

## **BAB IV**

### **METODE PENELITIAN**

#### **4.1. Lokasi dan Waktu Penelitian**

##### **4.1.1. Lokasi**

Tempat penelitian dilakukan pada Unit Pelayanan Teknis Dinas Balai Perbanyak Benih Tanaman Pangan dan Hortikultura Provinsi Sumatera Selatan di Jalan Residen Najamudin No.2 Palembang Sumatera Selatan 30961 Telp.(0711) 814872 .

##### **4.1.2. Waktu Penelitian**

Waktu penelitian berlangsung selama kurang lebih 2 bulan, dimulai dari 27 februari 2018 sampai minggu ke 2 bulan april 2018.

#### **4.2. Jenis Data**

##### **4.2.1. Data Primer**

Menurut Setyaningsih ( 2013:78), Data primer yaitu data yang diambil secara langsung dari obyek penelitian. Data primer diperoleh dengan cara mengamati secara langsung objek penelitian. Selain itu data primer dapat diperoleh dari hasil observasi dan wawancara sehinggadidapat informasi sesuai dengan kondisi fakta yang ada.

#### 4.2.2. Data Sekunder

Menurut Setyaningsih (2013:78), yaitu data-data yang diperoleh melalui literatur dan buku bacaan lainnya yang berhubungan dengan penelitian. Data sekunder dapat diperoleh dari referensi yang berasal dari berbagai sumber. Sumber data seperti perpustakaan, internet, literatur-literatur, atau sumber-sumber lain yang berkaitan dengan penelitian.

#### 4.3. Teknik Pengumpulan Data

##### 1) Wawancara

Menurut Hikmawati (2017:83) wawancara merupakan pertemuan dua orang untuk bertukar informasi dan ide melalui tanya jawab, sehingga dapat dikonstruksikan makna dalam satu topik tertentu.

Wawancara dilakukan kepada Tata Usaha yaitu, Bapak Lilik Rojali S.E, hal ini dilakukan untuk memperoleh data dan informasi. metode yang akan digunakan metode *Distribution Requirement Planing* dalam sistem pendistribusiannya, metode pengembangan sistem dan data yang digunakan untuk melakukan evaluasi menggunakan metode *Extreme Programing*.

Dalam wawancara ini penulis melakukan tanya jawab tentang UPTD-BPBTPH Provinsi Sumatera Selatan dan hal apa yang bisa diangkat untuk dijadikan penelitian, solusi apa yang akan

diberikan penulis dalam penelitian dan sistem apa yang dibutuhkan serta seberapa penting sistem tersebut untuk UPTD-BPBTPH Provinsi Sumatera Selatan. hasil wawancara penulis mendapatkan sistem yang sedang berjalan dalam hal pendistribusian berupa data produk atau tanaman apa saja yang ada di UPTD-BPBTPH Provinsi Sumatera Selatan dan BBI.

## **2) Observasi**

Menurut Kaelan (2012:101), Observasi adalah suatu pengamatan terhadap objek yang diteliti baik secara langsung maupun secara tidak langsung, untuk memperoleh data yang harus dikumpulkan dalam penelitian. Secara langsung artinya peneliti terjun ke lapangan dan mengamatinya, adapun secara tidak langsung pengamatan dapat melalui alat bantu seperti audio, kamera dan sebagainya. Penulis melakukan pengamatan dengan cara mengamati objek yang berhubungan dengan UPTD Balai Perbanyak Benih Tanaman Pangan dan Hortikultura provinsi Sumatera Selatan.

## **3) Dokumentasi**

Menurut Riduwan (2010:77), Dokumentasi adalah tujuan untuk memperoleh data langsung dari tempat penelitian, meliputi buku-buku *relevan*, laporan kegiatan dan foto-foto yang menjadi pendukung dalam penelitian.

Penulis melakukan dokumentasi dengan mendapatkan data-data berupa *file* dan laporan data UPTD Balai Perbanyak Benih Tanaman Pangan dan Hortikultura provinsi Sumatera Selatan.

#### **4) Studi Pustaka**

Menurut Permana (2017:110) Studi Pustaka adalah Melakukan penelaahan sumber referensi, buku, literatur dan jurnal untuk mendapat banyak bahan masukan tentang bagaimana merancang atau mengembangkan suatu sistem informasi yang terkait dengan pembahasan penelitian. Pada proses ini penulis mengumpulkan data yang berhubungan dengan pendistribusian data yang bersumber dari referensi buku, dan jurnal resmi, metodologi penelitian yang digunakan dan pengertian para ahli.

### **4.4. Jenis Penelitian**

#### **4.4.1. Penelitian Deskriptif**

Menurut Suryabrata (2012:76), Penelitian deskriptif adalah penelitian yang bermaksud untuk membuat pencandraan (deskripsi) mengenai situasi-situasi atau kejadian-kejadian. Penelitian ini digunakan untuk memecahkan atau menjawab permasalahan yang sedang di hadapi pada situasi sekarang. Penelitian deskriptif juga berarti penelitian yang dimaksudkan untuk menjelaskan fenomena atau karakteristik individual, situasi atau kelompok tertentu secara akurat.

## 4.5. Alat dan Teknik Pengembangan Sistem

### 4.5.1. Alat Pengembangan Sistem

#### 4.5.1.1. *Unified Modelling Language (UML)*

Menurut Salisah (2016:40), *Unified Modelling Language (UML)* adalah “bahasa” standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML merupakan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam Bahasa pemrograman apapun. UML lebih cocok untuk penulisan piranti lunak dalam Bahasa bahasa berorientasi objek seperti C++, Java, C# atau VB.

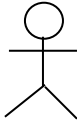
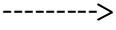
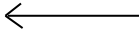
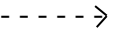
### 1. Pemodelan Proses

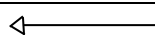
#### 1) *Use Case Diagram*


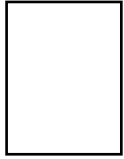



Menurut Salisah (2016:40) *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama

sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. Simbol-simbol *use case diagram* dapat dilihat pada tabel 4.1.

**Tabel 4.1. Simbol Use Case Diagram**

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifiakan himpunan peran yang pengguna mainkan ketika berintraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri ( <i>independent</i> ).
3.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
			Menspesifikasikan bahwa <i>use case</i> target memperluas








No	Gambar	Nama	Keterangan
5.		<i>Extend</i>	perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8..		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi actor.
9.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebihh besar dari jumlah dan elemen-elemennya.
10.		<i>Note</i>	Elemen yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sumber: Salisah (2016)

## 2) *Activity Diagram*

Menurut Salisah (2016:40) *Activity diagram* menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang terjadi pada beberapa eksekusi. Simbol-simbol *activity diagram* dapat dilihat pada tabel 4.2.

**Tabel 4.2. Simbol *Activity Diagram***

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berintraksi satu sama lain.
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi



No	Gambar	Nama	Keterangan
			beberapa aliran.


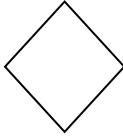
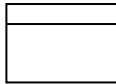

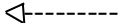
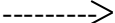
Sumber: Salisah (2016)


## 2. Pemodelan Data

### 1) *Class Diagram*

Menurut Salisah (2016:41) *Class diagram* akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class Diagram* menunjukkan interaksi antara kelas dalam sistem. Kelas mengandung informasi dan tingkah laku (*behavior*) yang berkaitan dengan informasi tersebut. Sebuah kelas pada diagram kelas dibuat untuk setiap tipe objek pada diagram sekuensial atau diagram kolaborasi. Simbol-simbol *class diagram* dapat dilihat pada tabel 4.3.

**Tabel 4.3. Simbol *Class Diagram***

No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri

No	Gambar	Nama	Keterangan
			( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
7.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Sumber: Salisah (2016)

## 4.5.2. Teknik Pengembangan Sistem

### 4.5.2.1. *Extreme Programming (XP)*

Menurut Mauludin (2017:16) *Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan-perubahan yang dibutuhkan sangat cepat.

Proses *Extreme Programming (XP)* memiliki 4 langkah yaitu sebagai berikut:

Keterangan gambar:

### 1. *Planning* (Perencanaan)

Pada tahap perencanaan ini dimulai dari pengumpulan informasi yang dibutuhkan untuk sistem yang akan dibuat, seperti:

a. Menganalisa dan mempelajari sistem yang sedang berjalan dan mengidentifikasi masalah serta pemecahannya.

b. Pengumpulan data.

1. Wawancara, mengumpulkan data dan informasi dengan melakukan proses tanya jawab dengan bapak Lilik Rojali selaku kepala TU (Tata Usaha) di UPTD-BPBTBH Provinsi Sumatera Selatan.

2. Observasi, penulis mengumpulkan data dengan melakukan pengamatan langsung ke UPTD-BPBTBH Provinsi Sumatera Selatan.

### 2. *Design* (Perancangan)

Pada tahap ini penulis melakukan proses perancangan, seperti merancang *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *Entity Relationship Diagram* (ERD), basis data *Database* dan perancangan *Interface*.

### 3. *Coding* (Pengkodean)

Tahapan pengkodean penulis menggunakan *Framework CodeIgniter*.

#### 4. *Testing* (Pengujian)

Pada tahap ini lebih fokus pada pengujian fitur dan fungsionalitas dari system, penulis menggunakan *BlackBox Testing*, dimana pengujian dilakukan terhadap fungsionalitas *input/output* dari suatu perangkat lunak.

### 4.6. Alat dan Teknik Pengujian Sistem

#### 4.6.1. *Black Box Testing*

Menurut Mustaqbal (2015:34), *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

*Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.