

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Panduan Implementasi

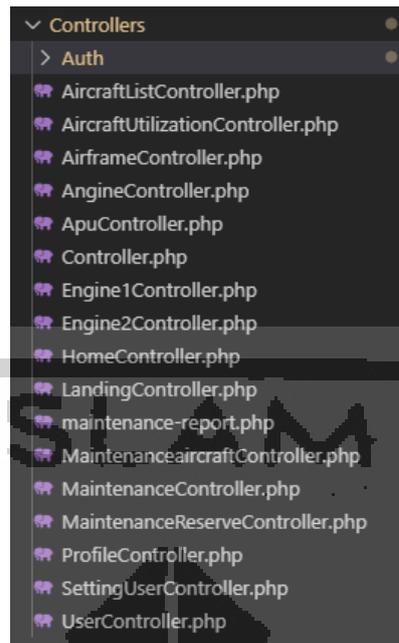
Tahap implementasi merupakan tahap mempresentasikan hasil perancangan yang telah dibuat pada tahap sebelumnya sehingga sistem siap untuk dioperasikan dan dilakukan pengujian dari sisi pengguna. Pengguna sistem ini merupakan pegawai manajemen *aircraft system*.

Pola MVC memecahkan sebuah aplikasi menjadi tiga modul asosiasi: model, view, dan controller. Model modul adalah logika bisnis dari aplikasi dan inti dari sebuah aplikasi. View adalah *user interface* dari controller. Yang merupakan muka umum untuk respon event pengguna. Komponen controller mengimplementasi flow yang mengontrol antara view dan model.

Model-View-Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) untuk meng-enkapsulasi data bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (controller) dan tampilan (view) untuk direpresentasikan pada sebuah *user interface*. Definisi teknis dari arsitektur MVC dibagi menjadi tiga lapisan.

A. Controller

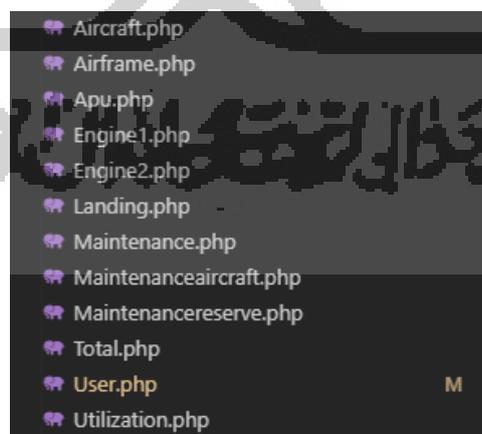
Controller menerima input dari pengguna dan mengintruksikan model dan view untuk melakukan aksi berdasarkan masukan tersebut. Sehingga, controller bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi. Didalam Pengembangan Sistem Manajemen Aset Pencatatan Utilitas Pesawat ini terdapat 16 controller. 16 controller tersebut antara lain, *AircraftListController*, *AircraftUtilizaionController*, *AirframeController*, *Engine1Controller*, *Engine2Controller*, *ApuController*, *LandingController*, *HomeController*, *MaintenanceController*, *Maintenance-reportController*, *MaintenanceReserveController*, *MaintenanceaircraftController*, *ProfileController*, *SettingController*, *RegisterController*, *UserController* yang terletak pada folder *Controllers*.



Gambar 4.1 Struktur file Controller

B. Model

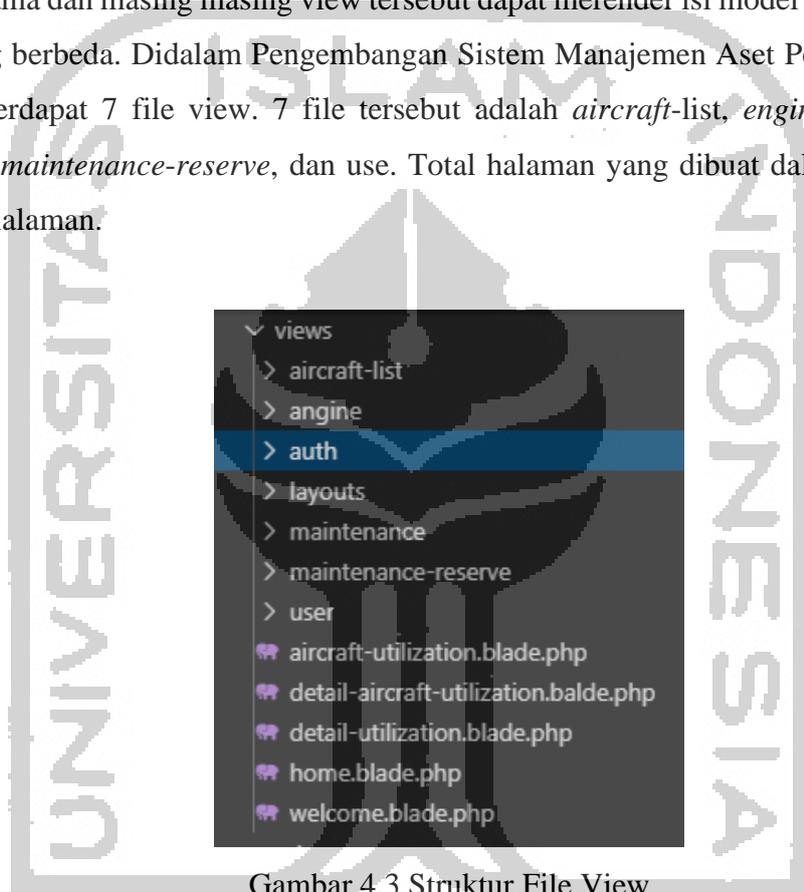
Model digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Hanya model yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data. Sebuah model meringkas lebih dari sekedar data dan fungsi yang beroperasi di dalamnya. Didalam Pengembangan Sistem Manajemen Aset Pencatatan Utilitas Pesawat ini terdapat 12 model. 12 model tersebut antara lain, *aircraft.php*, *airframe.php*, *apu.php*, *engine1.php*, *engine2.php*, *landing.php*, *maintenance.php*, *maintenancereserve.php*, *user.php*, *total.php*, *user.php* dan *utilization.php* yang terletak pada folder app.



Gambar 4.2 Struktur File Model

C. View

View bertanggung jawab untuk pemetaan grafis ke sebuah perangkat. View biasanya memiliki hubungan 1-1 dengan sebuah permukaan layar dan tahu bagaimana untuk membuatnya. View melekat pada model dan merender isinya ke permukaan layar. Selain itu, ketika model berubah, view secara otomatis menggambar ulang bagian layar yang terkena perubahan untuk menunjukkan perubahan tersebut. Terdapat kemungkinan beberapa view pada model yang sama dan masing masing view tersebut dapat merender isi model untuk permukaan tampilan yang berbeda. Didalam Pengembangan Sistem Manajemen Aset Pencatatan Utilitas Pesawat ini terdapat 7 file view. 7 file tersebut adalah *aircraft-list*, *engine*, *auth*, *layouts*, *maintenance*, *maintenance-reserve*, dan *use*. Total halaman yang dibuat dalam view adalah sebanyak 40 halaman.



Gambar 4.3 Struktur File View

4.2 Implementasi Sistem

4.2.1 Mengelola Pengguna

Halaman mengelola pengguna adalah halaman dimana manager untuk mengelola pengguna *system*. Hak akses penuh hanya diberikan kepada role manager. Manager dapat menambah, mengedit, melihat detail dan menghapus pengguna lain.

No.	Name	Email	Role	Created At	Updated At	Action
1	mochammad falshai	mochammadfalshai2@gmail.com	manager	2019-08-01 14:48:06	2019-08-01 14:48:06	Edit Detail Hapus
2	jaksen	jaksentiago2@gmail.com	manager	2019-08-07 14:21:42	2019-08-07 14:21:42	Edit Detail Hapus
3	codenesia	codenesia@gmail.com	manager	2019-10-03 18:00:49	2019-10-03 18:00:49	Edit Detail Hapus
4	falshai	falshai@gmail.com	manager	2019-10-03 18:02:44	2019-10-03 18:02:44	Edit Detail Hapus
5	hahaha	hahaha@gmail.com	manager	2019-10-19 05:30:40	2019-10-19 05:30:40	Edit Detail Hapus

Gambar 4.4 Halaman List Pengguna

Pada Gambar 4.5 Manager dapat menambahkan pengguna ke dalam sistem. Manager harus menginputkan name, email, role, dan password. Setelah mengisi semua inputan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list pengguna.

Gambar 4.5 Halaman Add Pengguna

Pada Gambar 4.6 Manager dapat mengedit pengguna yang terdapat dalam sistem. Manager dapat merubah name, email, role, dan password. Setelah merubah semua inputan sesuai kebutuhan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list pengguna.

Gambar 4.6 Halaman Edit Pengguna

Pada Gambar 4.7 Manager dapat melihat detail dari setiap pengguna. Dalam halaman detail pengguna manager dapat melihat informasi nama, email, role, employee number, created at, dan updated at.

Gambar 4.7 Halaman Detail Pengguna

4.2.2 Mengelola *Maintenance*

Halaman mengelola *maintenance* adalah halaman dimana manager untuk mengelola *maintenance* di dalam *system*. Hak akses penuh hanya diberikan kepada role manager. Manager dapat menambah, mengedit, melihat dan menghapus *maintenance*. Halaman list *maintenance* manager dapat melihat semua list *maintenance* yang sudah terdapat di dalam *system*.

No	Maintenance Name	Maintenance Type	Implementation	Paymen Type	Reserve Rate	Estimated Cost	Action
7	8 YR STRUCTURE	8 YR STRUCTURE	SETIAP 8 TAHUN	CALENDER	FIX PER BULAN	500000	Edit Hapus
8	10 YR STRUCTURE	10 YR STRUCTURE	SETIAP 10 TAHUN	CALENDER	FIX PER BULAN	800000	Edit Hapus
9	8 YR STRUCTURE	8 YR STRUCTURE	SETIAP 8 TAHUN	CALENDER	FIX PER TAHUN	30000	Edit Hapus

Gambar 4.8 Halaman Mengelola *Maintenance*

Pada Gambar 4.9 Manager dapat menambahkan *maintenance* ke dalam sistem. Manager harus menginputkan *maintenance name*, *maintenance type*, *implementation*, *payment type*, *reserve rate*, dan *estimated cost*. Setelah mengisikan semua inputan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *maintenance*.

Gambar 4.9 Halaman Add *Maintenance*

Pada Gambar 4.10 Manager dapat mengedit *maintenance* yang terdapat dalam sistem. Manager dapat merubah *maintenance name*, *maintenance type*, *implementation*, *payment type*, *reserve rate*, dan *estimated cost*. Setelah merubah semua inputan sesuai kebutuhan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *maintenance*.

Gambar 4.10 Halaman Edit *Maintenance*

4.2.3 Mengelola Engine

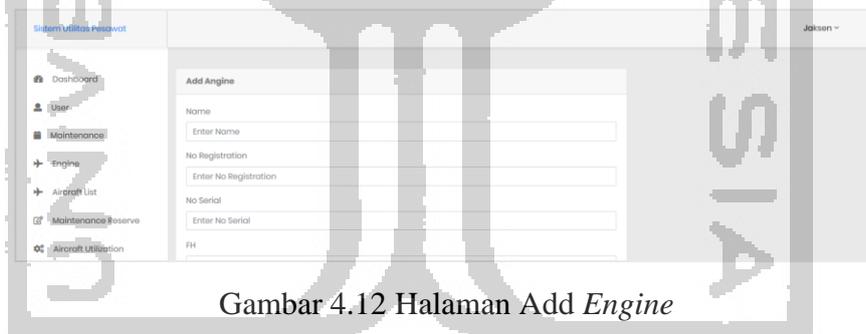
Halaman mengelola *Engine* adalah halaman dimana manager untuk mengelola *engine* di dalam *system*. Hak akses penuh hanya diberikan kepada *role* manager. Manager dapat menambah, mengedit, melihat dan menghapus *engine*. Pada halaman list *engine* manager dapat melihat semua list *engine* yang sudah terdapat di dalam *system*.



No	Name	No Registration	Serial	FH	FT	FC	SN	Billing Basis	Billing Rate	Amount Due	Date
1	engine 1	EP-04	EP-02	2987	379	768	1745	EFM	9835	940	1 September 2020
2	engine 2	EP-14	EP-78	5787	389	1880	2345	EFM	9835	127	1 Oktober 2020

Gambar 4.11 Halaman Mengelola *Engine*

Pada Gambar 4.12 Manager dapat menambahkan *engine* ke dalam sistem. Manager harus mengisi lis *engine*. Setelah mengisi semua inputan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *engine*.



Add Engine

Name
Enter Name

No Registration
Enter No Registration

No Serial
Enter No Serial

FH

Gambar 4.12 Halaman Add *Engine*

Pada Gambar 4.13 Manager dapat mengedit *engine* yang terdapat dalam sistem. Manager dapat merubah *engine name*, *serial no* dan *no registration*. Setelah merubah semua inputan sesuai kebutuhan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *engine*.

Gambar 4.13 Halaman Edit *Engine*

4.2.4 Mengelola *Aircraft*

Halaman mengelola *Aircraft* adalah halaman dimana manager untuk mengelola *aircraft* di dalam *system*. Hak akses penuh hanya diberikan kepada *role* manager. Manager dapat menambah, mengedit, melihat dan menghapus *Aircraft*. Pada halaman list *aircraft* manager dapat melihat semua list *Aircraft* yang sudah terdapat di dalam *system*.

No	Asset Name	Equipment Type	Serial No	Registration No	Lease Status	Lessor	Action
13	1377	Aircraft	PAK	15523	active	BORG	Edit Detail Hapus
14	AIRBUS 73	Aircraft	2330	82	active	AMERLAND	Edit Detail Hapus
15	hahaha	hahahaha	hahahaha	hahahaha	active	hahahaha	Edit Detail Hapus

Gambar 4.14 Halaman Mengelola *Aircraft*

Pada Gambar 4.15 Manager dapat menambahkan *aircraft* ke dalam sistem. Manager harus mengisi *page aircraft*, *page airframe*, *page engine1*, *page engine2*, *page APU*, dan *page landing gear*. Setelah mengisi semua inputan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *aircraft*.

The screenshot shows a web application interface for adding a new aircraft. The page title is 'Sistem Utilitas Pesawat'. On the left, there is a navigation menu with options: Dashboard, User, Maintenance, Aircraft List, Maintenance Reserve, and Aircraft Utilization. The main content area is titled 'Add Aircraft' and contains the following form fields:

- Asset Name:
- Equipment Type:
- Serial No:
- No Registration:
- Lease Status:
- Lessor:
- Start Date:
- End Date:

A blue 'Add' button is located at the bottom center of the form.

Gambar 4.15 Halaman Add Aircraft

Pada Gambar 4.16 Manager dapat mengedit *aircraft* ke dalam sistem. Manager dapat merubah isi page *aircraft*, *page airframe*, *page engine1*, *page engine2*, *page APU*, dan *page landing gear*. Setelah merubah semua inputan maka manager menekan tombol save untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *aircraft*.

The screenshot shows the 'Edit Aircraft' page in the 'Sistem Utilitas Pesawat' application. The page title is 'Sistem Utilitas Pesawat'. On the left, there is a navigation menu with options: Dashboard, User, Maintenance, Engine, Aircraft List, Aircraft Utilization, and Maintenance Reserve. The main content area is titled 'Aircraft' and contains the following tabs: Aircraft, Engine 1, Engine 2, Airframe, APU, and Landing Gear. The 'Add Aircraft' form is visible in the background, showing the same fields as in Gambar 4.15.

Gambar 4.16 Halaman Edit Aircraft

Pada Gambar 4.17 Manager dapat melihat detail dari setiap *Aircraft*. Dalam halaman detail *Aircraft* manager dapat melihat informasi lengkap *aircraft* seperti *detail airframe*, *detail engine1*, *detail engine2*, *detail APU*, *detail landing gear*.

Aircraft	Serial No	Registration No
Boeing 737	FW	18523210

Airframe	
No Serial	FX
No Registration	18523210
FH	358
FC	23
TSN	1056
CSN	400

Engine 1	
No Serial	FX

Gambar 4.17 Halaman Detail Aircraft

4.2.5 Mengelola *Maintenance Reserve*

Halaman mengelola *Maintenance Reserve* adalah halaman dimana manager dan admin *aircraft* untuk mengelola *maintenance* di dalam *system*. Hak akses penuh diberikan kepada role manager dan admin *aircraft*. Manager dan admin *aircraft* dapat menambah, mengedit, melihat dan menghapus *maintenance reserve*. Pada halaman list *maintenance reserve* manager admin *aircraft* dapat melihat semua list *maintenance reserve* yang sudah terdapat di dalam *system*. Role *user* hanya dapat melihat halaman list *maintenance reserve* dan tidak mendapatkan action apapun kedalam *system*.

No	Date	Aircraft Name	Serial No	Registration No	Lease Status	Amount Due	Action
1	1 September 2018	Boeing 737	XV	18523210	Active	500\$	Edit Detail Hapus
1	1 Oktober 2018	Airbus 737	VW	1852327	Active	800\$	Edit Detail Hapus

Gambar 4.18 Halaman Mengelola *Maintenance Reserve*

Pada Gambar 4.19 Manager dan admin *aircraft* dapat menambah *maintenance reserve* ke dalam sistem. Manager dapat menambah dengan mengisi *page aircraft*, *page airframe*, *page engine1*, *page engine2*, *page APU*, dan *page landing gear*. Setelah mengisi seluruh inputan maka manager dan admin *aircraft* menekan tombol *save* untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *maintenance reserve*.

Gambar 4.19 Halaman Add *Maintenance Reserve*

Pada Gambar 4.20 Manager dan admin *aircraft* dapat mengedit *maintenance reserve* dalam sistem. Manager dan admin dapat merubah isi tab *aircraft*, tab *airframe*, tab *engine1*, tab *engine2*, tab *APU*, dan tab *landing gear*. Setelah merubah semua inputan maka manager admin *aircraft* menekan tombol *save* untuk memasukan ke dalam sistem. Setelah menginputkan pengguna manager dapat melihat hasilnya di halaman list *maintenance reserve*.

Gambar 4.20 Halaman Edit *Maintenance Reserve*

4.2.6 Melihat *Aircraft Utilization*

Pada Gambar 4.21 Manager, admin *aircraft* dan *user* bisa melihat utilitas pesawat. Pengguna dapat melihat utilitas pesawat dengan menekan tombol melihat utilisasi sesuai dengan pesawat. Halaman *aircraft Utilization* akan menampilkan detail utilitas pesawat seperti detail *aircraft*, detail *airframe*, detail *engine1*, detail *engine2*, detail *APU*, detail *landing gear*, dan detail *maintenance* yang terdapat pada suatu *aircraft*.

Aircraft	Serial No	Registration No
UIR AIR	UI03	UI05

Engine 1	
Engine Name	ENGINE 1
No Serial	EE02
No Registration	EE44
PH	023
PC	21
TSN	235
CSN	103

Gambar 4.21 Halaman Melihat *Aircraft Utilization*

4.3 Pengujian

Pada bagian ini menjelaskan pengujian yang dilakukan yaitu dengan pengujian *blackbox* pada *fungsi*litas, pengujian *usabilitas* berdasarkan *use case* yang telah ditentukan, dan pengujian *non fungsi*litas berdasarkan komponen *Critical Safe Software*. Pengujian ini dilakukan untuk mengetahui apakah sistem aplikasi telah berjalan sesuai rencana.

4.3.1 Fungsionalitas (*Black Box*)

Pada tahap ini, pengujian dilakukan untuk melihat apakah fitur-fitur yang direncanakan telah dapat diwujudkan. Metode *Blackbox Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan (Mustaqbal, Firdaus, & Rahmadi, 2015). Hasil pengujian yang dilakukan dapat terlihat pada Tabel 4.1.

Tabel 4.1 Tabel Hasil Pengujian Fungsionalitas (*Black Box*)

No	Pengguna	Role	Hak Akses	Hasil	Kesimpulan
1.	Mengelola Pengguna	Manager	Lihat	Menampilkan list pengguna	Berhasil
			Tambah	Menampilkan form add pengguna	Berhasil
			Edit	Menampilkan form edit pengguna	Berhasil
			Hapus	Menghapus pengguna	Berhasil

		Admin <i>Aircraft</i>	Lihat	Menampilkan list pengguna	Berhasil
			Tambah	Menampilkan form add pengguna	Berhasil
			Edit	Menampilkan form edit pengguna	Berhasil
			Hapus	Menghapus pengguna	Berhasil
			<i>User</i>	Lihat	Menampilkan list pengguna
2.	Mengelola <i>Maintenance</i>	Manager	Lihat	Menampilkan list <i>maintenance</i>	Berhasil
			Tambah	Menampilkan form add <i>maintenance</i>	Berhasil
			Edit	Menampilkan form edit <i>maintenance</i>	Berhasil
			Hapus	Menghapus <i>maintenance</i>	Berhasil
		Admin <i>Aircraft</i>	Lihat	Menampilkan list <i>maintenance</i>	Berhasil
			Tambah	Menampilkan form add <i>maintenance</i>	Berhasil
			Edit	Menampilkan form edit <i>maintenance</i>	Berhasil
			Hapus	Menghapus <i>maintenance</i>	Berhasil
		<i>User</i>	Lihat	Menampilkan list <i>maintenance</i>	Berhasil
		3.	Mengelola Egine	Manager	Lihat
Tambah	Menampilkan form add <i>engine</i>				Berhasil
Edit	Menampilkan form edit <i>engine</i>				Berhasil
Hapus	Menghapus <i>engine</i>				Berhasil

		Admin <i>Aircraft</i>	Lihat	Menampilkan list <i>engine</i>	Berhasil
			Tambah	Menampilkan form add <i>engine</i>	Berhasil
			Edit	Menampilkan form edit <i>engine</i>	Berhasil
			Hapus	Menghapus <i>engine</i>	Berhasil
		<i>User</i>	Lihat	Menampilkan list <i>engine</i>	Berhasil
4.	Mengelola <i>Aircraft</i>	Manager	Lihat	Menampilkan list <i>aircraft</i>	Berhasil
			Tambah	Menampilkan form add <i>aircraft</i>	Berhasil
			Edit	Menampilkan form edit <i>aircraft</i>	Berhasil
			Hapus	Menghapus <i>aircraft</i>	Berhasil
		Admin <i>Aircraft</i>	Lihat	Menampilkan list <i>aircraft</i>	Berhasil
			Tambah	Menampilkan form add <i>aircraft</i>	Berhasil
			Edit	Menampilkan form edit <i>aircraft</i>	Berhasil
			Hapus	Menghapus <i>aircraft</i>	Berhasil
		<i>User</i>	Lihat	Menampilkan list <i>aircraft</i>	Berhasil
		5.	Mengelola <i>Maintenance Reserve</i>	Manager	Lihat
Tambah	Menampilkan form add <i>maintenance reserve</i>				Berhasil
Edit	Menampilkan form edit <i>maintenance reserve</i>				Berhasil
Hapus	Menghapus <i>maintenance reserve</i>				Berhasil

		Admin <i>Aircraft</i>	Lihat	Menampilkan list <i>maintenance reserve</i>	Berhasil
			Tambah	Menampilkan form add <i>maintenance reserve</i>	Berhasil
			Edit	Menampilkan form edit <i>maintenance reserve</i>	Berhasil
			Hapus	Menghapus <i>maintenance reserve</i>	Berhasil
		<i>User</i>	Lihat	Menampilkan list <i>maintenance reserve</i>	Berhasil
6.	Melihat Utilization	Manager	Lihat	Menampilkan list Utilization	Berhasil
		Admin <i>Aircraft</i>	Lihat	Menampilkan list Utilization	Berhasil
		<i>User</i>	Lihat	Menampilkan list Utilization	Berhasil

Berdasarkan pengujian *blakbox* yang sudah dilakukan, *system* sudah dapat berhasil menampilkan hasil halaman yang sesuai dengan *use case* yang telah dibuat. Sehingga dapat disimpulkan bahwa sistem yang telah dibuat memiliki fungsionalitas yang baik.

4.3.2 Non Fungsionalitas

Pada tahap ini, pengujian dilakukan untuk melihat apakah fitur-fitur yang direncanakan telah dapat diwujudkan. Hasil pengujian yang dilakukan dapat terlihat pada Tabel 4.1.

Tabel 4.2 Tabel Hasil Pengujian Non Fungsionalitas

No	Komponen	Hasil	Skenario	Kesimpulan
1.	Availability	Sistem yang dibuat harus selalu tersedia atau dapat diakses oleh seluruh pegawai maskapai pesawat dengan mudah dan cepat. Dengan menggunakan komponen	Sistem dideploy ke server sehingga dapat diakses banyak orang	Berhasil

		availability maka <i>system</i> haruslah di upload ke server agar seluruh pegawai dengan mudah memakainya.		
2.	Reliability	Sistem yang dibuat harus selalu tersedia atau dapat diakses oleh seluruh pegawai maskapai pesawat dengan mudah dan cepat. Dengan menggunakan komponen availability maka <i>system</i> haruslah di upload ke server agar seluruh pegawai dengan mudah memakainya.	Dilakukan testing Blackbox terhadap sistem	Berhasil
3.	Safety	Sistem akan dibuat untuk meminimalisir kesalahan yang menyebabkan resiko bagi pengguna. Karna kesalahan sedikit saja dalam menjalankan aplikasi akan berakibat fatal pada pengguna.	Memastikan hasil FH,FC sesuai dengan TSN dan CSN	Berhasil
4.	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.	Membangun sistem dengan Framework Laravel	Berhasil

Berdasarkan pengujian *non fungsionalitas* yang sudah dilakukan, *system* sudah dapat berhasil diimplementasikan sesuai dengan metode *critical software*. Sehingga dapat disimpulkan bahwa sistem yang telah dibuat memiliki asas *non fungsionalitas* yang baik.

4.3.3 User Test

Pengujian sistem adalah tahap akhir dalam penyelesaian sistem ini. Pada tahap ini telah dilakukan pengujian sistem kepada 5 orang untuk mendapatkan hasil, *system* diujikan kepada 1 orang yang berasal dari PT GARUDA INDONESIA dan 4 orang rekan developer. Pengujian sistem dilakukan secara online, dengan mengakses www.aircraft.asdita.com. Sebelum melakukan pengujian *system* pengguna diberi penjelasan singkat tentang sistem yang dibuat kepada calon pengguna. Setelah mencoba sistem calon pengguna diminta untuk mengisi kuesioner melalui *google form* sebagai alat ukur kelayakan sistem. Rakapitulasi hasil dari kuesioner dapat dilihat pada Tabel 4.1.

Tabel 4.3 Tabel Hasil Pengujian Usabilitas

No	Pertanyaan	Penilaian					Presentase
		STS	KS	C	S	SS	
1.	Sistem yang telah dibangun dapat dengan mudah untuk dipahami dan digunakan.			1	2	2	84%
2.	Dalam sistem dapat memasukkan <i>aircraft</i> /pesawat sesuai dengan komponennya (<i>engine1,engine2,airframe,APU,landing gear</i>).			1	1	3	88%
3.	Sistem dapat menampilkan detail <i>aircraft</i> dengan seluruh komponennya (<i>engine1,engine2,airframe,APU,landing gear</i>).			1		4	92%
4.	Sistem dapat menampilkan laporan <i>maintenance reserve</i> per tanggal sesuai utilitas pesawat.				2	3	92%
5.	Sistem yang telah dibangun memiliki tampilan antarmuka yang menarik.				3	2	92%
6.	Ukuran huruf pada sistem yang telah dibangun mudah untuk dilihat. Tidak terlalu besar dan tidak terlalu kecil.			1	3	1	88%

7.	Sistem yang telah dibangun dapat membantu petugas <i>maintenance</i> , dalam memasukan utilitas pesawat ke dalam form (FH,FC,TSC,CSN) .				2	3	92%
8.	Terdapat informasi jika mengalami kesalahan baik disengaja atau tidak.				4	1	84%
9.	Daftar menu yang ada dalam sistem mudah untuk dipahami.			1	4		84%
Total							88,4%.

Berdasarkan perhitungan skor pengujian kuesioner yang telah dilakukan, didapatkan bahwa skor pengujian usability sebesar 88,4%. Berdasarkan data skala Likert tersebut, hasil dari pengujian dapat digolongkan dalam kategori sangat baik. *User* sudah bisa memahami dengan mudah *system* yang telah dibuat dan *user* telah berhasil menjalankan fitur fitur yang terdapat di dalam *system*. Sehingga dapat disimpulkan bahwa sistem yang telah dibuat memiliki usability yang sangat baik.

4.4 Evaluasi Sistem

Berdasarkan hasil pengujian pada *system* diketahui kelebihan dan terdapat kekurangan yang didapatkan oleh pengguna. Kelebihan Sistem Manajemen Aset Pencatatan Utilitas Pesawat adalah ,sistem dapat membantu maskapai pesawat dalam manajemen aset pencatatan utilitas pesawat, sistem dapat membantu petugas dalam melihat jadwal *maintenance* pesawat, sistem dapat membantu petugas dalam melihat total dana yang harus dipersiapkan untuk *maintenance*, dan tampilan *system* dibuat simple sehingga mudah dipahami oleh pengguna.

Adapun kekurangan Sistem Manajemen Aset Pencatatan Utilitas Pesawat adalah , kurang fitur untuk dapat mengoptimalkan *system* dalam fitur *aircraft Utilization*, seperti fitur utilisasi berdasarkan tanggal, sistem yang sudah dibuat belum *support* untuk *mobile responsive*, dan tidak adanya pengingat untuk petugas dalam mengisi *form* utilisasi pesawat dalam fitur *maintenance reserve*.

Hasil *system* menunjukkan bahwa telah berhasil dibangun Sistem Manajemen Aset Pencatatan Utilitas Pesawat menggunakan kajian *Asset Management Software & Critical Safe Software*. Sistem yang sudah dibangun dapat membantu maskapai pesawat dalam manajemen

aset pencatatan utilitas pesawat, melihat jadwal *maintenance* pesawat, dan melihat total dana yang harus dipersiapkan untuk *maintenance*.

