

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini berisikan teori-teori pendukung penelitian yang didapatkan dari hasil studi literatur dan jurnal-jurnal yang penulis dapatkan. Setiap teori dalam bab ini akan menjadi landasa dalam perancangan dan implementasi sistem yang akan penulis buat.

#### **2.1 Pengertian – pengertian**

##### **2.1.1 Pengertian Perancangan**

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi perancangan sistem yang dapat dirancang dalam bentuk bangan alir sistem (*system flowchart*), yang merupakan alat bentuk grafis yang dapat digunakan untuk menunjukkan urutan – urutan proses dari system. <sup>13</sup>

##### **2.1.2 Pengertian Sistem**

Perancangan Sistem dapat didefinisikan sebagai penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi <sup>1</sup>.

##### **2.1.3 Pengertian Informasi**

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan <sup>2</sup>

##### **2.1.4 Sistem Informasi**

Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahantransaksi harian, mendukung operasi, bersifat manajerial dan

kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Jogiyanto.<sup>1</sup>

### **2.1.5 Pengertian Kelurahan**

Kelurahan adalah pembagian wilayah administratif di Indonesia setelah kecamatan. Kelurahan merupakan wilayah kerja lurah sebagai perangkat daerah kota. Kelurahan dipimpin oleh seorang lurah yang berstatus sebagai Pegawai Negeri Sipil. Kelurahan merupakan unit pemerintahan terkecil setingkat dengan desa.<sup>10</sup>

## **2.2 Teori Perancangan Sistem**

Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Tahap ini menyangkut mengkonfigurasi dari komponen - komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancangan bangun yang telah ditetapkan pada akhir tahap analisa sistem".<sup>1</sup>

### **2.2.1 Prinsip Dalam Perancangan Sistem**

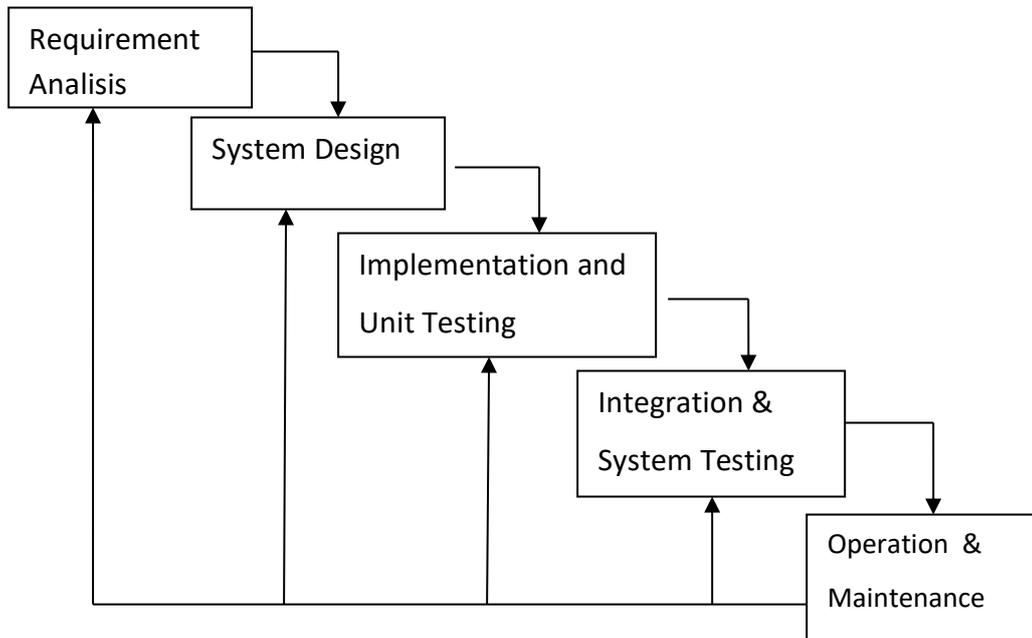
Beberapa prinsip dalam proses perancangan sistem adalah sebagai berikut:

1. Sistem yang dirancang adalah untuk manajemen.
2. Sistem yang dirancang memerlukan orang terdidik.
3. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses perancangan sistem.
4. Proses perancangan sistem terstruktur
5. Jangan takut membatalkan proyek.
6. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem.

### **2.2.2 Metode Waterfall**

Metode air terjun atau yang sering disebut metode waterfall sering dinamakan siklus – siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahap – tahapan perencanaan (*planing*),

permodelan (*modeling*), konstruksi (*deployment*) yang diakhiri dengan dukungan pada perangkat lunak lengkap<sup>3</sup>.



**Gambar 2.1 Tahapan Metode Waterfall**

Tahapan – tahapan dalam metode waterfall sebagai berikut :

### 1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

### 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini merealisasikan kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

### 3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### 4. Pengujian

Pengujian fokus pada perangkat lunak dari segi logic dan fungsional dan memastikan bahwasemua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak sudah ada, tapi tidak untuk membuat perangkat lunak baru.

## 2.2.3 Tahapan Perancangan Sistem

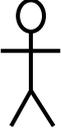
Tahapan perancangan system menggunakan bahasa pemodelan UML. Pengertian Bahasa Pemodelan Pengembangan Sistem (*UML*) Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Beberapa jenis diagram *UML* antara lain sebagai berikut: <sup>19</sup>

### 1. *Use Case Diagram*

*Use case* diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case*

mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat, simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.1 di bawah ini: <sup>19</sup>

**Tabel 2.1 Simbol Diagram Use Case**

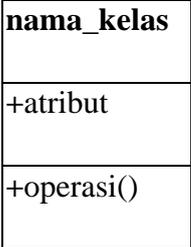
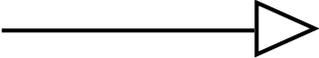
Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i></p>
<p>Asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i></p>
<p>Ekstensi/<i>extend</i></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan</p>

Simbol	Deskripsi
<<extend>> 	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> <<include>> 	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

## 2. *Class Diagram*

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 atau operasi,[8] simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.2 di bawah ini:

Tabel 2. 2 Simbol Class Diagram

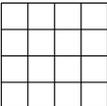
Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka/ <i>Interface</i>  <b>nama_interface</b>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Kebergantungan/ <i>dependecy</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/ <i>agregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

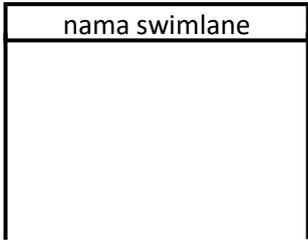
Sumber <sup>19</sup>

### 3. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem,<sup>19</sup> simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.4 di bawah ini :

**Tabel 2. 3 Simbol Activity Diagram**

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Simbol	Deskripsi
<p><i>Swimlane</i></p> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

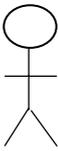
Sumber <sup>19</sup>

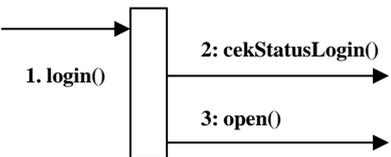
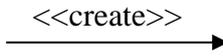
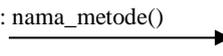
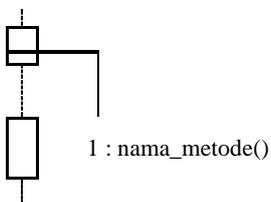
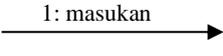
#### 4. Sequence Diagram

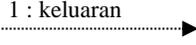
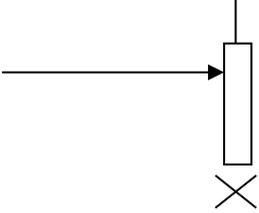
Sequence Diagram adalah diagram interaksi yang merinci bagaimana operasi dilakukan. Mereka menangkap interaksi antar objek dalam konteks kolaborasi. Sequence Diagram adalah fokus waktu dan mereka menunjukkan urutan interaksi secara visual dengan menggunakan sumbu vertikal diagram untuk mewakili waktu pesan apa yang dikirim dan kapan.

19

**Tabel 2. 4 Simbol Sequence Diagram**

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama aktor</p> <p>Atau</p>  <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata bendadi awal frase nama aktor.
Simbol	Deskripsi
<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.

<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>Nama objek : nama kelas</p> </div>	<p>Menyatakan objek yang berinteraksipesan.</p>
<p>Waktu aktif</p> <div style="border: 1px solid black; width: 20px; height: 60px; margin: 10px auto;"></div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe create</p> <div style="margin: 10px auto;"> <p>&lt;&lt;create&gt;&gt;</p>  </div>	<p>Menyatakan suatu ibjek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <div style="margin: 10px auto;"> <p>1: nama_metode()</p>  </div>	<p>Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan objek yang berinteraksi.</p>
<p>Pesan tipe send</p> <div style="margin: 10px auto;"> <p>1: masukan</p>  </div>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>

<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>

Sumber <sup>19</sup>

## 5. Data Kamus

Kamus data adalah penjelasan tertulis lengkap dari data yang diisikan ke dalam database.<sup>7</sup>

### 2.2.4 Pengertian MySQL

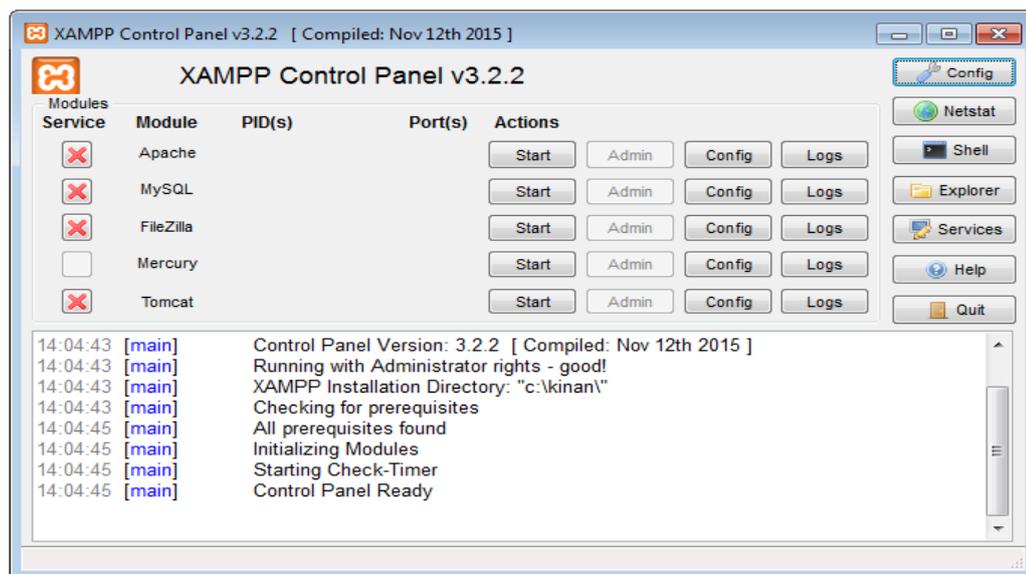
MySQL adalah software yang digunakan untuk menyimpan data yang diperlukan oleh aplikasi database.<sup>3</sup>

### 2.2.5 Hypertext Preprocessor.(PHP)

Salah satu bahasa pemrograman skrip yang dirancang untuk membangun suatu aplikasi web.<sup>3</sup>

### 2.2.6 XAMPP

Menurut Yogi wicaksono (14) “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* dikomputer local”. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet.



**Gambar 2.2 XAMPP**

### 2.2.7 Notepad++

Notepad++ adalah suatu text editor yang berjalan pada Operating System(OS) Windows. Notepad++ disini menggunakan komponen-komponen *Scintilla* agar dapat menampilkan dan menyunting text dan berkas source code berbagai bahasa pemrograman. Notepad++ didistribusikan sebagai Free Software (gratis) Proyek ini dilayani oleh [Sourceforge.net](http://Sourceforge.net) dengan telah diunduh lebih dari 27 juta kali dan dua kali memenangkan penghargaan *SourceForge Community Choice Award for Best Developer Tool*.<sup>15</sup>

### **2.3 Penelitian Sebelumnya**

Pada setiap penelitian tentunya dibutuhkan sebuah landasan teori, guna untuk memperkuat atau mendukung metode dan analisis yang dilakukan penelitian yang akan dilakukan ini merujuk pada penelitian terdahulu antara lain:

1. Perancangan sistem informasi pendataan penduduk kelurahan sungai jering berbasis web dengan object oriented programming.<sup>12</sup>
2. Perancangan sistem informasi pelayanan penduduk berbasis website di rw 010 Kelurahan Keagungan Kecamatan Tamansari - Jakarta Bara.<sup>15</sup>
3. Sistem Informasi Pelayanan Masyarakat Secara Online Pada Kelurahan Sungai Raya.<sup>11</sup>