

BAB II

TINJAUAN PUSTAKA

2.1 Penelitain-Penelitian Sebelumnya

Sebelum penelitian ini dilakukan terdapat peneitian-penelitian yang dapat menjadi acuan untuk menulis penelitian ini. Adapun, berbagai penelitian telah dilakukan dengan metode *Optical Character Recognition* (OCR) untuk mengkonversi gambar huruf menjadi karakter yang dikenali oleh komputer. Salah satu penelitian yang mencoba untuk mengkonversi gambar adalah penelitian Syahrullah (2018) [4], melalui metode *Optical Character Recognition* (OCR) yang berfokus pada konversi teks dari gambar menjadi file teks yang bisa diproses komputer, menyimpulkan bahwa dengan bantuan *Raspberry PI* sebagai mikrokontroller, dengan *Push Button* sebagai *trigger*, menghasilkan sebuah file gambar dengan *graphical capture* yang dilakukan oleh *Raspberry Pi Camera*, secara otomatis akan diolah dalam proses *Optical Character Recognition* (OCR) dan proses *Text to Speech* (TTS) yang mana berfungsi untuk mengubah *text* kedalam ucapan dalam format mp3 sehingga dapat didengar oleh penyandang tunanetra. Adapun keunggulan dari penelitian ini salah satunya adalah menciptakan suatu inovasi yang memiliki portabilitas tinggi sehingga dapat dibawa kemanapun dengan mudah dan juga sebagai pengganti atau alternatif dari buku braile. Adapun kekurangannya adalah tidak terdapatnya tombol yang terpisah untuk memotret dan mengeluarkan suara.

Kemudian menurut penelitian Kurnia, dkk (2020) [5], melalui hasil *research and development* (R&D) menggunakan metode *Optical Character Recognition* (OCR) pada Android, menyimpulkan bahwa tafsir yang mudah dipahami tunanetra adalah tafsir yang ringkas, bahasanya sederhana dan tidak menggunakan istilah-istilah yang rumit. Pengucapan jelas dan menghindari bahasa yang ambigu. Tunanetra dapat dengan mudah menggunakan teknologi yang kompatibel dengan ponsel Android. Hal ini dikarenakan Android sangat terkenal dan banyak aplikasi yang bisa digunakan. Meski tidak mudah menyajikan interpretasi melalui ucapan, namun tetap bisa dibantu oleh pembaca layar. Adapun

kekurangan dari penelitian ini adalah aplikasi tidak dapat mengeluarkan suara dengan otomatis.

Selain itu menurut penelitian Utami, dkk (2016) [6], melalui hasil inovasinya menggunakan perangkat Android dengan teknologi *Optical Character Recognition* (OCR), menyimpulkan bahwa pengujian aplikasi terjemahan (Inggris – Indonesia) dengan menggunakan OCR metode *white box* menghasilkan akurasi pengenalan karakter sebesar 97.5%. Hasil perhitungan persentase akurasi pengenalan karakter yang diberikan oleh aplikasi untuk seluruh sampel yang diuji menggunakan OCR dapat dinyatakan sangat baik.

Kemudian menurut penelitian Yudhistiro (2018) [7], melalui metode OCR yang digunakan untuk merestorasi citra yang tidak sempurna menjadi lebih baik menggunakan algoritma *Recurent Hopfiled* dengan *pemrograman Visual Basic* 2010, menyimpulkan bahwa hasil restorasi mampu memperbaiki vektor pola uji coba dan menampilkan pola mana yang mirip dengannya.

Selain itu menurut penelitian Ibnutama, dkk (2019) [8], melalui metode *Template Matching* pada *Optical Character Recognition* (OCR), menyimpulkan bahwa untuk meningkatkan akurasi deteksi pada plat kendaraan dibutuhkan suatu modifikasi dengan menggunakan metode *Template Matching* yang mana merupakan sebuah algoritma yang dapat diterapkan pada *Optical Character Recognition* (OCR) untuk meningkatkan persentase keberhasilan dengan mengenali karakter teks dari gambar yang diambil. Namun akurasi hasil pengenalan sangat mempengaruhi kualitas gambar yang diperoleh, seperti banyaknya *noise* atau pemilihan atau pendeteksian karakter yang tidak lengkap. Hasil penelitiannya menunjukkan bahwa hasil pengenalan meningkat sebesar 4,44% untuk sampel sebanyak 35 plat nomor.

Kemudian menurut penelitian Saputra, dkk (2021) [9], melalui hasil inovasinya dalam pengembangan *text to speech* media pembelajaran pengenalan anggota tubuh manusia untuk kelas 5 SD dengan *voices*, *3D images*, dan *videos* yang menarik untuk menambah minat para murid untuk belajar, menyimpulkan

bahwa aplikasi berhasil mengkonversi teks menjadi suara *menggunakan Cloud Text to Speech API* serta dengan penambahan *Cloud Google Translate API*. Adapun kelebihan dari aplikasi tersebut adalah dapat dijalankan tanpa harus menggunakan koneksi internet atau *offline* yang akan lebih mempermudah pengguna yang tidak memiliki koneksi internet.

Adapun menurut penelitian Sugeng & Syamsuddin (2020) [10], melalui hasil inovasinya dalam pengenalan plat nomor kendaraan khususnya plat nomor Indonesia menggunakan *image processing* sebagai alat tilang elektronik otomatis, menyimpulkan bahwa dengan metode optimasi pengenalan karakter menggunakan *image processing* mampu meningkatkan pengenalan plat nomor atau pembacaan karakter berdasarkan kelompok karakter yang ditetapkan, dengan peningkatan sebesar 14% yaitu persentase akurasi awal 80% naik menjadi 94% keberhasilan.

2.2 Teori Pendukung

2.2.1 Tunanetra

Tunanetra merupakan istilah yang biasa digunakan untuk orang yang mengalami gangguan pengelihatian baik buta maupun *low vision*. Penyandang tunanetra membutuhkan bantuan alat untuk dapat membaca, yaitu dengan menggunakan aksara *braille* [11].

2.2.2 Pengolahan Citra Digital

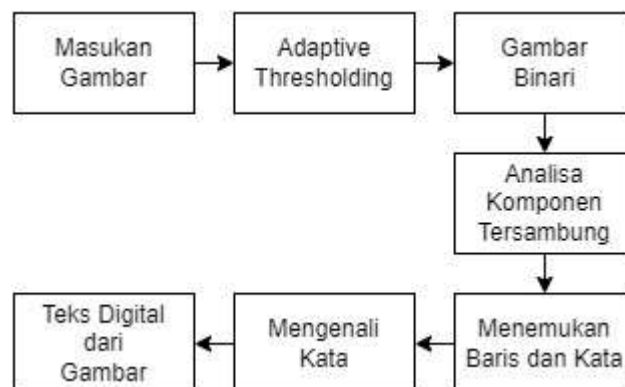
Pengolahan citra digital adalah sebuah proses dari gambar asli menjadi gambar lain yang sesuai dengan kebutuhan kita. Contoh sebuah gambar yang didapatkan terlalu gelap maka dengan pengolahan citra gambar tersebut bisa diproses sehingga mendapat gambar yang jelas. Prinsip dasar dari pengolahan citra adalah pengolahan warna RGB (*red*, *green*, dan *blue*) pada posisi tertentu. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00FFFFFF [12].



Gambar 2. 1 Konversi Gambar dengan OCR (Picture to Text)

2.2.3 *Optical Character Recognition (OCR)*

Optical character recognition (OCR) merupakan suatu sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (*image character*) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam database aplikasi. Dari proses OCR dihasilkan teks sesuai dengan gambar *output scanner* dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan [13]. Proses gambar ke teks yaitu proses yang terjadi pada *Tesseract OCR*. Pada pengerjaannya dilakukan beberapa proses, yaitu:

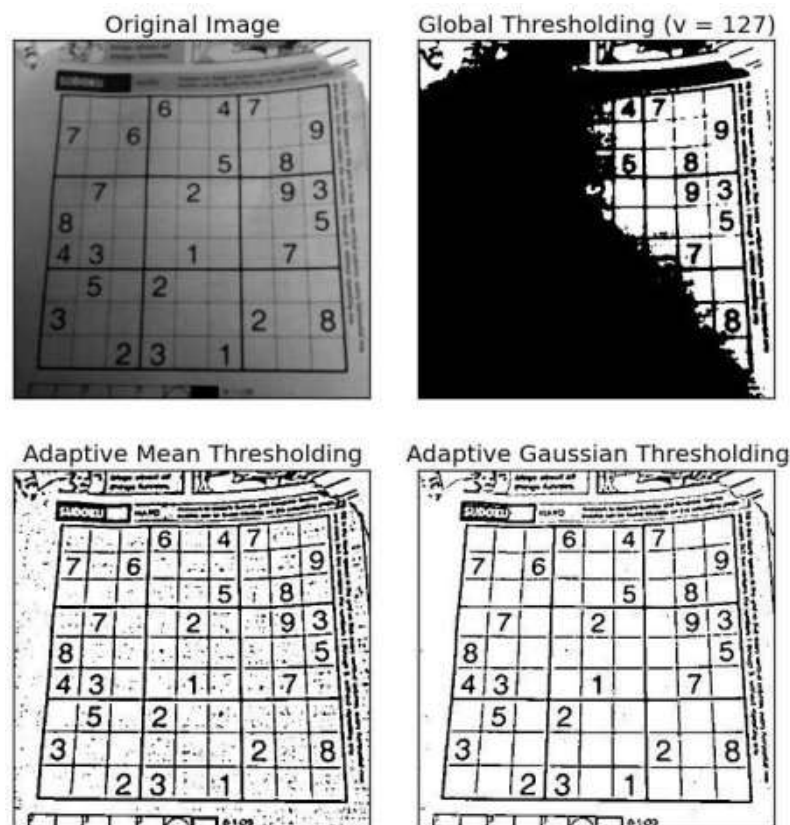


Gambar 2. 2 Diagram Konversi Gambar ke Teks

Adapun berikut adalah diagram konversi gambar ke teks, mulai dari tahap memasukan gambar menggunakan kemudian dilakuakn *adaptive thresholding*, dan didapatkan gambar binari yang dilanjutkan dengan Analisa komponen tersambung, selanjutnya menemukan baris kemudian mengenali kata sehingga didapatkan teks yang berhasil dikonversi dari gambar.

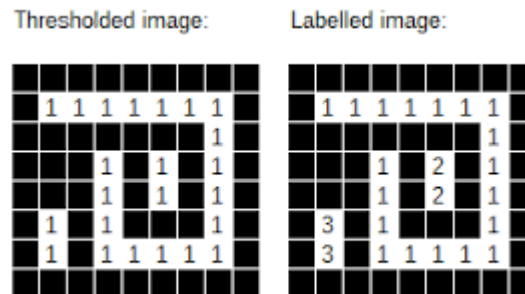
Gambar yang dimasukan, gambar yang menjadi *input* yaitu berwarna atau *grayscale* (keabu-abuan). *Tesseract* menerima input dengan ekstensi *.tiff* dan *.bmp* secara *native*, namun juga dapat memproses gambar dengan format ekstensi yang lainnya. Gambar ideal *tesseract engine* adalah gambar yang flat atau tidak memiliki efek 3D.

Adaptive Thresholding merupakan proses selanjutnya. Pada proses ini dilakukan reduksi dari *grayscale image* ke *binary image*. Algoritma memperkirakan gambar terdiri dari *foreground pixel* dan *background pixel*. Setelah itu, algoritma menghitung *threshold* yang optimal untuk memisahkan kedua *pixel*.



Gambar 2. 3 Adaptive Thresholding

Penandaan Komponen Tersambung (Connected-Component Labeling). *Tesseract* menghitung *pixel* dalam gambar, menentukan *foreground pixel*, dan di tandai sebagai *blob* atau *potential character*.



Gambar 2. 4 Connected-Component Labeling

Kemudian proses selanjutnya adalah Penemuan baris dan karakter potensial gambar digunakan untuk menemukan garis dari teks. Algoritma ini menentukan lokasi dengan memilih *pixel* kurang dari *threshold* tertentu. Hasilnya berupa area yang ditandai sebagai baris potensial.

Baseline Fitting Algorithm digunakan setelah itu. Setelah menentukan garis dari setiap teks, *tesseract* memeriksa garis teks untuk menghitung tinggi teks. Proses ini adalah langkah awal untuk mengenal karakter.

Tesseract engine menghitung lebar karakter. Nilai ini adalah nilai inkremental yang akan digunakan oleh *tesseract engine* dalam mengekstrak karakter demi karakter. Disebut dengan Fixed Pitch Detention. Jika ditemukan karakter yang tidak seragam dengan lingkungan disekitarnya, karakter tersebut akan diidentifikasi ulang secara terpisah atau Non-Fixed Pitch Spacing Delimiting.

Pengenalan Kata dilakukan selanjutnya. Setelah menentukan seluruh *possible character* dan *possible line* dalam gambar, *tesseract engine* melakukan *word recognition* dengan menelusuri garis. Kata kemudian dikirim kepada *contextual* dan *syntactical analyzer* untuk meningkatkan tingkat akurasi.

$$Cacc = \frac{\text{Jumlah kata yang dapat dikenali dengan baik}}{\text{Jumlah seluruh kata}} \times 100\% \quad \dots (2.1)$$

Setelah hasil dari pendeteksian OCR pada gambar teks didapatkan selanjutnya dilakukan perhitungan dengan menguji tingkat keakurasian pada

pengenalan karakter atau *Character Accuracy* (Cacc). Untuk menghitung banyaknya karakter teks yang dapat dideteksi dengan benar, maka dapat menggunakan persamaan (2.1).

2.2.4 *Tesseract OCR*

Tesseract OCR merupakan suatu *library* yang digunakan untuk pemrosesan citra biner menjadi teks, metode yang digunakan adalah dengan analisa distribusi piksel pada citra untuk mengenali karakter [14].



Gambar 2. 5 Tesseract dengan Python

2.2.5 *OpenCV*

OpenCV merupakan sebuah *library* yang digunakan untuk pemrosesan citra. *Library* ini berlisensi *open source* dan dapat digunakan secara gratis. Bahasa pemrograman yang digunakan juga beragam dari *C* samapi *Python*. Digunakan untuk mengubah citra dengan format RGB menjadi format biner. Beberapa proses dalam mengolah suatu citra seperti *plotting histogram*, perubahan format citra dapat dilakukan dengan menggunakan *OpenCV*. *OpenCV* dapat digunakan untuk pengembangan *computer vision*. *OpenCV* memiliki banyak fitur untuk membantu dalam membuat program yang akan dibuat seperti dalam alat bantu baca ini [15].

2.2.6 *Text to Speech* (TTS)

Text to Speech adalah sistem konversi perubahan sebuah teks menjadi sinyal suara dan lebih dikenal dengan sebutan TTS (*Text to Speech*). Adanya fasilitas TTS pada alat ini sebagai sistem konversi teks yang telah diidentifikasi dengan OCR menjadi suara. TTS mengubah suatu teks dengan membuatnya menjadi fonem-

fonem sebelum diubah menjadi kalimat. Digunakan *eSpeak library* sebagai TTS konversinya [16]. *eSpeak* diatur untuk mengeluarkan Bahasa Indonesia.

Text to Speech (TTS) yaitu proses pengubahan teks menjadi *audio digital* dan diucapkan. Tugas sistem TTS secara umum dapat dibagi dalam dua bagian besar, analisa teks dan sintesa ucapan.



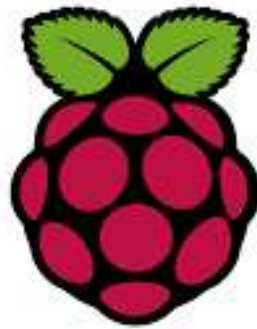
Gambar 2. 6 Diagram Konversi Teks ke Suara

Analisa teks mentransformasi teks masukan menjadi representasi linguistik, selanjutnya bagian sintesis ucapan mentransformasi representasi linguistik tersebut menjadi gelombang sinyal ucapan. Sistem *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu: Bagian Konverter Teks ke Fonem (*Text to Phoneme*), serta Bagian Konverter Fonem to Ucapan (*Phoneme to Speech*). Bagian Konverter text ke Fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta *pitch*-nya. Bagian Konverter Fonem ke Ucapan akan menerima masukan berupa kode-kode fonem serta pitch dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan.

2.2.7 *Raspberry Pi*

Raspberry Pi adalah *single board computer* berukuran kartu kredit yang dikembangkan di Inggris oleh Yayasan *Raspberry Pi* dengan tujuan untuk mempromosikan pengajaran ilmu pengetahuan dasar komputer di sekolah. Board

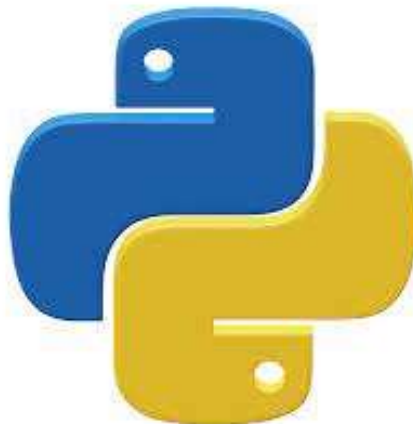
yang digunakan adalah *Raspberry Pi 3 B*. Board ini sudah dilengkapi dengan 30 pin GPIO yang dapat digunakan untuk menerima atau mengeluarkan input. Lalu sudah dilengkapi dengan prosesor 1.2 GHz dengan ram 1 GB yang dapat memudahkan pekerjaan board. *Raspberry Pi 3 B* juga sudah dilengkapi wireless LAN. Dengan modul kamera 5 MP yang memiliki resolusi 2592 x 1944 *pixel* [17].



Gambar 2. 7 Gambar Logo Raspberry Pi

2.2.8 Python

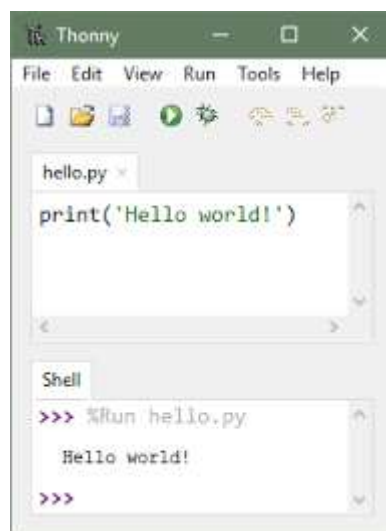
Python adalah bahasa pemrograman yang *freeware* yang berarti tidak ada batasan dalam pendistribusiannya atau penyalinannya. Lengkap dengan GUI, *source code* dan *debugger*. Bahasa pemrograman ini yang akan digunakan pada saat penelitian. *Python* digunakan karena memiliki struktur sintak yang lebih mudah dipahami, *python 3* digunakan sebagai perinterpretasi bahasa *python* [18].



Gambar 2. 8 Logo Python

2.2.9 Thonny Python IDE

Thonny Python IDE merupakan sebuah perangkat lunak untuk mengembangkan kode secara efisien. IDE ini digunakan dengan alasan dapat memvisualisasikan sintak-sintak dengan lebih detail. *Thonny* juga digunakan karena mempermudah *Raspberry Pi* saat digunakan, juga memiliki lisensi *open-source* [19].



Gambar 2. 9 Thonny Python IDE