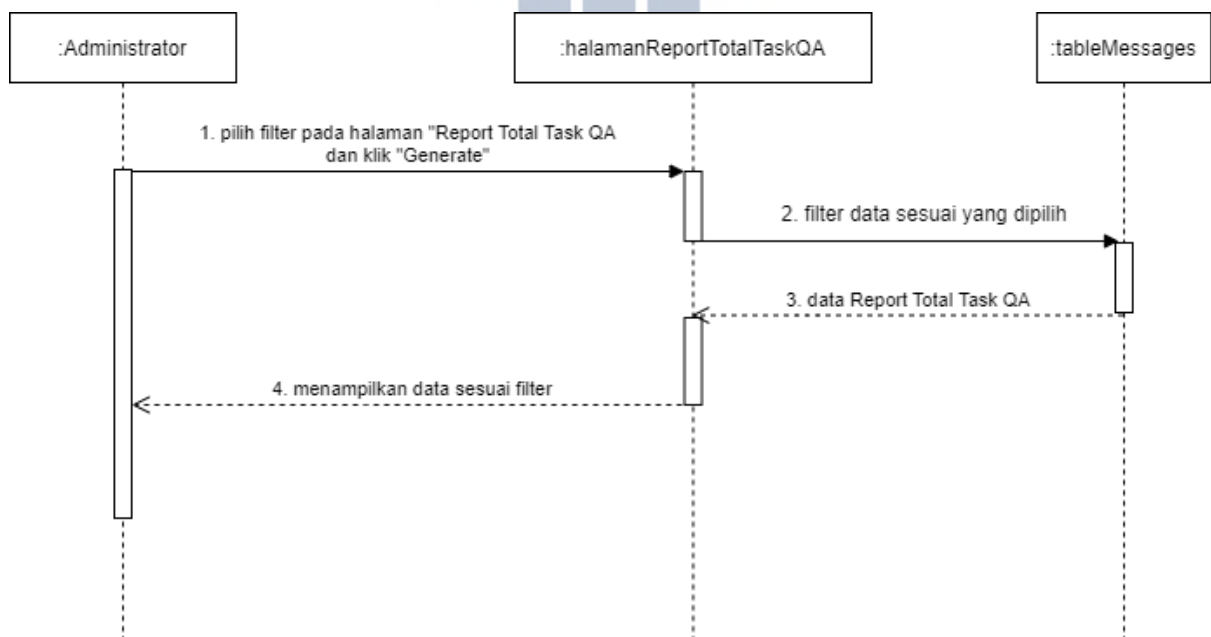
Gambar 4.26 Sequence Diagram Report Performance Review

### 3.5.9. Sequence Diagram Report Total Task Programmer



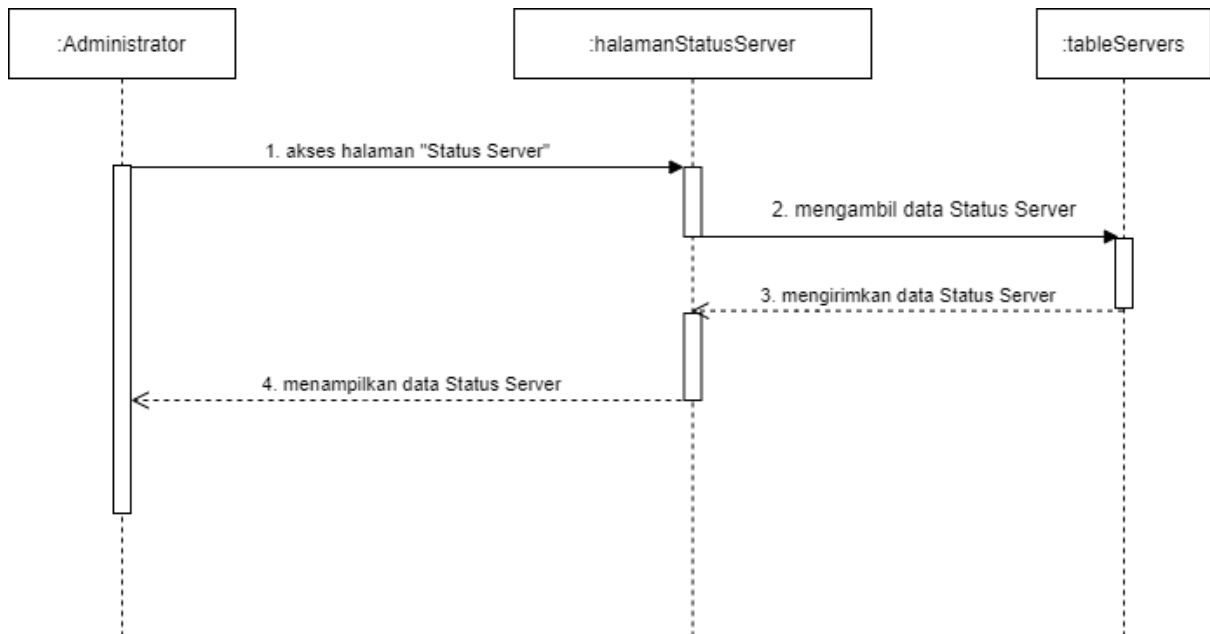
Gambar 4.27 Sequence Diagram Report Total Task Programmer

### 3.5.10. Sequence Diagram Report Total Task QA



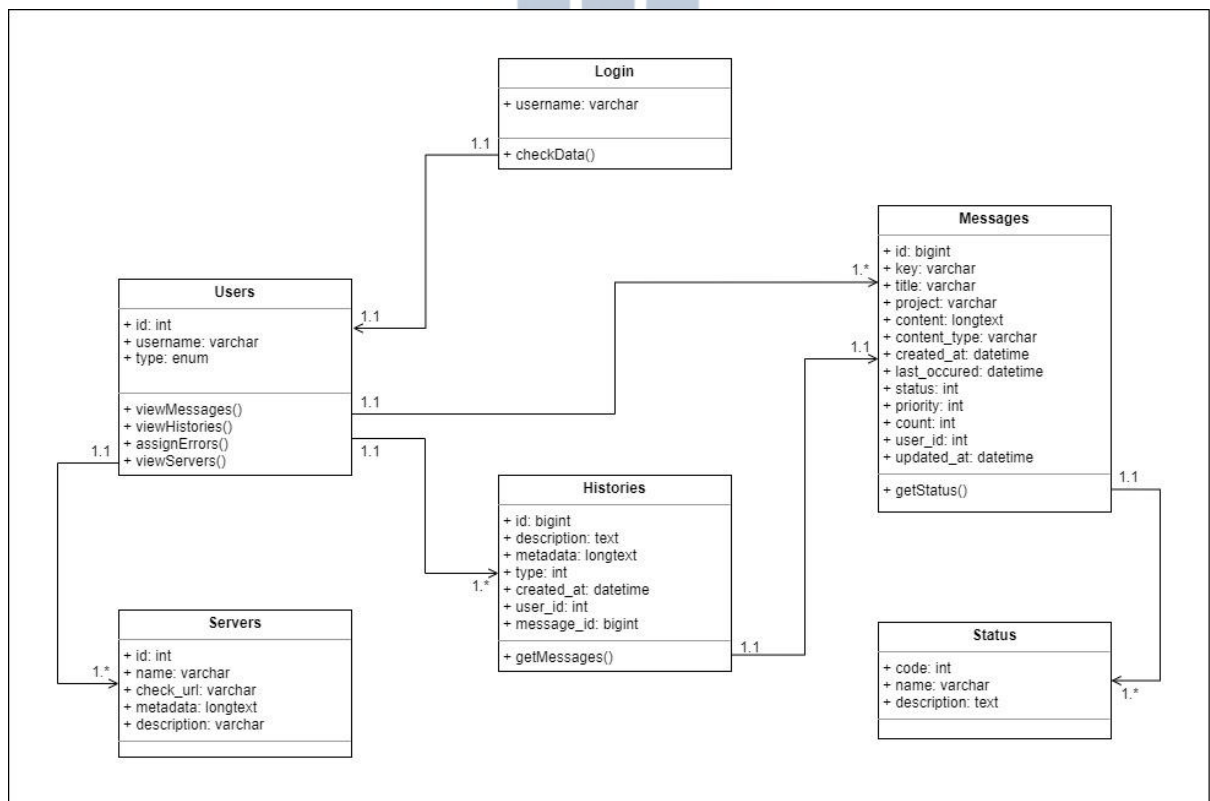
Gambar 4.28 Sequence Diagram Report Total Task QA

### 3.5.11. Sequence Diagram Status Server



Gambar 4.29 Sequence Diagram Status Server

### 3.6. Class Diagram



Gambar 4.30 Class Diagram

### 3.7. Perancangan Basis Data

#### 3.7.1. Tb\_temp\_login

Tabel 4.20 Tb\_temp\_login

No	Nama Atribut	Tipe Data	Panjang	Keterangan
1	username	varchar	25	

#### 3.7.2. Tb\_temp\_user

Tabel 4.21 Tb\_temp\_user

No	Nama Atribut	Tipe Data	Panjang	Keterangan
1	id	int	11	PRIMARY KEY
2	username	varchar	25	
3	password	text		
4	type	enum	('PROGRAMMER', 'SOFTWARE_TESTER')	

#### 3.7.3. Tb\_temp\_status

Tabel 4.22 Tb\_temp\_status

No	Nama Atribut	Tipe Data	Panjang	Keterangan
1	code	int	10	PRIMARY KEY
2	name	varchar	100	
3	description	text		

#### 3.7.4. Tb\_temp\_server

Tabel 4.23 Tb\_temp\_server

No	Nama Atribut	Tipe Data	Panjang	Keterangan
1	id	int	10	PRIMARY KEY
2	name	varchar	100	
3	check_url	varchar	255	
4	metadata	longtext		
5	description	varchar	255	

### 3.7.5. Tb\_temp\_messages

Tabel 4.24 Tb\_temp\_messages

No	Nama Atribut	Tipe Data	Panjang	Keterangan
1	id	bigint	20	Primary Key
2	key	varchar	100	
3	title	varchar	255	
4	project	varchar	100	
5	content	longtext		
6	content_type	varchar	100	
7	created_at	datetime		
8	last_occured	datetime		
9	status	int	10	FOREIGN KEY
10	priority	int	11	
11	count	int	10	
12	user_id	int	10	FOREIGN KEY
13	updated_at	datetime		

### 3.7.6. Tb\_temp\_histories

Tabel 4.25 Tb\_temp\_histories

No	Nama Atribut	Tipe Data	Panjang	Keterangan
1	id	bigint	20	Primary Key
2	description	text		
3	metadata	longtext		
4	type	int	10	
5	created_at	datetime		
6	user_id	int	10	FOREIGN KEY
7	message_id	bigint	20	FOREIGN KEY

## 3.8. Perancangan Antar Muka

### 3.8.1. Modul Login



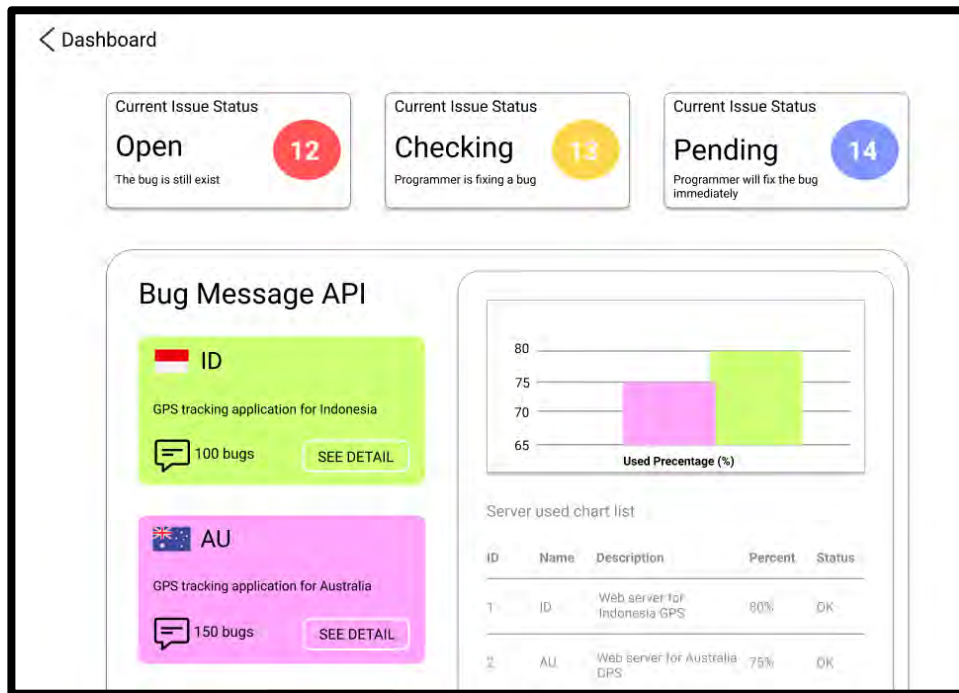
Sistem Pengecekan Bug  
Login

Username:

LOGIN

Gambar 4.31 Modul Login

### 3.8.2. Modul Dashboard



Gambar 4.32 Modul Dashboard

### 3.8.3. Modul Active Issue QA

The 'Bug List' module allows users to filter and view active issues. The table below shows a list of issues with their counts, project names, priorities, and current status.

Count	Project	Priority	Status
12	Sample	High	Checking by Sultan
20	Sample	Normal	Open
21	Sample	Normal	Open
22	Sample	Normal	Open
23	Sample	Normal	Open

Gambar 4.33 Modul Active Issue QA



### 3.8.4. Detail Active Issue QA

The screenshot shows a 'Bug List' interface. At the top, it says 'Bug List' and 'ALL ISSUE (250)'. Below this are search and sort controls. The main table has columns for Count, Project, Priority, and Status. The first row shows a count of 12, Project 'Sample', Priority 'Normal', and Status 'Checking by Sultan'. Below the table, there are fields for Assigned, Priority, Created At, Last Occured, User Info, Formatted, and Stack Trace, all with 'Sample' values. At the bottom, there is an 'Assignee' field and an 'ASSIGN USER' button.

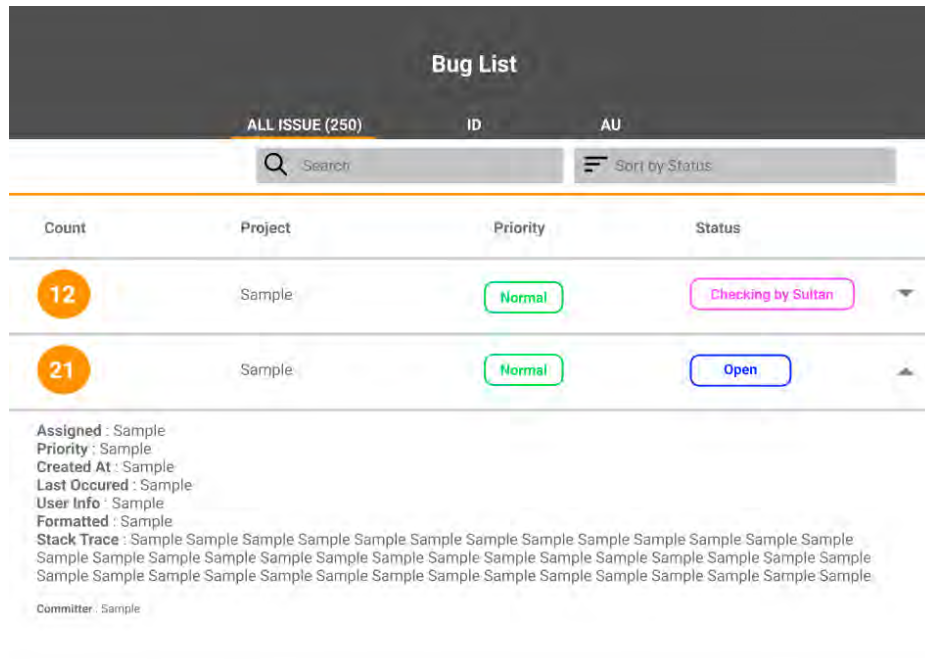
Gambar 4.34 Detail Active Issue QA

### 3.8.5. Modul My Task QA

The screenshot shows a 'Modul My Task QA' interface. At the top, it says '(1) Bug assigned by me'. Below this are search and sort controls. The main table has columns for Count, Project, Priority, and Status. The first row shows a count of 10, Project 'Sample', Priority 'Normal', and Status 'Testing'. Below the table, there are fields for Assigned, Priority, Created At, Last Occured, User Info, Formatted, and Stack Trace, all with 'Sample' values. At the bottom, there is an 'Assignee' field and buttons for 'RE-ASSIGN' and 'CLOSED'. Below this, there are three more rows in the table with counts 21, 12, and 23, all with Project 'Sample', Priority 'Normal', and Status 'Testing'.

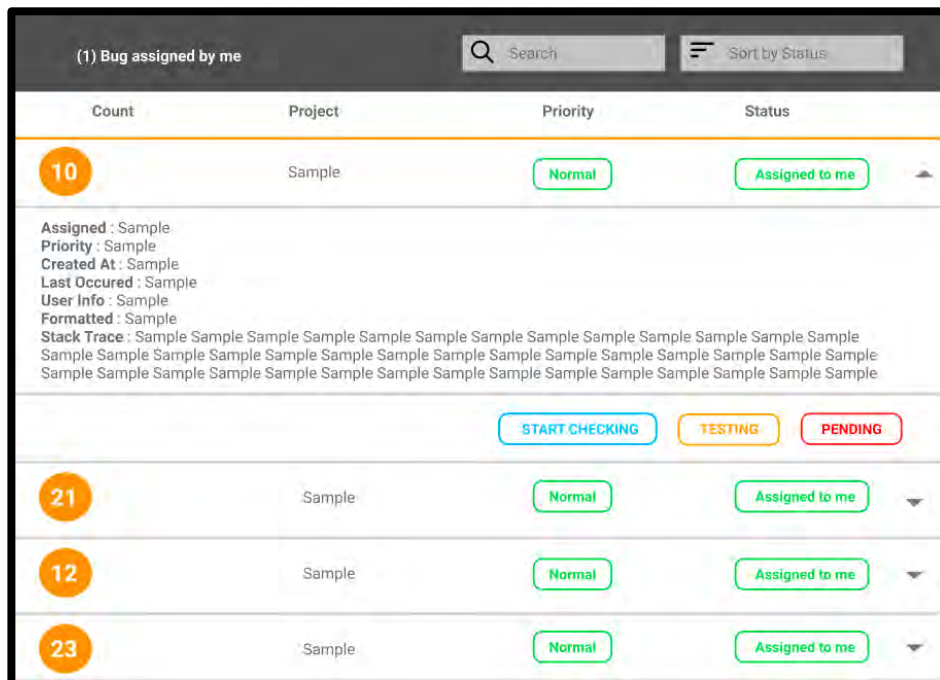
Gambar 4.35 Modul My Task QA

### 3.8.6. Modul My Task Programmer



Gambar 4.36 Modul My Task Programmer

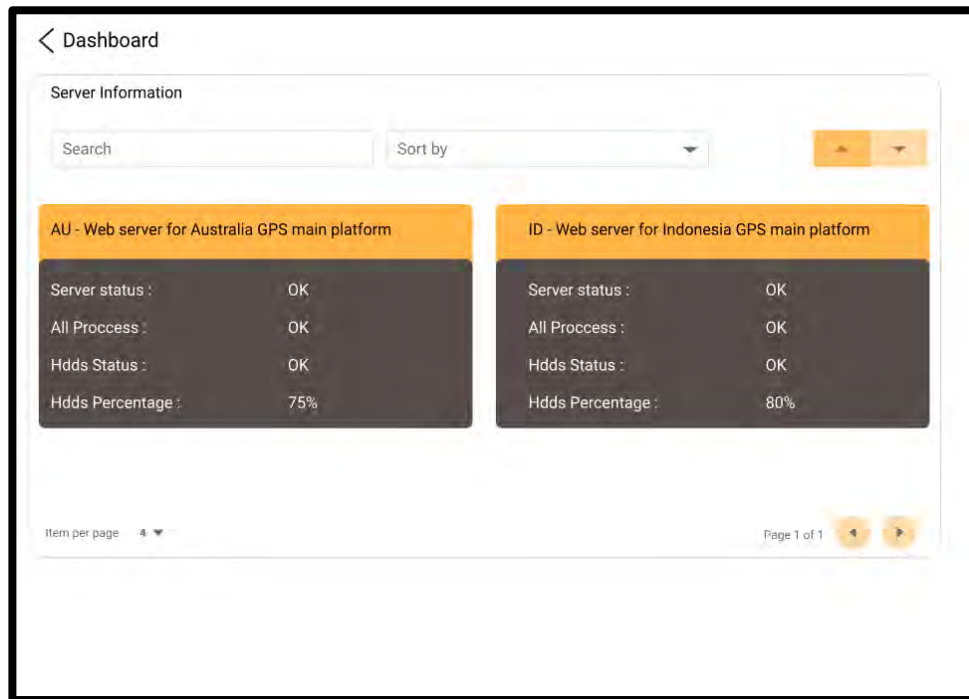
### 3.8.7. Detail My Task Programmer



Gambar 4.37 Detail My Task Programmer

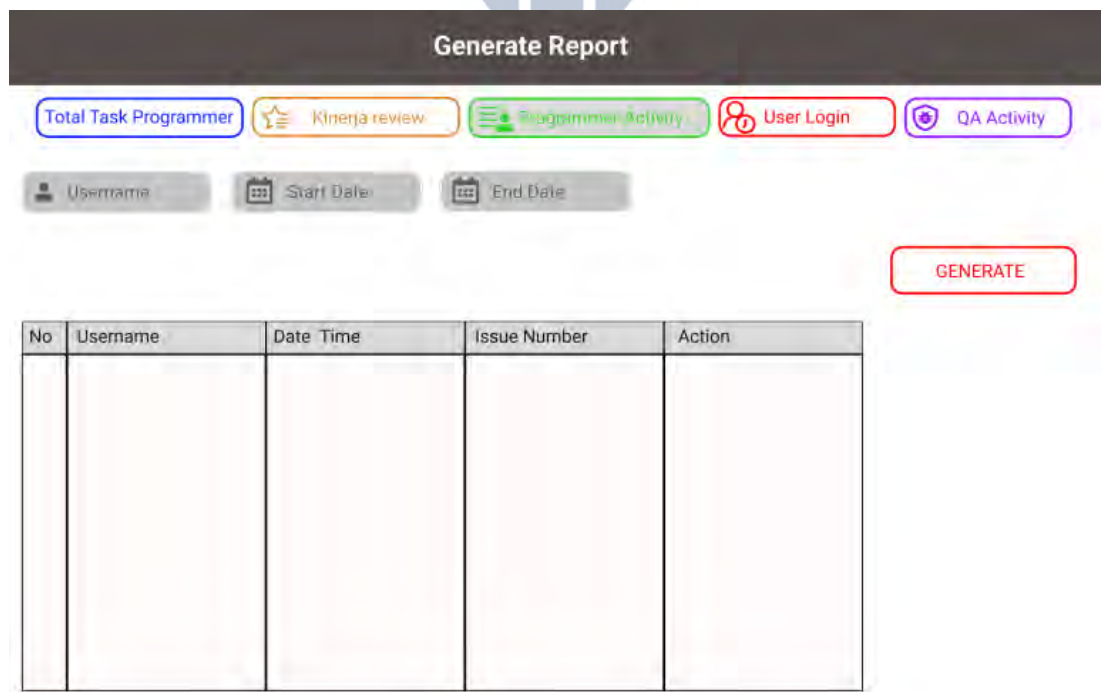


### 3.8.10. Modul Status Server QA dan Programmer



Gambar 4.40 Modul Status Servers QA dan Programmer

### 3.8.11. Modul Report Programer Activity



Gambar 4.41 Modul Report Programer Activity

### 3.8.12. Modul Report QA Activity

**Generate Report**

Total Task Programmer Kinerja review Programmer Activity User Login QA Activity

Username Start Date End Date

**GENERATE**

No	Username	Date Time	Issue Number	Action

Gambar 4.42 Modul Report QA Activity

### 3.8.13. Modul Report Review Kinerja

**Generate Report**

Total Task Programmer Kinerja review Programmer Activity User Login QA Activity

Username Start Date End Date

**GENERATE**

No	Username	Issue Number	Working Time

Gambar 4.43 Modul Report Review Kinerja

### 3.8.14. Modul Report Total Task Programmer

The screenshot shows a web interface for generating a report. At the top, there is a dark grey header with the text "Generate Report". Below the header, there are five colored buttons: "Total Task Programmer" (blue), "Kinerja review" (orange), "Programmer Activity" (green), "User Login" (red), and "QA Activity" (purple). Underneath these buttons are three input fields: "Username" (with a person icon), "Start Date" (with a calendar icon), and "End Date" (with a calendar icon). To the right of these fields is a red "GENERATE" button. Below the form is a table with the following columns: "No", "Username", "Pending Bug", "Close Bug", and "Total". The table is currently empty.

Gambar 4.44 Modul Report Total Task Programmer

### 3.8.15. Modul Report Programmer Activity

The screenshot shows a web interface for generating a report. At the top, there is a dark grey header with the text "Generate Report". Below the header, there are five colored buttons: "Total Task Programmer" (blue), "Kinerja review" (orange), "User Activity Summary" (green), "User Login" (red), and "QA Activity" (purple). Underneath these buttons are three input fields: "Username" (with a person icon), "Start Date" (with a calendar icon), and "End Date" (with a calendar icon). To the right of these fields is a red "GENERATE" button. Below the form is a table with the following columns: "No", "Username", "Date & Time", "IP Address", and "Browser". The table is currently empty.

Gambar 4.45 Modul Report Programmer Activity

Modul	Action	Test Type	Scenario test	Expected result	Actual Result	Status
<b>Login</b>	Login dengan memasukkan Username bukan nama programmer atau QA di PT Intimap	Negative Case	Buka web sistem pengecekan bug	Muncul pop up atau alert "Username doesnt matching"	Muncul pop up atau alert "Username doesnt matching"	Passed
			Masukkan username bukan nama programmer atau QA di PT Intimap			
			Klik tombol login			
	Login dengan memasukkan Username nama programmer atau QA di PT Intimap	Positive Case	Buka web sistem pengecekan bug	Berhasil login	Berhasil login	Passed
			Masukkan username nama programmer atau QA di PT Intimap			
			Klik tombol login			
	Klik tombol login tanpa mengisi kolom username	Negative Case	Klik tombol login	Muncul pop up atau alert "Username is required"	Muncul pop up atau alert "Username is required"	Passed
<b>Active Issue QA</b>	Mencari data bug dengan memasukkan macam-macam keyword	Negative Case	Buka halaman Active Issue	Tidak ada data yang muncul	Tidak ada data yang muncul	Passed
			Masukkan keyword apa saja di dalam kolom pencarian			
	Mencari data bug dengan memasukkan keyword nomor project	Positive Case	Buka halaman Active Issue	Muncul data bug sesuai dengan nomor project yang dicari	Muncul data bug sesuai dengan nomor project yang dicari	Passed
	Masukkan keyword nomor project di dalam kolom pencarian					
	Filter status bug	Positive Case	Buka halaman Active Issue	Muncul bug sesuai dengan statusnya	Muncul bug sesuai dengan statusnya	Passed
			Filter atau pilih status bug yang			

			ingin ditampilkan			
	Assign bug ke programmer bukan comitter	Negative Case	Buka halaman Active Issue Filter status bug menjadi open Pilih salah satu bug Assign bug ke programmer yang bukan comitter Klik tombol Assign User	Muncul bug tersebut dihalaman My Task Programmer sesuai dengan user yang dipilih	Muncul bug tersebut dihalaman My Task Programmer sesuai dengan comitter	Failed
	Assign bug ke programmer sesuai comitter	Positive Case	Buka halaman Active Issue Filter status bug menjadi open Pilih salah satu bug Assign bug ke programmer yang sesuai comitter Klik tombol Assign User	Muncul bug tersebut dihalaman My Task Programmer sesuai dengan user yang dipilih	Muncul bug tersebut dihalaman My Task Programmer sesuai dengan user yang dipilih	Passed
	Assign bug tanpa klik tombol Assign User	Negative Case	Buka halaman Active Issue Filter status bug menjadi open Pilih salah satu bug Assign bug ke programmer yang sesuai comitter	Bug tidak ter assign	Bug tidak ter assign	Passed
<b>Histori es QA</b>	Mencari data bug dengan memasukkan macam-macam keyword	Negative Case	Buka halaman Histories Masukkan keyword apa saja di dalam kolom pencarian	Tidak muncul data bug yang dicari	Tidak muncul data bug yang dicari	Passed



	Mencari data bug dengan memasukkan keyword format bug	Positive Case	Buka halaman Histories Masukkan keyword format bug di dalam kolom pencarian	Muncul data bug yang dicari	Muncul data bug yang dicari	Passed
	Menghapus keyword di kolom pencarian	Positive Case	Buka halaman Histories Masukkan keyword format bug di dalam kolom pencarian Hapus keyword	Kembali memunculkan semua histories bug	Kembali memunculkan semua histories bug	Passed
<b>My Task QA</b>	Re-assign user sesuai dengan programmer yang mengerjakan bug	Positive Case	Buka halaman My Task QA Pilih user Klik tombol Re-assign User	Muncul kembali bug dihalaman my task programmer sesuai dengan programmer yang mengerjakan bug	Muncul kembali bug dihalaman my task programmer sesuai dengan programmer yang mengerjakan bug	Passed
	Re-assign user tidak sesuai dengan programmer yang mengerjakan bug	Negative Case	Buka halaman My Task QA Pilih user Klik tombol Re-assign User	Muncul bug tersebut dihalaman My Task Programmer sesuai dengan user yang dipilih	Muncul kembali bug dihalaman my task programmer sesuai dengan programmer yang mengerjakan bug	Failed
	Re-assign user tanpa klik tombol re-assign user	Negative Case	Buka halaman My Task QA Pilih user	Bug tidak ter-re-assign	Bug tidak ter-re-assign	Passed
	Klik tombol Closed	Positive Case	Buka halaman My Task QA Klik tombol closed	Status bug berubah menjadi closed, dan bug tersebut muncul dihalaman histories	Status bug berubah menjadi closed, dan bug tersebut muncul dihalaman histories	Passed
<b>Active Issue Programmer</b>	Mencari data bug dengan memasukkan macam-macam keyword	Negative Case	Buka halaman Active Issue Masukkan keyword apa saja di dalam kolom pencarian	Tidak ada data yang muncul	Tidak ada data yang muncul	Passed
	Mencari data bug dengan memasukkan	Positive Case	Buka halaman Active Issue	Muncul data bug sesuai dengan nomor project yang dicari	Muncul data bug sesuai dengan	Passed

	keyword nomor project		Masukkan keyword nomor project di dalam kolom pencarian		nomor project yang dicari	
	Filter status bug	Positive Case	Buka halaman Active Issue	Muncul bug sesuai dengan statusnya	Muncul bug sesuai dengan statusnya	Passed
			Filter atau pilih status bug yang ingin ditampilkan			
<b>Histori es Programmer</b>	Mencari data bug dengan memasukkan macam-macam keyword	Negative Case	Buka halaman Histories	Tidak muncul data bug yang dicari	Tidak muncul data bug yang dicari	Passed
			Masukkan keyword apa saja di dalam kolom pencarian			
	Mencari data bug dengan memasukkan keyword format bug	Positive Case	Buka halaman Histories Masukkan keyword format bug di dalam kolom pencarian	Muncul data bug yang dicari	Muncul data bug yang dicari	Passed
Menghapus keyword di kolom pencarian	Positive Case	Positive Case	Buka halaman Histories	Kembali memunculkan semua histories bug	Kembali memunculkan semua histories bug	Passed
			Masukkan keyword format bug di dalam kolom pencarian			
			Hapus keyword			
<b>My Task Programmer</b>	Klik tombol Start Checking	Positive Case	Buka halaman My Task Programmer	Status bug berubah menjadi Start Checking		Passed
			Pilih salah satu bug Klik tombol Start Checking			
	Klik tombol Pending	Positive Case	Buka halaman My Task Programmer	Status bug berubah menjadi Pending		Passed

			Pilih salah satu bug			
			Klik tombol Pending			
	Klik tombol Testing	Positive Case	Buka halaman My Task Programmer	Status bug berubah menjadi Testing dan bug tersebut masuk ke halaman my task QA dengan status Testing		Passed
			Pilih salah satu bug			
			Klik tombol Testing			
	Klik tombol Testing, lalu klik tombol Start Checking kembali	Negative Case	Buka halaman My Task Programmer	Bug kembali masuk ke halaman my task programmer		Passed
			Pilih salah satu bug			
			Klik tombol Testing			
			Klik tombol Start Checking			
<b>Report Admin</b>	Klik tombol generate report tanpa pilih user dan tanggal di semua report	Negative Case	Buka halaman Report	Muncul warning harus pilih user dan tanggal	Tidak ada warning, namun muncul pop up "No data Available"	Failed
			Pilih salah satu report			
			Klik tombol generate report			
	Klik tombol generate report dengan memilih user dan tanggal di report QA Activity	Positive Case	Buka halaman Report QA Activity	Muncul data QA Activity	Muncul data QA Activity	Passed
			Pilih user			
			Pilih start and end date			
			Klik tombol generate report			
	Klik tombol generate report dengan memilih user dan tanggal di report Programmer Activity	Positive Case	Buka halaman Report Programmer Activity	Muncul data Programmer Activity	Muncul data Programmer Activity	Passed
			Pilih user			
			Pilih start and end date			
			Klik tombol generate report			

Klik tombol generate report dengan memilih user dan tanggal di report Review Kinerja	Positive Case	Buka halaman Report Review Kinerja	Muncul data Review Kinerja	Muncul data Review Kinerja	Passed
		Pilih user			
		Pilih start and end date			
Klik tombol generate report dengan memilih user dan tanggal di report total task QA	Positive Case	Buka halaman Report total task QA	Muncul data total task QA	Muncul data total task QA	Passed
		Pilih user			
		Pilih start and end date			
Klik tombol generate report dengan memilih user dan tanggal di report total task programmer	Positive Case	Buka halaman Report total task programmer	Muncul data total task Programmer	Muncul data total task Programmer	Passed
		Pilih user			
		Pilih start and end date			
		Klik tombol generate report			

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **4.1. Kesimpulan**

Tujuan dari penelitian ini adalah untuk memastikan bug pada aplikasi yang digunakan oleh user sudah terminimalisir sehingga meningkatkan kualitas aplikasi yang dibuat oleh PT Intimap. Tujuan tersebut telah berhasil dengan dibangunnya Sistem Informasi Pencatatan Bug sebagai media pencatatan atau pengumpulan temuan bug pada suatu sistem atau aplikasi yang terstruktur dan terperinci dengan data real time mengenai bug yang muncul. Serta perbaikan pada bug tercatat, sehingga mempermudah programmer dalam melakukan perbaikan bug. Sistem ini berbasis website, di mana semua informasi yang dibutuhkan dapat disimpan dan diakses secara terpusat. Tampilan website ini dibuat sesederhana mungkin agar dapat dengan mudah dipahami oleh semua pengguna sehingga mempercepat proses perbaikan suatu bug. Semua informasi terkait temuan bug pada web ditampilkan dalam bentuk tabel agar lebih rapi dan jelas. Serta terdapat report yang dapat digunakan untuk mengevaluasi perbaikan bug oleh programmer.

#### **4.2. Saran**

Berdasarkan kesimpulan di atas dan penelitian yang dilakukan, maka dapat menemukan beberapa saran yang diharapkan dapat menjadi bahan pertimbangan lebih lanjut dalam upaya peningkatan kualitas sistem yang telah dibuat. Adapun saran-saran yang ingin disampaikan adalah :

1. Melakukan pembaruan pada interface dan sistem informasinya karena seiring berjalannya waktu akan diperlukan fitur baru yang akan berguna untuk penggunaannya.
2. Penulis menyarankan agar user dapat menggunakan sistem informasi ini sesuai dengan tujuan dibangunnya sistem ini agar bisa dipergunakan sesuai fungsinya.
3. Penambahan fitur untuk mengeskport report dalam bentuk excel atau format lain jika diperlukan dikemudian hari.

## DAFTAR PUSTAKA

- [1] D. Purnomo, “Model Prototyping Pada Pengembangan Sistem Informasi,” *JIMP - J. Inform. Merdeka Pasuruan*, vol. 2, no. 2, pp. 54–61, 2017, doi: 10.37438/jimp.v2i2.67.
- [2] D. Andriansyah, “Pengujian Kotak Hitam Boundary Value Analysis Pada Sistem Informasi Manajemen Konseling Tugas Akhir,” vol. 7, no. 1, pp. 20–25, 2018, [Online]. Available: <http://ijns.org/journal/index.php/ijns/article/view/1496>.
- [3] R. SUSANTO, “PERBANDINGAN MODEL WATERFALL DAN PROTOTYPING UNTUK PENGEMBANGAN SISTEM INFORMASI,” vol. 14, 2016.
- [4] E. Pandara, S. Karouw, and M. Najooan, “Implementasi Proses Uji Sistem Informasi Admisi Pascasarjana Universitas Sam Ratulangi,” *J. Tek. Inform.*, vol. 4, no. 2, pp. 1–6, 2015, doi: 10.35793/jti.4.2.2014.6989.
- [5] G. Riyadi, “PEMBANGUNAN PERANGKAT LUNAK ERROR MONITORING PADA WEB UNTUK MENUNJANG PERAWATAN PERANGKAT LUNAK (Studi Kasus PT. Integra ...),” 2019, [Online]. Available: <http://repository.unpas.ac.id/id/eprint/41342>.
- [6] R. A. Rusdiansyah, “MONITORING ERROR PROYEK TEKNOLOGI INFORMASI SECARA REAL TIME MENGGUNAKAN WEB SERVICE SEBAGAI PENGUKUR KINERJA TIM PROYEK TEKNOLOGI INFORMASI,” *Angew. Chemie Int. Ed.* 6(11), 951–952., pp. 5–24, 2017.
- [7] M. Y. Riadi and B. T. Sartana, “Analisa dan Perancangan Sistem Informasi Quality Assurance Berbasis Web Pada PT.Codigo Cyberlin Metadata Menggunakan Unified Modeling Language,” *J. IDEALIS*, vol. 1, no. 1, pp. 284–292, 2018.
- [8] A. . Syamsudin, “Analisis Kesalahan Coding Pemrograman Java Pada Matakuliah Algoritma Pemrograman Mahasiswa Tadris Matematika Iain Kediri,” *Factor M*, vol. 2, no. 2, pp. 102–114, 2020, doi: 10.30762/f\_m.v2i2.1711.
- [9] G. S. M, “Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Berbasis Spring : Modul Core System dan Manajemen Source Code,” no. January, 2011.
- [10] PT Intimap, “API Errors Reporting,” *PT Intimap*, 2021. .
- [11] K. Relan, *Building REST APIs with Flask*. Apress, 2019.

- [12] D. Pembimbing *et al.*, “PREDIKSI WAKTU PERBAIKAN BUG BERDASARKAN LAPORAN,” 2018.
- [13] E. S. Asih *et al.*, “Mobile E-commerce website for technology-based buying selling services,” *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 884–888, 2019, doi: 10.35940/ijeat.F1167.0886S19.
- [14] S. Nunez, M. Kabalan, P. Singh, and V. Moncada, “Assessment of potential microgrid system comprising renewable energy in la Kasquita Community, Nicaragua,” *2015 IEEE Canada Int. Humanit. Technol. Conf. IHTC 2015*, pp. 386–400, 2015, doi: 10.1109/IHTC.2015.7238067.
- [15] M. Kramer, “Lifecycle : an Analyses Based on the,” *Rev. Bus. Financ. Stud.*, vol. 9, no. 1, pp. 77–84, 2018.
- [16] S. R. Pressman, *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*. 2015.
- [17] S. Ali, R. Alauldeen, and A. Ruaa, “What is Client-Server System: Architecture, Issues and Challenge of Client-Server System,” *HBRP Publ.*, no. February, pp. 1–6, 2020, doi: 10.5281/zenodo.3673071.
- [18] “Node.js - Introduction,” 2021. [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm).
- [19] K. Tatroe and P. MacIntyre, *Programming PHP Creating Dynamic Web Pages*, 4th ed. 2020.
- [20] J. Shen, G. Sun, and Y. Li, “Design and Implementation of E-Commerce Platform Based on Android,” *J. Comput. Commun.*, vol. 06, no. 08, pp. 92–100, 2018, doi: 10.4236/jcc.2018.68007.
- [21] M. Ma'mur, “Sistem Informasi Penjualan Barang ( Studi Kasus : Minimarket Gemart Bandar Lampung ),” vol. 13, no. 2, pp. 31–35, 2017.
- [22] Z. Yusdinata and M. A. Bora, “Analisis Penerapan Keselamatan Dan Kesehatan Kerja (K3) Dengan Menggunakan Metode Fishbone Diagram,” *J. Tek. Ibnu Sina*, vol. 3, no. 2, pp. 127–133, 2018, doi: 10.36352/jt-ibsi.v3i2.144.
- [23] I. Hoeronis, “Pendekatan bpmn dalam membuat abstraksi proses bisnis pengelolaan penelitian di universitas siliwangi,” *semantik*, vol. 4, no. December, pp. 83–90, 2018, doi: 10.5281/zenodo.1471134.

- [24] R. Akbar, F. Akbar, and W. Alifah, "Penerapan Aplikasi Berbasis Web Untuk Monitoring Pengobatan Pasien Gangguan Jiwa Pada UPT Puskesmas Pasar Usang," *J. Nas. Teknol. dan Sist. Inf.*, vol. 7, no. 3, pp. 130–137, 2021, doi: 10.25077/teknosi.v7i3.2021.130-137.
- [25] F. Procházka, "Refactoring of Kduby packages Bachelor ' s Thesis," 2017.
- [26] A. Saboury, P. Musavi, and F. Khomh, *An empirical study of code smells in JavaScript projects*, . 2017.
- [27] U. Rahardja, Q. Aini, and N. P. L. Santoso, "Pengintegrasian Yii Framework Berbasis API pada Sistem Penilaian Absensi," *Sisfotenika*, vol. 8, no. 2, p. 140, 2018, doi: 10.30700/jst.v8i2.403.
- [28] R. FAJRIN, "Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia," *J. Inform.*, vol. 11, no. 2, pp. 33–40, 2017, doi: 10.26555/jifo.v11i2.a6090.
- [29] Y. Hartanto, "Implementasi Bug Tracking System dengan Metodologi Scrum dan Algoritma Cosine Similarity," vol. 9, no. 1, pp. 24–32, 2022, doi: 10.30865/jurikom.v9i1.3861.
- [30] H. Ju, F. T. Industri, and U. I. Indonesia, "Sistem Informasi Pencatatan Bug Berbasis Web," 2021.
- [31] S. I. Mandala, L. Fitriani, and A. Sutedi, "Rancang Bangun Aplikasi Pengelolaan Bug Tracker Berbasis Web pada Pengembangan Software," *J. Algoritm.*, vol. 19, no. 1, pp. 180–191, 2022, doi: 10.33364/algoritma/v.19-1.1046.
- [32] A. Satyagraha, W. Witanti, and H. Ashaury, "Pembangunan Sistem Informasi Monitoring Logistik di PT Dirgantara Indonesia," *Sainteks*, vol. 13, pp. 142–146, 2020.
- [33] J. Timur, "SISTEM INFORMASI MONITORING PROYEK PADA PT . SURYAPRIMA KARYA ARRYVYA BERBASIS JAVA," pp. 1037–1041, 2021.
- [34] G. Maulani, H. Komara, and S. Meiliana, "Rancang Bangun Sistem Informasi Monitoring Dashboard Traffic Work Order Berbasis Web," *J. CERITA*, vol. 6, no. 2, pp. 137–146, 2020, doi: 10.33050/cerita.v6i2.1127.
- [35] S. N. Ambo, J. Jumail, and A. Kuspurnama, "Rancang Bangun Aplikasi Monitoring Kegiatan Implementasi Proyek Perangkat Lunak Berbasis Web," *Pros. Semnastek*, pp.



1–6, 2019, [Online]. Available:  
<https://jurnal.umj.ac.id/index.php/semnastek/article/view/5245>.

- [36] H. Hendri, “Implementasi Chatbot Untuk Laporan Bug / Error Reporting Dengan Natural Language Processing (Studi Kasus : CV. Solusi Prima),” *J. Process.*, vol. 16, no. 2, p. 135, 2021, doi: 10.33998/processor.2021.16.2.1026.



**LAMPIRAN 1**  
**KARTU BIMBINGAN**

Nama Mahasiswa : Diyana Zevira Pebriana

Rifanny Suci Syahda

Sultan Athariq Ismail

NIM : 41818110050

41818110129

41818110134

Judul MPTI : Rancang Bangun Sistem Pengecekan Bug Aplikasi Menggunakan  
API Berbasis Web

Dosen Pembimbing : Nur Ani, ST, MMSI





### KARTU ASISTENSI TUGAS AKHIR (SKRIPSI/THESIS/DISERTASI)

**NAMA** : SULTAN ATHARIQ ISMAIL **SEM/THN AKAD** : 9 / 2022  
**NIM** : 41818110134 **JENIS BIMBINGAN** : TUGAS AKHIR  
**FAKULTAS** : SISTEM INFORMASI **DOSEN PEMBIMBING** : Nur Ani, ST, MMSI  
**PROGRAM STUDI** : ILMU KOMPUTER

**JUDUL TUGAS AKHIR** : "RANCANG BANGUN SISTEM PENGECEKAN BUG APLIKASI MENGGUNAKAN API BERBASIS WEB (STUDI KASUS: PT INTIMAP)"

NO.	TANGGAL	KETERANGAN	PARAF	NO.	TANGGAL	KETERANGAN	PARAF
1.	02 Mar 2022	Pembahasan masalah dan metode via gmeet: <a href="http://meet.google.com/gou-mnvf-ykh">http://meet.google.com/gou-mnvf-ykh</a>		7.	15 Jul 2022	Review dan revisi minor bab1 sd 5: <a href="https://meet.google.com/zcf-qrjv-mdd">https://meet.google.com/zcf-qrjv-mdd</a>	
2.	21 Mar 2022	Pembahasan pokok masalah dan tinjauan pustaka via gmeet: <a href="https://meet.google.com/ocs-cxbr-gbm">https://meet.google.com/ocs-cxbr-gbm</a>		8.	09 Agu 2022	Pembahasan jurnal dan presentasi pra-sidang <a href="https://meet.google.com/jur-wwfk-ndf">https://meet.google.com/jur-wwfk-ndf</a>	
3.	05 Apr 2022	Pengecekan tinjauan pustaka dan metode penelitian: <a href="https://meet.google.com/isw-mrre-qii">https://meet.google.com/isw-mrre-qii</a>					
4.	30 Mei 2022	Pembahasan bab 3 dan bab 4 <a href="https://meet.google.com/yxc-norx-uuq">https://meet.google.com/yxc-norx-uuq</a>					
5.	04 Jun 2022	Membahas tentang Analisa dan UML <a href="https://meet.google.com/dxg-kusc-duy">https://meet.google.com/dxg-kusc-duy</a>					
6.	01 Jul 2022	Pembahasan Class diagram dan database <a href="https://meet.google.com/vxq-xeye-zgt">https://meet.google.com/vxq-xeye-zgt</a>					

## LAMPIRAN 2

### KEGIATAN BIMBINGAN *ONLINE*

