

**APLIKASI TIKET PESAWAT  
“SUMMER TRAVEL”**

**PROYEK AKHIR 1**

Oleh :

<b>Jeffri Bahisindra</b>	<b>3311201008</b>
<b>Viona Fitri Anjani</b>	<b>3311201009</b>
<b>Indita Brayudi</b>	<b>3311201023</b>
<b>Rama Dwi Satriawan</b>	<b>3311201033</b>



**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK NEGERI BATAM  
BATAM  
2013**

## DAFTAR ISI

BAB I PENDAHULUAN .....	1
I.1 Latar Belakang .....	1
I.2 Rumusan Masalah .....	2
I.3 Batasan Masalah .....	2
I.4 Tujuan .....	2
I.5 Sistematika Penulisan .....	3
BAB II TINJAUAN PUSTAKA .....	4
II.1 Visual Paradigm for UML .....	4
II.2 Java .....	4
II.3 OOP ( <i>Object Oriented Programming</i> ) .....	5
II.4 NetBeans IDE .....	6
II.5 MySQL .....	7
II.6 Use Case Diagram .....	7
BAB III ANALISIS DAN PERANCANGAN .....	9
III.1 Deskripsi Umum Sistem .....	9
III.2 Diagram Use Case .....	9
III.3 Skenario Use Case .....	10
III.3.1 Skenario Login .....	10
III.3.2 Skenario Melakukan Transaksi Pembayaran .....	10
III.3.3 Skenario Mengelola Transaksi .....	11
III.3.4 Skenario Mengelola Jadwal Penerbangan .....	11
III.4 Sequence Diagram .....	12
III.4.1 Sequence Login .....	12
III.4.2 Sequence Melakukan Transaksi Pembayaran .....	13
III.4.3 Sequence Mengelola Transaksi .....	13
III.4.4 Sequence Mengelola Jadwal Penerbangan .....	14
III.5 Diagram Kelas .....	15
III.5.1 Rancangan Kelas Rinci .....	15
III.5.1.1 Kelas FrameLogin .....	15
III.5.1.2 Kelas FrameTransaksi .....	16

III.5.1.3 Kelas FrameLaporan .....	16
III.5.1.4 Kelas FrameUpdate .....	17
<b>BAB IV IMPLEMENTASI .....</b>	<b>18</b>
IV.1 Implementasi Kelas .....	18
IV.2 Implementasi Antarmuka .....	19
IV.3 FrameLogin .....	20
IV.3.1 Rancangan Tampilan .....	20
IV.3.2 Deskripsi .....	20
IV.4 FrameAdmin .....	21
IV.4.1 Rancangan Tampilan .....	21
IV.4.2 Deskripsi .....	21
IV.5 FrameKasir .....	22
IV.5.1 Rancangan Tampilan .....	22
IV.5.2 Deskripsi .....	22
IV.6 FrameTransaksi .....	23
IV.6.1 Rancangan Tampilan .....	23
IV.6.2 Deskripsi .....	24
IV.7 FrameLaporan .....	26
IV.7.1 Rancangan Tampilan .....	26
IV.7.2 Deskripsi .....	27
IV.8 FrameUpdate .....	28
IV.8.1 Rancangan Tampilan .....	28
IV.8.2 Deskripsi .....	29
<b>BAB V PENUTUP .....</b>	<b>31</b>
V.1 Kesimpulan .....	31
V.2 Saran .....	31
<b>DAFTAR PUSTAKA .....</b>	<b>32</b>

## **DAFTAR GAMBAR**

Gambar 1 Diagram Use Case .....	9
Gambar 2 Sequence Login .....	12
Gambar 3 Sequence Melakukan Transaksi Pembayaran .....	13
Gambar 4 Sequence Mengelola Transaksi .....	13
Gambar 5 Sequence Mengelola Jadwal Penerbangan .....	14
Gambar 6 Diagram Kelas .....	15
Gambar 7 Kelas Frame Login .....	15
Gambar 8 Kelas Frame Transaksi .....	16
Gambar 9 Kelas Frame Laporan .....	16
Gambar 10 Kelas Frame Update .....	17
Gambar 11 Frame Login .....	20
Gambar 12 Frame Admin .....	21
Gambar 13 Frame Kasir .....	22
Gambar 14 Frame Transaksi Pembayaran .....	23
Gambar 15 Frame Laporan Transaksi .....	26
Gambar 16 Frame Update Daftar Penerbangan .....	28

## **DAFTAR TABEL**

Tabel 1 Implementasi Kelas .....	18
Tabel 2 Implementasi Antarmuka .....	19
Tabel 3 Deskripsi Frame Login .....	20
Tabel 4 Deskripsi Frame Admin .....	21
Tabel 5 Deskripsi Frame Kasir .....	22
Tabel 6 Deskripsi Frame Transaksi Pembayaran .....	24
Tabel 7 Deskripsi Frame Laporan Transaksi .....	26
Tabel 8 Deskripsi Frame Update Jadwal Penerbangan .....	28

# **BAB I**

## **PENDAHULUAN**

Pada bab pendahuluan ini akan dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan serta sistematika penulisan.

### **I.1 Latar Belakang**

Indonesia merupakan sebuah negara yang terdapat di Asia Tenggara. Negara ini merupakan negara kepulauan terbesar di dunia. Negara ini memiliki lebih dari 100.000 pulau yang tersebar sepanjang khatulistiwa. Oleh karena itu, transportasi udara merupakan salah satu transportasi utama di negara ini. Kondisi ekonomi yang terus berkembang membuat sarana transportasi udara menjadi salah satu pilihan bagi masyarakat yang menginginkan kecepatan dan ketetapan waktu perjalanan. Semenjak tahun 2000, peraturan mengenai penerbangan di Indonesia mulai dilonggarkan, hal ini menyebabkan banyaknya maskapai-maskapai penerbangan baru di Indonesia.

Maskapai penerbangan merupakan suatu organisasi yang menyediakan jasa penerbangan bagi penumpang dan barang, mempunyai jadwal teratur serta tarif yang tetap dan berlaku untuk umum. Saat ini banyak travel khususnya bagian kasir dapat melihat informasi jadwal penerbangan harus dengan membuka alamat situs maskapai. Hal ini membuat kasir pada travel merasa kesulitan karena harus membuka situs maskapai yang berbeda beda untuk mendapatkan informasi jadwal penerbangan.

Berdasarkan masalah yang dihadapi inilah penulis melakukan perancangan **APLIKASI TIKET PESAWAT** yang bernama **SUMMER TRAVEL**.

Dalam aplikasi **Summer Travel** data dari pembelian tiket pesawat di kontrol dan disimpan dalam database. Proses pengontrolan data hanya dapat dilakukan oleh admin. Dan proses pembayaran dilakukan oleh kasir.

## **I.2 Rumusan Masalah**

Setelah melakukan analisis maka dirinci kembali masalah seperti :

1. Bagaimana kasir menyediakan informasi jadwal penerbangan ?
2. Bagaimana admin mengontrol data transaksi?

## **I.3 Batasan Masalah**

Untuk memperkecil ruang lingkup PA yang dikerjakan, maka masalah yang akan ditangani akan di batasi sebagai berikut :

1. Aplikasi Summer Travel ini hanya menggunakan bahasa Indonesia.
2. Aplikasi Summer Travel ini tidak berhubungan langsung dengan pihak bank maka pembayaran dilakukan secara manual.
3. Aplikasi Summer Travel ini tidak menyediakan informasi mengenai posisi kursi sesuai yang diinginkan penumpang.
4. Aplikasi Summer Travel ini digunakan hanya untuk kasir dan admin pada perusahaan tersebut.
5. Hanya melayani pembelian tiket pesawat dalam negeri.
6. Aplikasi Summer Travel ini memerlukan pengontrolan terhadap jadwal penerbangan secara berkala.

## **I.4 Tujuan**

Berikut tujuan dibuat aplikasi ini adalah :

1. Kasir dapat menyediakan informasi jadwal penerbangan.
2. Admin dapat mengontrol data transaksi.

## **I.5 Sistematika Penulisan**

Sistematika penyusunan Proyek Akhir ini disusun berdasarkan:

- BAB I : Pendahuluan. Berisi tentang latar belakang, tujuan, batasan masalah dan sistematika penulisan untuk memberikan gambaran isi laporan Proyek Akhir I ini.
  
- BAB II : Deskripsi umum perangkat lunak. Berisi tentang deskripsi umum sistem, karakteristik pengguna dan lingkungan operasional dan lingkungan pengembangan.
  
- BAB III : Deskripsi perancangan. Berisi tentang peraturan penamaan dan penomoran, deskripsi data, deskripsi fungsional yang mencakup struktur menu dan spesifikasi fungsi layar serta rancangan implementasi fisik yang terdiri dari spesifikasi ketergantungan antar modul dan struktur direktori dan deskripsi *file*.
  
- BAB IV : Berisi tentang dekskripsi implementasi, pengujian serta perancangan sistem perangkat lunak dan perangkat keras, untuk perancangan sistem perangkat lunak terdiri dari kelas-kelas yang terkandung dalam aplikasi yang dijelaskan dalam diagram kelas.
  
- BAB V : Kesimpulan dan saran dari pembuatan dan pengembangan aplikasi.



## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini berisi tentang teori-teori yang berkaitan dengan Proyek Akhir dan juga penjelasan-penjelasan.

#### **II.1 Visual Paradigm for UML**

*Visual Paradigm for UML* (VP-UML) adalah alat pendukung UML 2, SysML dan *Business Process Modeling Notation* (BPMN) dari *Object Management Group* (OMG). Selain dukungan pemodelan, ia menyediakan pembuatan laporan dan kemampuan rekayasa kode termasuk generasi kode. Hal ini dapat membalikkan diagram *engineer* dari kode, dan memberikan *round-trip* rekayasa untuk berbagai bahasa pemrograman. Notasi yang digunakan dalam pemodelan adalah UML sebagai notasi standart dari pemodelan *software*. Dan juga, Visual paradigm mempunyai motto “*Build Quality Application Faster, Better and Cheaper*” yaitu tool program yang berguna jika membuat aplikasi dengan proses pemodelan terlebih dahulu.

#### **II.2 Java**

Java merupakan bahasa pemrograman yang diciptakan oleh James Gosling pada tahun 1996 dan mengklaim dirinya mengimplementasikan konsep PBO. Sampai saat ini pengembangan java berada dibawah *Sun Microsystem* walaupun akhir akhir tahun lalu java mulai di *open-source* kan. Java dapat diimplementasikan pada berbagai aspek kehidupan mulai dari komputer *mainframe*, PC, *handphone*, PDA, *smart Card* sampai dengan perlengkapan rumah tangga seperti mesin cuci dan TiVo.

Java menjanjikan sifat *platform independent* yang berarti program yang cukup ditulis satu kali dan di *compile* satu kali maka akan dapat di jalankan di mesin lain tanpa memerlukan pengubahan kode. Sampai saat ini java terbagi menjadi empat kategori yaitu *Java2 Platform Standart Edition* (J2SE) untuk aplikasi desktop, *Java2 Platform Edition* (J2EE) untuk aplikasi server kelas *enterprise* yang

biasanya berskala besar, *Java2 Platform Micro Edition* (J2ME) untuk aplikasi pada perangkat yang memiliki tingkat komputasi tidak setingkat komputer.

### **II.3 OOP (*Object Oriented Programming*)**

OOP adalah pemrograman yang tujuan utamanya membentuk objek dan mengatur interaksi objek dengan objek lainnya untuk memecahkan masalah.

Prinsip-prinsip OOP adalah :

#### 1. Encapsulation

*Encapsulation* adalah mekanisme pemrograman yang membungkus kode dan data yang dimanipulasi dan menjaga supaya terhindar dari interferensi dan penggunaan yang tidak perlu dan salah satu caranya dengan membentuk objek.

Manfaat *encapsulation* adalah :

- Menyembunyikan implementasi detail sebuah *class* dan menyediakan *public method*
- Memaksa pengguna untuk menggunakan *method* untuk mengakses data
- Membuat kode lebih terpelihara.

#### 2. Inheritance

*Inheritance* memungkinkan programmer meletakkan anggota yang sama dalam satu *class* dan *class* lain dapat mewarisi anggota tersebut. *Class* yang mengandung member yang sama dari beberapa *class* dinamakan *superclass* atau *parent class*. *Class* yang mewarisi dinamakan *subclass* atau *child class*. *Inheritance* menghasilkan class hirarki.

### 3. Polymorphisme

*Polymorphisme* artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphic* bila objek-objek itu mempunyai antarmuka yang identik namun mempunyai perilaku-perilaku yang berbeda.

## II.4 NetBeans IDE

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah!). Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama.

Saat ini terdapat dua produk : NetBeans IDE dan NetBeans Platform.

The NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

Tersedia juga NetBeans Platform; sebuah pondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi desktop yang besar. Mitra ISV menyediakan *plug-in* bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam *Platform* dan dapat juga digunakan untuk membuat kakas dan solusi sendiri.

Kedua produk adalah kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License* (CDDL).

## II.5 MySQL

*Database Relational* yang mengizinkan untuk melakukan *query* menjadi alasan menggunakannya. Keunggulan lain dari MySQL yaitu : gratis, cepat, dan mudah dipelajari (terutama jika dikaitkan dengan program yang akan saya gunakan untuk mengaksesnya, PHP). Kita akan menggunakan MySQL untuk menyimpan data informasi jadwal penerbangan yang nantinya akan diakses oleh aplikasi J2ME.

## II.6 Use Case Diagram

*Use case diagram* digunakan untuk memodelkan bisnis proses berdasarkan perspektif penggunaan sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan bentuk *elips* dengan nama operasi dituliskan didalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*. Untuk lebih lengkapnya akan dijelaskan sebagai berikut.

Komponen pembentuk *use case diagram* :

1. Actor

*Actor* merepresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi masukan pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki control atas *use case*. *Actor* digambarkan sengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.

2. Use Case

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga customer atau pengguna sistem memahami dan mengerti mengenai kegunaan sistem yang akan dibangun.

Cara menentukan *use case* dalam suatu sistem :

1. Pola perilaku perangkat lunak aplikasi
2. Gambaran tugas dari sebuah *actor*
3. Sistem atau "benda" yang memberikan sesuatu yang bernilai kepada *actor*
4. Apa yang dikerjakan oleh suatu perangkat lunak (\*bukan bagaimana cara mengerjakannya)

Ada beberapa relasi yang terdapat pada *use case diagram* :

1. *Association*, menghubungkan *link* antar-elemen
2. *Generalization*, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.
4. *Agregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

## BAB III

### ANALISIS DAN PERANCANGAN

Pada bab analisis dan perancangan ini akan dijelaskan tentang arsitektur jaringan dan deskripsi umum sistem, kategori pengguna aplikasi, batasan sistem dan fitur utama perangkat lunak, *use case* serta analisis kelas.

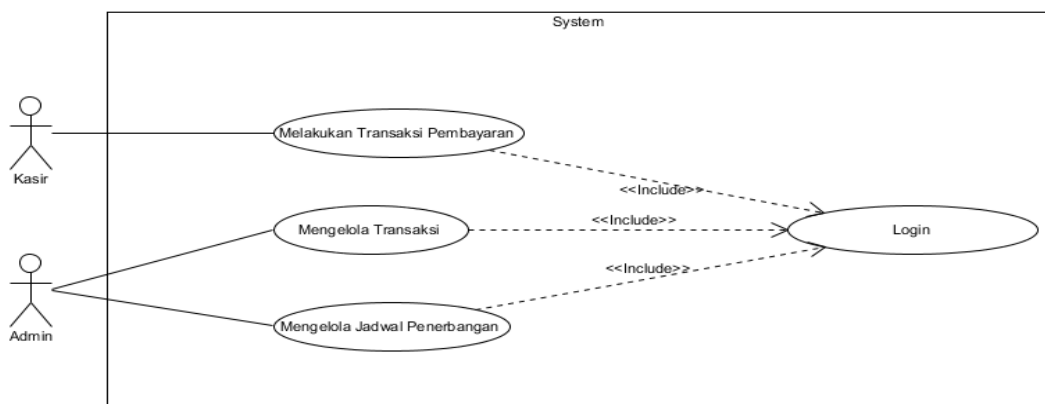
#### III.1 Deskripsi Umum Sistem

Summer Travel adalah perangkat lunak yang digunakan oleh travel untuk melayani transaksi pembayaran secara offline. Sistem ini dilengkapi dengan kemampuan mengelola transaksi, dan mengelola jadwal penerbangan.

Summer Travel ini dibangun dengan menggunakan bahasa Java dan hanya dapat digunakan oleh Kasir dan Admin pada perusahaan travel.

#### III.2 Diagram Use Case

Diagram *use case* adalah diagram yang menggambarkan interaksi antara *actor* dengan sistem. Dalam *use case* Summer Travel terdapat dua *actor* yaitu Kasir dan Admin. Kasir hanya dapat melakukan transaksi pembayaran dengan melakukan login terlebih dahulu, sedangkan Admin dapat mengelola transaksi dan mengelola jadwal penerbangan dengan melakukan login terlebih dahulu, maka dapat disimpulkan diagram *use case* Summer Travel adalah sebagai berikut :



Gambar 1 Diagram Use Case

### III.3 Skenario Use Case

#### III.3.1 Skenario Login

Kondisi Awal: -

Kondisi Akhir: Status memiliki hak akses sebagai Admin atau Kasir.

Skenario: Proses memasukkan *UserName* dan *password*, sistem melakukan *verified* jika *password* yang dimasukkan benar akan di berikan hak akses sebagai admin ataupun kasir, jika *password* yang dimasukkan salah maka akan diminta mengulangi memasukkan *password* yang tepat.

#### III.3.2 Skenario Melakukan Transaksi Pembayaran

Kondisi Awal: Menampilkan daftar transaksi.

Kondisi Akhir: Data transaksi pembayaran disimpan kedalam *database*.

Skenario: Kasir mengklik button tampilDaftar untuk melihat daftar penerbangan apa saja yang tersedia. Setelah *customer* selesai memilih, kemudian kasir akan mengklik jadwal yang dipilih oleh customer. Sistem akan menampilkan NoPenerbangan, Tujuan, Pesawat, Waktu Keberangkatan, Waktu Tiba , Harga dan kasir melakukan proses transaksi, kasir memasukkan NoTransaksi, Tanggal, Jenis Kelamin, Nama, No.KTP, Alamat dan berapa jumlah tiket yang dibeli dan berapa uang yang harus di bayar oleh *customer*. Setelah itu sistem akan menampilkan jumlah kembalian yang akan di berikan kepada *customer*. Kemudian Kasir mengklik button Insert untuk memasukkan data tersebut kedalam database. Data yang disimpan berupa NoTransaksi, Nama, NoKTP, JenisKelamin, Alamat, Tanggal, NoPen, Tujuan, Pesawat, Jumlah, Harga, dan Total.

### **III.3.3 Skenario Mengelola Transaksi**

Kondisi Awal: Belum menampilkan data transaksi.

Kondisi Akhir: Menampilkan data transaksi

Skenario: Admin mengklik button Tampil Daftar. Kemudian system menampilkan data transaksi yang telah terjadi. Admin lalu mencari berdasarkan tujuan atau tanggal ke sistem masukkan data. Data yang di tampilkan yaitu: NoTransaksi, Nama, NoKTP, JenisKelamin, Alamat, Tanggal, NoPen, Tujuan, Pesawat, Jumlah, Harga, dan Total.

### **III.3.4 Skenario Mengelola Jadwal Penerbangan**

Kondisi Awal: -

Kondisi Akhir: Data diupdate kedalam database

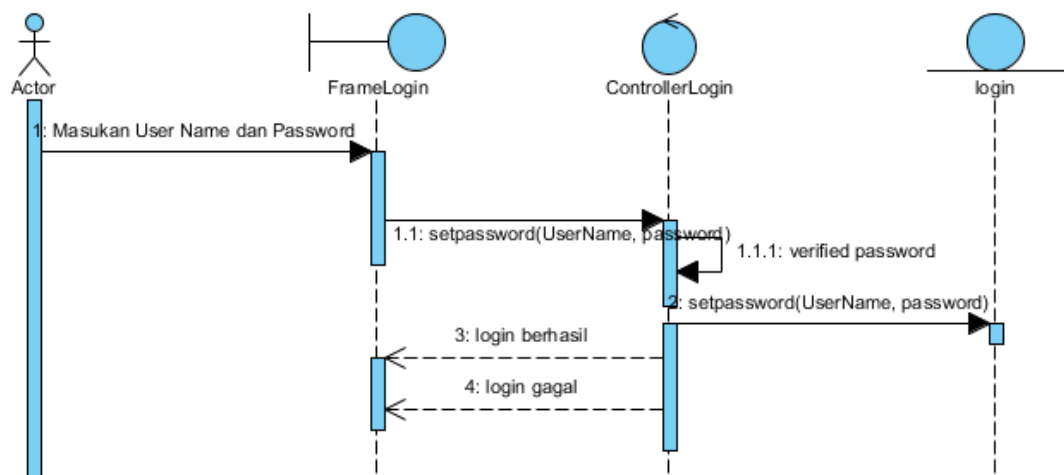
Skenario: Admin menginsert NoPen, Tujuan, Waktu\_Berangkat, Waktu\_Tiba, Tanggal, Harga, dan system menginsert kedalam database. Admin dapat mengupdate dan mendelete jadwal penerbangan.



### III.4 Sequence Diagram

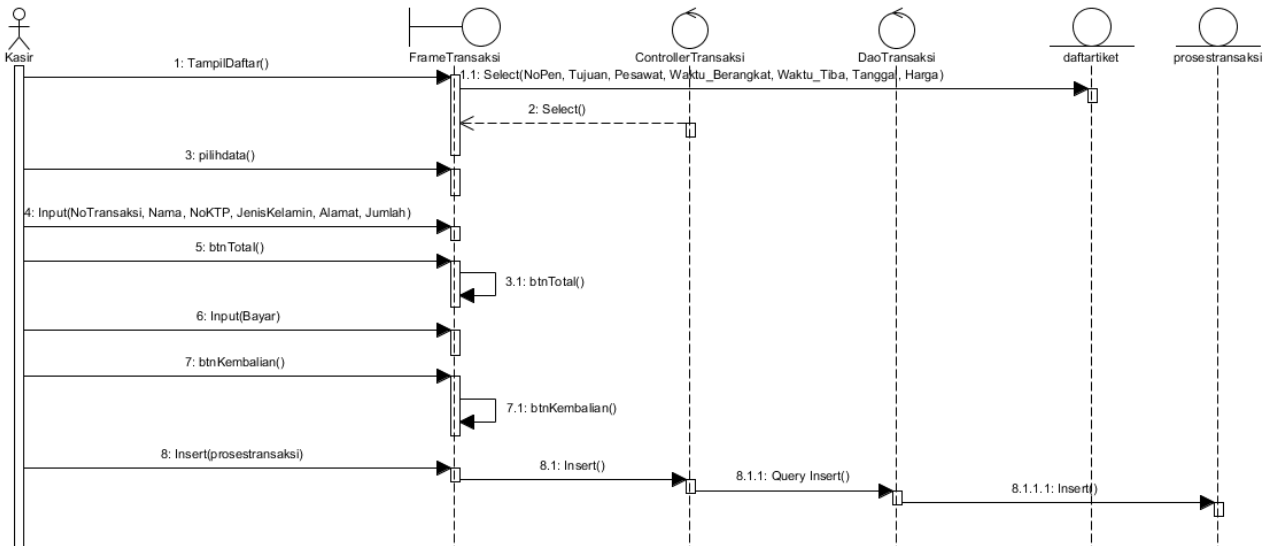
Sequence Diagram digunakan untuk menjelaskan lebih rinci jalannya aplikasi setiap use case. Diagram ini merupakan gambaran dari skenario use case pada sub bab sebelumnya yang merupakan respon dari sebuah even untuk menghasilkan output tertentu.

#### III.4.1 Sequence Login



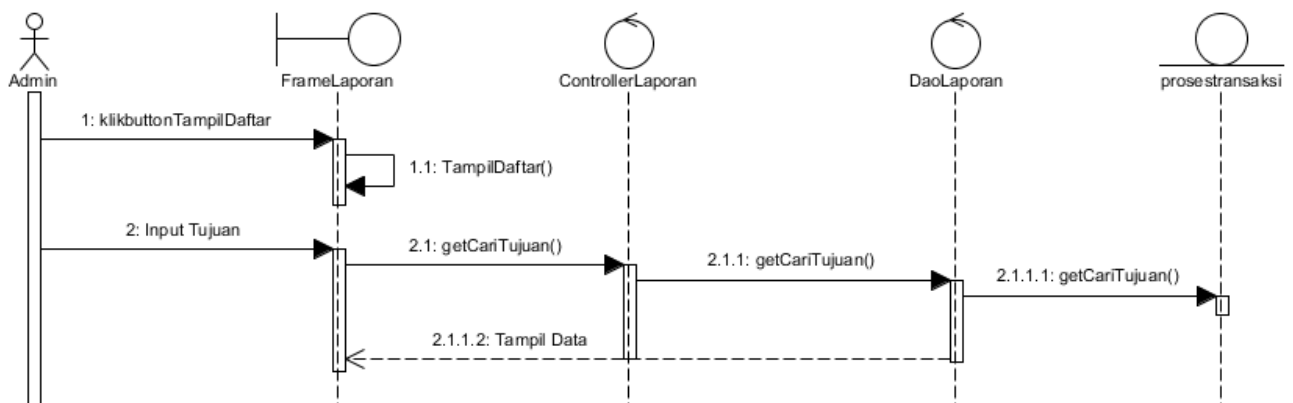
Gambar 2 Sequence Login

### III.4.2 Sequence Melakukan Transaksi Pembayaran



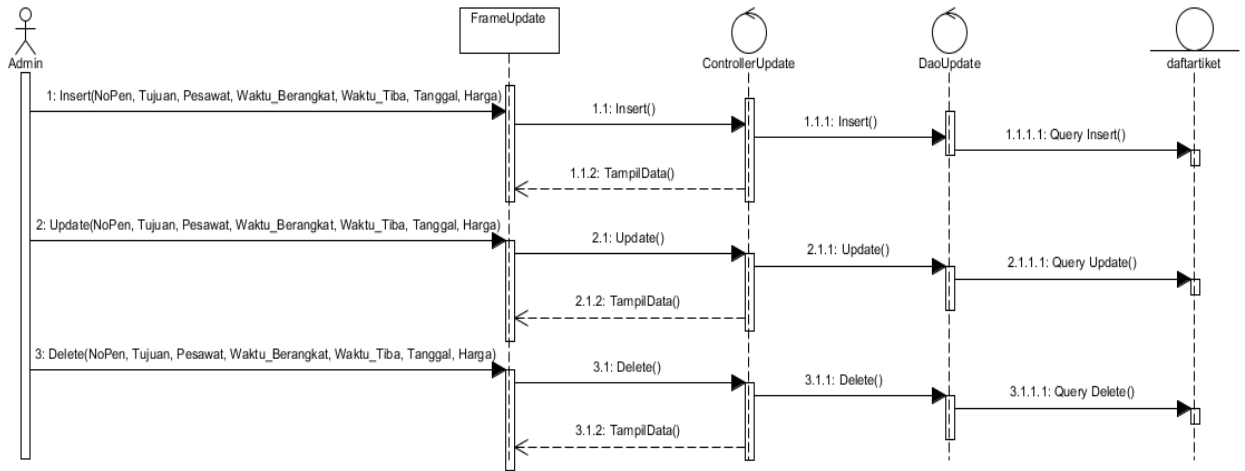
Gambar 3 Sequence Melakukan Transaksi Pembayaran

### III.4.3. Sequence Mengelola Transaksi



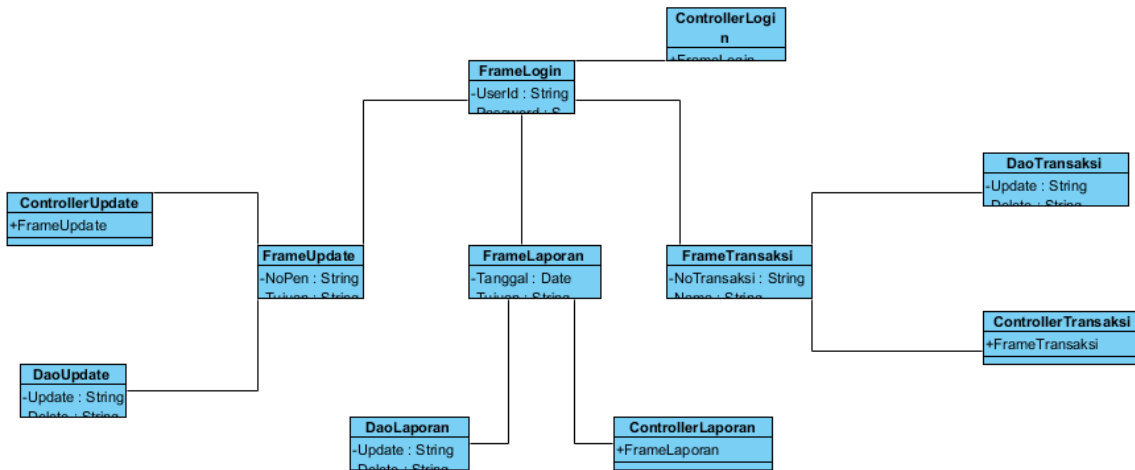
Gambar 4 Sequence Mengelola Transaksi

### III.4.4 Sequence Mengelola Jadwal Penerbangan



Gambar 5 Sequence Mengelola Jadwal Penerbangan

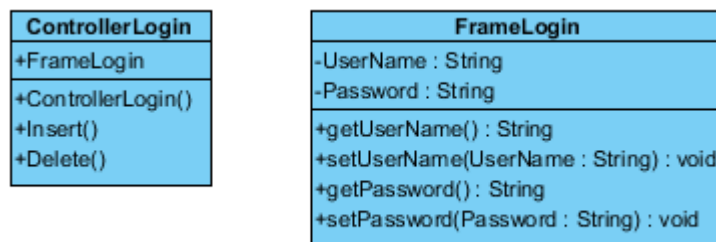
### III.5 Diagram Kelas



Gambar 6 Diagram Kelas

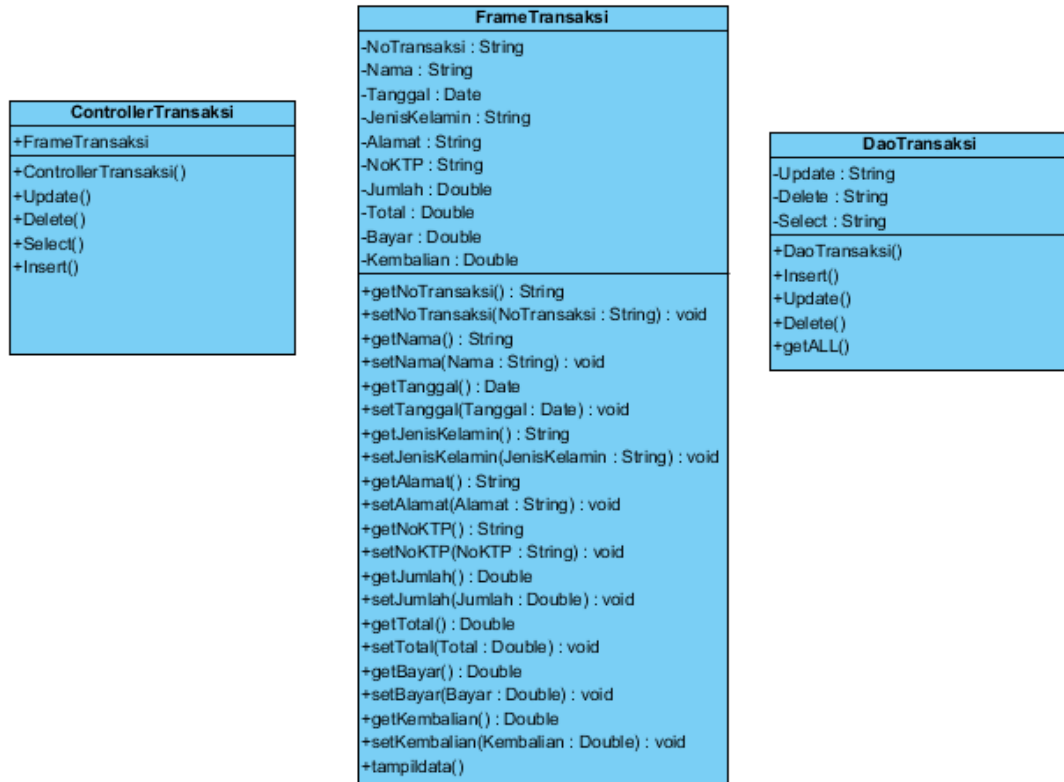
#### III.5.1 Rancangan Kelas Rinci

##### III.5.1.1 Kelas FrameLogin



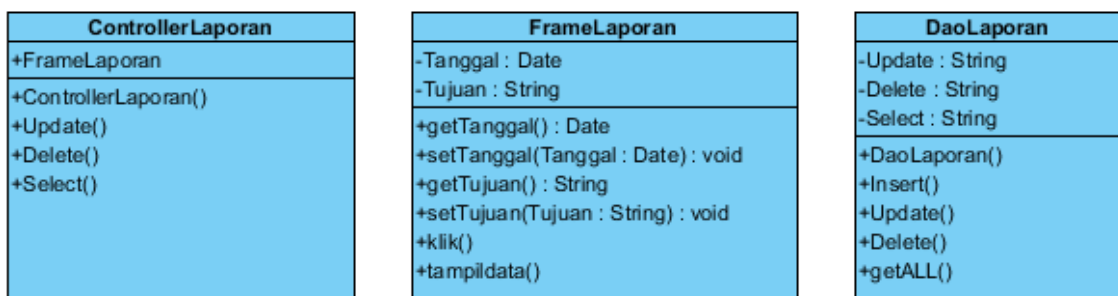
Gambar 7 Kelas Frame Login

### III.5.1.2 Kelas FrameTransaksi



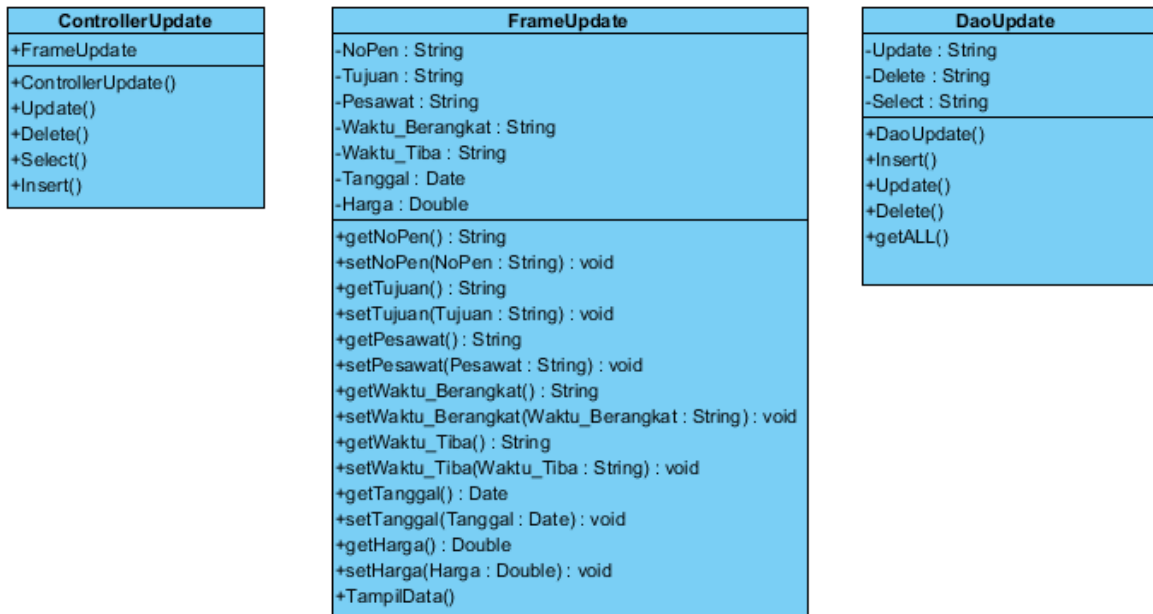
Gambar 8 Kelas Frame Transaksi

### III.5.1.3 Kelas FrameLaporan



Gambar 9 Kelas Frame Laporan

### III.5.1.4 Kelas FrameUpdate



Gambar 10 Kelas Frame Update

## BAB IV

### IMPLEMENTASI

Pada bab implementasi ini akan dijelaskan tentang implementasi kelas, implementasi antarmuka, rancangan tampilan , dan deskripsi dari rancangan tampilan

#### IV.1 Implementasi Kelas

Tabel 1 Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	controllertransaksi	controllertransaksi.java	controllertransaksi.class
2	controllerupdate	controllerupdate.java	controllerupdate.class
3	daotransaksi	daotransaksi.java	daotransaksi.class
4	daoupdate	daoupdate.java	daoupdate.class
5	implementtransaksi	implementtransaksi.java	implementtransaksi.class
6	implementupdate	implementupdate.java	implementupdate.class
7	koneksi	koneksi.java	koneksi.class
8	main	main.java	main.class
9	Laporan	Laporan.java	Laporan.class
10	Transaksi	Transaksi.java	Transaksi.class
11	Update	Update.java	Update.class

12	tablemodeltransaksi	tablemodeltransaksi.java	tablemodeltransaksi.class
13	tablemodelupdate	tablemodelupdate.java	Tablemodelupdate.class
14	FrameAdmin	FrameAdmin.java	FrameAdmin.class
15	FrameKasir	FrameKasir.java	FrameKasir.class
16	FrameLogin	FrameLogin.java	FrameLogin.class
17	FrameLaporan	FrameLaporan.java	FrameLaporan.class
18	FrameTransaksi	FrameTransaksi.java	FrameTransaksi.class
19	FrameUpdate	FrameUpdate.java	FrameUpdate.class

#### IV.2 Implementasi Antarmuka

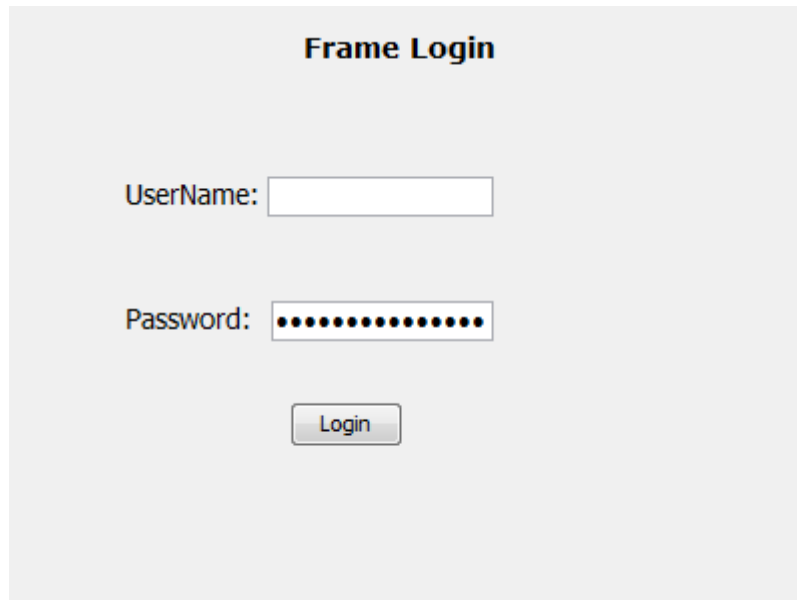
Tabel 2 Implementasi Antarmuka

No	Antarmuka	Nama File Fisik	Nama File Executable
1	FrameAdmin	FrameAdmin.java	FrameAdmin.class
2	FrameKasir	FrameKasir.java	FrameKasir.class
3	FrameLogin	FrameLogin.java	FrameLogin.class
4	FrameLaporan	FrameLaporan.java	FrameLaporan.class
5	FrameTransaksi	FrameTransaksi.java	FrameTransaksi.class
6	FrameUpdate	FrameUpdate.java	FrameUpdate.class



## IV.3 FrameLogin

### IV.3.1 Rancangan Tampilan



The image shows a login interface within a frame titled "Frame Login". It features two input fields: one for "UserName" and one for "Password". The "Password" field is masked with dots. Below the input fields is a "Login" button.

Gambar 11 Frame Login

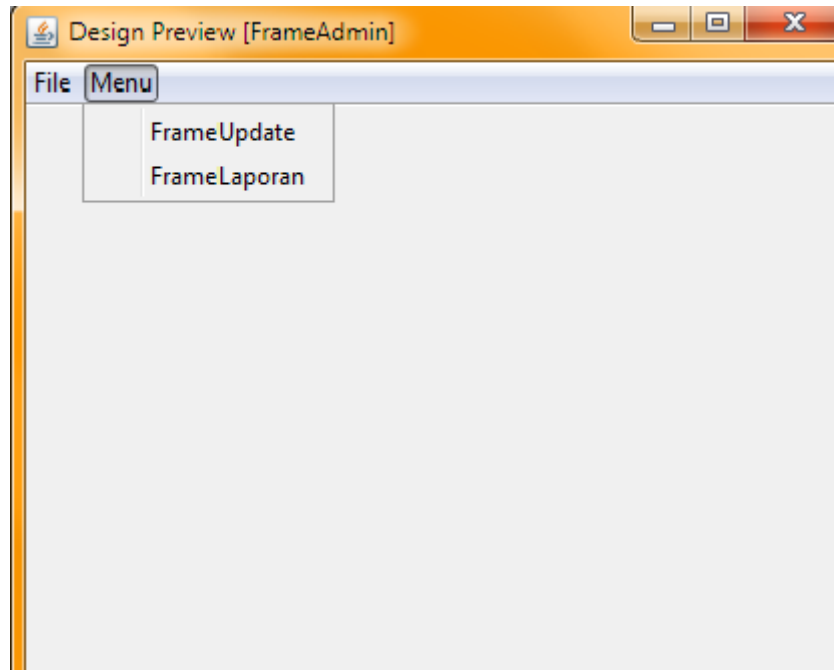
### IV.3.2 Deskripsi

Tabel 3 Deskripsi Frame Login

No	Jenis	Nama	Keterangan
1	Label	Frame Login	Menampilkan judul frame
2	Label	UserName	Menunjukkan Username
3	Label	Password	Menunjukkan Password
4	TextField	txtUserName	Tempat untuk mengisi data
5	TextField	txtPassword	Tempat untuk mengisi data
6	Button	btnLogin	Jika diklik, akan masuk ke sistem login

## IV.4 FrameAdmin

### IV.4.1 Rancangan Tampilan



Gambar 12 Frame Admin

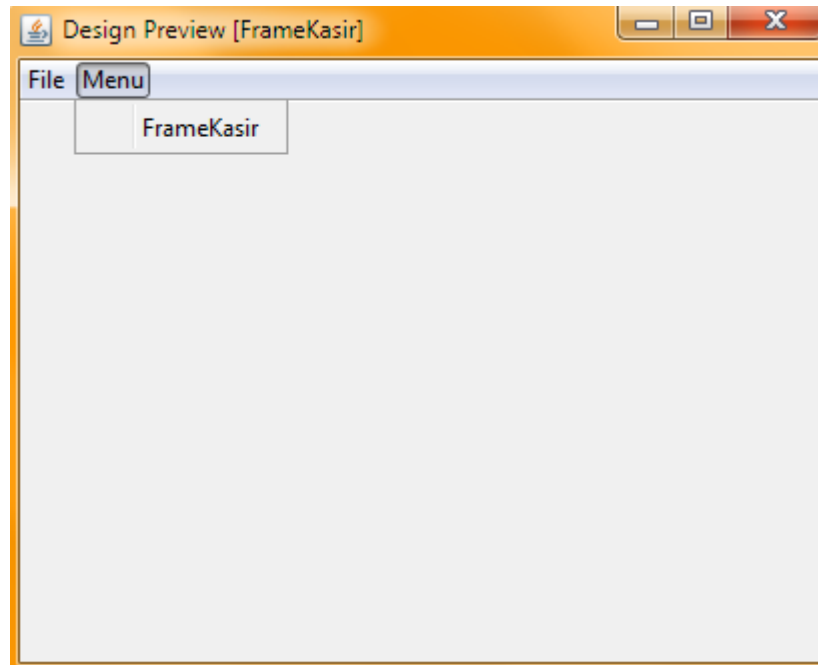
### IV.4.2 Deskripsi

Tabel 4 Deskripsi Frame Admin

No	Jenis	Nama	Keterangan
1	Menu Bar	menuFile	Menunjukkan menu pada tampilan bar
2	Menu Bar	menuMenu	Menunjukkan menu pada tampilan bar
3	Menu Item	itemUpdate	Menunjukkan menu item
4	Menu Item	itemLaporan	Menunjukkan menu item

## IV.5 FrameKasir

### IV.5.1 Rancangan Tampilan



Gambar 13 Frame Kasir

### IV.5.2 Deskripsi

Tabel 5 Deskripsi Frame Kasir

No	Jenis	Nama	Keterangan
1	Menu Bar	menuFile	Menunjukkan menu pada tampilan bar
2	Menu Bar	menuMenu	Menunjukkan menu pada tampilan bar
3	Menu Item	itemUpdate	Menunjukkan menu item

## IV.6 FrameTransaksi

### IV.6.1 Rancangan Tampilan

**Frame Transaksi**

Title 1	Title 2	Title 3	Title 4

     NoPen: -      Waktu Berangkat: -  
Tujuan: -      Waktu Tiba: -  
Pesawat: -      Harga: -

NoTransaksi:       Tanggal: -  
Tanggal:        Nama:   
NoKTP:

Jenis Kelamin:       Alamat:

Jumlah:

Bayar:

Title 1	Title 2	Title 3	Title 4

Gambar 14 Frame Transaksi Pembayaran

#### IV.6.2 Deskripsi

Tabel 6 Deskripsi Frame Transaksi Pembayaran

No	Jenis	Nama	Keterangan
1	Label	Frame Transaksi	Menampilkan judul frame
2	Table	tampildaftar	Tempat untuk menampilkan hasil inputan
3	Button	btntampildaftar	Jika diklik, akan menampilkan daftar transaksi
4	Label	NoPen	Menunjukkan nomor penerbangan
5	Label	Tujuan	Menunjukkan tujuan penerbangan
6	Label	Pesawat	Menunjukkan Pesawat
7	Label	Waktu Berangkat	Menunjukkan waktu keberangkatan
8	Label	Waktu Tiba	Menunjukkan waktu tiba di daerah tujuan
9	Label	Harga	Menunjukkan harga tiket
10	Label	Tanggal	Menunjukkan tanggal
11	Label	NoTransaksi	Menunjukkan nomor transaksi
12	Label	Tanggal	Menunjukkan tanggal
13	Label	Nama	Menunjukkan nama
14	Label	NoKTP	Menunjukkan no. KTP

15	Label	Jenis Kelamin	Menunjukkan jenis kelamin
16	Label	Alamat	Menunjukkan alamat
17	Label	Jumlah	Menunjukkan jumlah tiket
18	Label	Bayar	Menunjukkan jumlah yang harus dibayar
19	Button	btnKembalian	Jika diklik, menunjukkan jumlah uang kembali
20	Button	btnTotal	Jika diklik, menunjukkan total harga tiket
21	Button	btnInsert	Jika diklik, akan mengaktifkan fungsi insert
22	Button	btnUpdate	Jika diklik, akan mengaktifkan fungsi update
23	Button	btnReset	Jika diklik, akan mengaktifkan fungsi reset
24	Button	btnDelete	Jika diklik, akan mengaktifkan fungsi delete
25	TextField	txtNoTransaksi	Tempat untuk mengisi data
26	TextField	txtTanggal	Tempat untuk mengisi data
27	TextField	txtJK	Tempat untuk mengisi data
28	TextField	txtNama	Tempat untuk mengisi data
29	TextField	txtNo.KTP	Tempat untuk mengisi data
30	TextField	txtAlamat	Tempat untuk mengisi data

31	TextField	txtJumlah	Tempat untuk mengisi data
32	TextField	txtBayar	Tempat untuk mengisi data
33	TextField	txtKembalian	Tempat untuk mengisi data
34	Table	tabletransaksi	Tempat untuk menampilkan hasil inputan

## IV.7 FrameLaporan

### IV.7.1 Rancangan tampilan

**Frame Laporan Transaksi**

Title 1	Title 2	Title 3	Title 4

Cari Data Berdasarkan Tujuan:

Cari Data Berdasarkan Tanggal:

Gambar 15 Laporan Transaksi

#### IV.7.2 Deskripsi

Tabel 7 Deskripsi Frame Laporan Transaksi

No	Jenis	Nama	Keterangan
1	Label	Frame Laporan Transaksi	Menampilkan judul frame
2	Table	jTabel1	Tempat untuk menampilkan hasil inputan transaksi
3	Button	btnTampilData	Jika diklik, menampilkan data laporan transaksi
4	Label	Cari Data Berdasarkan Tujuan	Menunjukkan tujuan yang akan diisi pada txtTujuan()
5	TextField	txtTujuan	Tempat untuk mengisi data
6	Label	Cari Data Berdasarkan Tanggal	Menunjukkan tanggal yang akan diisi pada txtTanggal()
7	Textfield	txtTanggal	Tempat untuk mengisi data



## IV.8 FrameUpdate

### IV.8.1 Rancangan tampilan

**Frame Update Daftar Penerbangan**

No\_Pen:

Tujuan:

Pesawat:

Waktu Berangkat:

Waktu Tiba:

Tanggal:

Harga:

Cari Berdasarkan Tujuan:

Title 1	Title 2	Title 3	Title 4

Gambar 16 Frame Update Daftar Penerbangan

#### IV.8.2 Deskripsi

Tabel 8 Deskripsi Frame Update Daftar Penerbangan

No	Jenis	Nama	Keterangan
1	Label	Frame Update Daftar Penerbangan	Menampilkan judul frame
2	Label	No_Pen	Menunjukkan no penerbangan
3	Label	Tujuan	Menunjukkan tujuan penerbangan
4	Label	Pesawat	Menunjukkan nama pesawat
5	Label	Waktu Berangkat	Menunjukkan waktu keberangkatan
6	Label	Waktu Tiba	Menunjukkan waktu tiba di daerah tujuan
7	Label	Tanggal	Menunjukkan tanggal
8	Label	Harga	Menunjukkan harga tiket
9	Button	Insert	Jika diklik, akan mengaktifkan fungsi insert
10	Button	Update	Jika diklik, akan mengaktifkan fungsi update
11	Button	Delete	Jika diklik, akan mengaktifkan fungsi delete
12	Button	Reset	Jika diklik, akan mengaktifkan

			fungsi reset
13	Button	Cari	Jika diklik,
14	Label	Cari Berdasarkan Tujuan	Menunjukkan pencarian berdasarkan tujuan
15	Tabel	jTable1	Tempat hasil inputan
16	TextField	txtNoPen	Tempat untuk mengisi data
17	ComboBox	cbTujuan	Tempat untuk memilih
18	ComboBox	cbPesawat	Tempat untuk memilih
19	TextField	txtWaktuBerangkat	Tempat untuk mengisi data
20	TextField	txtWaktuTiba	Tempat untuk mengisi data
21	TextField	txtTanggal	Tempat untuk mengisi data
22	TextField	txtHarga	Tempat untuk mengisi data
23	TextField	txtCari	Tempat untuk mengisi data

## **BAB V**

### **PENUTUP**

#### **V.1 Kesimpulan**

Setelah dilakukan tahap analisis, perancangan, dan implementasi pada aplikasi Summer Travel, maka kesimpulan yang didapat adalah :

3. Aplikasi Tiket Pesawat “Summer Travel“ dapat menyediakan informasi jadwal penerbangan kepada kasir.
4. Aplikasi Tiket Pesawat “Summer Travel“ dapat melakukan pengontrolan data yang di kelola oleh admin.

#### **V.2 Saran**

Adapun saran yang dapat diberikan untuk penyempurnaan dari Aplikasi Tiket Pesawat “Summer Travel” adalah sebagai berikut :

1. Aplikasi dapat menggunakan bahasa asing.
2. Aplikasi dapat menyediakan informasi penerbangan luar negeri.

## **DAFTAR PUSTAKA**

1. Pascal Roques, “UML in Practice”, John Wiley & Sons, 2001
2. Patric Grassle, Henriette Baumann, Philippe Baumann, “UML 2.0 in Action”, Packt publishing, 2005
3. Doug Rosenberg, Matt Stephens, “Use Case Driven Object Modeling with UML”, Apress, 2007