

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Selama Praktikan melaksanakan KP pada LPMU, Praktikan ditugaskan pada *Back-End developer* aplikasi. Pada divisi tersebut praktikan mendapatkan tugas untuk membuat aplikasi penghubung unit kerja dan program studi di Universitas Pembangunan Jaya, menganalisa pemecahan masalah yang akan digunakan dan melakukan perancangan sebuah output berupa aplikasi reporting ketika *Monitoring* dan Evaluasi (MONEV) berlangsung. Dalam merancang aplikasi ini, praktikan bekerja sama dengan 1 orang lainnya secara tim. Dibawah ini tercantum divisi dan nama-nama dari tim yang termasuk dalam pembuatan Aplikasi, yaitu:

Muhammad Farhan Syah: *Front-End Developer*

Selama berlangsungnya pembuatan aplikasi selama kerja profesi, praktikan selalu berkoordinasi dengan tim agar tidak terjadinya kesalahpahaman ketika pembuatan dan pembagian tugas berlangsung. Khususnya praktikan bekerja sama dengan *divisi front-end developer* dalam merancang desain aplikasi ini.

3.2 Pelaksanaan Kerja

Pelaksanaan KP dilaksanakan kurang lebih 3 bulan yang dimulai dari tanggal 21 Juni 2021 sampai dengan 31 Agustus. Praktikan sebagai *System Analyst dan developer* dan praktikan bekerja yang sesuai dengan **Tabel 3.1** dengan *timeline* yang sudah ditentukan untuk merancang dan membuat Aplikasi.

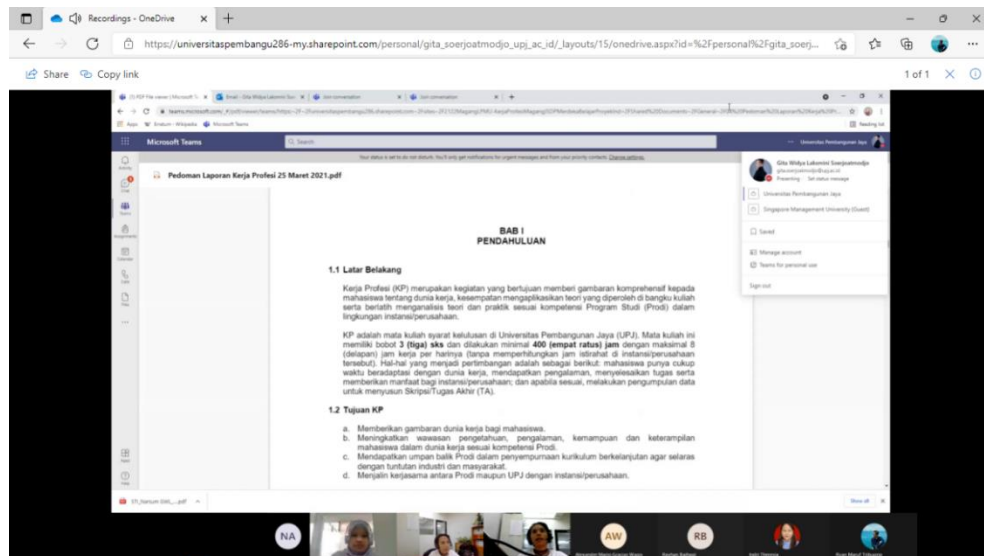
Tabel 3.1 *Timeline Pelaksanaan Kerja Profesi*

No	Kegiatan	Bulan												PIC
		Juni				Juli				Agustus				
		1	2	3	4	1	2	3	4	1	2	3	4	
1	Pengenalan mengenai LPMU	█												Analyst, UI/UX, & Web Dev
2	Analisa dan Perancangan Aplikasi	█	█	█										Analyst
3	Pembuatan design tampilan <i>website</i>				█									UI/UX
4	Development <i>website</i>					█	█	█	█	█	█			Web Dev
5	<i>Testing and implementation Website</i>												█	Analyst, UI/UX, & Web Dev
6	Perbaikan <i>Bugs</i>												█	Web Dev
7	<i>Training</i>												█	Analyst, UI/UX, & Web Dev

Sumber: Dokumentasi praktikan

3.2.1 **Online Kick-off Meeting Kerja Profesi**

Tahapan pelaksanaan KP dimulai dengan melakukan *Online Kick-off Meeting* dengan Bagian LPMU UPJ. Pada tahap ini, Praktikan diberi pengenalan oleh Kepala Bagian LPMU tentang *Job Description* yang dilaksanakan oleh masing masing mahasiswa



Gambar 3.1 Online Kick-off Meeting

Sumber: Dokumentasi praktikan

during the professional work process. This meeting is conducted online through *Microsoft Teams* as seen in **Gambar 3.1** below.

3.2.2 Perencanaan Sistem

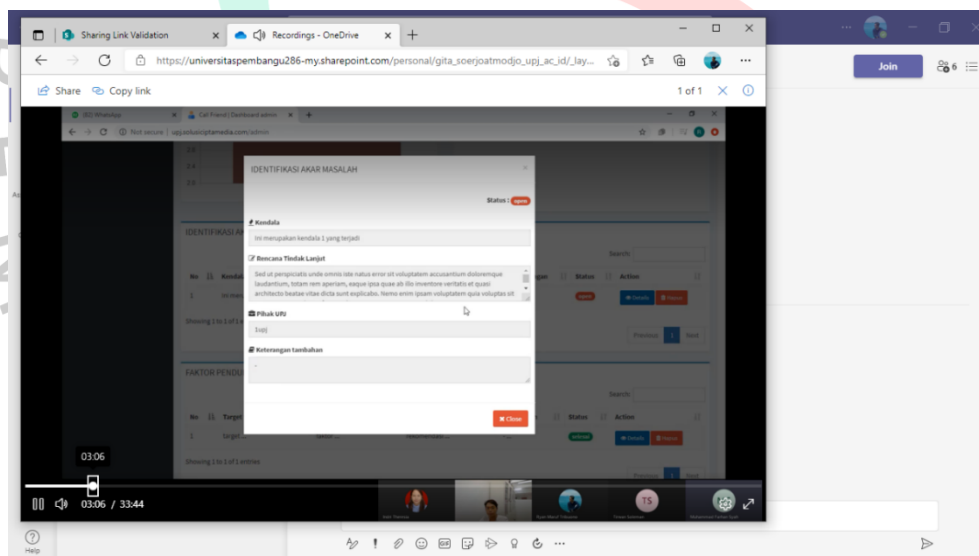
The first stage is to determine *user requirement* in this practical and the team will work together directly to make the *Reporting* application system, the purpose of this is to facilitate the practical in carrying out the work. The practical together with the team members who act as *Front-End developer* discuss with the head of the department and LPMU staff in designing the *Reporting* application system. In this stage, the schedule that has been prepared by the team that has been explained above, in carrying out the system work to know how long the time needed to build the system or application as expected.

In the design of the *Reporting* application, there are system design stages (*planning*), namely *Requirement Analyst*. In this case, the team applies the *OOAD (Object Oriented Analyst Design)* method in the system development process so that the time of implementation and

pekerjaan yang dilakukan oleh tim ini diharapkan dapat diselesaikan dengan tepat waktu yang sesuai dengan waktu yang di tetapkan di dalam pedoman kerja profesi.

3.2.2.1 User Story

Pada **Gambar 3.1** Kepala Lembaga unit LPMU (Bu Gita Widya Laskmini) yang berkoordinasi dan menjelaskan kebutuhan sistem dan proses alur aplikasi kepada praktikan. Dengan harapan aplikasi yang dikerjakan sesuai dengan kebutuhan dan bermanfaat bagi suatu unit kerja dan program studi UPJ ketika sesi Monev berlangsung. Namun untuk praktikan ditugaskan mengerjakan aplikasi *Reporting* bersama dengan tim yang sudah ditentukan, dengan adanya tim ini diharapkan pekerjaan bisa selesai dengan tepat waktu sesuai dengan waktu berlakunya kerja profesi.



Gambar 3. 2 Pemaparan Aplikasi *Reporting*

Sumber: Dokumentasi praktikan

3.2.2.2 Requirement Analyst

Tahap *Requirement analyst* (analisa kebutuhan) adalah tahap interaksi antara analisis sistem dengan pemakai sistem yaitu *end-user*. Pada tahap ini Praktikan menggunakan metode OOAD, dengan metode OOAD di tahap ini bersama tim Front-End mengumpulkan apa saja yang sesuai untuk diterapkan sebagai sistem aplikasi. Hal ini praktikan sebagai *Back-End Developer* atau Perancang aplikasi mendapatkan dokumen-dokumen dalam membantu pekerjaan aplikasi *Reporting*.

3.2.3 Tahap Pengerjaan

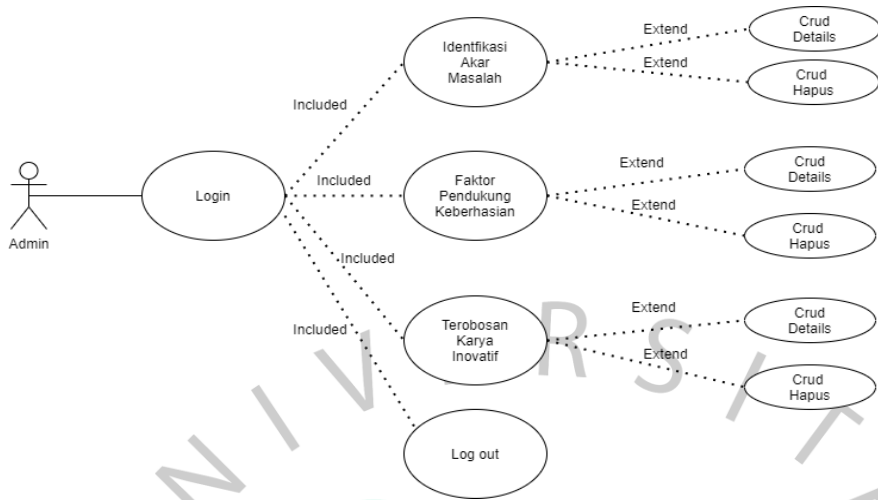
Pada tahap ini, Praktikan mengerjakan Analisa sistem aplikasi berdasarkan hasil analisis yang sudah dilakukan sebelumnya. Praktikan mengerjakan tahap ini dimulai dari Desain dan Implementasi sistem yang disarankan oleh bagian unit kerja LPMU, sampai melakukan pengujian sistem untuk mengetahui apakah sistem dan alur aplikasi tersebut berjalan dengan baik dan sesuai dengan arahan dari Kepala Lembaga LPMU.

3.2.3.1 Analisa Sistem

Setelah tahap perencanaan, pengalokasian tugas telah disepakati oleh tim, maka tahap selanjutnya adalah Analisa Sistem pada aplikasi *Reporting*. Tahap ini bertujuan untuk menganalisa siapa saja aktor yang dapat hak akses menggunakan sistem dan alur proses sistem tersebut dapat dijalankan. Berikut ini tahapan-tahapan dalam menggabungkan bisnis proses dan Analisa sistem.

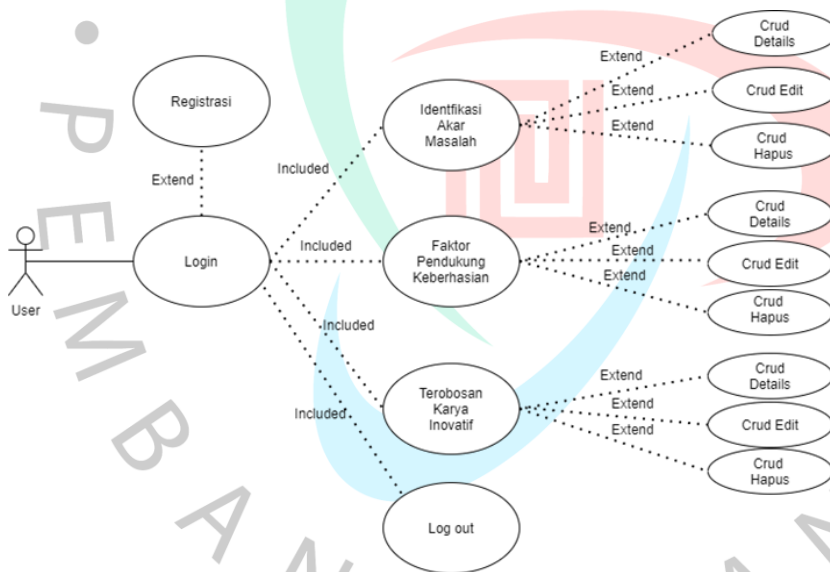
A. Use Case

Use Case aplikasi *Reporting*, aplikasi ini memiliki 2 (dua) aktor yaitu *user* dan *admin* LPMU. Setiap *user* maupun *admin* LPMU memiliki perannya masing-masing. Setelah menentukan aktor, praktikan melakukan Analisa kegiatan atau interaksi sistem dan aktor seperti pada **Gambar 3.2** dan **Gambar 3.3**.



Gambar 3.3 Use Case Admin

Sumber: Dokumentasi praktikan



Gambar 3. 4 Use Case User

Sumber: Dokumentasi praktikan

B. Spesifikasi Use Case

Setelah menganalisa *Use Case* yang sudah dibuat selanjutnya, Praktikan membuat Spesifikasi *Use Case*, untuk menjelaskan proses dari setiap hubungan proses alur aplikasi yang terlibat dalam sistem, berikut ini adalah Spesifikasi *Use Case* yang telah Praktikan rancang pada **Tabel 3.2** sampai **Tabel 3.8**.

Use Case Name	Registrasi Akun	
Actors	User	
Trigger	Menjadi seorang <i>user</i> yang telah terdaftar di dalam sistem	
Preconditions	-	
Postcondition	Jumlah <i>user</i> bertambah	
Success Scenario	Actor	System
	1. User mendaftar di website	Menampilkan form daftar user
	2. User mengisi data diri secara lengkap	-
	3. Submit form daftar user yang sudah diisi secara lengkap	Sistem akan menyimpan data user yang baru saja mendaftar di dalam website
Alternative Flows	Jika data diri belum diisi secara lengkap, maka akan terdapat notifikasi yang menyatakan bahwa terdapat kolom yang belum diisi.	

Tabel 3.2 Registrasi *User*

Sumber: Dokumentasi praktikan

Use Case Name	Login	
Actors	<i>User</i>	
Trigger	<i>User</i> dapat masuk ke dalam sistem <i>website</i>	
Preconditions	<i>User</i> sudah <i>login</i> dan sudah bisa melakukan pengisian form Identifikasi akar masalah, Faktor mendukung keberhasilan dan Terobosan karya inovatif	
Success Scenario	Actor	System
	1. <i>User</i> mengetik url <i>website</i>	Menampilkan <i>form login</i>
	2. <i>User</i> mengisi <i>username</i> dan <i>password</i>	Jika <i>username</i> dan <i>password</i> salah maka akan terdapat sebuah alert " <i>Username & password</i> tidak tepat" dan tampilan akan tetap berada di halaman <i>login</i>
Alternative Flows	Jika <i>user</i> salah menginput <i>username</i> dan <i>password</i> , sistem akan memberikan <i>warning</i> Yang Menyatakan " <i>Username & password</i> tidak tepat", dan sistem akan meminta <i>user</i> memasukkan kembali <i>username</i> dan <i>password</i>	

Tabel 3.3 Spesifikasi *Login User*

Sumber: Dokumentasi praktikan

Use Case Name	Lihat dan Pengisian form task	
Actors	<i>User</i>	
Trigger	<i>User ingin melihat form task serta ingin mengisi form task</i>	
Preconditions	<i>User harus berada di web agar dapat melihat dan mengisi form task dokumen.</i>	
Success Scenario	Actor	System
	1. <i>User masuk ke website</i>	Menampilkan halaman utama <i>website</i>
	2. <i>User meng-klik menu tambah task</i>	Sistem mengalihkan ke halaman <i>form task</i>
	3. <i>User telah mengisi form task</i>	<i>Form task</i> sudah terisi
Alternative Flows		

Tabel 3.4 Spesifikasi *Form Task User*

Sumber: Dokumentasi praktikan

Use Case Name	Logout	
Actors	<i>User</i>	
Trigger	<i>Sudah selesai melakukan keperluan di website</i>	
Preconditions	<i>User melakukan login</i>	
Postcondition	<i>Sudah selesai dengan urusannya sebagai user</i>	
Success Scenario	Actor	System
	1. <i>Klik logout</i>	<i>Redirect</i> halaman <i>login</i>
Alternative Flows		

Tabel 3. 5 Spesifikasi *Log out user*

Sumber: Dokumentasi praktikan

Use Case Name	Login	
Actors	Admin	
Trigger	Tambah, <i>update</i> dan <i>delete form</i> identifikasi akar masalah, Faktor pendukung keberhasilan dan terobosan karya inovatif	
Preconditions	Admin dapat masuk ke dalam sistem <i>website</i>	
Postcondition	Admin harus sudah login dan sudah bisa melakukan kegiatan yang berhubungan dengan <i>website</i> .	
Success Scenario	Actor	System
	1. Admin membuka halaman <i>website</i>	Menampilkan <i>form login</i>
	2. Admin mengisi <i>username</i> dan <i>password</i>	Jika <i>username</i> dan <i>password</i> yang dimasukkan salah maka akan terdapat alert " <i>Username & password tidak tepat</i> " dan kembali ke halaman <i>login</i> , jika benar akan masuk kehalaman utama <i>admin</i>
Alternative Flows	Jka <i>admin</i> salah menginput <i>username</i> dan <i>password</i> , sistem akan memberikan <i>alert</i> yang menyatakan " <i>Username & password tidak tepat</i> ", dan sistem akan meminta <i>admin</i> memasukan kembali <i>username</i> dan <i>password</i> secara tepat	

Tabel 3.6 Spesifikasi *Login Admin*

Sumber: Dokumentasi praktikan

Use Case Name	Melihat details dan menghapus <i>form task</i>	
Actors	Admin	
Trigger	Tambah, <i>update</i> dan <i>delete</i> dokumen	
Preconditions	Admin memasuki halaman <i>login</i>	
Postcondition	<i>Form task</i> berhasil <i>ter-update/terhapus</i> .	
Success Scenario	Actor	System
	1. Admin membuka halaman website	Menampilkan <i>form login</i>
	2. Admin mengisi <i>username</i> dan <i>password</i>	Jika <i>username</i> dan <i>password</i> yang dimasukkan salah maka akan terdapat <i>alert "Username & password tidak tepat"</i> dan kembali ke halaman <i>login</i> , jika benar akan masuk kehalaman utama <i>admin</i>
	3. Pilih menu <u><i>form task</i></u>	Menampilkan <i>form task</i>
	4. Klik <i>button "action task"</i>	Menampilkan <i>form action task</i>
	5. Terdapat <i>detail</i> dan hapus <i>task</i>	Menampilkan kendala, rencana tindak lanjut, pihak upj terkait dan keterangan tambahan
	6. Pilih <i>task</i> yang mau di hapus dan klik hapus dokumen	Menampilkan <i>alert "Apakah anda yakin ingin menghapus dokumen ini?"</i> Jika iya maka <i>form</i> yang sudah terisi akan terhapus sedangkan jika tidak maka halaman akan kembali ke <i>form task</i> pada awalnya

Alternative Flows		
--------------------------	--	--

Tabel 3.7 Spesifikasi *Form Task Admin*

Sumber: Dokumentasi praktikan

Use Case Name	Logout	
Actors	Admin	
Trigger	Sudah selesai melakukan keperluan di website	
Preconditions	Admin melakukan <i>login</i>	
Postcondition	Sudah selesai dengan urusannya sebagai <i>admin</i> website	
Success Scenario	Actor	System
	1. Klik <i>logout</i>	<i>Redirect</i> halaman <i>logout</i>
Alternative Flows		

Tabel 3.8 Spesifikasi *Log Out Admin*

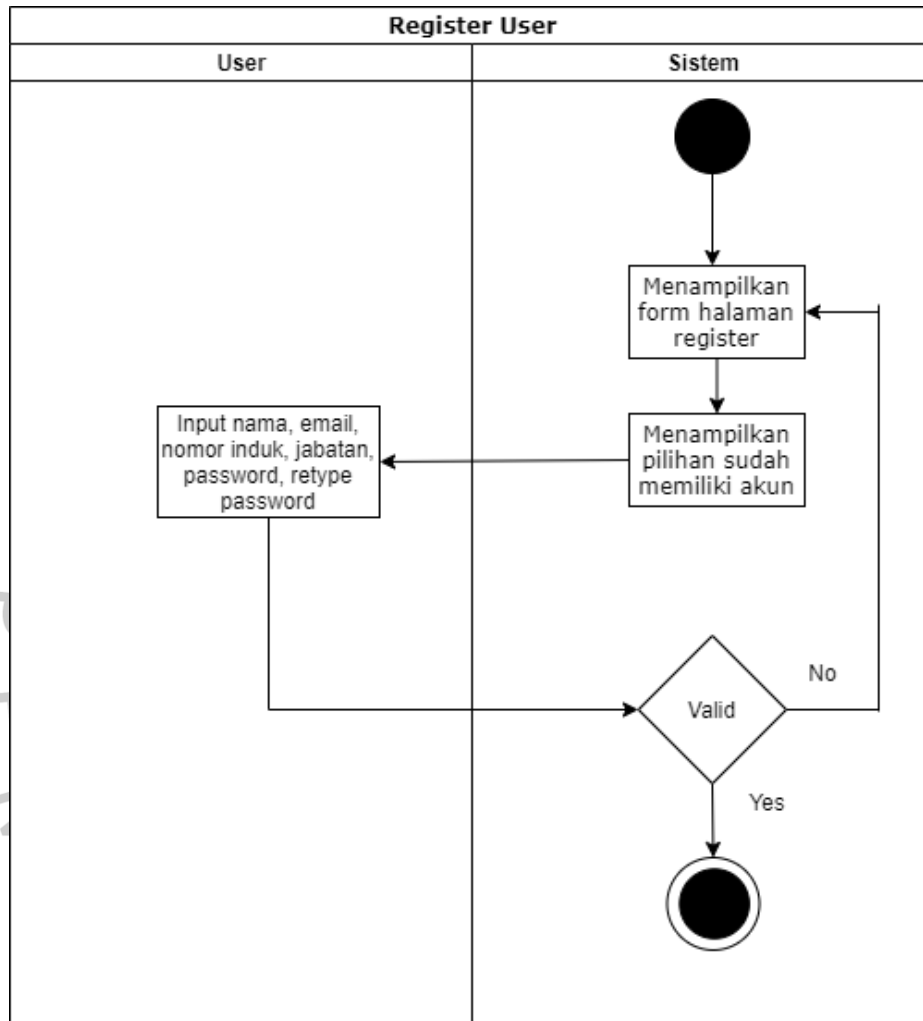
Sumber: Dokumentasi praktikan

C. Activity Diagram

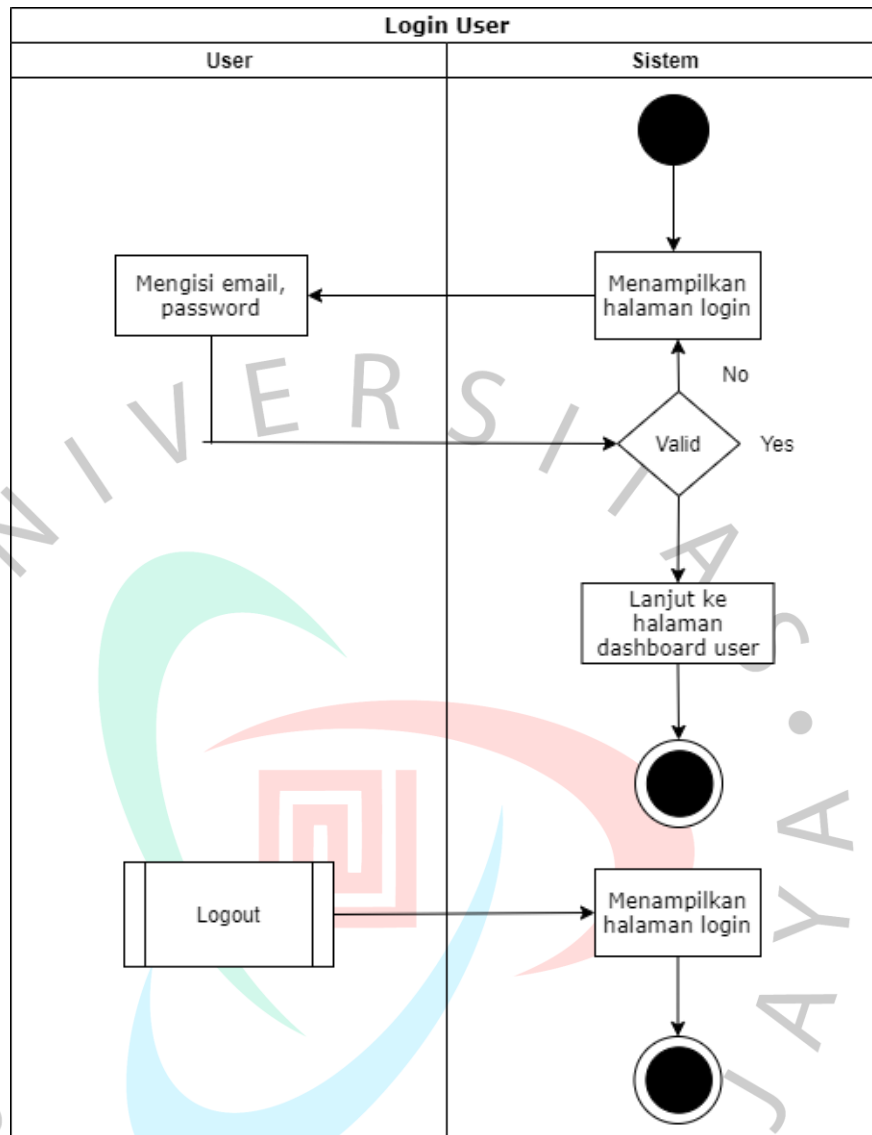
Activity diagram merupakan suatu rancangan struktur alur aktivitas atau Workflow dalam sebuah sistem yang akan dikerjakan. Activity diagram juga di pakai untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. Activity diagram atau diagram aktifitas memiliki komponen dengan bentuk yang berhubungan dengan tanda panah atau simbol, tanda atau simbol itu memiliki artian mengarahkan keurutan aktifitas yang terjadi dari awal hingga akhir.

Dalam membuat analisa ini terdapat dua *actor*

yang berbeda dalam berkomunikasi dengan sistem yaitu ada *User* dengan sistem dan ada *admin* dengan sistem seperti pada **Gambar 3.5** sampai **Gambar 3.9.3**.

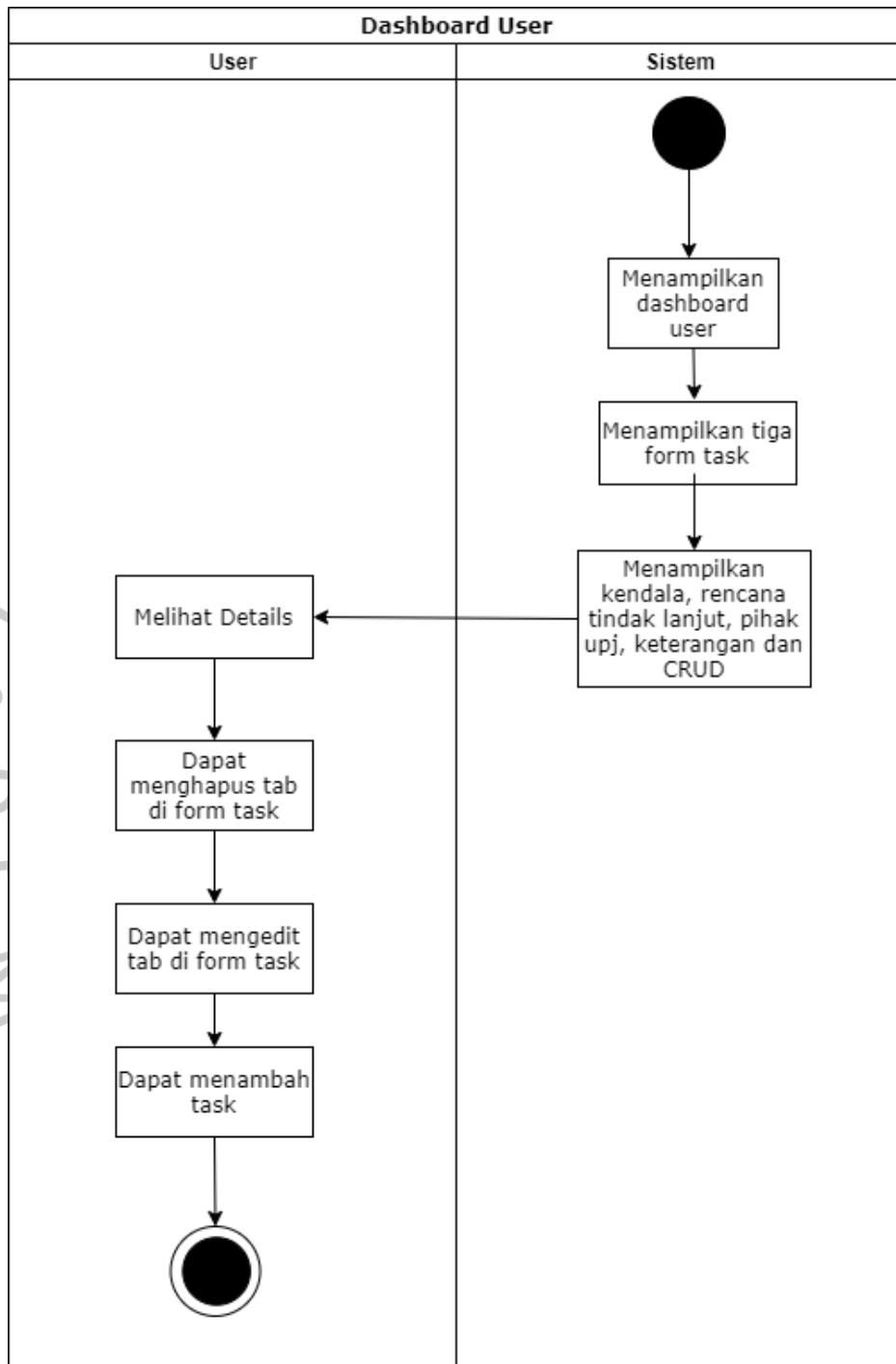


Gambar 3.5 Activity Diagram Register User
Sumber: Dokumentasi praktikan



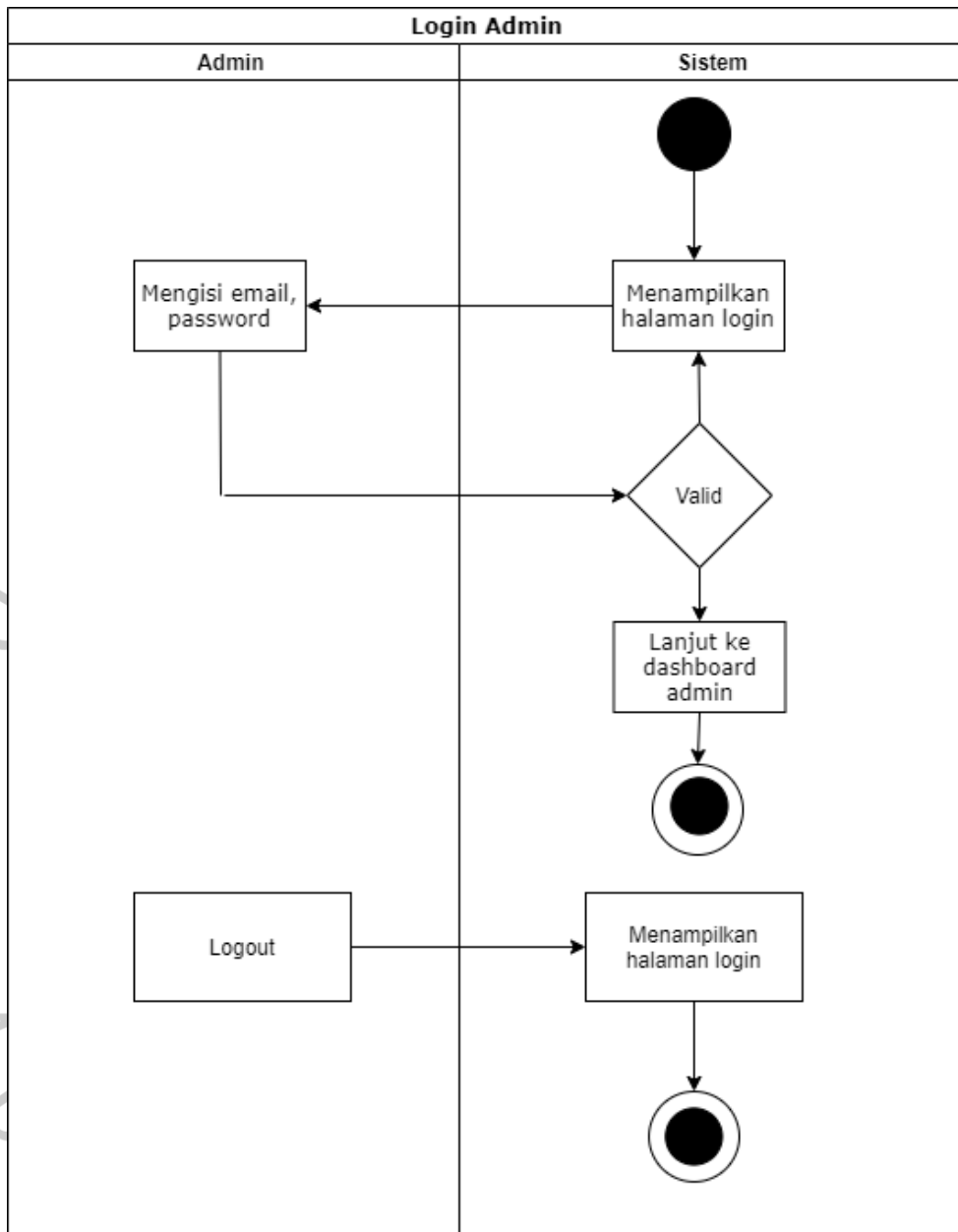
Gambar 3.6 Diagram Activity Login User

Sumber: Dokumentasi praktikan



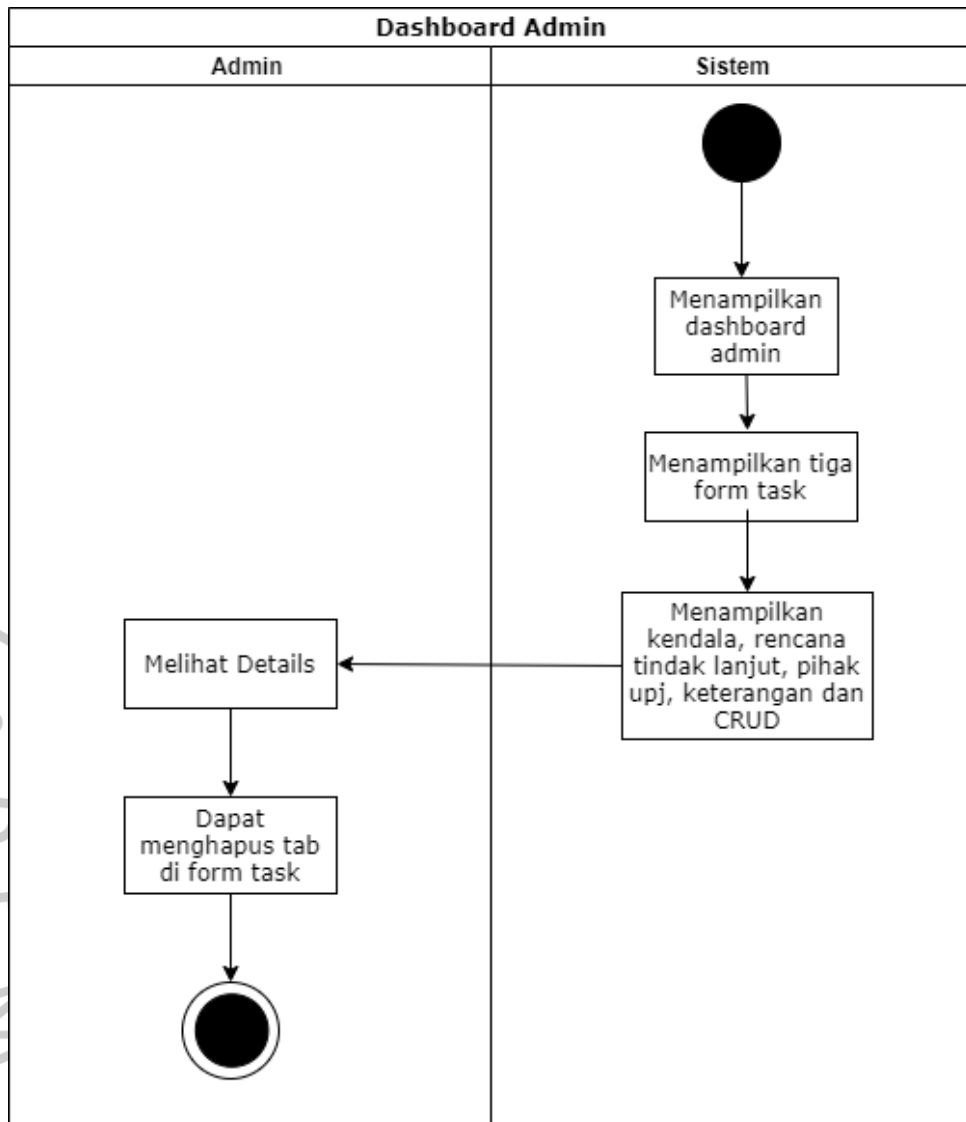
Gambar 3.7 Activity Diagram Dashboard User

Sumber: Dokumentasi praktikan



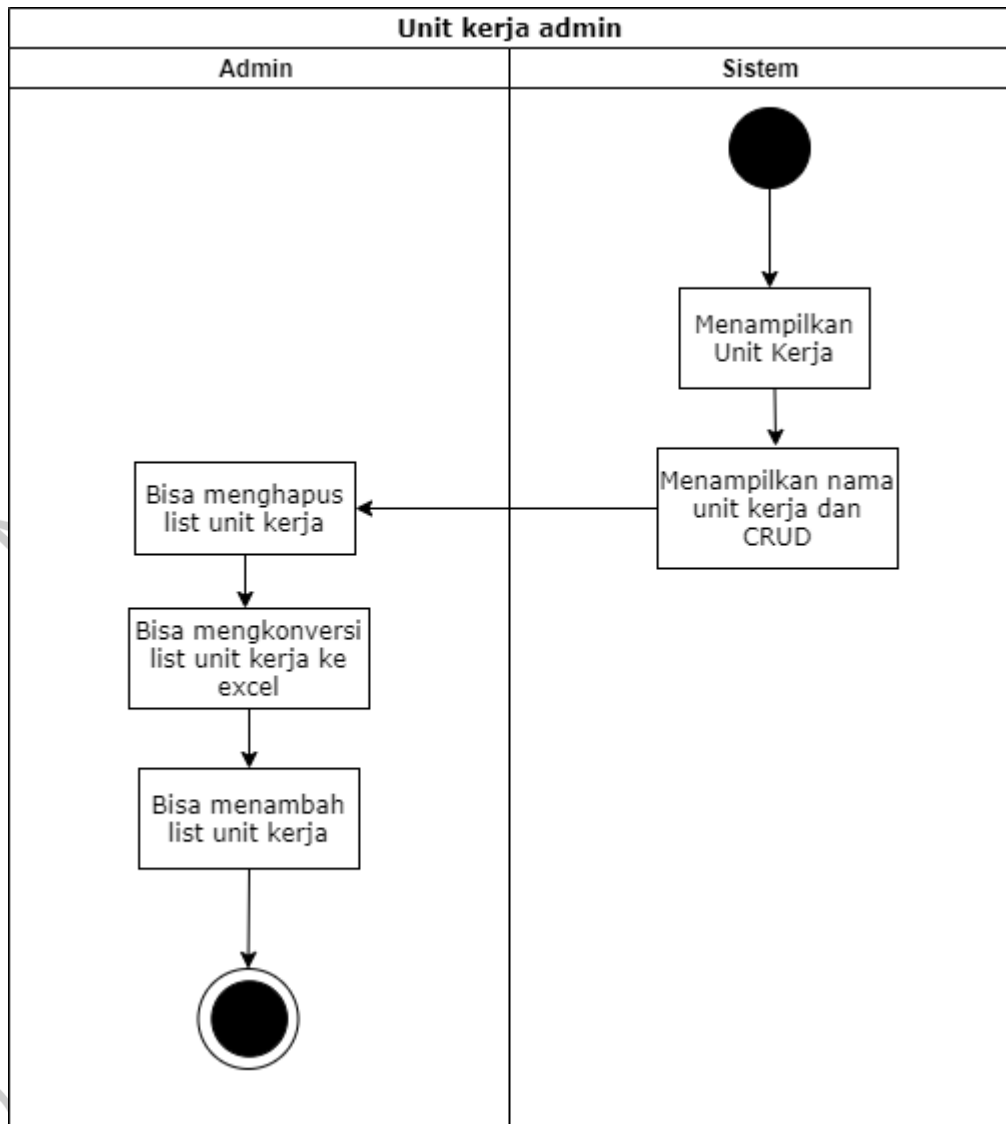
Gambar 3.8 Activity Diagram Login Admin

Sumber: Dokumentasi praktikan



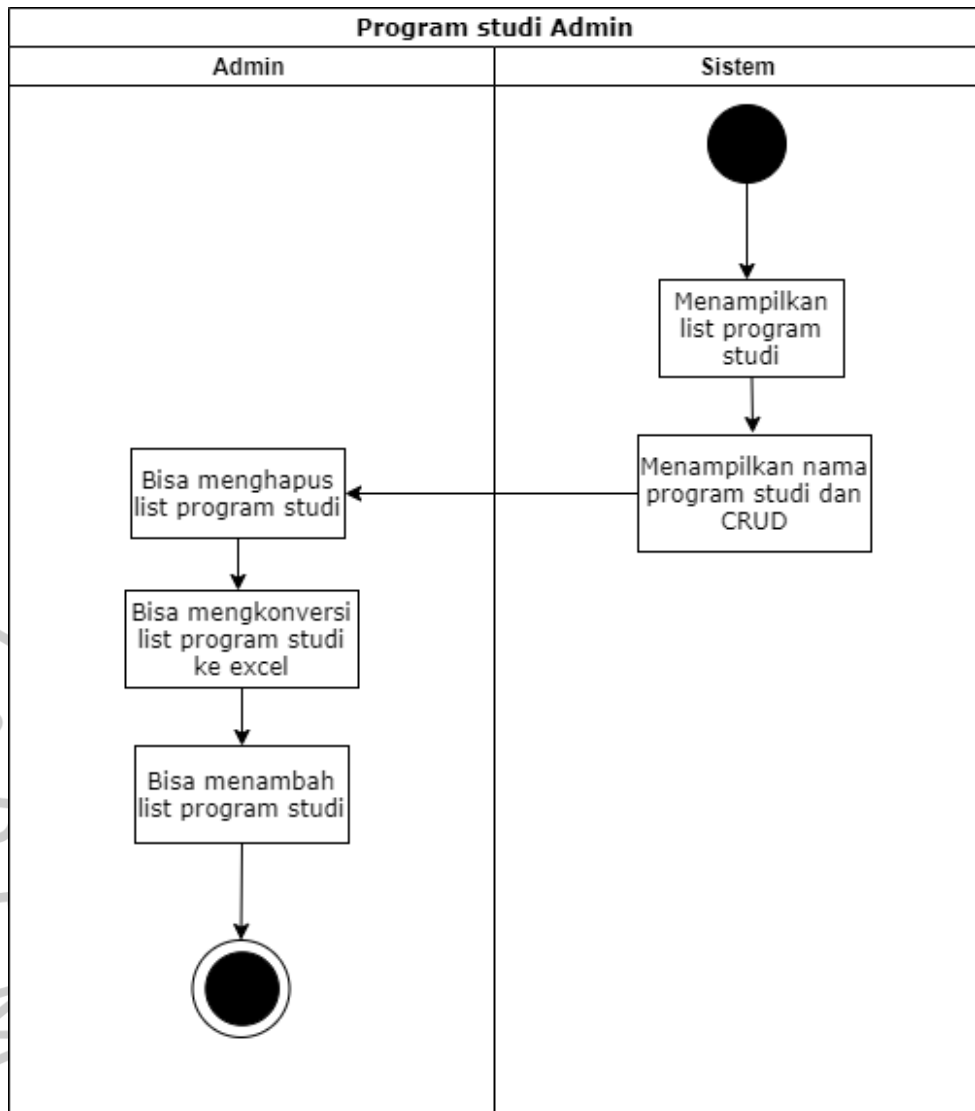
Gambar 3.9 Activity Diagram Dashboard Admin

Sumber: Dokumentasi praktikan



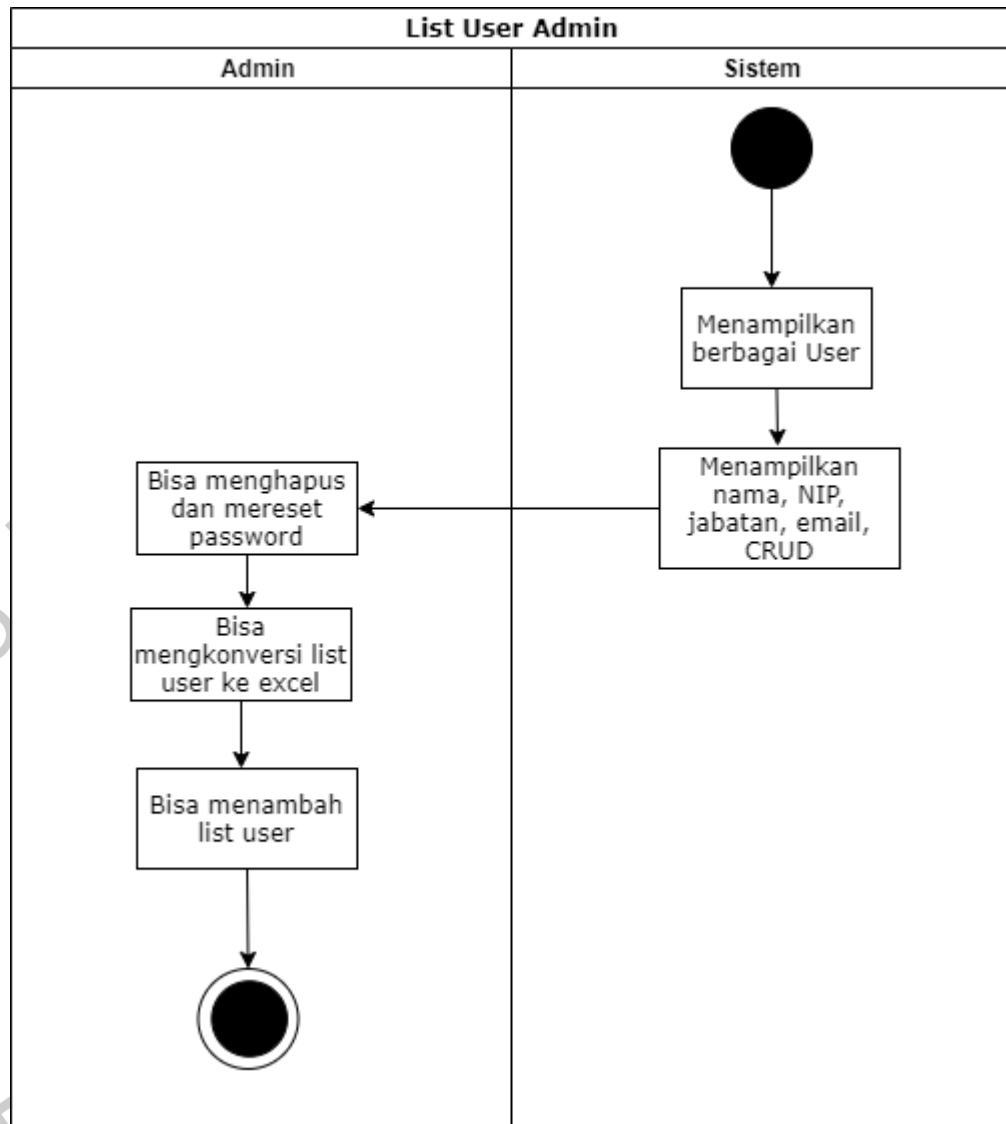
Gambar 3.9.1 Activity Diagram Unit Kerja

Sumber: Dokumentasi praktikan



Gambar 3.9.2 Activity Diagram Program Studi Admin

Sumber: Dokumentasi praktikan

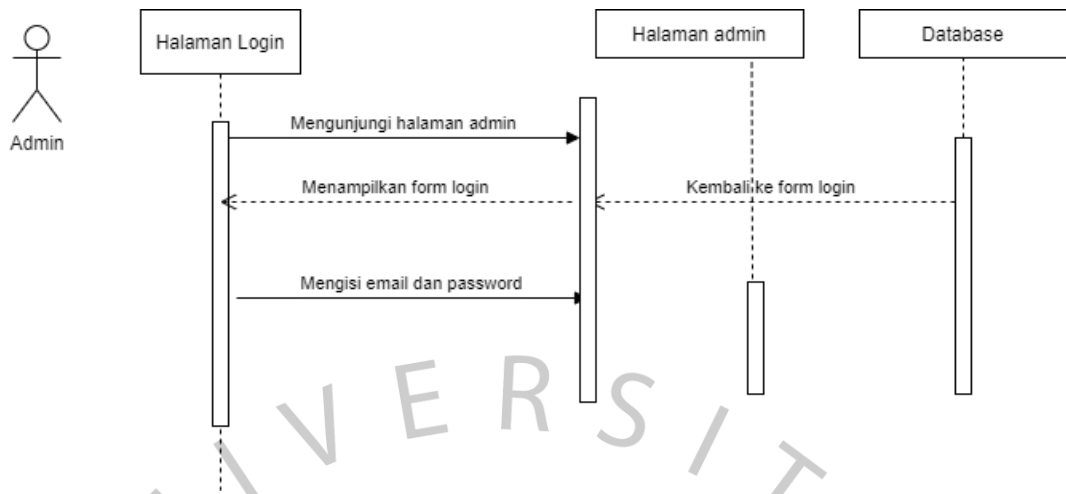


Gambar 3.9.3 Activity Diagram Program List User Admin

Sumber: Dokumentasi praktikan

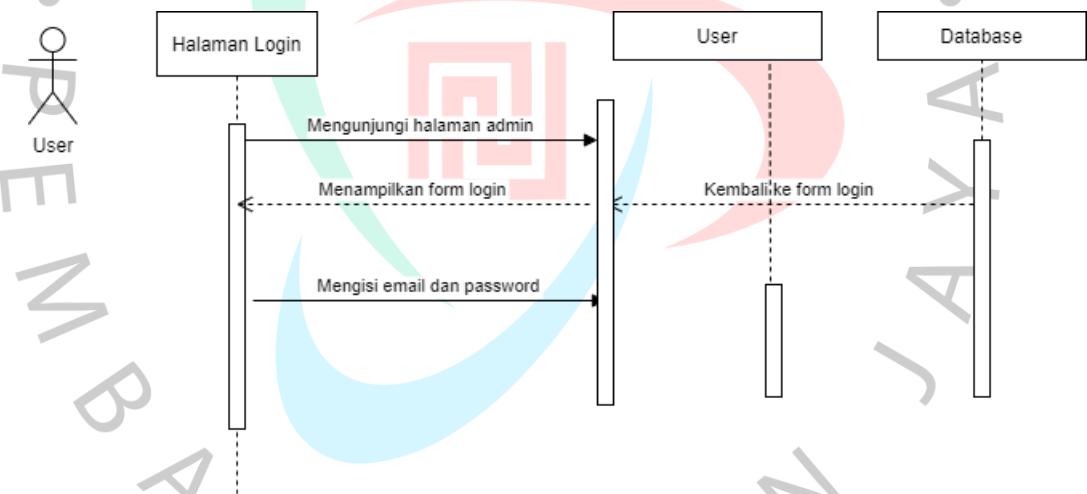
D. Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan alur diagram yang dinamis di beberapa objek. Tujuannya untuk menampilkan serangkaian pesan yang ditujukan kepada objek dan interaksi antar objek (Pratama, 2019). Berikut ini adalah contoh *Sequence* diagram pada **Gambar 3.9.4** sampai **Gambar 3.9.5**.



Gambar 3.9.4 *Sequence Diagram Admin*

Sumber: Dokumentasi praktikan



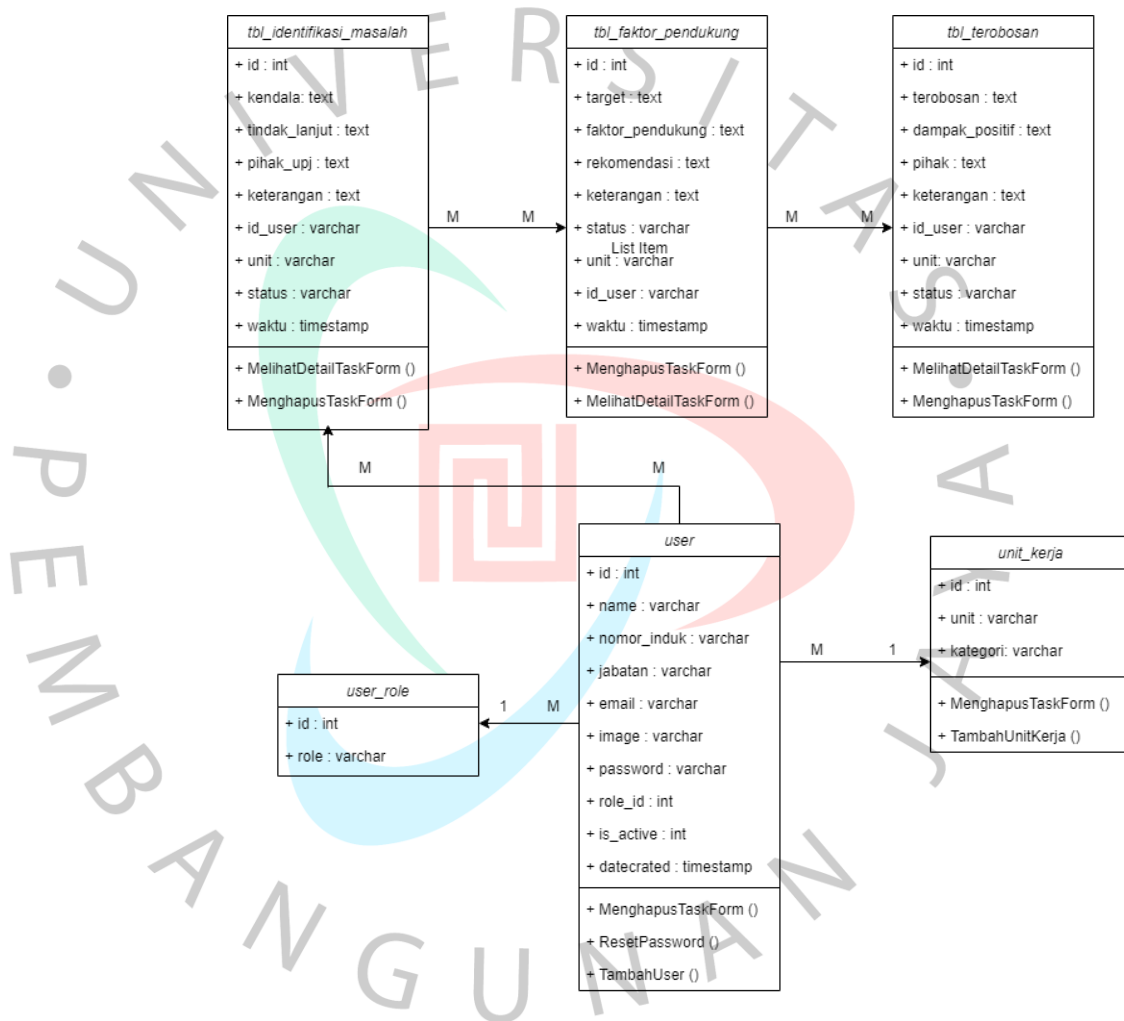
Gambar 3.9.5 *Sequence Diagram User*

Sumber: Dokumentasi praktikan

E. Class Diagram

Class Diagram merupakan gambaran dari struktur program sistem dengan cara memodelkan

kelas sesuai dengan struktur kelas, memiliki atribut dan hubungan masing-masing antar objek, *Class diagram* merupakan alur perancangan database pada sebuah aplikasi yang dirancang (Imanudin, 2021). **Gambar 3.9.6** Terdapat *class diagram* yang sudah di buat praktikan sebagai berikut:



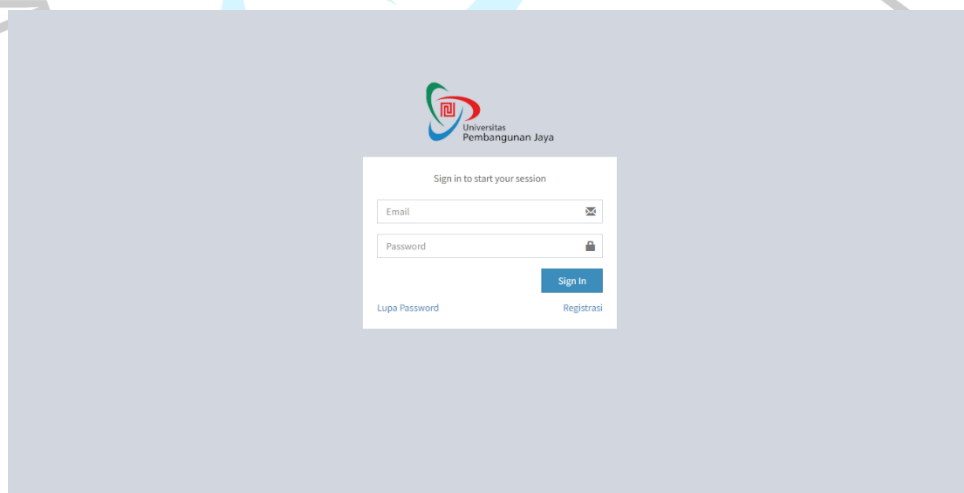
Gambar 3.9.6 *Class Diagram*

Sumber: Dokumentasi praktikan

3.2.3.2 Desain dan Implementasi Sistem

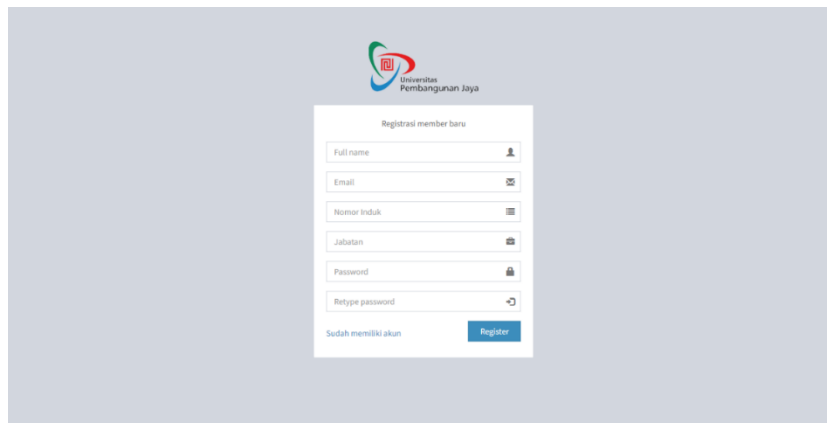
Pada tahap ini, bersama rekan tim perancang aplikasi *Reporting* menentukan tampilan yang digunakan oleh sistem aplikasi *Reporting*. Bersama *Front-End developer* memberikan desain tampilan aplikasi *Reporting* yang diinginkan kepada Kepala bagian dan *staff* LPMU. Apabila desain yang diberikan sesuai dan terpenuhi kemudian diimplementasikan oleh *front-end developer*. Selain itu bila tampilan tidak diharapkan dengan Lembaga unit kerja LPMU, praktikan bersama *front-end developer* akan menerima masukan dan diskusi dalam ketidaksesuaian desain yang diharapkan, dari masukan tersebut Praktikan akan melakukan perubahan untuk tercapainya kesepakatan dalam perancangan desain aplikasi.

Setelah tahap Implementasi berhasil dan sesuai kebutuhan, Praktikan bersama rekan tim melakukan perancangan aplikasi berikut rancangan yang di kerjakan:



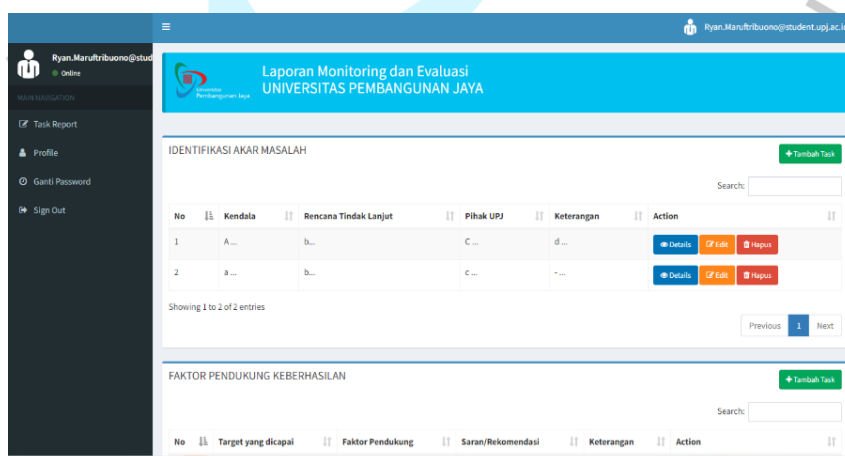
Gambar 3.9.7 Login Aplikasi *Reporting*

Sumber: Dokumentasi praktikan



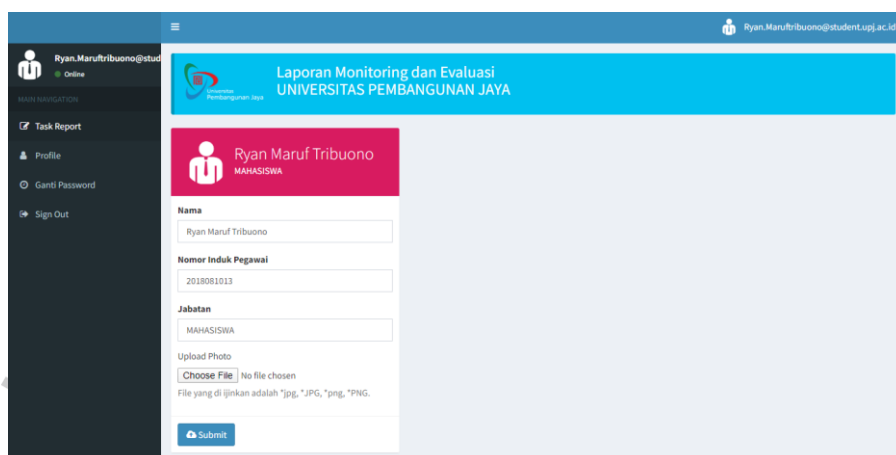
Gambar 3.9.8 Tampilan Registrasi Aplikasi *Reporting*
 Sumber: Dokumentasi praktikan

Pada **Gambar 3.9.7** dan **3.9.8** merupakan tampilan pada halaman *login* dan registrasi. Pada halaman registrasi terdapat *first name* untuk *user* akses, lalu terdapat *email*, nomor induk, jabatan, *password* dan *Retype password*. Ketika sudah mengisi registrasi lakukan ada pemberitahuan bahwa registrasi sudah dilakukan. Setelah melakukan registrasi user mendapatkan email untuk verifikasi akun, setelah verifikasi akun user bisa lakukan *login* dengan mengisi *email*, *password* pada halaman *login*.



Gambar 3.9.9 Tampilan *Dashboard* aplikasi *Reporting*
 Sumber: Dokumentasi praktikan

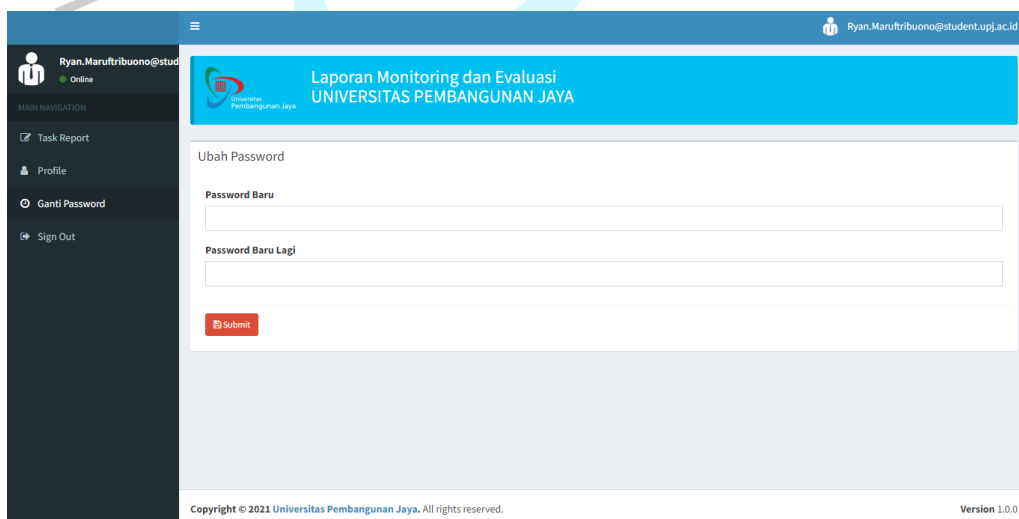
Pada **Gambar 3.9.9** merupakan pengisian *Form Task Report* yang harus dilengkapi oleh *user*, pengisian data tersebut terdiri dari *Form* Identifikasi Akar Masalah, Faktor Pendukung Keberhasilan, dan Terobosan Karya Inovatif.



Gambar 4.0 Tampilan *Profile* Aplikasi *Reporting*

Sumber: Dokumentasi praktikan

Pada **Gambar 4.0** merupakan pengisian *Form Profile* yang harus dilengkapi oleh *user*, pengisian data tersebut terdiri dari *Form* Nama, Nomor Induk, dan Foto.



Gambar 4.1 Tampilan *Ganti Password* Aplikasi *Reporting*

Sumber: Dokumentasi praktikan

Pada **Gambar 4.1** merupakan pengisian *Form Ganti* yang harus dilengkapi oleh user, pengisian data tersebut terdiri dari *Password Baru*, dan *Password Baru Lagi* akan tetapi fitur ini sudah disediakan juga di *Forgot Password* pada tampilan awal *Login aplikasi*.

3.3 Kendala yang Dihadapi

Kendala yang dialami Praktikan selama pelaksanaan Kerja Profesi adalah sebagai berikut:

1. Pada saat pelaksanaan Kerja Profesi, sedang terjadi pandemi Covid-19 banyak perusahaan maupun instansi yang menerapkan sistem kerja ini sehingga mengharuskan Praktikan bekerja secara *Work from Home* (WFH) sehingga sulit untuk berdiskusi dan berkoordinasi secara langsung dengan Bagian LPMU UPJ.
2. Praktikan kurang memahami dan terkadang misskomunikasi sesuai dengan permintaan dan arahan yang diberikan dari kepala bagian LPMU.

3.4 Cara Mengatasi Kendala

Praktikan mengatasi kendala yang terdapat di atas dengan cara:

1. Menggunakan aplikasi WhatsApp dan Microsoft Teams untuk berkomunikasi yang berkaitan dengan Kerja Profesi dan melakukan presentasi.
2. Berdiskusi dengan rekan selama kerja profesi yaitu Muhammad Farhan Syah selaku *Front-End Developer* untuk mencari solusi dalam mengatasi permasalahan ketika perancangan aplikasi.
3. Mencari berbagai referensi melalui internet, forum, dan github untuk membantu menyelesaikan masalah pada pembuatan sistem aplikasi yang dikerjakan penulis.

3.5 Pembelajaran yang Diperoleh dari Kerja Profesi

Selama pelaksanaan Kerja Profesi, Praktikan mempelajari hal-hal baru dan mendapatkan pengetahuan yang sebelumnya tidak terlalu dikuasai oleh Praktikan sebelumnya, mulai dari teknik-teknik pembuatan website, Praktikan

juga dapat mengimplementasikan kemampuan yang di dapat selama perkuliahan berlangsung yaitu, membangun sebuah aplikasi website, dan mendapatkan pengetahuan lebih mengenai pengembangan desain yang baik untuk sebuah aplikasi. Penulis juga mengetahui bagaimana proses berjalannya Monev yang di selenggarakan oleh Bagian LPMU Universitas Pembangunan Jaya.



