

BAB III METODE PENELITIAN

3.1. Bahan Penelitian

Bahan penelitian yang akan digunakan untuk sistem pengenalan teks menggunakan *optical character recognition* berbasis *text to speech realtime* ini. Yaitu, *Data input* berupa *image* yang berisikan teks dari media tercetak. Media tersebut berupa buku paket pembelajaran, koran majalah serta teks berupa hasil cetak komputer dan juga tulisan tangan.

3.2. Peralatan Penelitian

Penelitian ini menggunakan peralatan berupa perangkat lunak dan perangkat keras, yaitu :

3.2.1. Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan antara lain :

Tabel 3.1 Perangkat Lunak

No	Nama	Keterangan
1	Android Studio	Versi 3.5.3 menggunakan Bahasa pemrograman <i>Java</i>
2	<i>Library</i>	<i>Library Text To Speech</i>
3	<i>Application Programing Interface</i>	Google API database

3.2.2. Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan antara lain :

Tabel 3.2 Perangkat Keras

No	Nama	Keterangan
1	Laptop	Membuat program
2	Printer	<i>Sample</i> teks media tercetak
3	Penggaris	Mengukur jarak
4	<i>Smartphone</i>	<i>Testing</i> program Kamera 16 megapixel.

3.3. Lokasi Penelitian dan Waktu Penelitian

3.3.1. Lokasi Penelitian

Penelitian ini akan dilakukan oleh penulis di laboratorium riset *networking* Fakultas Teknik dan Ilmu Komputer Universitas Buana Perjuangan Karawang dengan hasil analisis data dan persiapan kebutuhan yang sudah dipersiapkan.

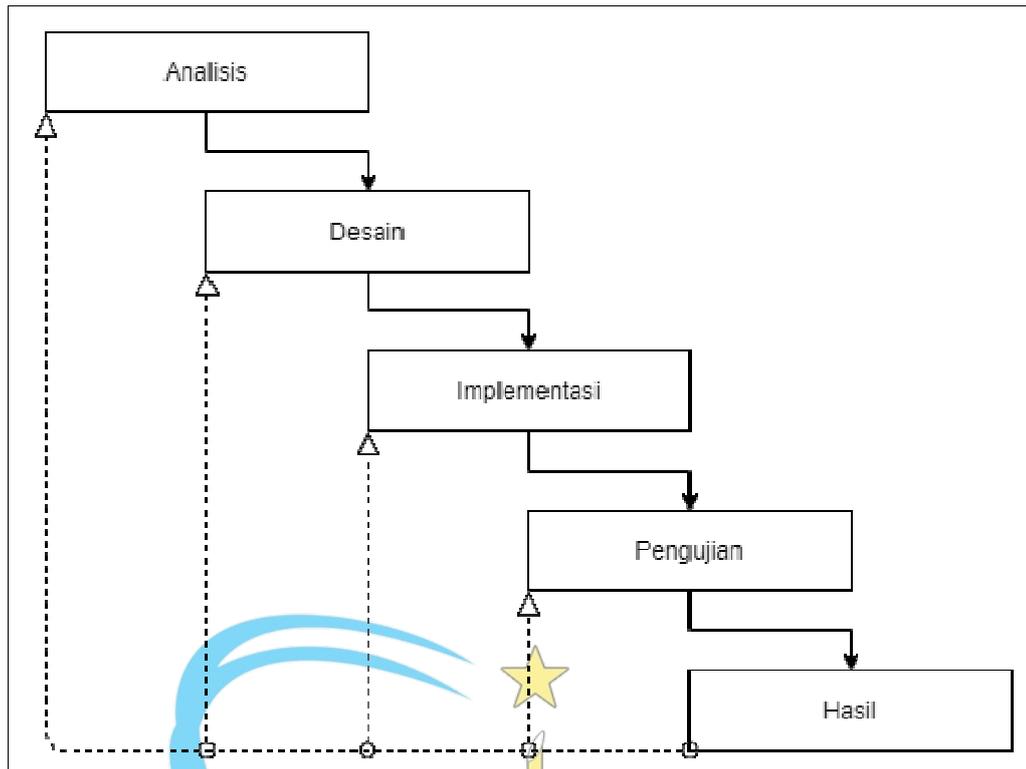
3.3.2. Waktu Penelitian

Tabel 3.3 Waktu Penelitian

No	Keterangan	Bulan ke 1				Bulan ke 2				Bulan ke 3				Bulan ke 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi literatur	█															
2	Analisis data	█															
3	Analisis sistem	█															
4	Perancangan sistem	█															
5	Implementasi sistem	█															
6	Pengujian sistem	█															
7	Evaluasi sistem	█															
8	Hasil	█															

3.4. Prosedur Penelitian

Prosedur percobaan dalam pengembangan sistem penelitian ini akan menggunakan model (*Software Development Life Cycle*) SDLC, SDLC adalah proses metodologi yang digunakan untuk pembuatan dan juga pengembangan dalam sebuah sistem yang terdiri dari beberapa tahapan. Model SDLC yang akan digunakan pada penelitian ini adalah model *Waterfall* atau yang biasa dikenal dengan model air terjun (Wicaksono, 2012). Menurut Bassil (2012) disebut *waterfall* karena tahapan – tahapan yang dilalui harus menyelesaikan tahapan sebelumnya atau berurutan. Adapun tahapan model *waterfall* tersebut.



Gambar 3.1 Alur Kerja *Waterfall*

Penelitian ini akan dibuat sebuah sistem aplikasi yang mampu mengenal sebuah tulisan teks, dimana proses *input* sebuah *image text* hingga mengeluarkan *output* sebuah suara menggunakan *text to speech*. Seperti pada Gambar 3.1 adalah tahapan pertahapan untuk mencoba mengembangkan sistem pada penelitian ini. Adapun penjelasan pada tahapan percobaan ini antara lain :

3.4.1. Analisis

Analisis data pada penelitian ini dilakukan dengan berdasarkan beberapa jurnal yang sudah dibaca dan juga *sample text* yang akan dilakukan untuk pengujian. Pada jurnal proses pengenalan teks seringkali terjadi sebuah masalah yaitu kesulitan mendeteksi yang disebabkan karena banyaknya ragam ukuran dan jenis *font* yang berbeda. Hal tersebut menyulitkan sistem untuk memproses dan mencocokkan huruf-huruf yang ada. Sedangkan data yang disiapkan pada penelitian ini yaitu beberapa kalimat teks yang akan dijadikan *sample* uji serta *library* dan *google API* yang akan diterapkan pada sistem OCR.

3.4.2. Desain

Perancangan atau desain sistem adalah tahapan yang dilakukan setelah persiapan dan analisis data hingga akan dilakukannya sebuah implementasi pada sistem, namun sistem yang akan dibuat harus dirancang sedemikian rupa terlebih dahulu dengan sebuah tampilan *mockup design*. Perancangan atau desain sistem OCR berbasis TTS ini bisa dilihat sebagai berikut :

1. *Design* Perancangan Aplikasi

Perencanaan awal yang akan dibuat adalah persiapan untuk tata cara ataupun tampilan input untuk melakukan deteksi teks, dengan tampilan yang sederhana. Bisa dilihat pada Gambar 3.2.



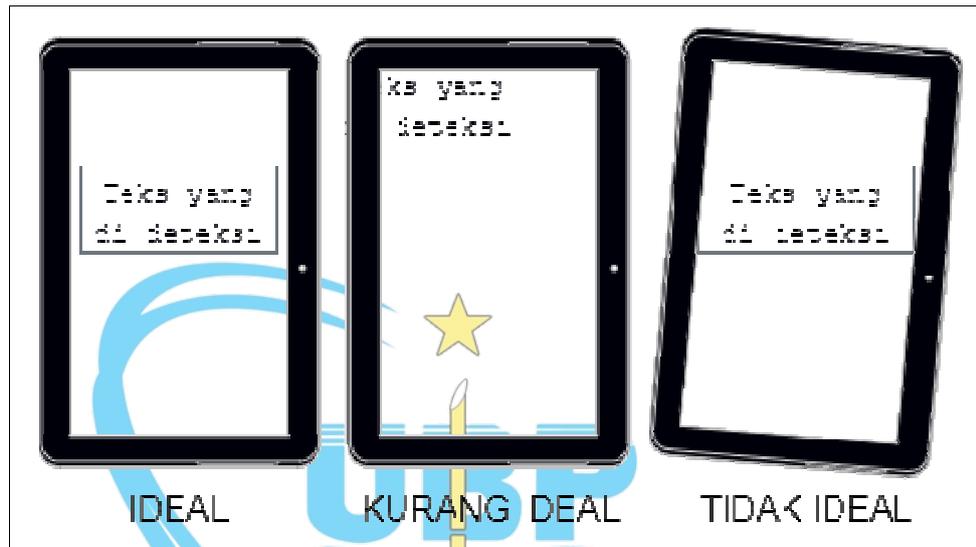
Gambar 3.2 Desain Perancangan Aplikasi

Gambar 3.2 merupakan desain awal yang menggambarkan contoh untuk mendeteksi tulisan yang berada dikertas dengan sistem OCR berbasis TTS yang ada pada *smartphone*.

Gambar 3.2 merupakan contoh proses awal pada tahapan sistem OCR ini, dimana *sample text* yang akan diuji dipersiapkan terlebih dahulu untuk melakukan tahapan selanjutnya. Dengan persiapan perancangan pada sistem maka dilakukan desain tampilan terlebih dahulu sebelum di implemetasikan.

2. *Design* Posisi Tampilan Aplikasi

Perancangan atau desain pada tampilan proses sistem OCR berbasis TTS secara *realtime* ini akan mempersiapkan rancangan untuk posisi tata letak *smartphone* ataupun *sample input* yang akan dilakukan saat pendeteksian, dimana posisi *smartphone* sangat berpengaruh pada proses pengambilan gambar.



Gambar 3.3 Posisi Deteksi Teks

Gambar 3.3 menunjukkan posisi Ideal saat melakukan pendeteksian sebuah teks. Dimana posisi Ideal adalah posisi paling tepat untuk melakukan pendeteksian sistem OCR, sedangkan pada posisi Kurang Ideal sistem OCR hanya bisa mendeteksi bagian - bagian teks yang terlihat jelas saja dan pada posisi Tidak Ideal ini, sistem OCR tidak bisa mendeteksi teks, karena sistem OCR hanya mampu mendeteksi teks dengan posisi kamera *vertical*.

3. *Design* Tampilan Aplikasi

Desain tampilan aplikasi pada sistem OCR berbasis TTS ini adalah *layout* yang akan menampilkan kalimat teks yang terdeteksi serta kamera saat melakukan pendeteksian. Bisa dilihat seperti pada Gambar 3.4.

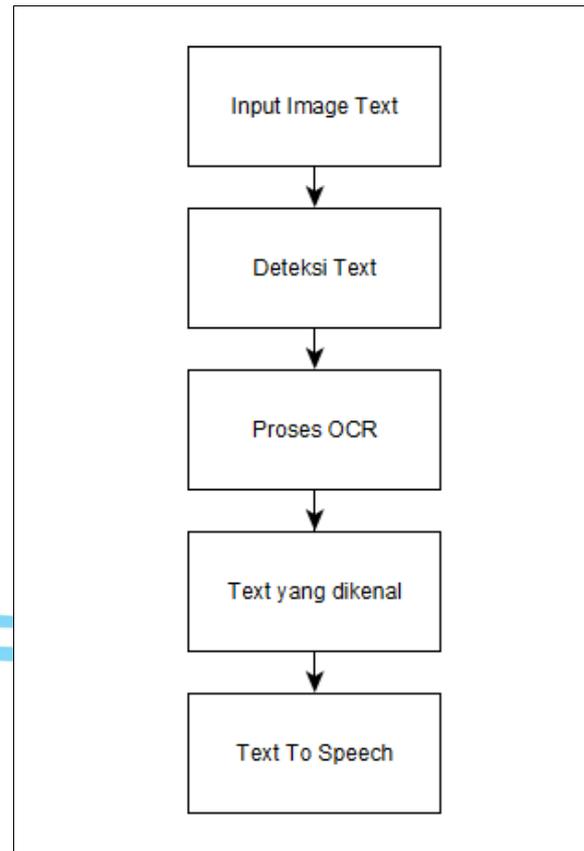


Gambar 3.4 Desain Tampilan Aplikasi

Gambar 3.4 adalah rencana atau perancangan yang akan dibuat untuk tampilan aplikasi sistem OCR berbasis TTS secara *realtime* ini. *Text view* yang berada dibawah dengan *background* biru akan menampilkan huruf-huruf yang sudah dicocokkan. Dan tampilan putih diatas *text view* adalah *open camera* yang akan berfungsi untuk mendeteksi objek.

3.4.3. Implementasi Sistem

Implementasi adalah suatu tahapan setelah sebelumnya melakukan tahapan perancangan atau desain sistem. Penelitian ini menggunakan Bahasa *Java* untuk membuat program sebuah sistem dan menggunakan perangkat lunak *Android Studio IDE* serta menggunakan *library* dan *google API* untuk melengkapi sistem dengan tahapan sebagai berikut :

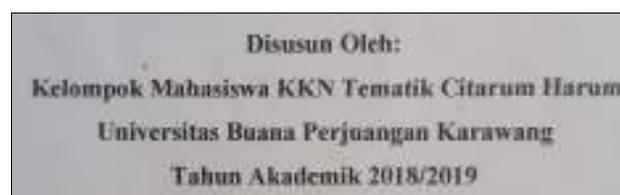


Gambar 3.5 *Flowchart* Implementasi Sistem

Flowchart pada Gambar 3.5 menjelaskan tahapan sistem dari *input*, proses pendeteksian *optical character ecognition* hingga mengeluarkan sebuah *output* suara atau *text to speech*.

1. *Input Image Text*

Input image text adalah tahapan awal untuk mempersiapkan segala lembar tulisan ataupun *sample text* yang akan diuji pada penelitian ini, contoh *input* suatu gambar teks pada sistem OCR berbasis TTS *realtime* ini bisa dilihat pada Gambar 3.6.



Gambar 3.6 *Input Image Text*

2. *Deteksi Text*

Sistem OCR akan melakukan suatu deteksi pada *sample* yang sudah disiapkan oleh tahapan sebelumnya, pada sistem OCR berbasis TTS *realtime* ini, untuk mendapatkan proses pendeteksian yang maksimal maka posisi kamera dilakukan seperti pada contoh Gambar 3.7.



Gambar 3.7 Deteksi Teks

3. Proses *Optical Character Recognition*

Pengenalan teks pada sistem ini menggunakan metode OCR atau *Optical Character Recognition*. Dimana metode OCR memiliki sebuah tahapan proses seperti pada Gambar 3.8 berikut :



Gambar 3.8 Alur Proses Sistem OCR

A. *File Input*

Tahapan awal ini adalah tahapan dimana persiapan awal untuk data-data ataupun *sample text* yang akan di *input*.

B. *Preprocessing*

Tahapan yang memiliki suatu proses untuk menghilangkan bagian bagian yang tidak diperlukan seperti pada *background* gambar.

C. *Segmentasi*

Tahapan ini adalah proses yang memisahkan setiap karakter yang terdeteksi.

D. *Normalisasi*

Tahapan ini yaitu proses merubah dimensi tiap karakter dan juga ukuran serta ketebalan karakter yang terdeteksi.

E. *Ekstraksi Fitur*

Tahapan ini adalah penelitian pada sistem untuk mengambil sebuah ciri-ciri tertentu dari karakter yang terdeteksi.

F. Recognition

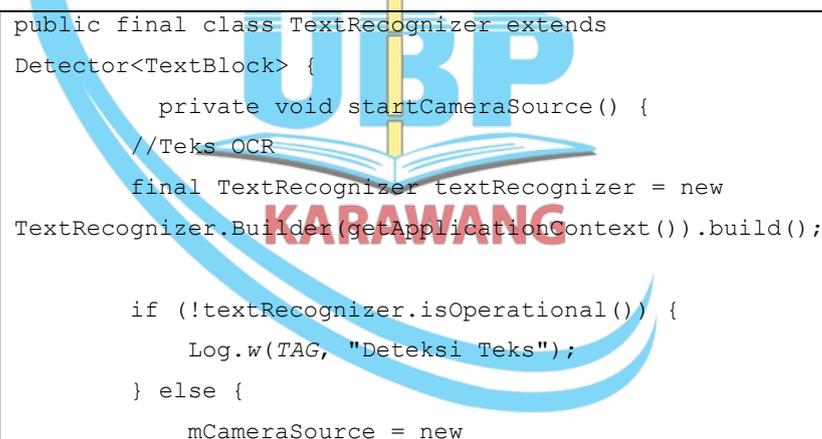
Proses ini adalah tahapan untuk mengenali serta menentukan karakter dengan cara mengamati dan membandingkan ciri-ciri karakter yang terdeteksi dengan ciri-ciri karakter yang ada pada *library*.

G. ASCII Text

Tahapan akhir ataupun sebuah *output* dari semua proses - proses yang sebelumnya telah dilakukan oleh tahapan yang lain hingga mengeluarkan sebuah *output* dari persamaan *sample text* yang sudah dicocokkan oleh proses *recognition*.

4. Text Yang Dikenal

Pencocokan teks pada sistem OCR menggunakan *library* yang sudah disediakan oleh *software* android studio. Bisa dilihat contoh *script* pada Gambar 3.9.



```
public final class TextRecognizer extends
Detector<TextBlock> {
    private void startCameraSource() {
        //Teks OCR
        final TextRecognizer textRecognizer = new
TextRecognizer.Builder(getApplicationContext()).build();

        if (!textRecognizer.isOperational()) {
            Log.w(TAG, "Deteksi Teks");
        } else {
            mCameraSource = new
```

Gambar 3.9 *script API google*

Source code yang ada pada Gambar 3.9 adalah contoh *script API google* yang berfungsi untuk pencocokan pada suatu proses pendeteksian kamera OCR yang sudah di *input* untuk dijadikan sebuah *output*.

Disusun Oleh:
Kelompok Mahasiswa KKN Tematik Citarum
Harum
Universitas Buana Perjuangan Karawang
Tahun Akademik 2018/2019

Gambar 3.10 Teks Yang Dikenal

Source code yang ada pada Gambar 3.9 akan melakukan pencocokan teks dengan yang ada pada *database* dan sistem ini bekerja secara dibelakang layar sistem hingga mengeluarkan sebuah *output* teks seperti pada Gambar 3.10. Hasil dari proses OCR tersebut pun akan terlihat pada *layout smartphone*.

5. *Text To Speech*

Tahapan ini adalah *output* terakhir dari sistem dimana teks yang di *input* di proses hingga tercocokkan oleh teks yang berada pada sistem hingga ditampilkan pada *text view output* lalu dibaca oleh *text to speech* dengan menggunakan *script* seperti pada Gambar 3.11.

```

textToSpeech=new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
    @Override
    public void onInit(int status) {
        if (status == TextToSpeech.SUCCESS) {
            int result =
textToSpeech.setLanguage(new Locale("id", "ID"));
            speakOut();

```

Gambar 3.11 *Text To Speech*.

Untuk memanfaatkan *text to speech* pada sistem memerlukan beberapa langkah atau menerapkan pemanggilan sebuah *script* seperti yang ada pada Gambar 3.11, *script* yang berfungsi untuk memanggil google API yang berada pada *library* android studio.

3.4.4. Pengujian Sistem

Tahapan pengujian ini akan dilakukan secara 3 kali dengan deteksi teks secara normal, variasi jarak dan juga deteksi teks pada layar komputer menggunakan perangkat keras berupa *smartphone* dengan *resolusi* kamera 16 *mega pixel*. Pengujian yang dilakukan secara normal yaitu letak kamera *smartphone* didekatkan dengan teks yang berada dikertas hingga menghasilkan hasil maksimal, sedangkan pengujian dengan variasi jarak sistem OCR berbasis TTS akan meniru penglihatan manusia saat

membaca. Menurut Kompasiana.com, (2015), Jika membaca terlalu dekat tidak baik untuk kesehatan mata dan berdampak juga pada tubuh kita. Jarak normal untuk membaca yaitu 25-30 centimeter (CM). maka sistem OCR ini akan dirancang dan di implementasikan hingga mampu melakukan pengujian deteksi *sample* teks dengan jarak manusia membaca normal dengan mengukur jarak secara manual menggunakan alat bantu ukur.

Percobaan pada pengujian ini pun terbagi pada beberapa tahapan dimana tahapan ini yang akan menentukan hasil dari proses perancangan dan implementasi, antara lain :

1. *Input*

Proses ini dilakukan sebelum sistem aplikasi mendeteksi sebuah teks dimana harus disiapkan beberapa *sample* teks dengan tulisan tipe *font*, *size font* dan *background* serta kamera yang berbeda.

2. *Proses*

Kertas yang berisi tulisan teks akan secara langsung di tangkap oleh sistem aplikasi melalui kamera *smartphone* dengan jarak yang sesuai dan jarak kamera hingga 25-30 cm. Proses pada kamera *smartphone* harus disesuaikan secara *vertical*. Kemudian teks yang sudah terdeteksi oleh sistem akan diolah pada proses pengenalan karakter dalam sistem aplikasi.

3. *Output*

Tahapan ini adalah hasil dari proses kerja sistem aplikasi pengenalan teks menggunakan OCR berbasis *text to speech*, dimana *image* tulisan teks yang sudah terdeteksi oleh sistem dan diproses dengan metode OCR sehingga menentukan hasil teks yang memiliki tingkat ciri-ciri kesamaannya dengan huruf-huruf yang berada pada *database library text recognition* sehingga dicocokkan oleh sistem OCR.

Sistem OCR terlebih dahulu mengeluarkan sebuah *output* teks yang sudah diproses sebelumnya untuk dicocokkan, dan *text to speech*

akan bekerja untuk membaca *text view* yang ada pada sistem, ini adalah tahapan *klasifikasi* akhir pada sistem OCR ini.

