

## BAB 2 LANDASAN TEORI

### 2.1 Rancang Bangun

**Definisi 1.** Rancang bangun adalah kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada [4].

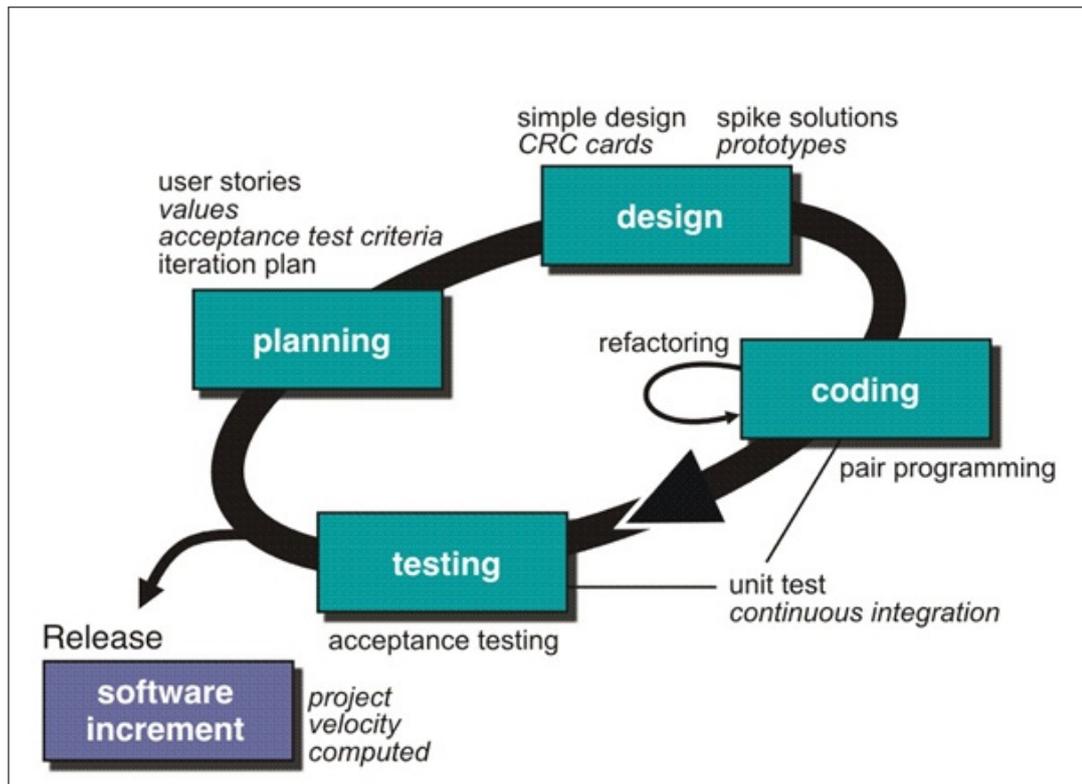
**Definisi 2.** Rancang Bangun adalah tahap awal dari membuat gambaran dan bentuk sketsa yang belum pernah dibuat sama sekali lalu dikelolah menjadi gambaran atau sketsa yang memiliki fungsi yang diinginkan [5].

### 2.2 Metode Extreme Programming

**Definisi 1.** Metode pengembangan *Extreme Programming* ini sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel. *Expert Programming* bukan hanya berfokus pada coding tetapi meliputi seluruh area pengembangan perangkat lunak. *Expert Programming* mengambil pendekatan 'ekstrim' dalam iterative development [1]. kelebihan dari penggunaan metode pengembangan *Expert Programming* dibandingkan dengan metode pengembangan lain adalah pembangunan sistem atau aplikasi dibuat lebih cepat, serta meminimalisir biaya yang diperlukan jika ada perubahan dalam pengembangan perangkat lunak.[1]

Terdapat empat kerangka kegiatan dalam tahapan pelaksanaan extreme programming. Empat kerangka kegiatan dari tahapan extreme programming adalah sebagai berikut.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.1. Kerangka Extreme Programming

Source: <https://repository.bsi.ac.id/index.php/unduh/item/254651/1043-2463-3-PB.pdf>

### 1. Planning

Pada tahapan digunakan untuk memahami konsep bisnis, pengumpulan kebutuhan sistem, menggambarkan output, fitur-fitur, dan fungsionalitas apa saja yang akan digunakan pada pengembangan *software* [6].

### 2. Design

Pada tahap ini data yang telah diperoleh dari tahap *Planning* dirancang menjadi bentuk UML sebagai bentuk pemodelan sistem yang sudah memiliki standar, tahap ini digunakan untuk memastikan *software* yang dibangun selesai dengan waktu yang tepat, sesuai anggaran, dan sesuai dengan spesifikasi yang ditentukan pada tahap *planning* [6].

### 3. Coding

Tahapan ini melakukan koding *software* yang menggunakan, tim kecil atau *personal person* akan bekerja secara bertahap dengan panduan alur sistem yang sudah dirancang pada tahap *design* modul per modul. Dengan menggunakan *refactoring* agar dapat mudah dibaca dan dimodifikasi, sehingga hasil yang diharapkan pada pengembangan *software* menjadi cepat [6].

#### 4. **Testing**

Tahap testing merupakan tahapan akhir dari metode ini yang dilakukan dengan cara beta test untuk mendapat feedback dari end-user yang telah melakukan pengujian fungsional terhadap software [6].

### 2.3 **Black Box Testing**

**Definisi** Black Box Testing merupakan pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji kode atau sisi internal program. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik [7].

Terdapat beberapa teknik dalam melakukan black box testing. Berikut merupakan teknik dalam melakukan black box testing[?]:

#### 1. **All pair testing**

Teknik all pair testing ini dikenal juga dengan pairwise testing. Pengujian ini digunakan untuk menguji semua kemungkinan kombinasi dari seluruh pasangan berdasarkan input parameternya.

#### 2. **Boundary value analysis**

Teknik ini berfokus pada pencarian error dari luar atau sisi dalam perangkat lunak.

#### 3. **Cause-effect graph**

Berikutnya adalah teknik cause-effect graph. Teknik pengujian ini menggunakan grafik sebagai patokannya. Grafik ini menggambarkan relasi antara efek dan penyebab dari error.

#### 4. **Equivalence partitioning**

Teknik ini bekerja dengan cara membagi data input dari beberapa perangkat lunak menjadi beberapa partisi data.

#### 5. **Fuzzing**

Fuzzing merupakan teknik pencarian bug dalam perangkat lunak dengan memasukan data yang tidak sempurna.

#### 6. **Orthogonal array testing**

Selanjutnya adalah orthogonal array testing. Teknik ini digunakan jika input berukuran kecil, akan tetapi cukup berat jika digunakan dalam skala yang besar.

## 7. State transition

Terakhir adalah state transition. Teknik ini berguna untuk melakukan pengujian terhadap mesin dan navigasi dari UI dalam bentuk grafik.

## 2.4 EUCS

**Definisi** End User Computing Satisfaction (EUCS) merupakan evaluasi secara keseluruhan dari pengguna sistem informasi yang berdasarkan pengalaman pengguna setelah menggunakan sistem tersebut. Model evaluasi EUCS lebih menekankan kepuasan pengguna akhir terhadap aspek teknologi, dengan menilai isi (Content), keakuratan (Accuracy), format, waktu (Timeliness), dan kemudahan dalam penggunaan (ease to use). Berikut merupakan penjelasan setiap aspek didalam EUCS :

### 1. Isi (Content)

Aspek isi atau kelengkapan informasi, merupakan aspek yang digunakan dalam melakukan pengukuran kepuasan dari sisi isi dalam sebuah konten aplikasi atau sistem. Konten dari aplikasi tersebut berupa fungsi atau modul yang didapatkan oleh pengguna aplikasi atau sistem serta informasi yang akan disampaikan dari aplikasi tersebut.

### 2. Akurasi (Accuracy)

Aspek akurasi, merupakan aspek yang digunakan sebagai pengukuran kepuasan dari sisi keakuratan data, ketika aplikasi atay sistem menerima sebuah masukan atau inputan, kemudian diolah menjadi sebuah informasi. Dalam aspek akurasi atau keakuratan aplikasi diukur dengan seberapa sering aplikasi atau sistem menghasilkan output yang salah ketika menerima masukan dari pengguna.

### 3. Format

Aspek format, yaitu aspek terkait penyajian informasi, dimana aspek ini mengukur kepuasan pengguna dari sisi tampilan dan antarmuka sistem atau aplikasi, apakah tampilan dari aplikasi atau sistem menarik dan mempermudah pengguna dalam menggunakan aplikasi yang telah dibuat.

### 4. Waktu (Timeliness)

Aspek waktu atau ketepatan waktu, pada aspek ini mengukur kepuasan

pengguna dari sisi ketepatan waktu ketika menyajikan data dan informasi yang dibutuhkan oleh pengguna.

#### 5. Kemudahan (Ease to Use)

Aspek kemudahan, maksud dari aspek kemudahan ini adalah aspek yang mengukur kepuasan pengguna dari kemudahan dalam menggunakan aplikasi atau sistem dalam melakukan proses memasukan data, pengolahan data.

### 2.5 Skala Likert

**Definisi** Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti [8]. Dalam penggunaan skala Likert, terdapat dua bentuk pertanyaan, yaitu bentuk pertanyaan positif untuk mengukur skala positif, dan bentuk pertanyaan negatif untuk mengukur skala negatif. Pertanyaan positif diberi skor 5, 4, 3, 2, dan 1; sedangkan bentuk pertanyaan negatif diberi skor 1, 2, 3, 4, dan 5 atau -2, -1, 0, 1, 2. Bentuk jawaban skala Likert antara lain: sangat setuju, setuju, ragu-ragu, tidak setuju, dan tidak setuju [8].

Berikut merupakan bentuk tabel dari skala likert [9].

Tabel 2.1. Tabel Skala Likert

Kategori	Skor
Sangat Tidak Setuju	1
Tidak Setuju	2
Sederhana Setuju	3
Setuju	4
Sangat Setuju	5

Source: [www.researchgate.net/publication/331429872](http://www.researchgate.net/publication/331429872)

Dengan rumus perhitungan skor pada skala likert berdasarkan dari jawaban responden.

$$\text{Persentase} = \frac{(STS * 1) + (TS * 2) + (N * 3) + (S * 4) + (SS * 5)}{5 * responden} * 100\% \quad (2.1)$$

Hasil dari pengukuran tadi, dikelompokkan sesuai dengan kriteria.

Tabel 2.2. Tabel interval Skala Likert

<b>Kategori</b>	<b>Interval</b>
Sangat Tidak Setuju	0% - 19.9%
Tidak Setuju	20% - 39.9%
Sederhana Setuju	40% - 59.9%
Setuju	60% - 79.9%
Sangat Setuju	80% - 100%

