

## **BAB III**

### **METODE PENELITIAN**

#### **3.1. Model Pengembangan Perangkat Lunak**

Tahapan Pengumpulan Lunak, dalam penelitian ini dilakukan dengan menggunakan metode yang telah di pilih, yaitu Metode Prototype. Proses ini juga dilakukan secara intensif untuk menspesifikasi kebutuhan perangkat lunak agar dapat memenuhi keinginan *User* (Pengguna). Maka dari itu dalam proses ini diperlukan pengumpulan data sebagai Berikut

##### **3.1.1 Komunikasi (*Communication*)**

Komunikasi memegang peranan yang sangat penting dalam mendapatkan Informasi dari pengguna . Komunikasi harus dilakukan dengan cara yang tepat. Data Objektif dan Relevan dengan pokok pembahasan menjadi Indikator Keberhasilan suatu penelitian.

Setelah mendapatkan informasi yang cukup, langkah selanjutnya adalah analisis kebutuhan untuk menentukan kebutuhan yang spesifik untuk mengembangkan perangkat lunak.

##### **3.1.1.1 Perencanaan (*Planning*)**

Tahapan perencanaan memudahkan pengembang dalam menyusun jadwal- jadwal kerja. Jadwal-jadwal kerja yang harus disusun meliputi waktu yang dibutuhkan untuk menyusun dan menganalisis kebutuhan, mengembangkan produk, hingga pengujian produk. Jadwal-jadwal pengembangan dibuat menggunakan tabel yang berisi progres pengerjaan proyek dan durasi pengerjaan.

##### **3.1.1.2 Pemodelan (*Modelling*)**

Langkah awal yang dilakukan dalam tahap modeling adalah membuat desain *User Experience* (UX) dan *User Interface* (UI). Untuk merancang UI dilakukan dengan membuat *prototype* tampilan perangkat lunak dan menu-menu yang ada di perangkat lunak yang sesuai dengan analisis kebutuhan. Sedangkan untuk UX dirancang menggunakan diagram *UML* yang bertujuan menggambarkan alur sistem yang akan dikembangkan.

### **3.1.1.3 Konstruksi (Construction)**

Kegiatan ini merupakan inti dari tahap pengembangan perangkat lunak. Tahap ini mengatur semua logika dan arsitektur sistem dengan menggunakan sebuah bahasa pemrograman. Perintah-perintah akan ditulis dan digabungkan sesuai dengan fungsinya masing-masing. Dalam tahap ini dilakukan pengujian menggunakan standar internasional *ISO/IEC 25010*. Pengujian bertujuan untuk menemukan kekeliruan-kekeliruan/kesalahan-kesalahan pada perangkat lunak. Sehingga, pada tahapan ini menjalankan dua kegiatan yaitu membangun perangkat lunak dan melakukan pengujian.

## **3.2 Metode Pengumpulan Data**

Untuk mengumpulkan data dalam penelitian ini dilakukan dengan beberapa cara yaitu observasi, wawancara, dan kuesioner (angket). Tujuan dari pengumpulan data adalah untuk menentukan kebutuhan apa saja yang diperlukan dalam sistem dan digunakan untuk keperluan pengujian dari sisi pengguna.

Berikut penjelasan metode yang digunakan untuk mengumpulkan data.

### **3.2.1 Observasi**

Metode Pengumpulan data ini dilakukan dengan cara melakukan pengamatan langsung pada Dinas Perumahan Kawasan Permukiman dan Cipta Karya Provinsi Lampung.

### **3.2.2 Wawancara**

Metode ini dilakukan dengan Cara bertemu langsung dan melakukan proses Tanya jawab dengan Kepala Dinas dan Developer mengenai System yang akan di buat

## **2. Instrumen Reliability**

Web Application Load, Stress and Performance Testing (WAPT) melakukan simulasi dengan beberapa *virtual user* yang mengakses web secara simultan dalam

beberapa waktu tertentu, sehingga mendapatkan hasil berupa presentase keberhasilan eksekusi sebuah *website*.

### 3. Instrumen Maintainability

Perhitungan *Maintainability Index* (MI). Perhitungan MI berdasarkan pada perhitungan dari *McCabe's Cyclomatic Complexity* (CC), *Halstead's Volume* (V), dan *Lines of Code* (LOC) menggunakan *tool* PHPMetrics.

### 4. Instrumen Usability

Pengujian *Usability* digunakan untuk menilai apakah perangkat lunak yang dibuat mudah untuk dipahami, dipelajari, dan bisa membantu pengguna dalam menyelesaikan masalahnya. Pengujian ini menggunakan kuesioner *USE (Usefulness, Satisfaction, and Ease of use) Questionnaire* oleh Arnold M. Lund yang sudah mencakup subkarakteristik *usability ISO 25010*. Instrumen *USE Questionnaire* tersaji pada tabel 3.

Tabel 3. Instrumen Pengujian *Usability*

No	Pernyataan	Skala Penilaian				
		STS	TS	RG	S	SS
<b>USEFULNESS</b>						
1.	Software ini membantu saya bekerja lebih efektif					
2.	Software ini membantu saya bekerja lebih produktif					
3.	Software ini sangat berguna bagi penyelesaian					
<b>Dst...( Angket lengkap terlampir)</b>						

## G. Teknik Analisis Data

### 1. Analisis Pengujian Functional Suitability

Rumus yang digunakan untuk menghitung hasil dari pengujian *functional suitability*. Hasil dari pengujian ini adalah presentase kelayakan perangkat lunak. Berikut rumus perhitungannya sebagai berikut.

$$P_{\text{Kelayakan}}(\%) = \frac{S_{\text{gagal}}}{S_{\text{total}}} \times 100\%$$

$$P_{\text{Kelayakan}}(\%) = \frac{S_{\text{gagal}}}{S_{\text{total}}}$$

$$S_{\text{gagal}} = \dots$$

$$S_{\text{total}} = \dots$$

Setelah didapatkan presentase kelayakan, kemudian akan dikonversi sesuai dengan interpretasi yang sudah ditentukan. Sudaryono (2014) mengatakan bahwa hasil hitungan akan dikonversikan ke dalam pernyataan sesuai dengan interpretasi yang sudah ditentukan. Tabel konversi kelayakan tersaji pada tabel 4 berikut.

Tabel 4. Konversi Presentase Kelayakan

No	Presentase	Interpretasi
1	0% - 20%	Sangat Tidak Layak
2	21% - 40%	Tidak Layak
3	41% - 60%	Cukup Layak
4	61% - 80%	Layak
5	81% - 100%	Sangat Layak

## **2. Analisis Pengujian Reliability**

Pengujian reliability dilakukan dengan stress testing menggunakan WAPT. Guritno (2011) mengatakan bahwa presentase keberhasilan yang dihitung adalah

*sessions*, *pages*, dan *hits*. Rumus hitung presentase keberhasilan ditunjukkan

dibawah ini:

$S_{\text{keberhasilan}} = \frac{R}{ax} \times 100\%$

$$R = \frac{S_{\text{keberhasilan}}}{ax} \times 100\%$$

Dimana:

R = Reliability

Telcordia mengatakan bahwa standar presentase keberhasilan uji *reliability* adalah 95% (Asthana & Oliviera, 2009). Sehingga software dapat dikatakan lolos apabila uji pada karakteristik *reliability* minimal memiliki presentase sebesar 95% ketika uji *stress testing* menggunakan WAPT.

### 3. Analisis Pengujian Maintainability

Pengujian *maintainability* dilakukan dengan menghitung *Maintainability Index*

(MI). Perhitungan nilai MI dapat dihitung dengan rumus berikut:

$$MI = 171 - 5.2 * \ln(aveV) - 0.23 * aveV(g) - 16.2 * \ln(aveLOC)$$

(Najm, 2014)

**Keterangan:**

*MI* = *Maintainability Index*

*aveV* = Rata-rata *Hasted Volume*

*aveV(g)* = Rata-rata *Cyclomatic Complexity* setiap modul

*aveLOC* = Rata-rata *Line of Code* setiap modul



Untuk standar nilai yang ditentukan pada pengujian *Maintainability* adalah harus diatas 65. Perbandingan hasil perhitungan *Maintainability Index* tersaji dalam tabel 5 berikut.

Tabel 5. *Maintainability Index*

<b>Maintainability Index</b>	<b>Level</b>
80-100	<i>High Maintainable</i>
66-85	<i>Moderate Maintainable</i>
0-65	<i>Difficult to Maintain</i>

#### 4. Analisis Pengujian Usability

Pengujian *usability* menggunakan kuesioner menggunakan kuesioner *USE (Usefulness, Satisfaction, and Ease of use) Questionnaire* oleh Arnold M. Lund yang sudah mencakup subkarakteristik *usability ISO 25010*. Kuesioner akan dibagikan kepada 30 responden sebagai pengguna yang terdiri dari 29 siswa dan 1 petugas perpustakaan. Setiap jawaban yang dihasilkan akan diberi skor untuk dianalisis. Bobot skor jawaban tersaji pada tabel 6 berikut.

Tabel 6. Bobot Skor Jawaban

NO	Jawaban	Bobot Skor
1	Sangat Setuju (SS)	5
2	Setuju (S)	4
3	Kurang Setuju (KS)	3
4	Tidak Setuju (TS)	2
5	Sangat Tidak Setuju (STS)	1

Untuk mengetahui hasil dari pengujian *usability* maka jumlah rata-rata skor jawaban yang didapatkan akan dihitung (Sugiyono, 2012). Untuk menghitung jumlah skor total, setiap jawaban yang diperoleh dikalikan dengan bobot skor yang sudah ditentukan. Perhitungan dapat dilakukan dengan melihat rumus dibawah ini.

$$\text{Skortotal} = (\text{JSS} \times 5) + (\text{JS} \times 4) + (\text{JRG} \times 3) + (\text{JTS} \times 2) + (\text{JSTS} \times 1)$$

Keterangan:

JSS = jumlah responden menjawab Sangat Setuju

JS = jumlah responden menjawab Setuju

JRG = jumlah responden menjawab Ragu-Ragu

JTS = jumlah responden menjawab Tidak Setuju

JSTS = jumlah responden menjawab Sangat Tidak Setuju

Kemudian mencari skor untuk mendapatkan kriteria interpretasi skor hasil pengujian *usability* dengan rumus:

$$P \text{ \textasciitilde\textasciitilde\textasciitilde\textasciitilde} = \frac{S \text{ \textasciitilde\textasciitilde\textasciitilde\textasciitilde}}{i \text{ \textasciitilde\textasciitilde\textasciitilde\textasciitilde}} \times 100\%$$

Keterangan:

Skor total = skor total hasil jawaban responden

i = jumlah pertanyaan

$r =$  jumlah responden

Presentase kelayakan kemudian akan dikonversi dalam pernyataan sesuai dengan tabel konversi kelayakan yang tersaji pada tabel 4 diatas.

Setelah dihitung persentase kelayakannya kemudian kuesioner akan dihitung *cronbach's alpha*-nya. Tujuan menghitung *cronbach's alpha* adalah untuk mengetahui reliabilitas instrume. Gilem (2003) menyatakan bahwa perhitungan *Alpha Cronbach* dihitung menggunakan *software* SPSS dengan interpretasi nilai *Alpha Cronbach* yang tersaji pada Tabel 7 (Heru & Handaru, 2017).

<b><i>Cronch's Alpha</i></b>	<b><i>Internal Consistency</i></b>
$\alpha \geq .9$	<i>Excellent</i>
$\alpha \geq .9$	<i>Good</i>
$.8 > \alpha \geq .7$	<i>Acceptable</i>
$.7 \alpha \geq .6$	<i>Questionable</i>
$.5 > \alpha$	<i>Unacceptable</i>

Tabel 7. *Internal Consistency Cronbach's*

