

Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem. Setelah terintegrasi seluruh sistem, akan dilakukannya pengujian sistem dengan menggunakan metode black box.

e. Operation & Maintenance

Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya dan dievaluasi apakah sistem sudah sesuai dengan rancangan awal serta proses bisnis yang sudah ditentukan.

BAB III ANALISIS PERENCANAAN

3.1 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk proses pengumpulan data yang dibutuhkan untuk dapat digunakan sebagai pendukung aplikasi yang akan dibuat.

3.1.1 Analisis Fungsionalitas

Pada tahap ini dilakukan identifikasi data dan informasi terhadap apa yang dibutuhkan pada sistem dalam merancang sebuah aplikasi. Data yang digunakan adalah data dari hasil wawancara dan data *record maintenance reserve* yang petugas isi. Dari hasil wawancara diambil poin-poin penting terkait fitur yang akan dibuat.

Setelah melakukan wawancara untuk menentukan fitur aplikasi dibutuhkan data *record maintenance reserve* yang petugas isi. Petugas *maintenance* mengisi *Flight Hours (FH)*, *Flight Cycle (FC)*, *Time Since New (TSN)*, dan *Cycle Since New (CSN)* dalam kurung waktu satu bulan sekali.

Berdasarkan hasil wawancara dan mendapatkan data *record maintenance reserve* yang petugas isi diperoleh gambaran dalam pembuatan fitur aplikasi yang ditampilkan pada *table* fitur.

Tabel 3.1 Tabel Fitur

No	Fitur	Penjelasan
----	-------	------------

1.	Mengelola Pengguna	Fitur mengelola pengguna memungkinkan pengguna untuk membuat, mengedit, melihat <i>user</i> . Sebelum menggunakan aplikasi harus mendaftar terlebih dahulu lalu login masuk ke dalam aplikasi.
2.	Mengelola <i>Maintenance</i>	Fitur mengelola <i>maintenance</i> memungkinkan pengguna untuk membuat, mengedit, melihat, dan menghapus <i>maintenance</i> . Setelah mengisi <i>maintenance</i> , pengguna mencocokkan jenis <i>aircraft</i> /pesawat dengan jenis <i>maintenance</i> .
3.	Mengelola <i>Engine</i>	Fitur mengelola <i>engine</i> memungkinkan pengguna untuk membuat, mengedit, melihat, dan menghapus <i>engine</i> . Setelah mengisi <i>engine</i> , pengguna mencocokkan jenis <i>aircraft</i> /pesawat dengan jenis <i>engine</i> .
4.	Mengelola <i>Aircraft</i>	Fitur mengelola <i>aircraft</i> memungkinkan pengguna untuk membuat, mengedit, melihat <i>aircraft</i> . Setiap <i>aircraft</i> wajib mempunyai jenis <i>engine</i> dan jenis <i>maintenance</i> .
5.	Mengelola <i>Maintenance Reserve</i>	Fitur mengelola <i>maintenance reserve</i> memungkinkan pengguna untuk membuat, mengedit, melihat, dan menghapus <i>maintenance reserve</i> . Pengguna mengisi utilisasi pesawat sesuai pemakaian pesawat tersebut.
6.	Melihat <i>Aircraft Utilization</i>	Fitur melihat <i>Aircraft Utilization</i> memungkinkan pengguna untuk melihat utilitas pesawat berdasarkan jenis pesawat dan tanggalnya.

3.1.2 Analisis Non Fungsionalitas

Pencatatan utilitas pesawat mengacu kepada *Critical Safe Software* yaitu komponen *availability, reability, safety, dan security*. Sistem akan dibuat untuk meminimalisir kesalahan yang menyebabkan resiko bagi pengguna.

a. Availability

Sistem yang dibuat harus selalu tersedia atau dapat diakses oleh seluruh pegawai maskapai pesawat dengan mudah dan cepat. Dengan menggunakan komponen availability maka *system* haruslah di *upload* ke *server* agar seluruh pegawai dengan mudah memakainya.

b. Reliability

Untuk memastikan sistem berjalan sesuai yang diinginkan aplikasi harus dilakukan testing terlebih dahulu. Aplikasi akan menjalankan Blackkbox testing sebelum digunakan agar mengetahui apakah aplikasi sudah sesuai dengan yang diinginkan.

c. Safety

Sistem akan dibuat untuk meminimalisir kesalahan yang menyebabkan resiko bagi pengguna. Karna kesalahan sedikit saja dalam menjalankan aplikasi akan berakibat fatal pada pengguna.

d. Security

Dengan menggunakan *framework* Laravel Menghilangkan kekhawatiran tentang *low-level security*, karena *framework* laravel telah di desain khusus untuk menangani masalah security yang umumnya menjadi masalah.

3.1.3 Analisis Pengguna

Pengguna yang dapat menggunakan aplikasi ini adalah sebagai berikut :

a. Manager

Manager adalah pengguna yang dapat mengelola semua fitur yang terdapat dalam aplikasi. Manager dapat mengelola *user, aircraft, engine, maintenance* ,dan *maintenance reserve*.

b. Admin Aircraft

Admin Aircraft adalah pengguna yang dapat melihat *user, aircraft, engine, dan maintenance*. *Admin Aircraft* berfungsi untuk mengisi *Flight Hours (FH), Flight Cycle (FC), Time Since New (TSN)*, dan *Cycle Since New (CSN)* di dalam fitur *maintenance reserve* dalam kurung waktu satu bulan sekali. Setelah *Admin Aircraft*

mengisi form sesuai yang sudah di buat di dalam aplikasi , akan muncul *invoice* dana cadangan yang harus disiapkan.

c. *User*

User adalah pengguna di luar department management *aircraft*. *User* hanya dapat melihat *user* , utilisasi pesawat dan melihat *maintenance*.

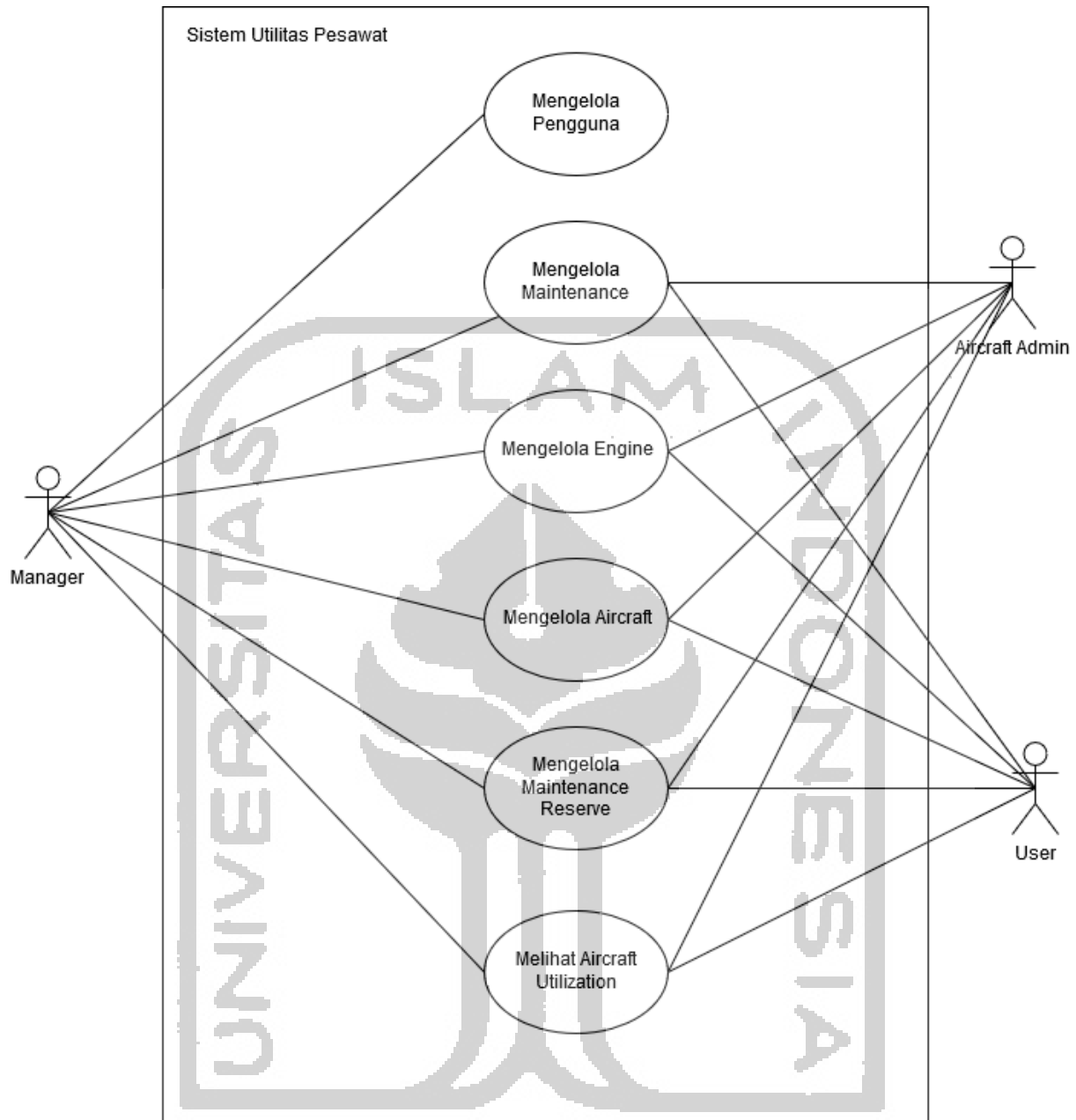
3.2 Use Case Diagram

Use case diagram merupakan diagram yang digunakan untuk menggambarkan tugas yang dikerjakan oleh sistem dan aktor. Merupakan rancangan diagram use case yang dibuat dalam penelitian ini, terdapat tiga *actor* yang berinteraksi dengan sistem aplikasi, yaitu manager , admin *aircraft*, dan *user*. Semua actor harus melakukan *register* dan *login* agar dapat menggunakan aplikasi.

Manager adalah pengguna yang dapat mengelola semua fitur yang terdapat dalam aplikasi. Manager dapat mengelola *user*, *aircraft*, *engine*, *maintenance* ,dan *maintenance reserve*. Manager adalah kepala dari department Management *Aircraft* yang bertanggung jawab atas utilitas pesawat yang dimiliki oleh maskapai. Manager dapat membuatkan akun untuk Admin *Aircraft* dan *User*. Manager juga dapat menghapus Admin *Aircraft* dan *User* jika sudah tidak digunakan.

Admin Aircraft adalah pengguna yang dapat melihat *user*, *aircraft*, *engine*, dan *maintenance*. *Admin Aircraft* berfungsi untuk mengisi *Flight Hours* (FH), *Flight Cycle* (FC), *Time Since New* (TSN), dan *Cycle Since New* (CSN) di dalam fitur *maintenance reserve* dalam kurung waktu satu bulan sekali. Setelah *Admin Aircraft* mengisi *form* sesuai yang sudah dibuat di dalam aplikasi , akan muncul *invoice* dana cadangan yang harus disiapkan. Tugas utama dari *Admin Aircraft* adalah mengisi utilitas pesawat.

User adalah pengguna di luar *department management aircraft*. *User* hanya dapat melihat *user* , utilisasi pesawat dan melihat *maintenance*. *Role* ini berfungsi jika department lain hanya ingin melihat utilitas pesawat dan *maintenance* pesawat. Desain *Usecase* diagram dapat dilihat pada gambar 3.1.



Gambar 3.1 Contoh gambar *use case*

Dalam fitur terdapat 4 *action* yang bisa dilakukan oleh pengguna yaitu lihat, tambah, edit, dan hapus. Masing-masing *role* pengguna mempunyai action yang berbeda dalam mengelola suatu fitur. Berikut tabel yang menunjukkan role pengguna dalam mengelola fitur.

fitur mengelola pengguna hanya manager yang mendapat hak akses penuh yaitu lihat, tambah, edit, dan hapus pengguna. *Role* admin *aircraft* dan *user* hanya mendapatkan hak akses lihat. Fitur mengelola *maintenance* terdapat dua *actor* yang mendapatkan akses penuh lihat, tambah, edit, dan hapus *maintenance* yaitu manager dan admin *aircraft*. *Role user* hanya

mendapatkan hak akses lihat. Fitur mengelola *engine* terdapat dua *actor* yang mendapatkan akses penuh lihat, tambah, edit, dan hapus *engine* yaitu *manager* dan *admin aircraft*. *Role user* hanya mendapatkan hak akses lihat. Fitur mengelola *aircraft* terdapat dua *actor* yang mendapatkan akses penuh lihat, tambah, edit, dan hapus *aircraft* yaitu *manager* dan *admin aircraft*. *Role user* hanya mendapatkan hak akses lihat. Fitur mengelola *maintenance reserve* terdapat dua *actor* yang mendapatkan akses penuh lihat, tambah, edit, dan hapus *maintenance reserve* yaitu *manager* dan *admin aircraft*. *Role user* hanya mendapatkan hak akses lihat. Fitur melihat *Utilization* *role manager, admin aircraft, dan user* hanya mendapatkan hak akses lihat. Fitur melihat *utilization* hanya menampilkan utilisasi pesawat berdasarkan *filter*. Berikut ringkasan *role* pengguna pada tabel *role* pengguna.

Tabel 3.2 Tabel Role Pengguna

No	Pengguna	Role	Hak Akses
1.	Mengelola Pengguna	<i>Manager</i>	lihat, tambah, edit, hapus
		<i>Admin Aircraft</i>	lihat, tambah, edit, hapus
		<i>User</i>	Lihat
2.	Mengelola <i>Maintenance</i>	<i>Manager</i>	lihat, tambah, edit, hapus
		<i>Admin Aircraft</i>	lihat, tambah, edit, hapus
		<i>User</i>	Lihat
3.	Mengelola <i>Egine</i>	<i>Manager</i>	lihat, tambah, edit, hapus
		<i>Admin Aircraft</i>	lihat, tambah, edit, hapus
		<i>User</i>	Lihat
4.	Mengelola <i>Aircraft</i>	<i>Manager</i>	lihat, tambah, edit, hapus
		<i>Admin Aircraft</i>	lihat, tambah, edit, hapus
		<i>User</i>	Lihat
5.	Mengelola <i>Maintenance Reserve</i>	<i>Manager</i>	lihat, tambah, edit, hapus
		<i>Admin Aircraft</i>	lihat, tambah, edit, hapus
		<i>User</i>	Lihat
6.	Melihat <i>Utilization</i>	<i>Manager</i>	Lihat
		<i>Admin Aircraft</i>	Lihat
		<i>User</i>	Lihat

3.3 Activity Diagram

Activity diagram dibuat untuk menggambarkan aliran aktivitas yang terjadi pada sistem di setiap *use case* yang ada. Pada desain *activity diagram* di bawah ini dibentuk berdasarkan desain *use case* yang sebelumnya telah dibentuk. Berikut merupakan desain activity diagram yang digunakan pada *system*.

3.3.1 Activity Diagram Mengelola Pengguna

Gambar 3.2 merupakan gambar *activity diagram* dari *use case* mengelola pengguna. Pengunjung yang telah terdaftar sebagai manager dapat mengelola pengguna dengan memilih klik *add user*, lalu *system* akan mengarahkan ke halaman *add user*, lalu manager mengisi *form add user* dan klik *add*. Setelah di proses database dan berhasil akan di arahkan ke halaman *user*.

3.3.2 Activity Diagram Mengolala Maintenance

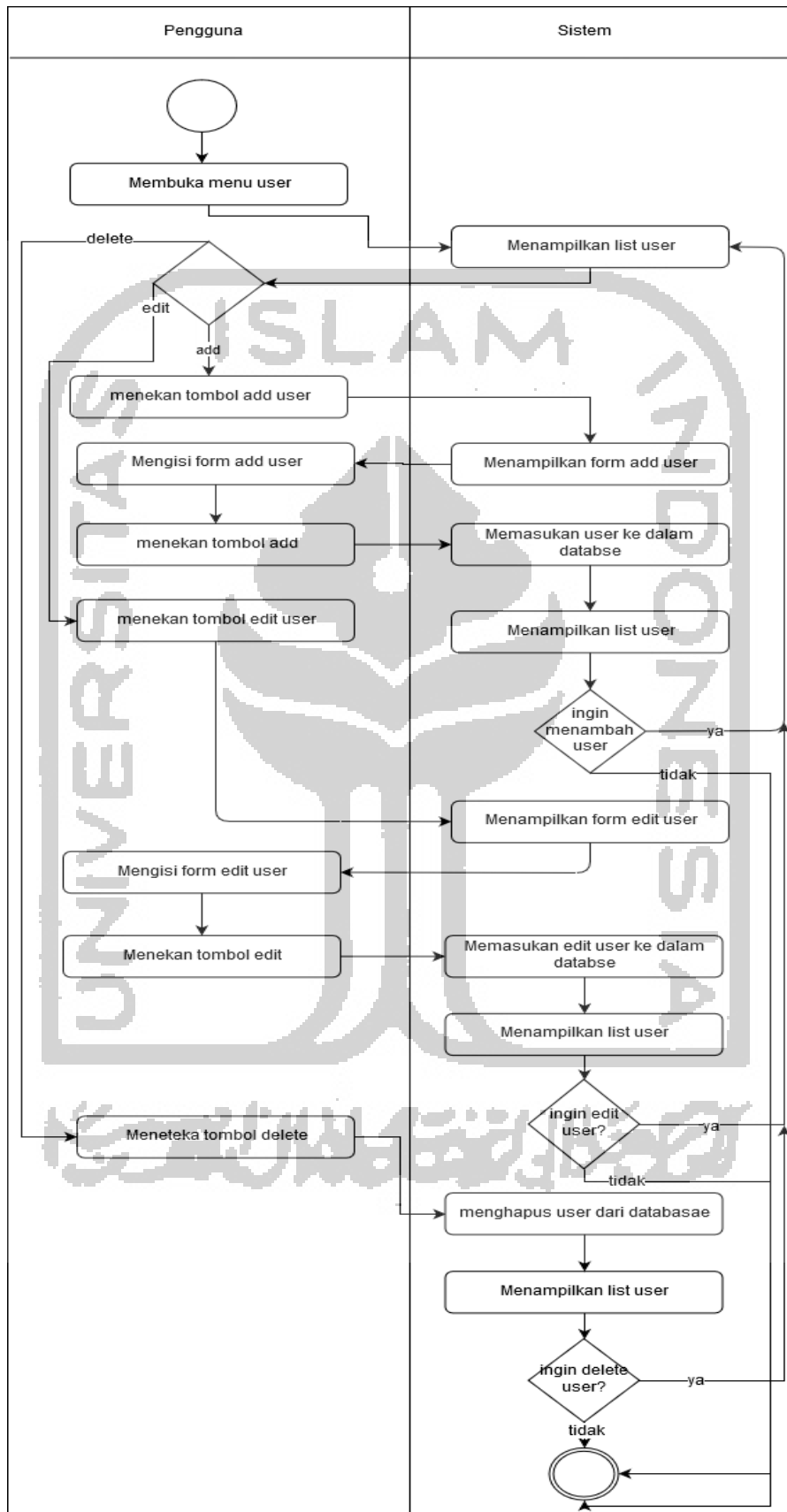
Gambar 3.3 merupakan gambar *activity diagram* dari *use case* mengelola *maintenance*. Pengunjung yang telah terdaftar sebagai pengguna dapat mengelola *maintenance* dengan memilih klik *add maintenance*, lalu *system* akan mengarahkan ke halaman *add maintenance*, lalu pengguna mengisi *form add maintenance* dan klik *add*. Setelah di proses database dan berhasil akan di arahkan ke halaman *maintenance*.

3.3.3 Activity Diagram Mengelola Engine

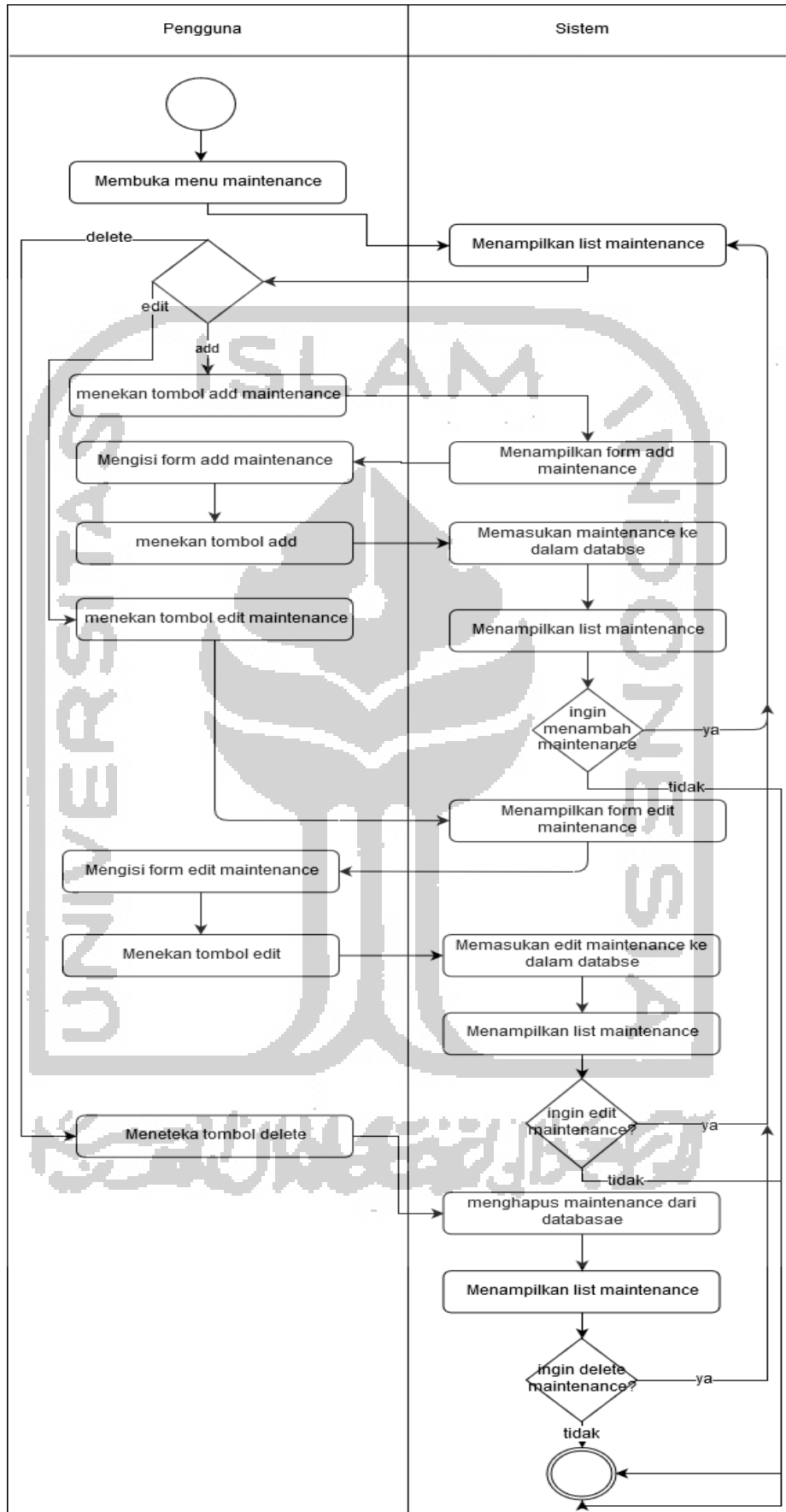
Gambar 3.4 merupakan gambar *activity diagram* dari *use case* mengelola *engine*. Pengunjung yang telah terdaftar sebagai pengguna dapat mengelola *engine* dengan memilih klik *add engine*, lalu *system* akan mengarahkan ke halaman *add engine*, lalu pengguna mengisi *form add engine* dan klik *add*. Setelah di proses database dan berhasil akan di arahkan ke halaman *engine*.

3.3.4 Activity Diagram Mengolala Aircraft

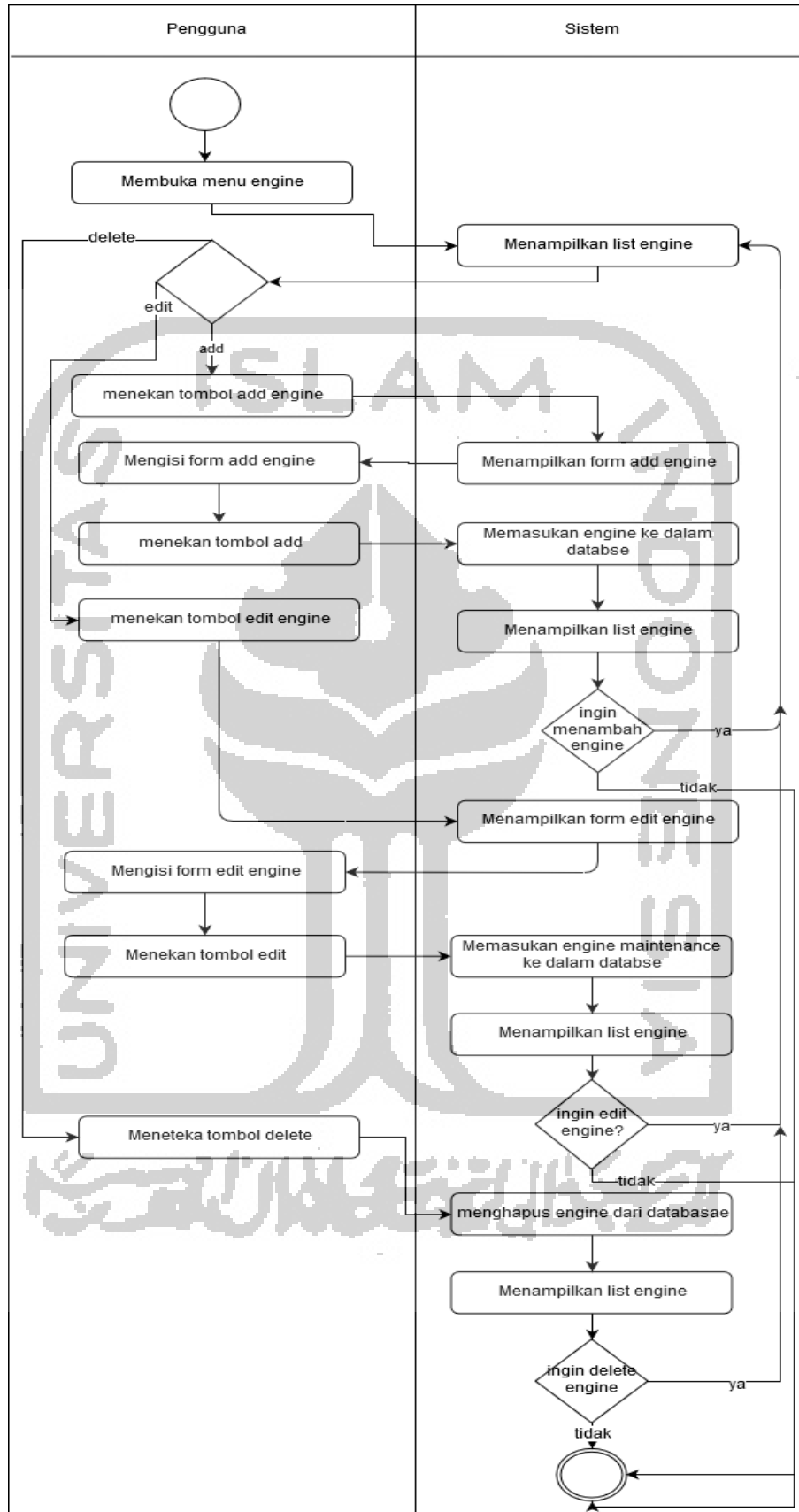
Gambar 3.5 merupakan gambar *activity diagram* dari *use case* mengelola *aircraft*. Pengunjung yang telah terdaftar sebagai pengguna dapat mengelola *aircraft* dengan memilih klik *add aircraft*, lalu *system* akan mengarahkan ke halaman *add aircraft*, lalu pengguna mengisi form *add aircraft* dan klik *add*. Selanjutnya pengguna mengisi *form aircraft, engine 1, engine 2, APU, dan landing gear* dan klik *add*. Setelah di proses database dan berhasil akan di arahkan ke halaman *aircraft*.



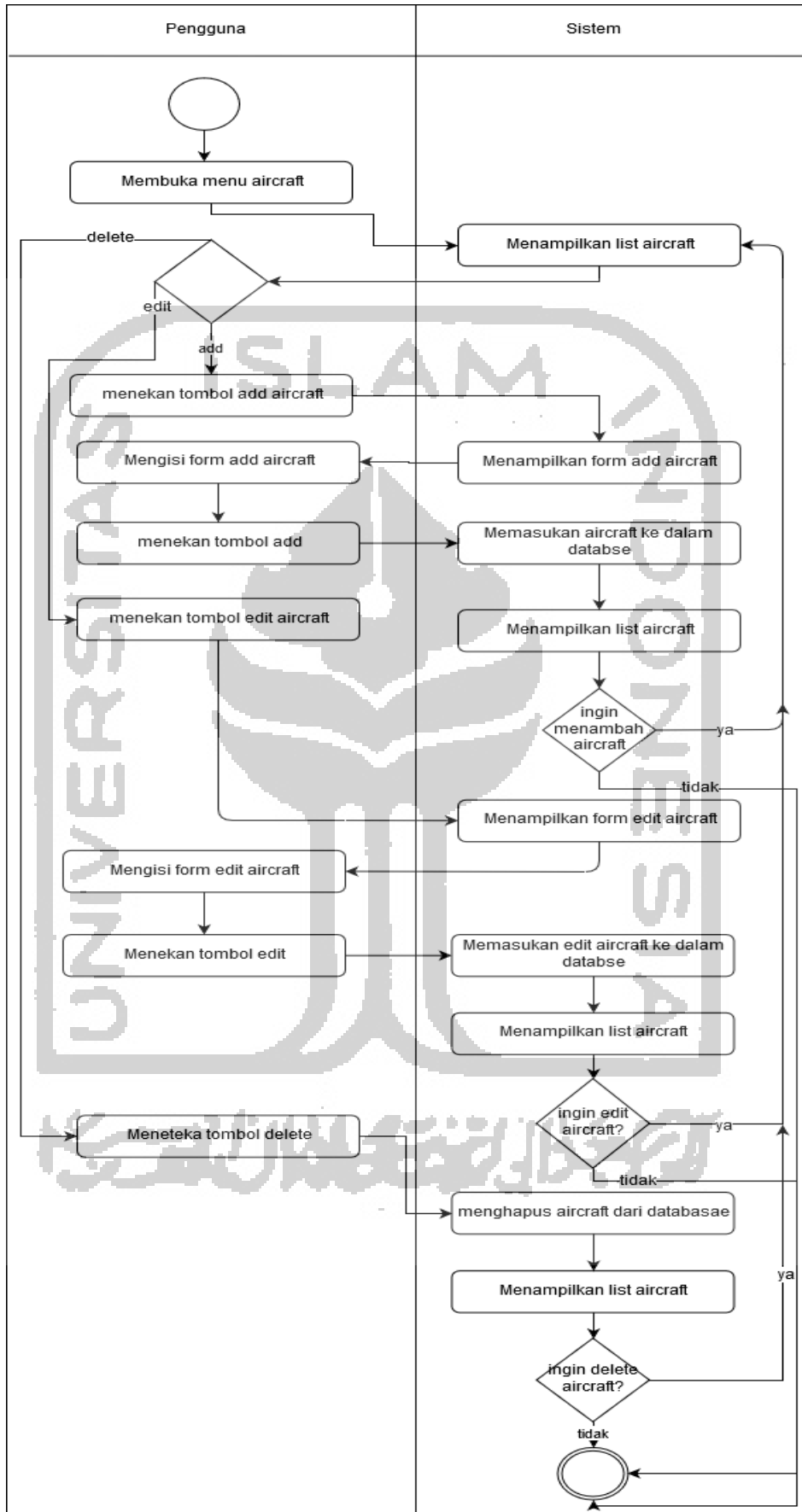
Gambar 3.2 Contoh gambar Activity Diagram mengelola Pengguna



Gambar 3.3 Contoh gambar Activity Diagram Login Maintenance



Gambar 3.4 Contoh gambar Activity Diagram Mengelola Engine



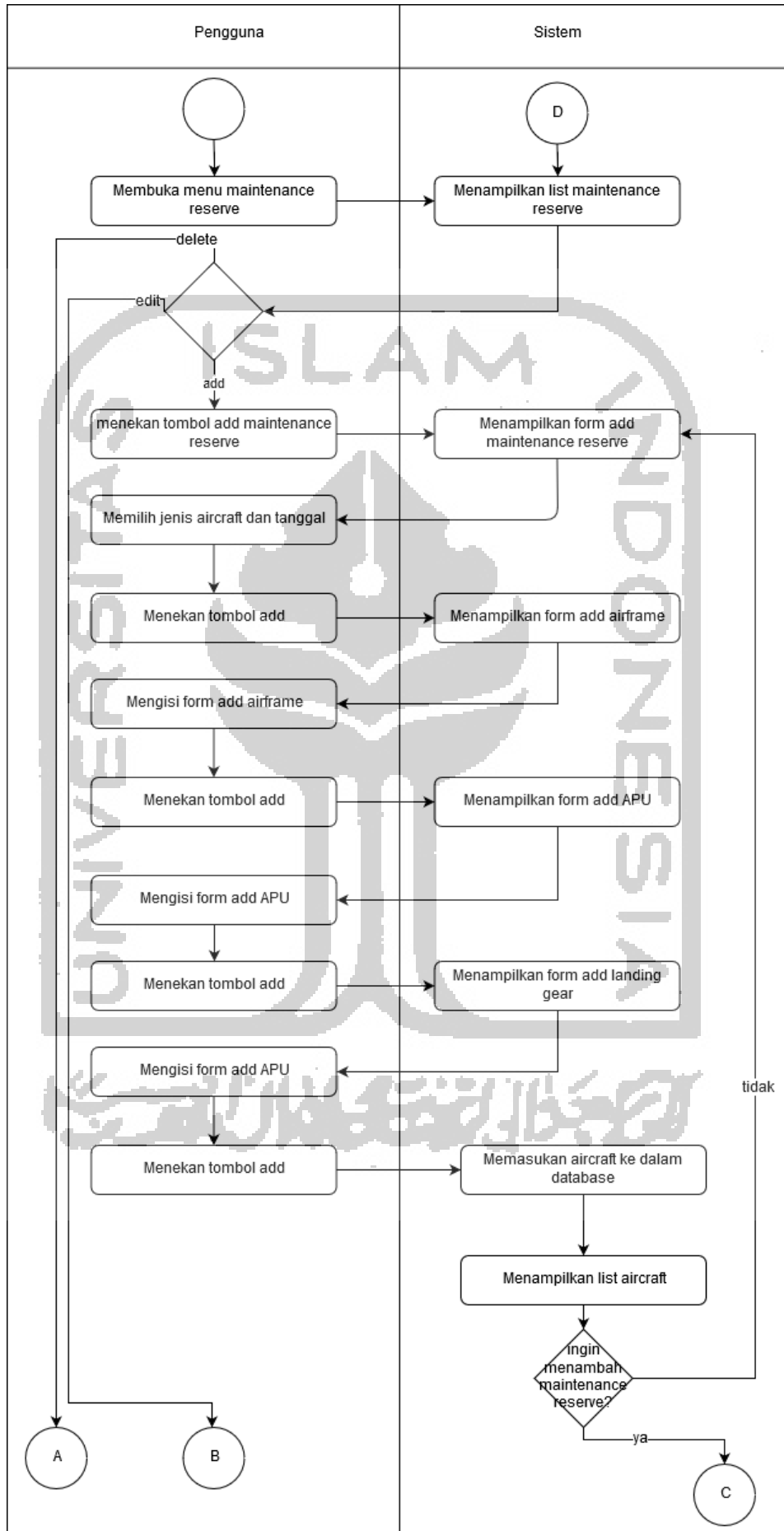
Gambar 3.5 Contoh gambar Activity Diagram Mengelola Aircraft

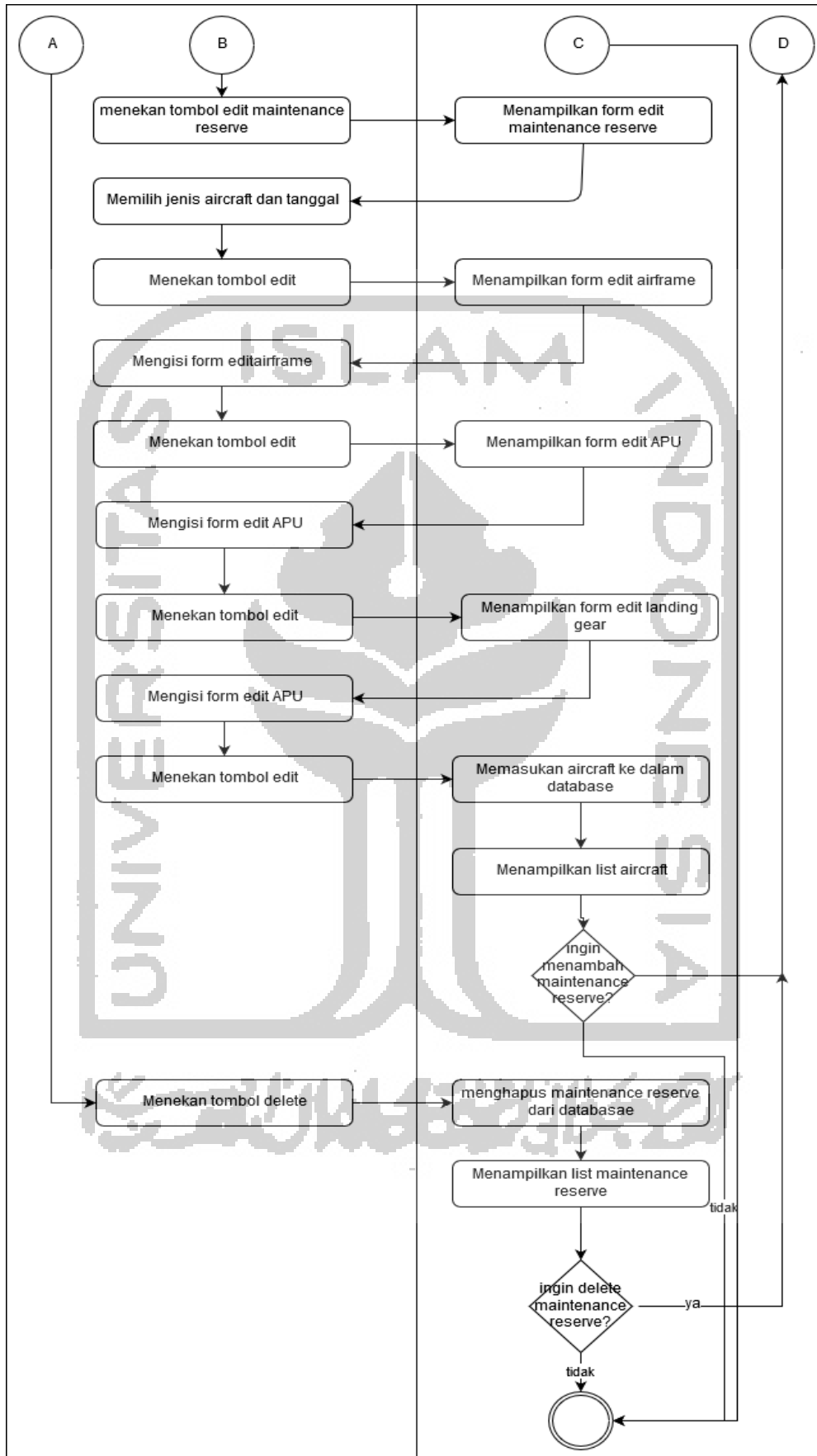
3.3.5 Activity Diagram Mengelola Maintenance Reserve

Gambar 3.6 merupakan gambar *activity diagram* dari *use case* mengelola *maintenance reserve*. Pengunjung yang telah terdaftar sebagai pengguna dapat mengelola *maintenance reserve* dengan memilih klik *add maintenance reserve*, lalu *system* akan mengarahkan ke halaman *add maintenance reserve*, lalu pengguna mengisi form *add maintenance reserve* dan klik *add*. Selanjutnya pengguna memilih jenis *aircraft* dan mengisi utilitas dari *engine 1*, *engine 2*, *APU*, dan *landing gear* dan klik *add*. Setelah di proses database dan berhasil akan di arahkan ke halaman *maintenance reserve*.

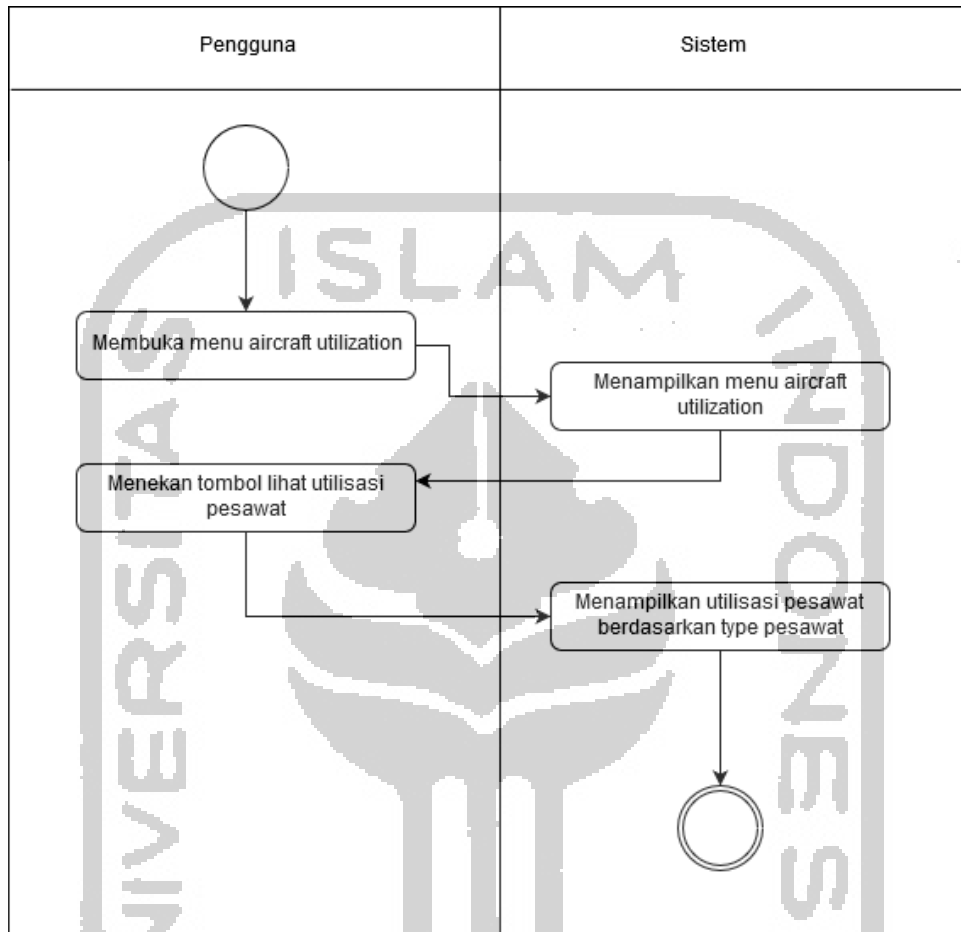
3.3.6 Activity Diagram Melihat Aircraft Utilization

Gambar 3.7 merupakan gambar *activity diagram* dari *use case* mengelola *maintenance reserve*. Pengunjung yang telah terdaftar sebagai pengguna dapat mengelola *maintenance reserve* dengan memilih klik *add maintenance reserve*, lalu *system* akan mengarahkan ke halaman *add maintenance reserve*, lalu pengguna mengisi form *add maintenance reserve* dan klik *add*. Selanjutnya pengguna memilih jenis *aircraft* dan mengisi utilitas dari *engine 1*, *engine 2*, *APU*, dan *landing gear* dan klik *add*. Setelah di proses database dan berhasil akan di arahkan ke halaman *maintenance reserve*.





Gambar 3.6 Contoh gambar *Activity Diagram* Mengelola *Maintenance Reserve*

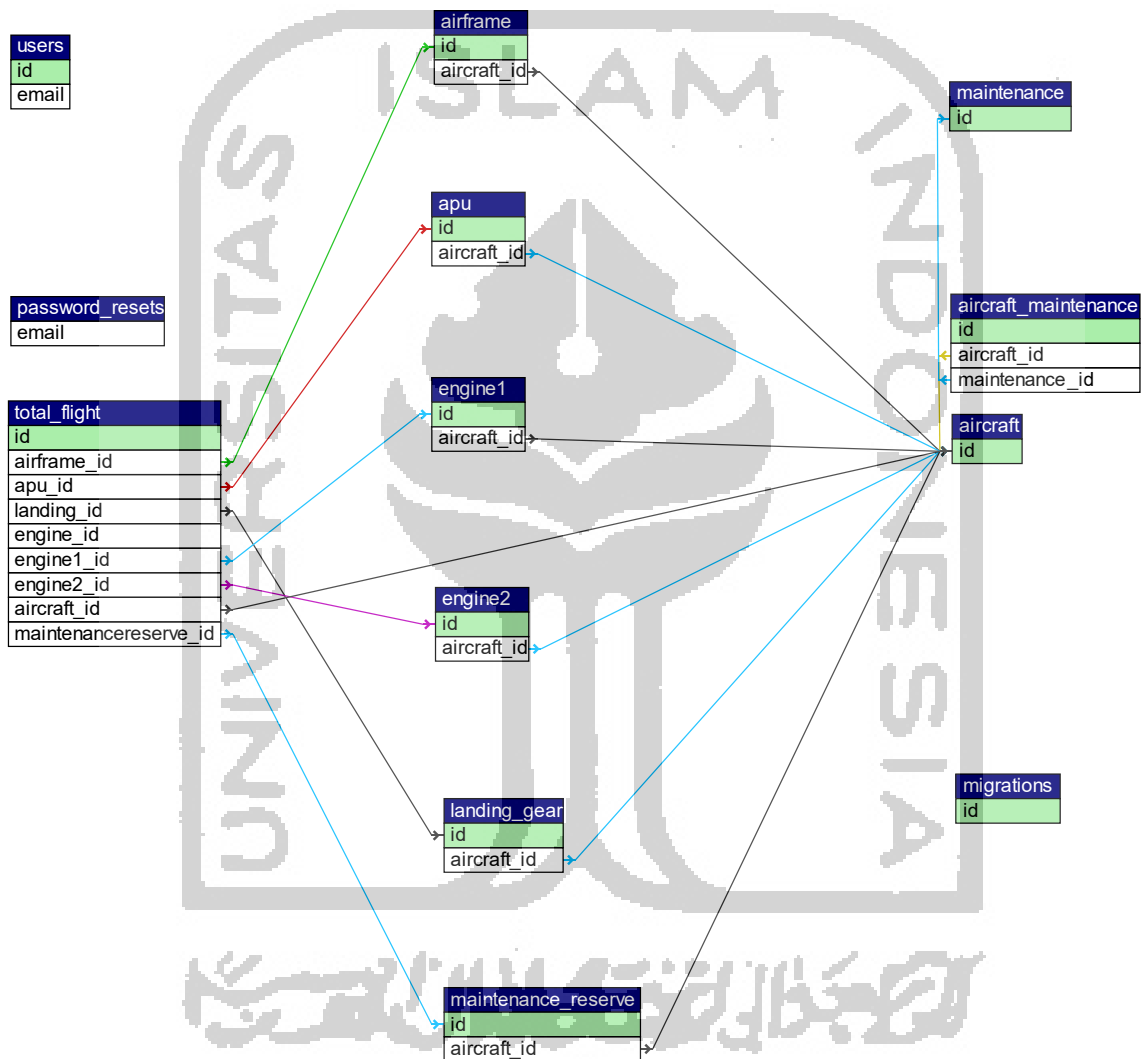


Gambar 3.7 Contoh gambar Activity Diagram Melihat Aircraft Utilization

3.4 Relational Schemas

Gambar 3.8 menunjukkan relasi yang digunakan pada sistem yang akan dibangun. Dalam desain basisdata yang digunakan menghasilkan 11 tabel yang beberapa memiliki relasi antara satu dengan yang lain. 11 tabel tersebut antara lain, *table users*, *table aircraft*, *table aircraft_maintenance*, *table maintenance_reserve*, *table airframe*, *table APU*, *table landing_gear*, *table maintenance*, *table engine1*, *table engine2* dan *table total fight*. Tabel *user* berisikan informasi pengguna *system*, terdapat tiga pengguna *system* yaitu *manager*, *admin aircraft*, dan *user*. Tabel *aircraft* digunakan untuk menyimpan semua jenis pesawat/*aircraft* ke dalam *system*. Tabel *aircraft_maintenance* berisikan relasi antara *aircraft* dengan jenis *maintenance* yang terdapat di dalam pesawat/*aircraft* tersebut. Tabel *maintenance* adalah tabel yang berisikan jenis *maintenance*. Tabel *engine1* dan *engine2* digunakan untuk memasukan

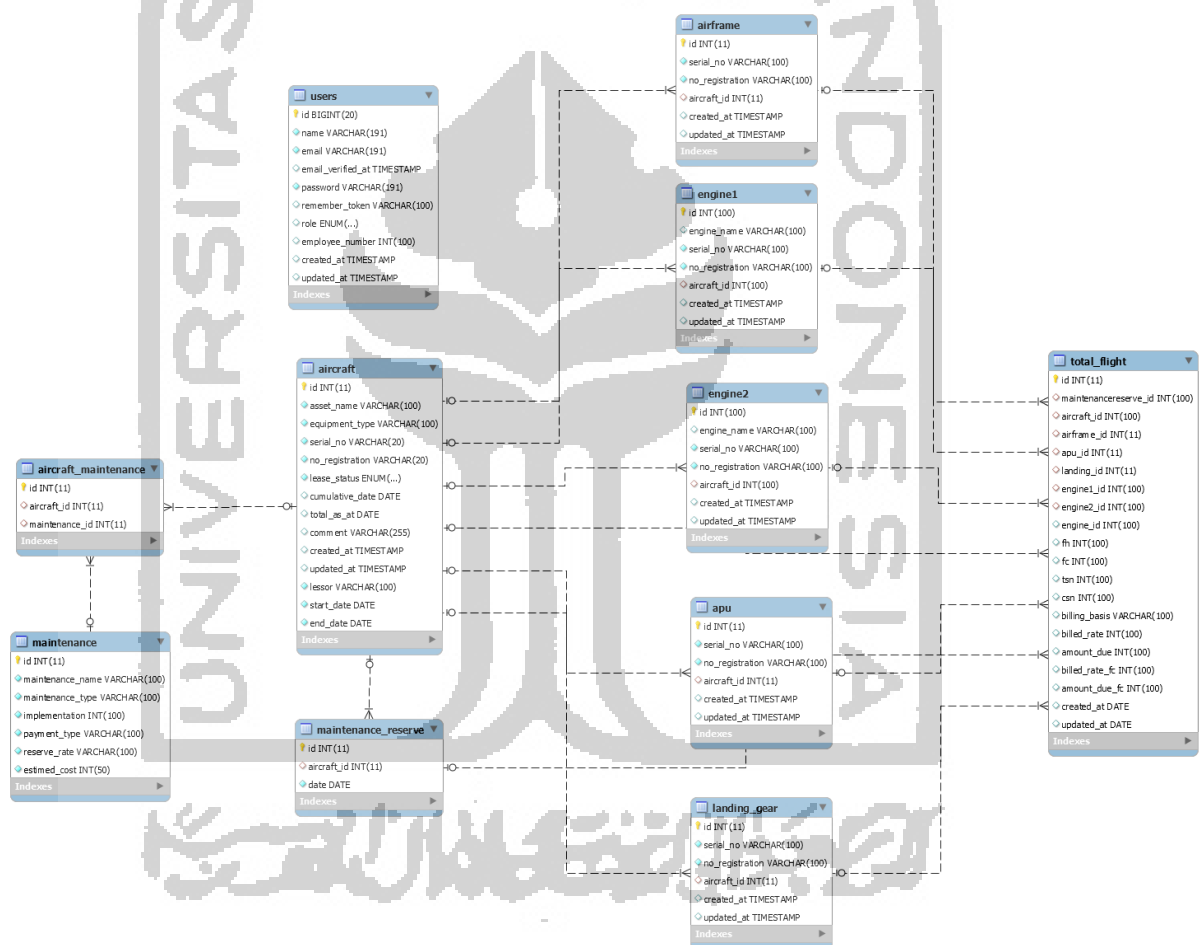
semua jenis *engine* yang terdapat disebuah maskapai. *Table airfarme*, *APU*, dan *landing_gear* adalah komponen dari pesawat/*aircraft*. *Table total_flight* adalah *table* yang bersikan utilitas dari komponen dari sebuah pesawat/*aircraft* yang terdiri dari *engine*, *airframe*, *APU*, dan *landing gear*. *Table maintenance_reserve* berfungsi untuk menyimpan data *maintenance* pesawat per bulan.



Gambar 3.8 Contoh Gambar *Relational Schemas*

3.5 Database Relationship Diagram

Berdasarkan Gambar 3.9 Terdapat *table aircraft* memiliki relasi *one-to-one* kepada *table air_frame*, *table APU*, dan *table landing_gear*, *table engine1*, dan *engine2*. Tabel *aircraft* memiliki relasi *many-to-many* terhadap *table maintenance* yang membuat *table* baru yaitu *aircraft_maintenance*. *Table aircraft_maintenance* memungkinkan banyak *aircraft* memiliki banyak *maintenance*. *Table engine1*, *table engine2*, *table airframe*, *table APU*, dan *table landing_gear* mempunyai relasi *one-to-one* terhadap *table total_flight* yang berarti komponen tersebut hanya mempunyai satu *fligh hour*.



Gambar 3.9 Contoh Gambar Database Relationship Diagram

3.6 Struktur Table Database

3.6.1 Table User

Tabel 3.3 menunjukkan penggunaan Tabel *User* pada database untuk menyimpan data credential seperti data login. Tabel ini memiliki field *id*, *first_name*, *last_name*, *email*, *password*, *join_date*, *phone_number*, *fax_number*, dan *address*. Saat login atau registrasi data yang diproses akan disimpan pada *table user*.

Tabel 3.3 Tabel *User*

Kolom	tipe data
<i>id_user</i> (PK)*	int
<i>first name</i> *	varchar
<i>last name</i>	varchar
<i>email</i> *	varchar
<i>password</i> *	varchar
<i>join_date</i> *	date
<i>phone_number</i> *	int
<i>fax_number</i>	int
<i>address</i> *	varchar

3.6.2 Table Aircraft

Table aircraft menunjukkan penggunaan *table aircraft* pada database untuk menyimpan *aircraft*/pesawat. *Table aircraft* mempunyai filed *id_aircraft*, *asset_name*, *asset_type*, *equipment_type*, *serial_no*, *lease_status*, *cumulative_date*, *total_as_at*, *comment*, *created_at*, dan *edit_at*. Terdapat 3 field foreign key pada *table aircraft* yaitu *id_airframe*, *id_apu*, dan *id_landing_gear*. Sebuah pesawat memiliki satu *airframe*, satu APU, dan satu *landing gear*.

Tabel 3.4 Tabel *Aircraft*

kolom	tipe data
<i>id_aircraft</i> (PK)*	int
<i>asset_name</i> *	varchar
<i>asset_type</i> *	enum('engine', 'aircraft')
<i>equipment_type</i> *	varchar
<i>serial_no</i> *	varchar
<i>no_registration</i> *	varchar
<i>lease_status</i> *	enum('active', 'non- active')
<i>cumulative_date</i>	date

total_as_at	date
Comment	varchar
Id_ <i>airframe</i>	int
Id_ apu (FK)*	Int
Id_ landing_gear (FK)*	int
created_at (FK)*	date
edit_at	date

3.6.3 Table *Airframe*

Table *Airframe* menunjukkan penggunaan *table airframe* pada *database* untuk menyimpan *airfarme*. Di dalam *table airframe* terdapat field *id*, *created_at*, dan *update_at*.

Tabel 3.5 Tabel *Airframe*

Kolom	tipe data
id (PK)*	int
serial_no	varchar
no_registration	varchar
created_at	date
update_at	date

3.6.4 Table APU

Table APU menunjukkan penggunaan *table apu* pada *database* untuk menyimpan APU. Di dalam *table apu* terdapat field *id*, *created_at*, dan *update_at*.

Tabel 3.6 Tabel APU

kolom	tipe data
id_ (PK)*	int
serial_no	varchar
no_registration	varchar
created_at	date
update_at	date

3.6.5 Table *Landing gear*

Table *Landing gear* menunjukkan penggunaan *table landing-gear* pada *database* untuk menyimpan *landing_gear*. Di dalam *table airframe* terdapat field *id*, *created_at*, dan *update_at*.

Tabel 3.7 Tabel *Landing gear*

kolom	tipe data
id_landing_gear(PK)*	int
serial_no	varchar
no_registration	varchar
created_at	date
update_at	date

3.6.6 *Table Engine1*

Table Engine menunjukkan penggunaan *table engine1* pada *database* untuk menyimpan *airfarme*. Di dalam *table engine* terdapat field *id*, *created_at*, dan *update_at*.

Tabel 3.8 Tabel *Engine1*

kolom	tipe_data
id_engine(PK)*	int
serial_no	varchar
no_registration	varchar
created_at	date
update_at	date

3.6.7 *Table Engine2*

Table Engine menunjukkan penggunaan *table engine1* pada *database* untuk menyimpan *airfarme*. Di dalam *table engine* terdapat field *id*, *created_at*, dan *update_at*.

Tabel 3.9 Tabel *Engine2*

kolom	tipe_data
id_engine(PK)*	int
serial_no	varchar
no_registration	varchar
created_at	date
update_at	date

3.6.8 *Table Maintenance*

Table Maintenance menunjukkan penggunaan *table maintenance* pada *database* untuk menyimpan *maintenance*. Di dalam *table engine* terdapat field *id*, *maintenance_name*, *maintenance_type*, *implementation*, *payment_type*, *reserve_rate*, dan *estimed_cost*. Setelah

admin *aircraft* memasukan jenis pesawat makan petugas wajib memasukan *maintenance* apa saja yang berkaitan dengan pesawat tersebut.

Tabel 3.10 Tabel *Maintenance*

Kolom	tipe data
id (PK)*	Int
<i>maintenance_name</i>	Varchar
<i>maintenance_type</i>	Varchar
Implementation	Varchar
<i>payment_type</i>	Varchar
<i>reserve_rate</i>	Varchar
<i>estimated_costs</i>	Int

3.6.9 Table Aircraft Maintenance

Table Aircraft Maintenance menunjukkan penggunaan *table aircraft_maintenance* pada *database* untuk menyimpan *maintenance* yang digunakan oleh *aircraft*. Di dalam *table aircraft_maintenance* terdapat field *id*, *id_aircraft*, dan *id_maintenance*. *Aircraft* dapat memiliki beberapa *maintenance*. *Id_aircraft* dan *id_maintenance* adalah *foreign key*.

Tabel 3.11 *Table Aircraft Maintenance*

kolom	tipe data
id(PK)*	int
<i>id_aircraft</i> (FK)*	int
<i>id_maintenance</i> (FK)*	int

3.6.10 Table Maintenance Reserve

Table Maintenance Reserve menunjukkan penggunaan *table maintenance_reserve* pada *database* untuk menyimpan *maintenance reserve* yang digunakan oleh *aircraft*. Di dalam *table maintenance_reserve* terdapat field *id*, *id_aircraft*, *id_maintenance*, *total_amount_due* dan *maintenance_date*. Dalam setiap bulan petugas wajib mengisi *maintenance reserve*. *Id_aircraft* dan *id_maintenance* adalah *foreign key*.

Tabel 3.12 *Table Maintenance Reserve*

kolom	tipe data
id(PK)*	int
<i>id_aircraft</i> (FK)*	int
<i>id_maintenance</i> (FK)*	int
<i>total_amount_due</i>	int

<i>maintenance_date</i>	date
-------------------------	------

3.6.11 Table Total Flight

Table Total Flight menunjukkan penggunaan *table* total_flight pada *database* untuk menyimpan utilitas yang digunakan oleh *aircraft*. Di dalam *table* total_flight terdapat field id, id_airframe, id_apu, id_landing_gear, id_engine, flight_hour, flight_cycle, time_since_new, cycle_since_new, billing_basis, billing_rated, created_at, dan update_at. Id_airframe, id_engine, id_apu, dan id_landing_gear adalah *foreign key*. Seluruh utilitas dari komponen pesawat dimasukkan ke dalam database total_flight.

Tabel 3.13 Table Total Flight

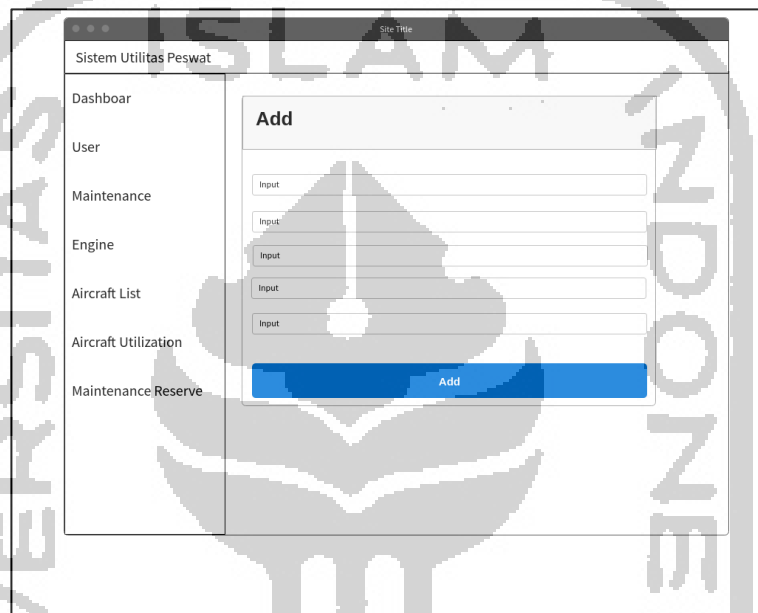
kolom	tipe data
id(PK)*	int
id_airframe(FK)*	int
id_apu(FK)*	int
id_landing_gear(FK)*	int
id_(FK)engine*	int
flight_hour	int
Flight_cycle	int
Time_since_new	int
Cycle_since_new	int
Billing_bassis	varchar
Billed_rate	int
Amount_due	int
Created_at	date
Update_at	date

3.7 Perancangan Mockup Antarmuka (interface)

Rancangan antarmuka dibuat berdasarkan *use case* yang sudah dibuat sebelumnya. *Use case* tersebut terdiri dari mengelola pengguna, mengelola *maintenance*, mengelola *engine*, mengelola *aircraft*, mengelola *maintenance reserve*, melihat *aircraft Utilization*. Rancangan ini dibutuhkan untuk pedoman dan batasan agar implementasi dilakukan terarah. Berikut merupakan rancangan antarmuka Pengembangan Sistem Manajemen Aset Pencatatan Utilitas Pesawat.

3.7.1 Halaman *Form Add*

Gambar 3.10 merupakan rancangan halaman *form add*. Rancangan halaman *form add* berlaku disemua fitur yang terdapat *form add*. *Form add* berfungsi untuk menambahkan inputan ke dalam *system*. Halaman *form add* muncul di *use case* mengelola pengguna, mengelola *maintenance*, mengelola *engine*, mengelola *aircraft*, mengelola *maintenance reserve*.

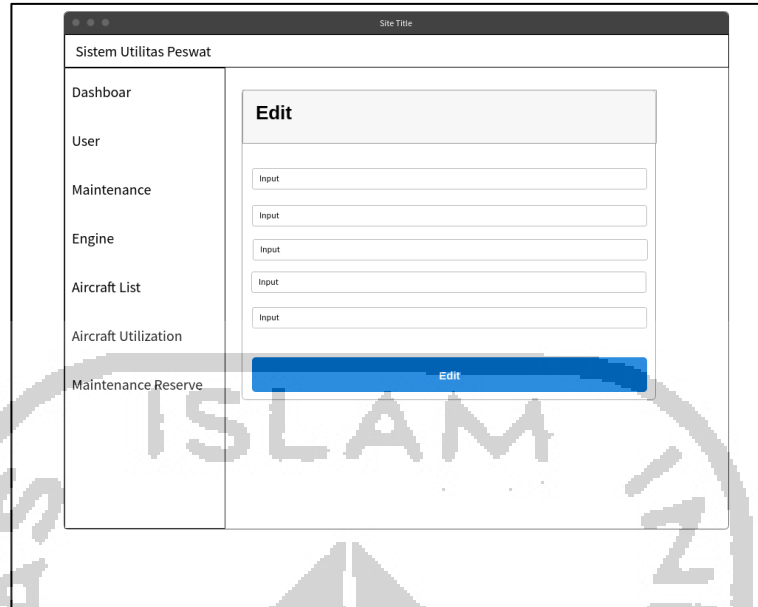


The image shows a web application interface for 'Sistem Utilitas Peswat'. On the left is a sidebar menu with the following items: Dashboard, User, Maintenance, Engine, Aircraft List, Aircraft Utilization, and Maintenance-Reserve. A modal window titled 'Add' is open in the center, containing five 'Input' fields stacked vertically and a blue 'Add' button at the bottom.

Gambar 3.10 Halaman *Login*

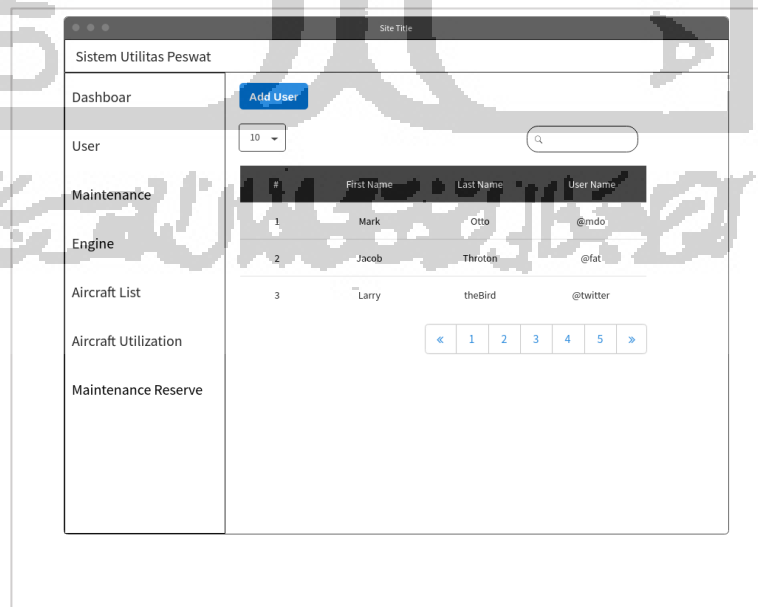
3.7.2 Halaman *Form Edit*

Gambar 3.11 merupakan rancangan halaman *form edit*. Rancangan halaman *form edit* berlaku disemua fitur yang terdapat *form edit*. *Form edit* berfungsi untuk mengubah data inputan ke dalam *system*. Halaman *form edit* muncul di *use case* mengelola pengguna, mengelola *maintenance*, mengelola *engine*, mengelola *aircraft*, mengelola *maintenance reserve*.

Gambar 3.11 Halaman *Login*

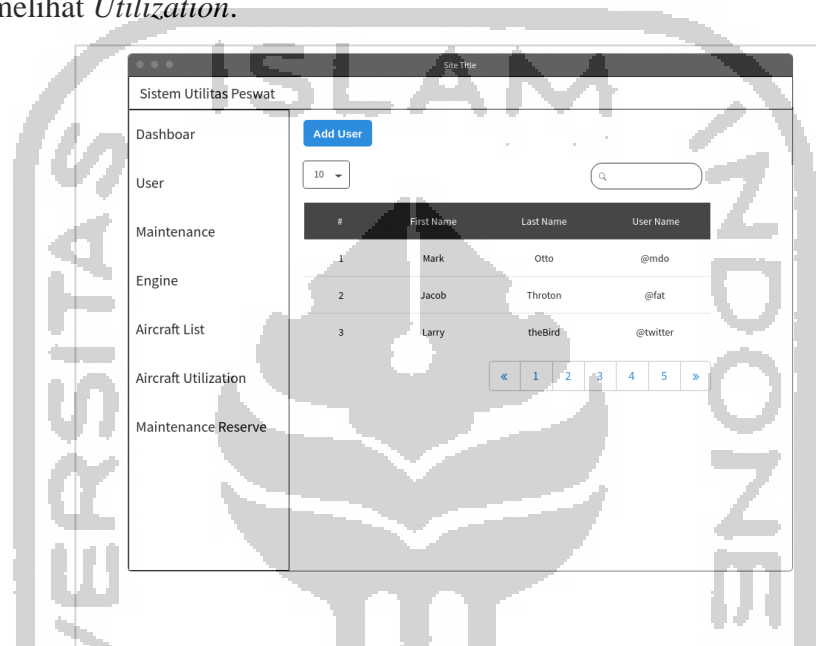
3.7.3 Halaman Detail

Gambar 3.12 adalah rancangan antarmuka dari halaman *Detail*. Halaman ini berisikan detail dari suatu informasi. Halaman detail terdapat di dalam 3 fitur antara lain halaman *aircraft*, halaman *maintenance reserve* dan halaman *Utilization*. Halaman *detail* muncul di *use case* mengelola pengguna, mengelola *engine*, mengelola *aircraft*, mengelola *maintenance reserve*, dan melihat *utilization*.

Gambar 3.12 Halaman Manajemen *user*

3.7.4 Halaman List

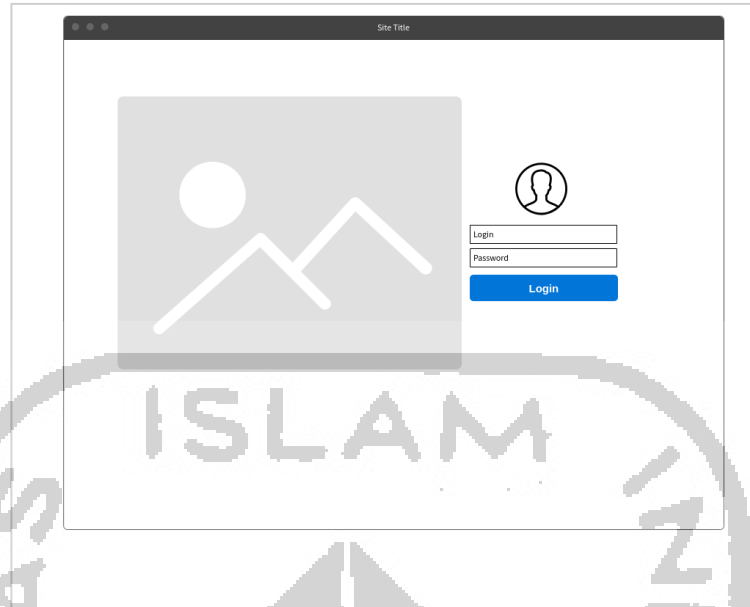
Gambar 3.13 merupakan rancangan halaman *list*. Rancangan halaman *list* berlaku disemua fitur yang terdapat halaman *list*. *Halaman list* berfungsi untuk melihat semua list yang terdapat didalam *system*. Halaman *form add* muncul di *use case* mengelola pengguna, mengelola *maintenance*, mengelola *engine*, mengelola *aircraft*, mengelola *maintenance reserve*, dan melihat *Utilization*.



Gambar 3.13 Halaman Manajemen *user*

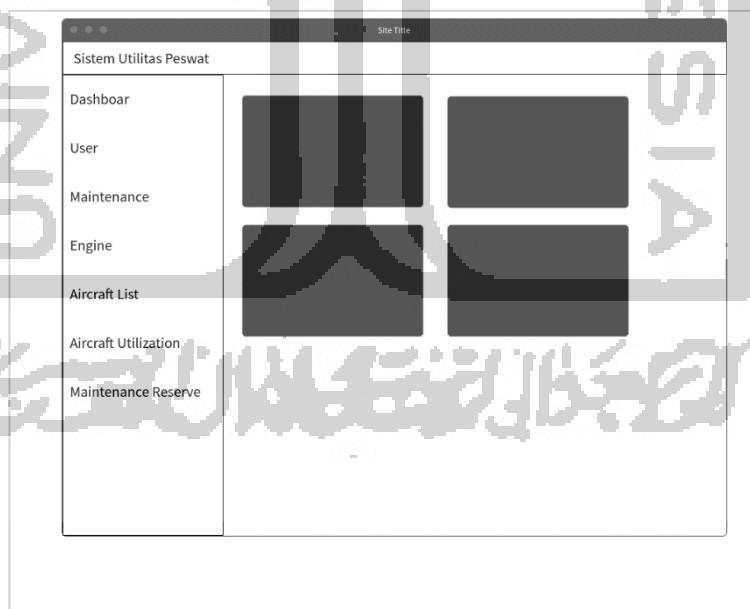
3.7.5 Halaman Login

Gambar 3.14 merupakan rancangan halaman login pengguna yang telah terdaftar. Terdapat 3 pengguna di dalam aplikasi, yaitu *manager*, *admin aircraft*, dan *user*. Ketika telah masuk maka pengguna dapat menggunakan aplikasi sesuai dengan rolenya.



Gambar 3.14 Halaman *Login*

Setelah login, pengguna akan masuk halaman *dashboard*. Gambar 3.15 adalah rancangan antarmuka dari halaman *dashboard* pengguna. Ketika pengguna telah benar memasukan data login, maka halaman ini yang akan ditampilkan. Halaman *dashboard* berisikan rangkuman total *user*, *total maintenance*, *total aircraft*, dan *total maintenance reserve*.



Gambar 3.15 Halaman *Dashboard*