

**BAB IV**  
**METODE PENELITIAN**

**4.1 Lokasi dan Waktu Penelitian**

**4.1.1 Lokasi**

Lokasi penelitian dilakukan pada Dusun I Desa Lumpatan Kecamatan Sekayu Kabupaten Musi Banyuasin.

**4.1.2 Waktu Penelitian**

Kegiatan akan dimulai pada bulan Oktober 2017 dan direncanakan akan berakhir pada bulan Januari 2018. Dan tempat penelitian di laksanakan di Dusun 1 Desa Lumpatan Kecamatan Sekayu, Kabupaten Musi Banyu Asin. Dengan mengikuti tahapan-tahapan metode *Rational Unified Process* (RUP). Tabel jadwal kegiatan penelitian dapat dilihat pada Tabel 4.1

**Tabel 4.1 Jadwal Kegiatan Penelitian**

Kegiatan	2017															
	Oktober				November				Desember				Januari			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Inception																
Elaboration																
Construction																
Trantition																

## **4.2 Jenis Data**

### **1. Data Primer**

Data primer adalah data yang dikumpulkan dan diolah sendiri oleh organisasi yang menerbitkan atau yang menggunakannya. . Data primer merupakan data sensus karena dikumpulkan, diolah, serta diterbitkan oleh Biro Pusat Statistik (Pasolong, 2013:70).

Data yang diperoleh melalui wawancara peneliti dengan narasumber mengenai reservasi tiket pada CV. Alerya Travel.

### **2. Data Sekunder**

Data sekunder adalah semua data yang diperoleh secara tidak langsung dari objek penelitian. Jadi data sekunder adalah data yang dikumpulkan atau digunakan oleh organisasi yang bukan pengelolahnya. (Pasolong, 2013:70)

Data yang diperoleh dari catatan atau dokumentasi seperti data perusahaan, struktur organisasi dan data karyawan CV. Alerya Travel Dusun I Desa Lumpatan Kecamatan Sekayu Kabupaten Musi Banyuasin.

## **4.3 Teknik Pengumpulan Data**

Dalam penyusunan laporan ini, penulis menggunakan beberapa teknik dalam mengumpulkan data yang akurat yaitu:

### **1. Observasi (Pengamatan)**

Menurut Sujarweni (2014:75) observasi adalah pengamatan dan pencatatan secara sistematis terhadap gejala yang tampak pada objek penelitian. Penulis melakukan observasi langsung pada CV. Alerya. Didalam observasi ini kami mendapatkan beberapa berkas berupa tanda daftar perusahaan persekutuan komanditer (CV), berkas pendirian perseron komanditer CV. Alerya, data pelanggan, dan transaksi barang.

### **2. Wawancara**

Menurut Narbuko (2012:83) wawancara adalah proses tanya-jawab dalam penelitian yang berlangsung secara lisan dalam mana dua orang atau lebih bertatap muka mendengarkan secara langsung informasi-informasi atau keterangan-keterangan. Dengan menggunakan metode wawancara, penulis mendapatkan secara detail tentang desain *website* yang diinginkan, dan data-data apa saja yang diperlukan untuk *website reservasi*. Peneliti melakukan wawancara kepada pemilik CV. Alerya Travel bapak Erwin Yakub dan ibu Latipah. Didalam wawancara bersama bapak Erwin Yakub sebagai pemilik perusahaan kami mendapatkan bukti berupa data jadwal keberangkatan, laporan pengiriman tiket, tanda daftar perusahaan persekutuan kamanditer dan surat izin usaha.

### **3. Studi Pustaka**

Menurut Afrizal (2015:122) studi pustakan merupakan bagian dari sebuah proposal penelitian yang berisikan informasi-informasi yang

diperoleh dari jurnal, buku dan kertas kerja (*working paper*). Dalam hal ini penulis melakukan studi pustaka dengan cara membaca buku, jurnal atau makalah ilmiah yang berkaitan dengan topik dan pembahasan.

#### **4.4 Alat dan Teknik Pengembangan Sistem**

##### **4.4.1 Alat Pengembangan Sistem**

###### **4.4.1.1 Pemodelan Proses**

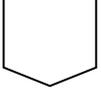
Menurut Susanto dan Mariana (2013) UML (*Unified Modeling Language*) adalah sebuah bahasa untuk melakukan spesifikasi, visualisasi, membangun dan pendokumentasian dari sebuah sistem *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas kelas dalam bahasa program yang spesifik, skema *database* dan komponen - komponen yang diperlukan dalam sistem *software*.

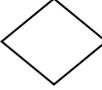
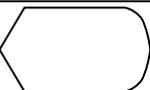
Pada penelitian ini penulis menggunakan model proses *flowchart* dan *activity diagram*.

###### **1. Flowchart**

Menurut Kristanto (2008:75), *flowchart* atau berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Adapun simbol-simbol yang sering digunakan dapat dilihat pada tabel 4.1.

**Tabel 4.2 Simbol *Flowchart***

Simbol	Keterangan
 <i>Document</i>	Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendek atau cetakan.
 <i>Multi Documents</i>	Multi Dokumen.
 <i>Manual Operation</i>	Proses Manual.
 <i>Process</i>	Proses yang dilakukan oleh komputer.
 <i>Manual File</i>	Menandakan dokumen yang diarsipkan (arsip manual).
 <i>Database</i>	Data penyimpanan ( <i>data storage</i> )
 <i>Predefined process</i>	Proses apa saja yang tidak terdefinsi termasuk aktivitas fisik.
 <i>Off-page Reference</i>	Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman lain.
 <i>On-page Reference</i>	Terminasi yang mewakili simbol tertetu yang digunakan pada aliran lain yang sama.
	Terminasi yang menandakan awal dan akhir dari

 <i>Terminator</i>	suatu aliran.
 <i>Decision</i>	Pengambilan keputusan.
 <i>Display</i>	Layar peraga ( <i>monitor</i> )
 <i>Manual Input</i>	Pemasukan data secara manual.

## 2. Activity Diagram

Menurut Munawar (2005 : 109) *Activity diagram* adalah teknik untuk mendeskripsikan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedang *flowchart* tidak bisa.

**Tabel 4.2 Simbol Activity Diagram**

Nama dan Simbol	Keterangan
 Start	Mendeskripsikan suatu tindakan sebelum aktivitas dimasukan.

 Arus Kegiatan	Mendeskripsikan ke mana aliran kegiatan.
 Proses / Kegiatan	Mendeskripsikan tentang suatu tindakan aktivitas, proses kegiatan.
 <i>Decisions</i>	Mendeskripsikan tentang suatu tindakan untuk menghasilkan keputusan.
 <i>Fork</i>	Menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
 <i>Final Activity</i>	Mendeskripsikan suatu tindakan sesudah aktivitas.

#### 4.4.1.2 Pemodelan Data

Pada penelitian ini penulis menggunakan model data *Class Diagram* dan *Entity Relationship Diagram*.

##### 1. *Class Diagram*

Menurut Rosa & Shalahuddin (2013) diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

**Tabel 4.3 Class Diagram**

Nama	Simbol	Deskripsi
Kelas ( <i>Class</i> )		Kelas pada struktur sistem
Asosiasi ( <i>Association</i> )		Realasi antar kelas dengan makna umum. Biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah ( <i>Direct Association</i> )		Relasi antar kelas dengan makna kelas satu digunakan oleh kelas lainnya. Biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi		Relasi antar kelas dengan makna generalisasi-spesialisasi.
Kebergantungan ( <i>Dependency</i> )		Relasi antar kelas dengan makna kelas kebergantungan

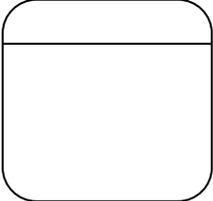
		antar kelas.
Agregasi ( <i>Aggregation</i> )		Relasi antar kelas dengan makna semua-sebagia.

## 2. Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) adalah pemodelan awal yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relasional (Rosa, 2013, h.50).

Simbol dasar dalam pembuatan ERD dari James Martin yang mendasari semua model data ditunjukkan pada Tabel 4.4.

**Tabel 4.5 Entity Relationship Diagram**

Simbol	Keterangan
	<i>Entity</i> atau entitas menunjukkan orang, tempat, objek, peristiwa atau konsep tentang apa yang digunakan untuk meng- <i>capture</i> dan menyimpan data.
	<i>Attributes</i> atau atribut merupakan sifat atau karakteristik deskriptif suatu entitas.

#### 4.4.2 Teknik Pengembangan Sistem

Menurut Rosa (2016:125) RUP (*Rational Unified Process*) adalah sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) inkremental (bertahap dengan progres menaik). Iteratif bisa dilakukan di dalam setiap tahap, iteratif yang inkremental bertambah menari dimana setiap interaksi akan memperbaiki interaksi berikutnya. Salah satu *Unified Process* yang terkenal adalah RUP (*Rational Unified Process*).

Menurut Rosa (2016:128-131) RUP memiliki empat buah tahap atau *fase* yang dapat dilakukan secara iteratif.

Tahapan-tahapan metode *Rational Unified Process (RUP)*:

a. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modelling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut ini tahap yang dibutuhkan pada tahap ini:

- 1) Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya)
- 2) Membangun kasus bisnis yang dibutuhkan.

Hasil yang diharapkan dari tahap ini adalah memenuhi *Lifecycle Objective Milestone* (batas/tonggak objektif dari siklus) dengan kriteria berikut :

- 1) Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, dan perkiraan jadwal.
- 2) Kebutuhan dimengerti dengan pasti (dapat dibuktikan) dan sejalan dengan kasus primer yang dibutuhkan.
- 3) Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, resiko, dan proses pengembangan.
- 4) Ruang lingkup purwarupa (*prototype*) yang akan dikembangkan.
- 5) Membangun garis dasar dengan membandingkan perencanaan aktual dengan perencanaan yang direncanakan.

b. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

Hasil yang diharapkan dari tahap ini adalah memenuhi *Lifecycle Architecture Milestone* (batas/tonggak arsitektur dari siklus) dengan kriteria berikut:

- 1) Model kasus yang digunakan (*use case*) dimana kasus dan aktor yang terlibat telah diidentifikasi dan sebagian besar kasus harus dikembangkan. Model *use case* harus 80 persen lengkap dibuat.
- 2) Deskripsi dari arsitektur perangkat lunak dari proses pengembangan sistem perangkat lunak telah dibuat.
- 3) Rancangan arsitektur yang dapat diimplementasikan dan mengimplementasikan *use case*.
- 4) Kasus bisnis atau proses bisnis dan daftar resiko yang sudah mengalami perbaikan (revisi) telah dibuat.
- 5) Rencana pengembangan untuk seluruh proyek telah dibuat.
- 6) Purwarupa (*prototype*) yang dapat didemonstrasikan untuk mengurangi setiap resiko teknis yang diidentifikasi.

c. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

d. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat

lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk implementasi proyek, pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*. Produk perangkat lunak juga disesuaikan dengan kebutuhan yang didefinisikan pada tahap *inception*. Jika semua kriteria objektif terpenuhi maka dianggap sudah memenuhi *Produk Release Milestone* (batas/tonggak peluncuran produk) dan pengembangan perangkat lunak selesai dilakukan.

#### **4.5 Alat dan Teknik Pengujian**

##### **1. *Black Box Testing***

Teknik pengujian yang penulis gunakan adalah *Black-BoxTesting*. Menurut Rosa (2016:275) *Black Box Testing* yaitu menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

