

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Model Pembangunan Perangkat Lunak**

Model pengembangan produk yang digunakan dalam penelitian ini adalah penelitian dan pengembangan (*Research and Development*) model ADDIE yaitu *Analysis* (analisis), *Design* (desain atau perencanaan), *Development* (pengembangan), *Implementation* (implementasi) dan *Evaluation* (evaluasi) (Piskurich (dalam Gafur, 2012: 39)).

##### **3.1.1 Analisis Perangkat Lunak**

Analisis kebutuhan diperlukan dalam pengembangan aplikasi untuk membantu proses perancangan. Masukan atau input dari aplikasi yang akan dikembangkan adalah citra digital dengan format \*.jpg, \*.jpeg, dan \*.bmp. Pengolahan citra digunakan untuk mengenali citra sehingga dapat dimengerti oleh mesin untuk menentukan suatu keputusan. Aplikasi koreksi lembar jawaban komputer dengan pengolahan citra memiliki fungsi memasukan kunci jawaban, mengenali dan mengoreksi lembar jawaban, menyimpan hasil koreksi kedalam *database* sesuai identitas lembar jawaban. Keluaran atau output dari aplikasi ini adalah hasil koreksi yang diekspor dalam program pengolahan angka.

##### **3.1.2 Desain Perangkat Lunak**

Desain perangkat lunak meliputi basis data, arsitektur perangkat lunak, antarmuka dan algoritma. Tahap ini dikenal dengan istilah membuat rancangan (blue print), ibarat bangunan maka sebelum dibangun harus ada rancang bangun diatas kertas terlebih dahulu.

##### **3.1.3 Pengembangan Perangkat Lunak**

Merupakan proses mewujudkan blue print alias desain tadi menjadi kenyataan. Artinya pada tahap ini segala sesuatu yang dibutuhkan atau yang akan mendukung proses pengembangan perangkat lunak semuanya harus disiapkan. Pada tahap ini

terdapat proses pengkodean. Pengkodean bertujuan untuk menerjemahkan keperluan perangkat lunak kedalam bentuk sebenarnya yang dapat dimengerti oleh komputer. Dalam pengembangan aplikasi ini digunakan metode *thresholding* sebagai algoritma dalam *image processing*.

#### a. Algoritma *Grayscale*

##### 1. Lightness

Algoritmanya adalah mencari nilai tertinggi dan terendah dari nilai R G B, kemudian nilai tertinggi dan terendah tersebut dijumlahkan lantas dikalikan dengan 0.5. Secara matematis dapat dirumuskan:

$$\mathbf{Grayscale} = (\mathbf{max(R,G,B)}) + (\mathbf{min(R,G,B)}) * 0.5$$

##### 2. Average

Algoritmanya adalah dengan menjumlahkan seluruh nilai R G B, kemudian dibagi 3, sehingga diperoleh nilai rata-rata dari R G B, nilai rata-rata itulah yang dapat dikatakan sebagai grayclass. Rumus matematisnya adalah:

$$\mathbf{Grayscale} = (\mathbf{R} + \mathbf{G} + \mathbf{B}) / 3$$

##### 3. Luminosity

Algoritmanya adalah dengan mengalikan setiap nilai R G B dengan konstanta tertentu yang sudah ditetapkan nilainya, kemudian hasil perkalian seluruh nilai R G B dijumlahkan satu sama lain. Rumus matematisnya adalah:

$$\mathbf{Grayscale} = (\mathbf{0.299} * \mathbf{R}) + (\mathbf{0.587} * \mathbf{G}) + (\mathbf{0.114} * \mathbf{B})$$

#### b. Algoritma *Global Threshold*

$$f(x,y)' = \begin{cases} \alpha_1, & f(x,y) < T \\ \alpha_2, & f(x,y) \geq T \end{cases}$$

Keterangan :  $f(x,y)'$  = citra biner

$f(x,y)$  = citra *grayscale*

$\alpha_1 = 0$

$\alpha_2 = 1$

T = nilai *threshold* (Purba, 2010)

Menurut Hariadi (2012) nilai ambang *threshold* sangat mempengaruhi hasil identifikasi karena besar nilai *threshold* mempengaruhi jumlah piksel hitam pada grid. Nilai *threshold* yang baik digunakan adalah sebesar 190.

### c. Algoritma *Adaptive Threshold*

Pada metode *threshold* Sauvola [7], *threshold*  $T(x,y)$  dihitung menggunakan mean  $m(x,y)$  dan standar deviasi  $\delta(x,y)$  untuk window dengan ukuran  $w \times w$  seperti pada persamaan (3.1) berikut:

$$T(x,y) = m(x,y) [1 + k(\delta(x,y) / R-1)] \dots\dots\dots(3.1)$$

Dimana  $R$  adalah nilai maksimum standar deviasi, dan  $k$  adalah bias, yang mempunyai nilai positif pada rentang  $[0.2, 0.5]$ . Pada penelitiannya, Sauvola menggunakan nilai 0.5 begitu juga dengan Sezgin (2004). Rerata lokal  $m(x,y)$  dan standar deviasi  $\delta(x,y)$  menyesuaikan nilai *threshold* berdasarkan nilai kontras pada piksel tetangga lokal.

Menurut Badekas et al (2005), nilai yang paling optimal adalah 0.34. Tentu itu terbatas pada data penelitian yang mereka gunakan karena secara umum algoritma ini tidak sensitif terhadap perubahan nilai.

#### 3.1.4 Implementasi (*Implementation*)

Implementasi adalah langkah nyata untuk menerapkan perangkat lunak yang telah dibuat. Artinya, pada tahap ini semua yang telah dikembangkan diinstal atau diset sedemikian rupa sesuai dengan peran atau fungsinya agar bisa diimplementasikan. Pada tahap ini dilakukan uji coba pemakaian aplikasi oleh user dengan mengoreksi 3 kelas lembar jawaban komputer. Apabila aplikasi koreksi lembar jawaban komputer yang dibuat belum sesuai maka dilakukan evaluasi kembali.

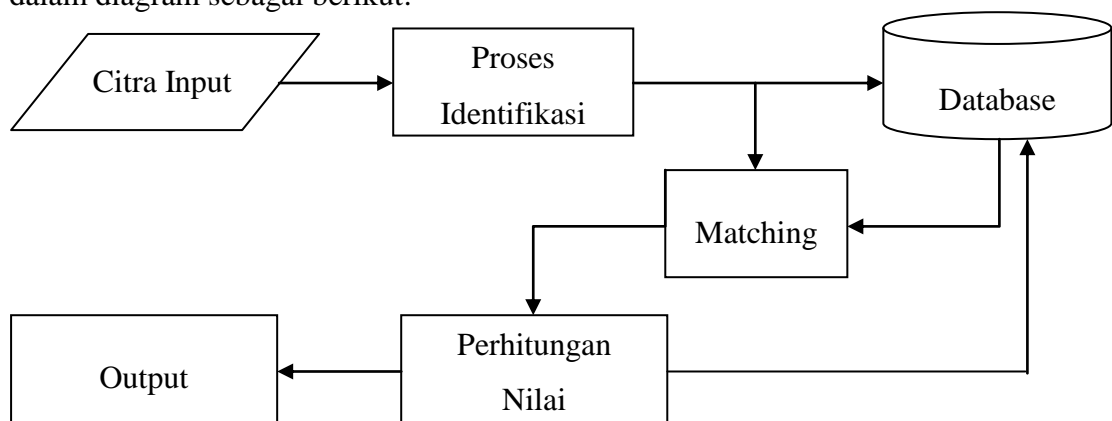
#### 3.1.5 Evaluasi

Evaluasi adalah proses untuk melihat apakah perangkat lunak yang sedang dibangun berhasil, sesuai dengan harapan awal atau tidak. Sebenarnya tahap evaluasi bisa terjadi pada setiap empat tahap di atas. Evaluasi yang terjadi pada

setiap empat tahap diatas itu dinamakan *evaluasi formatif*, karena tujuannya untuk kebutuhan revisi. Misal, pada tahap rancangan, mungkin kita memerlukan salah satu bentuk *evaluasi formatif* misalnya review ahli untuk memberikan input terhadap rancangan yang sedang kita buat. Pada tahap pengembangan, mungkin perlu uji coba dari produk yang kita kembangkan. Begitupun pada tahap implementasi akan dilakukan evaluasi untuk melihat apakah aplikasi yang dibangun sudah sesuai atau tidak.

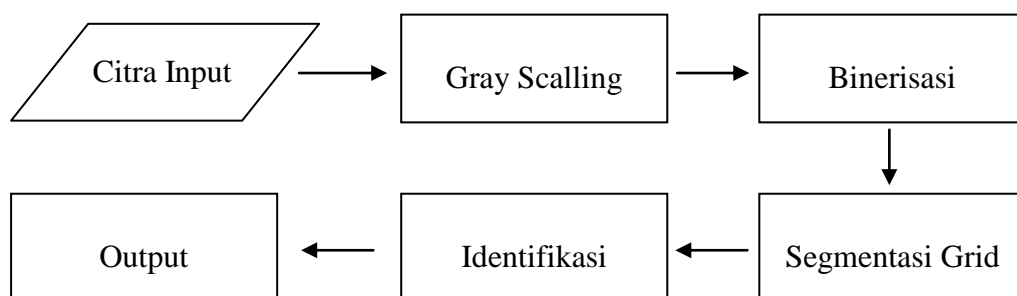
### 3.2 Pemodelan dan Perancangan Sistem

Sistematika kerja aplikasi dalam penelitian ini secara garis besar dimodelkan dalam diagram sebagai berikut:



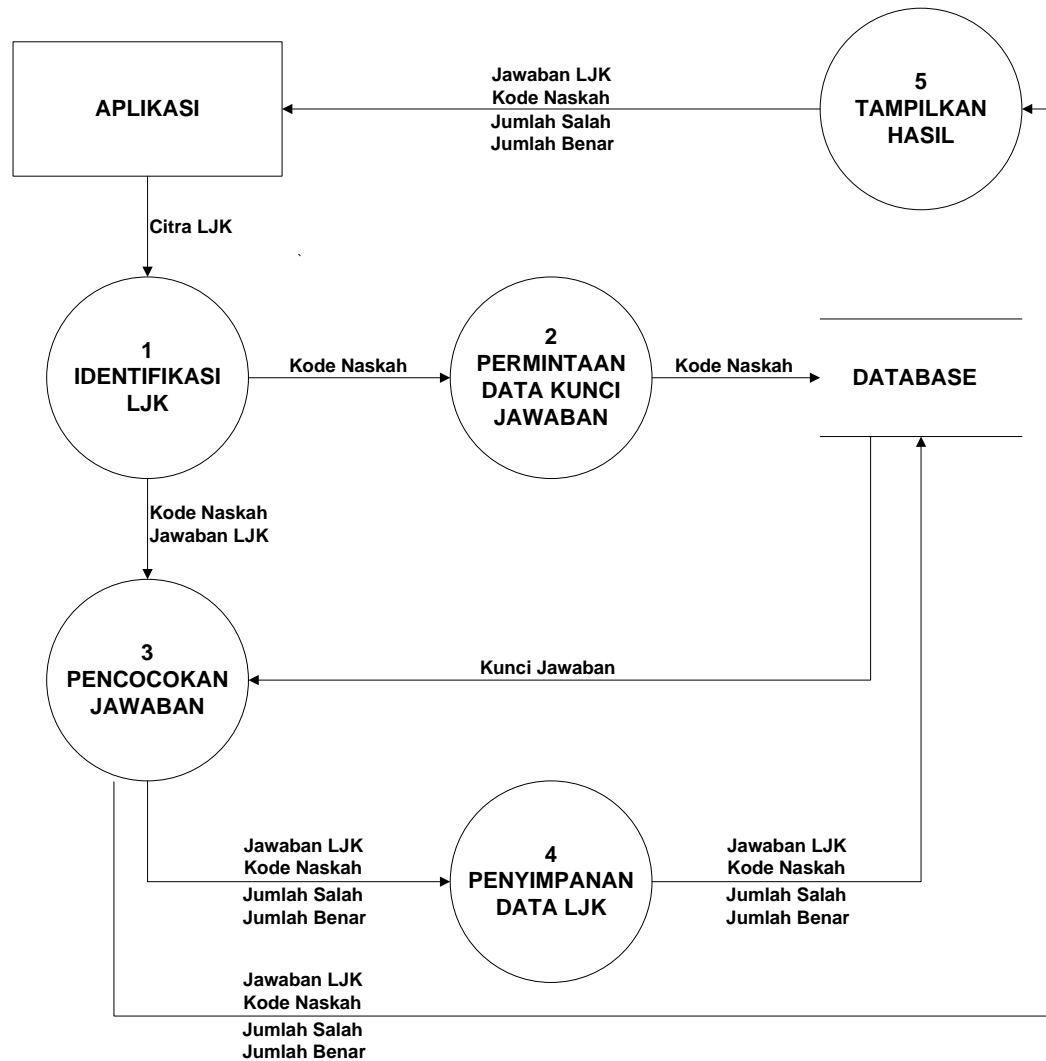
Gambar 3.1 Sistematika Kerja Sistem

Input merupakan citra LJK hasil scan yang kemudian dilakukan proses identifikasi menggunakan segmentasi citra. Hasil dari segmentasi citra merupakan pilihan jawaban yang dihitamkan akan disimpan ke dalam database dan dilakukan proses matching, yaitu pencocokan jawaban dengan kunci jawaban pada database. Hasil keluaran system akan berupa jawaban yang dihitamkan, jumlah benar dan jumlah salah.



Gambar 3.2 Blok Diagram Identifikasi LJK

Proses identifikasi citra LJK akan melalui proses *Grayscale*, sehingga gambar menjadi citra *grayscale*. Selanjutnya akan menjadi citra hitam putih (biner) lewat proses *binerisasi*. Citra biner tersebut kemudian akan disegmentasi sehingga pilihan jawaban yang dihitamkan akan diketahui.



Gambar 3.3 DFD Proses Identifikasi LJK

Berdasarkan DFD di atas dapat dijelaskan proses dalam aplikasi Koreksi LJK sebagai berikut:

- a. Proses 1 yaitu proses pengidentifikasian citra LJK oleh *system*.
- b. Proses 2 akan mengirimkan permintaan kunci jawaban yang sesuai dengan hasil kode naskah pada proses 1.

- c. Proses 3 merupakan proses pencocokan jawaban hasil identifikasi dengan kunci jawaban. Proses ini akan menghitung jumlah jawaban yang salah dan benar.
- d. Proses 4 merupakan proses penyimpanan jawaban hasil identifikasi, kode naskah, jumlah jawaban salah dan jumlah jawaban benar ke dalam database.
- e. Proses 5 merupakan proses akhir, dimana proses 5 akan memberikan kode naskah, jawaban hasil identifikasi, jumlah jawaban salah dan jumlah jawaban benar untuk ditampilkan pada aplikasi.

### 3.3 Tempat dan Waktu Penelitian

#### 3.3.1 Tempat Penelitian

Penelitian ini dilakukan di SMP Negeri 3 Sekampung Udik Lampung Timur yang beralamat di Jln. Gatot Subroto Plong 1 Desa Brawijaya Kecamatan Sekampung Udik Kabupaten Lampung Timur tahun pelajaran 2016-2017.

#### 3.3.2 Waktu Penelitian

Waktu penelitian dilaksanakan pada bulan Nopember 2016 sampai dengan Februari 2017. Berikut adalah jadwal dan waktu pelaksanaan penelitian dapat dilihat pada tabel 3.1 dibawah ini.

Tabel 3.1 Jadwal dan waktu pelaksanaan penelitian

No	Kegiatan	Bulan			
		Nop	Des	Jan	Feb-Mar
1	Analisis				
2	Desain				
3	Pengembangan				
4	Implementasi				
5	Evaluasi				