

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi Sistem

Implementasi sistem merupakan tahap pembuatan perangkat lunak yang dilakukan berdasarkan hasil dari analisis dan perancangan yang telah dilakukan. Selanjutnya hasil dari analisis dan perancangan akan dilakukan pengkodean ke dalam Bahasa pemrograman HTML, CSS, PHP, JavaScript. Adapun implementasi sistem yang dilakukan terdiri dari implementasi perangkat keras, implementasi perangkat lunak, dan implementasi antarmuka.

##### 4.1.1 Implementasi Perangkat Keras

Perangkat keras yang akan digunakan untuk membangun sistem *Optical Character Recognition* (OCR) Menggunakan *Support Vector Machine* (SVM) dan *Zoning* Pada Sertifikat dengan pertimbangan kebutuhan dan kemampuan perangkat keras untuk membangun program. Spesifikasi perangkat yang digunakan dapat dilihat pada Tabel 4.1 dibawah ini:

**Tabel 4.1 Implementasi Kebutuhan Perangkat Keras**

No	Perangkat Keras	Spesifikasi
1	Processor	Intel Core i5-8250U
2	Memory (RAM)	8 GB
3	Harddisk	1 TB
4	VGA	NVIDIA GeForce 940MX
5	Monitor	14'' dengan Resolusi 1920 x 1080

##### 4.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi dapat dilihat pada tabel 4.2 dibawah ini:

**Tabel 4.2 Implementasi Kebutuhan Perangkat Lunak**

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 10
2	Bahasa Pemrograman	HTML, CSS, PHP, JavaScript
3	Software Pendukung	Sublime Text 3, Google Chrome
4	Web Server	XAMPP ver 3.2.2
5	DBMS	MySQL 5.0.12

#### 4.1.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan pada setiap tampilan program yang telah dirancang. Berikut ini adalah penjelasan implementasi antarmuka, dapat dilihat pada tabel 4.3 dibawah ini:

**Tabel 4.3 Implementasi Antarmuka**

No	Nama Antarmuka	Deskripsi	Nama File
1	Beranda	Halaman utama dari sistem	beranda.php
2	Pengolahan Data	Digunakan untuk mengolah data serta memberi label pada data yang nantinya akan dijadikan data training	pengolahan_data.php
3	Training	Digunakan untuk melihat hasil dari pengolahan data dan melatih data yang sudah ada menggunakan SVM	training.php
4	Testing	Digunakan untuk melakukan pengujian pengenalan karakter	testing.php

#### 4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa sistem yang dibuat sesuai dengan rancangan sistem yang telah dibuat sebelumnya.

Pengujian sistem yang dilakukan meliputi tahap pengujian fungsionalitas serta pengujian akurasi.

#### 4.2.1 Pengujian *Black Box*

Berikut ini merupakan pengujian menggunakan metode *Black box*. Pengujian ini berfokus pada persyaratan fungsional perangkat lunak.

##### 4.2.1.1 Rencana Pengujian

Pengujian *black box* digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang dirancang. Berikut adalah tabel rencana pengujian *black box*.

**Tabel 4.4 Rencana Pengujian *Black Box***

No	Komponen Yang Diuji	Butir Uji	Jenis Pengujian
1	<i>Preprocessing</i>	Memasukkan citra latih	<i>Black Box</i>
		Melakukan proses pengolahan citra	<i>Black Box</i>
		Memberi label pada citra	<i>Black Box</i>
2	<i>Training</i>	Menekan tombol <i>training</i>	<i>Black Box</i>
3	<i>Testing</i>	Memasukkan citra uji	<i>Black Box</i>
		Melakukan proses pengolahan serta mengenali karakter	<i>Black Box</i>

##### 4.2.1.2 Pengujian *Black Box*

Berdasarkan rencana pengujian sebelumnya, maka dapat dilakukan pengujian *black box* pada aplikasi yang dibangun. Tabel 4.5 dibawah ini merupakan pengujian memasukkan citra yang dilakukan pada aplikasi.

**Tabel 4.5 Pengujian Memasukkan Citra Latih**

<b>Data Masukkan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<b>Kasus dan Hasil Uji (Data Benar)</b>			
data latih.jpg	Data ditampilkan dalam aplikasi	Data dapat ditampilkan dalam aplikasi	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
data latih.txt	Menampilkan pesan “Data harus berformat .jpg dan .png”	Tidak menampilkan pesan	[ <input type="checkbox"/> ] Diterima [ <input checked="" type="checkbox"/> ] Ditolak



Tabel 4.6 merupakan pengujian proses pengolahan citra yang dilakukan pada aplikasi.

**Tabel 4.6 Pengujian Proses Pengolahan Citra**

<b>Data Masukkan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<b>Kasus dan Hasil Uji (Data Benar)</b>			
data latih.jpg	Menampilkan tahapan segmentasi	Dapat menampilkan tahapan segmentasi	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
data latih.txt	Tidak Menampilkan tahapan segmentasi	Tidak menampilkan tahapan segmentasi	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak

Tabel 4.7 merupakan pengujian pemberian label pada citra yang telah di segmentasi yang dilakukan pada aplikasi.

**Tabel 4.7 Pengujian Pemberian Label Pada Citra**

<b>Data Masukkan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<b>Kasus dan Hasil Uji (Data Benar)</b>			
 Karakter = B	Data label dapat disimpan ke dalam database serta citra disimpan ke dalam folder <i>image</i>	Data label dapat disimpan ke dalam database serta citra disimpan ke dalam folder <i>image</i>	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
 Karakter = Null	Menampilkan pesan “Please fill out this field.”	Menampilkan pesan “Please fill out this field.”	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

Tabel 4.8 merupakan pengujian dengan menekan tombol *training* yang dilakukan pada aplikasi.

**Tabel 4.8 Pengujian Menekan Tombol Training**

<b>Data Masukkan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<b>Kasus dan Hasil Uji (Data Benar)</b>			
Menekan Tombol <i>Training</i>	Membuat model SVM dari data yang ada serta	Dapat membuat model SVM serta	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

	menampilkan pesan berhasil	menampilkan pesan berhasil	
<b>Kasus dan Hasil Uji (Data Salah)</b>			
Tidak Menekan Tombol <i>Training</i>	Tidak membuat model SVM serta tidak menampilkan pesan berhasil	Tidak membuat model SVM serta tidak menampilkan pesan berhasil	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak

Tabel 4.9 dibawah ini merupakan pengujian memasukkan citra yang dilakukan pada aplikasi.

**Tabel 4.9 Pengujian Memasukkan Citra Uji**

<b>Data Masukkan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<b>Kasus dan Hasil Uji (Data Benar)</b>			
data uji.jpg	Data ditampilkan dalam aplikasi	Data dapat ditampilkan aplikasi	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
data uji.txt	Menampilkan pesan “Data harus berformat .jpg dan .png”	Tidak menampilkan pesan	[ <input type="checkbox"/> ] Diterima [ <input checked="" type="checkbox"/> ] Ditolak

Tabel 4.10 merupakan pengujian proses pengolahan citra yang dilakukan pada aplikasi.

**Tabel 4.10 Pengujian Proses Pengolahan Citra**

<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<b>Kasus dan Hasil Uji (Data Benar)</b>			
data uji.jpg	Menampilkan hasil pengenalan karakter	Dapat menampilkan tahapan segmentasi	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
data uji.txt	Tidak menampilkan hasil pengenalan karakter	Tidak menampilkan hasil pengenalan karakter	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak

#### 4.2.2 Pengujian Akurasi

Pada pengujian akurasi akan dihitung nilai ketepatan atau kecocokkan dari hasil aplikasi pengenalan karakter pada sertifikat. Pengujian dilakukan apabila data *training* sudah melalui proses pelatihan.

Dalam penelitian ini pengujian akurasi digunakan beberapa kombinasi parameter pada data latih dan data uji. Tabel 4.11 dibawah ini merupakan kombinasi parameter yang digunakan.

**Tabel 4.11 Kombinasi Parameter Pengujian**

<b>Pengujian Ke-</b>	<b>Nilai <i>Threshold</i></b>	<b>Zona yang digunakan</b>
1	64	3
2	128	3
3	192	3
4	64	5
5	128	5

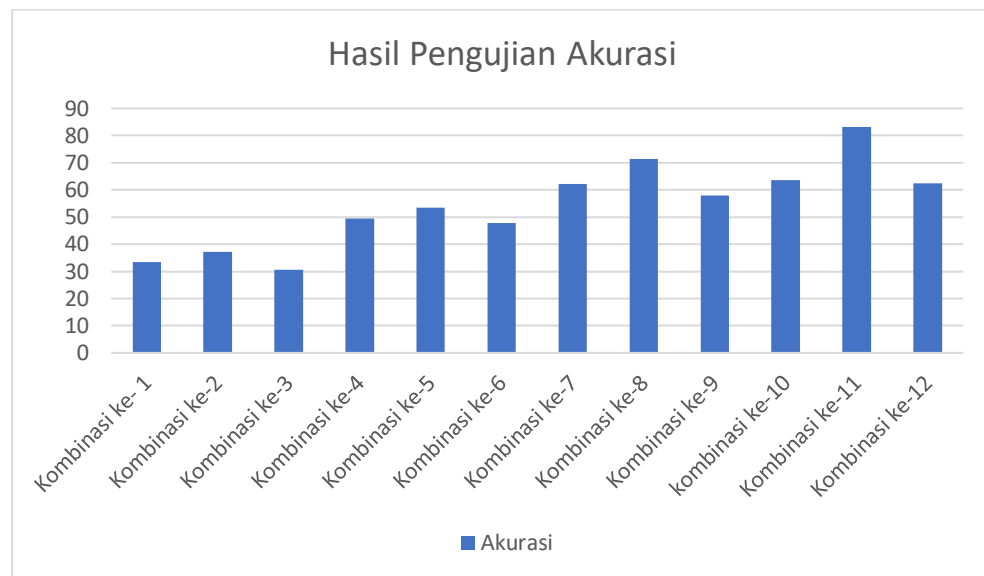
6	192	5
7	64	9
8	128	9
9	192	9
10	64	25
11	128	25
12	192	25

Untuk melihat detail dari pengujian akurasi sistem ini dapat dilihat pada lampiran E. Berikut ini adalah tabel serta grafik hasil dari pengujian akurasi:

**Tabel 4.12 Hasil Pengujian**

<b>Pengujian ke-</b>	<b>Akurasi</b>
1	33,52903%
2	37,14347%
3	30,62087%
4	49,41350%
5	53,51257%
6	47,86438%
7	62,12804%
8	71,34923%
9	57,93863%
10	63,50068%
11	83,27952%
12	62,53437%





### 4.2.3 Kesimpulan Pengujian

Kesimpulan dilakukan untuk mengetahui apakah sistem yang dibuat sesuai dengan pada saat perancangan serta untuk mengetahui ketepatan atau kecocokkan metode SVM dalam pengenalan karakter pada sertifikat.

#### 4.2.3.1 Kesimpulan Pengujian *Black Box*

Hasil dari pengujian *black box* yang telah dilakukan menunjukkan aplikasi yang dibangun memiliki fungsionalitas yang baik, karena hampir semua pengujian yang dilakukan menghasilkan hasil yang sesuai dengan harapan. Hasil yang tidak sesuai harapan adalah pada saat memasukkan data dengan format .txt kedalam pengolahan citra, dimana aplikasi menerima masukan format data masukan yang lain selain format .jpg dan .png.

#### 4.2.3.2 Kesimpulan Pengujian Akurasi

Berikut adalah kesimpulan dari pengujian akurasi yang sudah dilakukan terhadap keseluruhan kombinasi parameter. Adapun kombinasi parameter terbaik adalah kombinasi parameter ke-11 dengan akurasi rata-rata yang didapatkan sebesar 83,27952%. Kombinasi parameter yang digunakan yaitu nilai *threshold* 128 dengan 25 zona. Akurasi yang didapatkan dipengaruhi oleh *preprocessing* serta data uji yang digunakan karena pada *background* pada sebuah sertifikat tidak polos melainkan

terdapat gambar atau pola yang mengakibatkan pada saat proses segmentasi gambar atau pola tersebut dikenali sebagai karakter. Kemudian apabila ada dua karakter yang berdempet maka gabungan dari huruf tersebut hanya dikenali sebagai satu karakter saja. Serta susunan huruf yang harus dilakukan secara manual dikarenakan proses segmentasi memindai piksel dari ujung kiri atas hingga kanan bawah yang mengakibatkan apabila ada karakter yang memiliki titik piksel paling atas maka karakter tersebut akan di deteksi sebagai objek pertama bukan dilihat dari susunan karakter berdasarkan data masukannya.

Hasil temuan yang didapatkan pada saat pengujian adalah terdapat beberapa kekurangan dari metode *preprocessing* khususnya pada metode segmentasi. Berikut adalah kekurangan dari metode segmentasi yang diimplementasikan pada sistem ini:

1. Pada penelitian ini seharusnya objek yang dikenali itu hanya huruf dan angka saja, akan tetapi setelah dilakukan pengujian, *noise* yang ada pada sertifikat dikenali sebagai sebuah objek juga.
2. Metode ini tidak dapat menangani tulisan yang berdempet. Karena pada sertifikat, jenis huruf yang ada terkadang ada huruf yang berdempet dengan huruf lainnya. Hasilnya apabila ada beberapa huruf yang berdempet maka akan dibaca sebagai satu objek/huruf saja.
3. Metode ini tidak dapat menangani sebuah huruf yang memiliki dua objek yang terpisah, contohnya seperti huruf i (I kecil) dan huruf j (J kecil). Oleh karena itu, huruf tersebut tidak dikenali sebagai satu objek melainkan dua objek.