

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 1.1 Tinjauan Pustaka

Hutrianto (2014) meneliti sebuah sistem berbasis web yang berjudul “Sistem Informasi E-Musrenbang (Elektronik–Musyawarah Rencana Pembangunan) Pada Kantor BAPPEDA Musi Banyuasin Sumatera Selatan). Web tersebut diteliti dengan tujuan untuk memudahkan pelaksanaan musrenbang supaya lebih efektif dan efisien karena data akan tersimpan dalam basis data yang sewaktu-waktu dapat diperoleh dengan cepat dan mudah. Pada penelitian tersebut penulis menggunakan model *web engineering* yang artinya suatu proses yang digunakan untuk menciptakan suatu sistem aplikasi berbasis web dengan menggunakan prinsip-prinsip manajemen dan pendekatan sistematis.

Dwi Purnomo (2013) merancang sebuah sistem berbasis web yang berjudul “Kelurahanin Sistem Informasi Pengolahan Data Musrenbang Badan Perencanaan Pembangunan Daerah”. Sistem tersebut dirancang dengan tujuan untuk mengolah data musyawarah rencana pembangunan daerah sehingga mempercepat proses pembuatan laporan musrenbang mulai dari tingkat kecamatan sampai dengan Bappeda sebagai pengguna akhir dari sistem.

Berdasarkan pengujian dan analisis yang telah dilakukan oleh masing-masing peneliti, kedua penelitian tersebut membangun aplikasi perancangan Musrenbang berbasis web untuk mendukung kinerja pemerintahan daerah agar lebih efektif dan efisien.

Namun dalam kedua penelitian tersebut terdapat kekurangan yaitu laporan masih di cetak dalam bentuk gambar dan pdf, hal tersebut membuat kinerja admin menjadi lama dan

tidak efektif apabila membutuhkan laporan dalam bentuk dokumen. Kelebihan dari Aplikasi Musrenbang ini adalah dapat mengekspor data dalam bentuk file excel sehingga memudahkan dalam penyusunan laporan.

## **1.2 Landasan Teori**

### **1.2.1. Musrenbang**

Musrenbang atau Musyawarah Perencanaan Pembangunan adalah forum antar pelaku dalam rangka menyusun rencana pembangunan nasional dan rencana pembangunan daerah (). Dengan memanfaatkan teknologi informasi pelaksanaan musrenbang akan lebih efektif dan efisien karena data akan tersimpan dalam basis data yang sewaktu-waktu dapat diperoleh dengan cepat dan mudah.

### **1.2.2. Musrenbang Kelurahan Kricak**

Musrenbang Kelurahan Kricak adalah forum musyawarah tahunan masyarakat kelurahan Kricak yang dilaksanakan secara partisipatif oleh para pemangku kepentingan (*stakeholders*) kelurahan untuk menyepakati rencana kegiatan pembangunan pada tahun anggaran berikutnya. Musrenbang Kelurahan Kricak diharapkan menghasilkan formulasi usulan kebutuhan pembangunan yang terpadu yang didalamnya tidak saja memvalidasi kebutuhan pembangunan hasil musrenbang tahun sebelumnya yang belum tertangani serta mengakomodasi kebutuhan-kebutuhan pembangunan tahun yang akan datang dari masing-masing RT/RW di kelurahan yang bersangkutan, tetapi juga kebutuhan-kebutuhan lintas RW antar dua kelurahan yang berbeda yang harus diusulkan lebih lanjut ke Musrenbang tingkat Kecamatan.

Pada Musrenbang Kelurahan Kricak ditentukan prioritas usulan program/ kegiatan dari masing-masing RT/RW serta pengklasifikasian usulan program/ kegiatan baik dari sifat kegiatannya (fisik dan non fisik) dan sumber-sumber pembiayaannya (APBD, APBD Propinsi maupun APBN) serta kegiatan-kegiatan lain yang sesuai (PKMK dan sebagainya).

Musrenbang Kricak memiliki beberapa bidang usulan diantaranya:

- a. Ekonomi
- b. Sosial dan budaya
- c. Fisik

Musrenbang kelurahan Kricak memiliki beberapa tujuan yaitu :

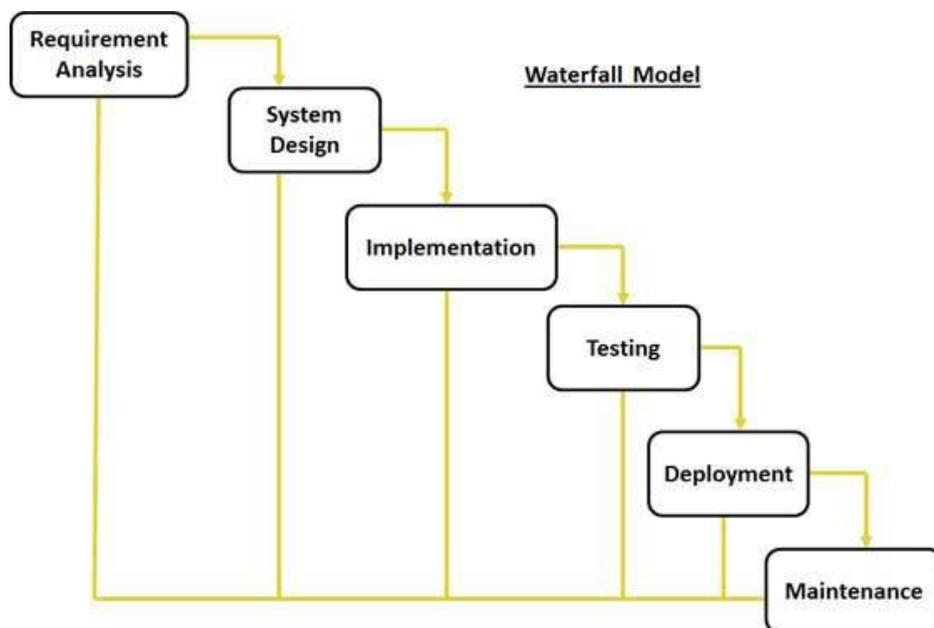
1. Menampung dan membahas usulan kegiatan prioritas RW yang diperoleh dari musyawarah perencanaan RW;
2. Menetapkan kegiatan prioritas pembangunan yang akan dibiayai melalui alokasi dana Kelurahan yang berasal dari APBD Kota maupun sumber pendanaan lainnya;
3. Menetapkan usulan kegiatan prioritas kelurahan yang akan diajukan dan dibahas pada forum Musrenbang Kecamatan;
4. Menetapkan wakil/delegasi kelurahan yang akan mengikuti Musrenbang Tingkat Kecamatan.

### **2.2.3 Software Development Life Cycle**

Dalam penelitian ini, metode yang digunakan adalah memakai model SDLC (Software Development Life Cycle). Menurut Al Fatta, Hanif (2007:25), System SDLC

merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan Kelurahanin.

Menurut Sommerville (2011:30-31), tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing, dan operation and maintenance. *Waterfall model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE).



**Gambar 2. 1** *Waterfall model*

Berikut adalah penjelasan dari tahapan-tahapan tersebut :

**1) Requirement Analysis and Definition**

Merupakan tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

**2) System and Software Design**

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.

**3) Implementation and Unit Testing**

Dalam tahapan ini, hasil dari Keluruhanin perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

**4) Integration and System Testing**

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

**5) Operation and Maintenance**

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru. Dengan memakai metode *Waterfall* ini tahap demi tahap harus dilakukan secara beruntun.

#### **2.2.4 *Unified Modeling Language (UML)***

Menurut Whitten dan Bentley (2007), UML (Unified Modeling Language) adalah seperangkat konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam hal objek. Model UML yang dipakai dalam pengembangan aplikasi antara lain adalah Use Case Diagram, Activity Diagram dan Class Diagram.

#### **2.2.5 *Use Case Diagram***

Menurut Whitten dan Bentley (2007), *use-case diagram* adalah sebuah diagram yang menggambarkan interaksi antara sistem dan sistem eksternal dan pengguna. dengan kata lain, grafis yang menggambarkan siapa yang akan menggunakan sistem dan dalam cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Simbol-simbol yang digunakan *use case diagram* dapat dilihat pada Tabel 2.1

**Tabel 2. 1** Simbol-simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<p><i>Actor</i></p>	<p>Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>.</p>
2		<p><i>Dependency</i></p>	<p>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).</p>

No	Gambar	Nama	Keterangan
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> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

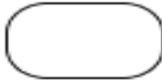
No	Gambar	Nama	Keterangan
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

No	Gambar	Nama	Keterangan
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

### 2.2.6 Activity Diagram

Menurut Whitten dan Bentley (2007), *activity diagram* adalah diagram yang dapat digunakan untuk grafis menggambarkan aliran proses bisnis, langkah-langkah dari kasus penggunaan, atau logika perilaku objek. Simbol-simbol yang digunakan *activity diagram* dapat dilihat pada Tabel 2.2.

**Tabel 2. 2** Simbol-simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Node Awal untuk memulai sebuah proses activity.
4		<i>Activity Final Node</i>	Node akhir untuk mengakhiri sebuah proses activiti.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

### 2.2.7 *Microsoft SQL Server*

Menurut Connolly dan Begg (2010, p113), SQL merupakan bahasa yang dirancang untuk menggunakan relasi untuk mengubah masukan menjadi keluaran yang diharapkan.

Menurut O'Brien dan Marakas (2010, p148), SQL adalah bahasa query yang ditemukan di berbagai paket manajemen *database*. SQL dimaksudkan untuk memenuhi keputusan berikut :

- a. Membuat *database* dan struktur relasi;
- b. Melakukan tugas dasar manajemen data, seperti pemasukan, modifikasi dan penghapusan data dari relasi;
- c. Melakukan query sederhana dan kompleks. Standar SQL memiliki dua komponen
  - Data Definition Language (DDL) untuk menetapkan struktur *database* dan mengontrol akses ke data;
  - Data Manipulation Language (DML) untuk mendapatkan kembali (retrieve) dan memperbaharui data.

### 2.2.8 *Netbeans*

Menurut Nishom (2012), Netbeans merupakan sebuah aplikasi *Integrated Development Environment (IDE)* yang berbasiskan Java dari *Sun Microsystems* yang berjalan di atas *swing*.

### 2.2.9 *Java Server Pages (JSP)*

*Java Server Pages* pertama kali diperkenalkan pada tahun 1999 oleh SUN Microsystems. Menurut Chopra et al. (2001, p34), *Java Server Pages (JSP)* sendiri adalah teknologi berbasis Java yang dijalankan pada server untuk memfasilitasi pemrosesan web berbasis request, untuk memformat dan menampilkan data yang ingin

anda lihat. Semenjak itu mereka menyadari bahwa tag tambahan akan berguna dan JSP Standard Tag Library (JSTL) lahir. JSTL adalah koleksi perpustakaan tag kustom yang merangkum fungsionalitas dari banyak standar JSP, sehingga menghilangkan pengulangan dan membuat aplikasi lebih kompak.

#### **2.2.10 *Server Glassfish***

*Server glassfish* adalah aplikasi proyek open-source dimulai oleh Sun Microsystems untuk platform Java EE dan sekarang disponsori oleh Oracle Corporation. Versi ini didukung disebut Oracle GlassFish Server. GlassFish adalah perangkat lunak bebas, dual-berlisensi di bawah dua lisensi perangkat lunak bebas: Pembangunan Umum dan Lisensi Distribusi (CDDL) dan GNU General Public License (GPL) dengan pengecualian classpath. GlassFish adalah implementasi referensi Java EE. Hal ini memungkinkan pengembang untuk membuat aplikasi enterprise yang portabel dan terukur, dan yang terintegrasi dengan teknologi klien-server. Komponen opsional juga dapat diinstal untuk layanan tambahan.

#### **2.2.11 *CSS (Cascading Style Sheet )***

Menurut Shalahuddin dan A.S Rosa (2010), *CSS atau Cascading Style Sheets* adalah suatu fasilitas untuk mempermudah pemeliharaan sebuah halaman web, dengan menggunakan *CSS* sebuah halaman web dapat diubah tampilannya tanpa harus mengubah dokumen *HTML-nya*.

#### **2.2.12 *JavaScript***

Menurut Shalahuddin dan A.S Rosa (2010), *JavaScript* adalah bahasa script (bahasa pemrograman yang dapat memegang kontrol aplikasi) yang berbasis pada pemrograman *Java*, namun *JavaScript* bukanlah bagian teknologi *Java* dari Sun. *Java*

merupakan bahasa pemrograman berorientasi objek murni sedangkan *JavaScript* digunakan secara prosedural.