

BAB IV

METODE PENELITIAN

4.1. Lokasi dan Jadwal Penelitian

4.1.1. Lokasi

Lokasi penelitian dilakukan pada Kantor Kecamatan Lalan di Bandar Agung P.16 Karang Agung Tengah Palembang Sumatera Selatan (30758). Telp. (0711) 8986031.

4.1.2. Jadwal Penelitian

Adapun Jadwal Penelitian yang diajukan untuk pembuatan *website* seperti yang tercantum pada Tabel 4.1.

Tabel 4.1 Jadwal Kegiatan

Kegiatan	Maret				April				Mei				Juni			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. Persiapan	■	■	■	■												
2. Pencarian data dan <i>literature</i>	■	■	■	■	■	■										
3. Ujian Proposal							■									
4. Revisi Proposal								■	■							
5. <i>Planning</i> (Perencanaan)	■	■	■	■	■	■										
6. <i>Design</i> (Perancangan)							■	■	■							
7. <i>Coding</i> (Pengkodean)									■	■	■	■				
8. <i>Testing</i> (Pengujian)												■	■			
9. Penyusunan laporan hasil penelitian															■	■

4.2. Jenis Data

Dalam hal ini perlu diuraikan apakah data dalam penelitian ini adalah data primer atau data sekunder.

4.1.3. Data Primer

Data yang diperoleh peneliti secara langsung dari objek penelitian yang berkaitan langsung dengan masalah efektifitas pelayanan pada kantor kecamatan lalan kabupaten musi banyuasin.

4.1.4. Data Skunder

Data yang diperoleh dari kantor kecamatan lalan, yang berkaitan dengan laporan penelitian, yang terdiri dari profil kecamatan dan struktur organisasi.

4.3. Teknik Pengumpulan Data

Dalam penelitian ini penulis menggunakan beberapa teknik untuk mendapatkan data, yaitu:

4.3.1. Observasi

Menurut Chairina, dkk (2015:40) Observasi merupakan salah satu teknik pengumpulan data dengan melakukan pengamatan baik secara langsung maupun tidak langsung terhadap subjek yang diteliti. Dalam penerapan metode ini penulis melakukan observasi dengan datang langsung ke kantor kecamatan lalan dan mendapatkan permasalahan dalam prosedur penyampaian informasi,

karena informasi yang disampaikan kepada masyarakat membutuhkan beberapa tahapan, yaitu staf kecamatan memberikan informasi kepada kepala desa, kemudian kepala desa memberikan informasi tersebut kepada kepala dusun/Rw, lalu kepala dusun memberikan informasi tersebut kepada ketua Rt untuk disampaikan ke masyarakat. sehingga proses penyampaian informasi pelayanan membutuhkan waktu beberapa hari untuk sampai kemasyarakat.

4.3.2. Wawancara

Menurut Widodo, dkk (2016:86) Wawancara adalah metode dimana pengumpulan data dilakukan dengan cara tanya jawab dengan petugas terkait. Teknik pengumpulan data dengan wawancara bertujuan untuk mendapatkan data secara langsung mengenai masalah yang nantinya dijadikan bahan untuk solusi sistem yang akan dirancang. Dalam penerapan metode ini penulis melakukan wawancara kepada Bapak Sukardi, S.I.P kasubbag umum di kantor kecamatan lalan medapatkan informasi, seperti informasi pelayanan IMB dan SITU. Untuk mendapatkan suatu pelayanan IMB dan SITU dari kecamatan membutuhkan waktu hingga dua hari, karena masyarakat harus datang ke kantor kecamatan dengan kondisi jalan yang rusak, jarak lokasi yang berbeda-beda dan harus menunggu antrian untuk mendapatkan pelayanan.

4.3.3. Dokumentasi

Menurut Rizal, dkk (2014:50) Metode ini digunakan sebagai dasar untuk menganalisa data dalam penelitian ini, data yang digunakan berupa Laporan Pertanggung jawaban dan Prosedur Pembayaran *Retribusi* di Kantor Pelayanan Terpadu Kabupaten Lumajang.

4.3.4. Studi Pustaka

Menurut Lukman dan Sunoto (2016:226) studi pustaka (*Library Research*) adalah pengumpulan data-data dengan cara mempelajari berbagai bentuk bahan-bahan tertulis seperti buku-penunjang kajian, catatan-catatan maupun referensi lain yang bersifat tertulis. Sumber pustaka yang didapat oleh penulis dari buku-buku atau hasil penelitian seperti skripsi disertai sumber-sumber lainnya seperti internet yang berkaitan dengan *website*.

4.4. Jenis Penelitian

4.4.1. Penelitian Terapan

Menurut Fuspita, dkk (2014 : 47), penelitian terapan adalah penyelidikan hati-hati, sistematis, dan terus menerus terhadap suatu masalah dengan tujuan untuk digunakan dengan segera untuk keperluan tertentu. Hasil penelitian tidak perlu sebagai satu penemuan baru tetapi merupakan dari penelitian yang telah ada.

4.5. Alat dan Teknik Pengembangan Sistem

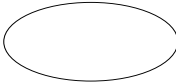


4.5.1. Alat Pengembangan Sitem

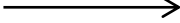

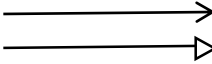
4.5.1.1. Model Proses

1. *Usecase Diagram*

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 4.2 Simbol *Use Case Diagram*

Simbol	Deskripsi
	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor.
	Aktor, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.
	Asosiasi Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi.
Simbol	Deskripsi



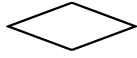


	<i>Ekstensi/extend</i> Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan.
	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.
	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya.

Sumber: Shalahuddin dan Rosa (2013)

2. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 4.3 Simbol *Activity Diagram*


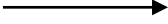

Simbol	Deskripsi
	Status awal aktivitas pada sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

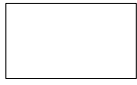

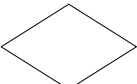
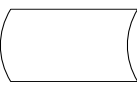

Sumber: Shalahuddin dan Rosa (2013)

3. Flowchart

Menurut Rahardja, dkk (2016 : 43) *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program.

Tabel 4.4 Simbol *Flowchart*

No	Simbol	Keterangan
1		<i>Terminator</i> untuk permulaan (<i>star</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
2		<i>Flow Line</i> adalah garis alur program
3		<i>Manual Operation</i> menjukan pengolahan yang tidak dilakukan oleh computer.

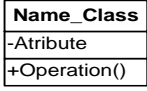


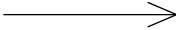



No	Simbol	Keterangan
4		<i>Processing</i> menunjukkan pengolahan yang Dilakuan oleh computer.
5		<i>Documen</i> Dokumen
6		<i>Decision</i> pemilihan proses berdasarkan kondisi yang ada.
7		<i>Storet Data</i> berfungsi untuk menyatakan input yang berasal dari disk atau disimpan kedisk
8		<i>Display</i> adalah untuk menyatakan <i>output</i> peralatan yang digunakan untuk layar, <i>plotter</i> , <i>pirinter</i> , dan <i>sebagainya</i>

Sumber: Taupik, dkk (2013:38)

4.5.1.2. Model Data

Class diagram adalah inti *forwardengineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model.

Tabel 4.5 Simbol *Class Diagram*

Simbol	Deskripsi
	Kelas pada struktur <i>system</i>
	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
	Relasi antarkelas dengan makna kebergantungan antarkelas.
	Relasi antarkelas dengan makna semua-bagian.

Sumber: Shalahuddin dan Rosa (2013)

4.5.2. Teknik Pengembangan Sitem

4.5.2.1. *Extreme Programming*

Menurut fatoni, dkk (2016:17), *Extreme programming* adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih *efisien*, *adaptif* dan *fleksibel*. Nilai dasar metode *extreme programming* :

1. *Communication*

Memfokuskan komunikasi yang baik antara *programmer* dengan *user* maupun antar *programmer*.

2. *Courage*

Pengembang perangkat lunak harus selalu memiliki keyakinan, keberanian dan integritas dalam melakukan tugasnya.

3. *Simplicity*

Lakukan semua dengan sederhana.

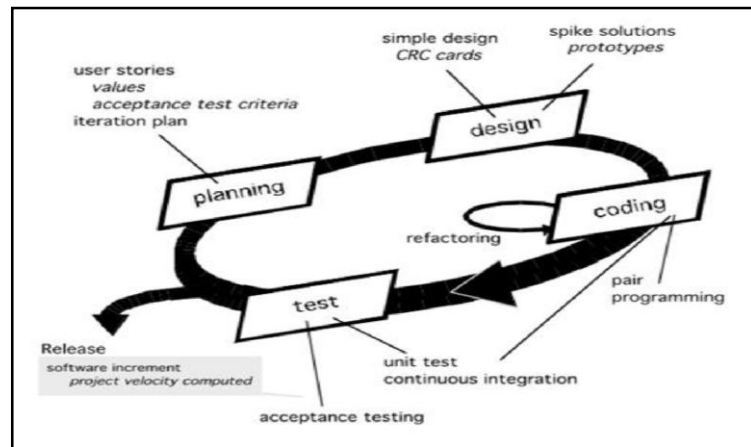
4. *Feedback*

Mengandalkan *feedback* sehingga dibutuhkan anggota tim yang berkualitas.

5. *Quality Work*

Proses berkualitas berimplikasi pada perangkat lunak yang berkualitas sebagai hasil akhirnya.

Menurut Andhika dalam jurnal Hamadi, Lumenta dan Putro (2017:2) Metode *extreme programming* merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat, karena sifat dari aplikasi yang di kembangkan dengan cepat. Metode *Extreme Programming* mempunyai 4 proses tahapan yang dapat dilihat pada gambar 4.1.:



Gambar 4.1 *Extreme Programming*

1. Perencanaan (*Planning*)

Pada tahap perencanaan ini dimulai dari pengumpulan informasi yang dibutuhkan untuk aplikasi game yang akan dibuat agar pengembangan mengerti bisnis konten dari perangkat lunak yang akan dikembangkan.

2. Perancangan (*Design*)

Perancangan XP dengan ketat mengikuti prinsip “tetap sederhana”. Sebuah hasil perancangan yang sederhana selalu lebih disukai dari pada gambaran yang lebih kompleks. Selain itu, perancangan XP akan memberikan panduan implementasi untuk suatu cerita ketika cerita itu ditulis, tidak kurang, tidak lebih. Rancangan dan fungsionalitas-fungsionalitas tambahan tidak terlalu disarankan.

3. Pengkodean (*Coding*)

Tahapan pengkodean ada pada *extreme programming* adalah *pair programming*, melibatkan lebih dari satu orang untuk menyusun kode. Kode ini bertujuan agar para pengembangan dapat berkonsentrasi pada apa yang akan diimplementasikan.

4. Pengujian (*Test*)

Tahapan yang terakhir ialah pengujian, ditahap ini lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi. Pengujian ini dilakukan untuk mengetahui apakah aplikasi dapat berjalan sesuai harapan untuk dicapai.

4.6. Alat dan Teknik Pengujian

Menurut Selfina Pare (2013:226), *Black Box* Metode ujicoba *black box* memfokuskan pada keperluan fungsional dari sistem, karena itu ujicoba *black box* memungkinkan pengembang sistem atau aplikasi untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *black box* bukan merupakan alternatif dari ujicoba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*.

Ujicoba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya; fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database eksternal* (jika ada), kesalahan performa, serta kesalahan inisialisasi dan terminasi.