

BAB II. LANDASAN TEORI

2.1 Penelitian Terdahulu

Pada penelitian yang dilakukan oleh Kristina Apriyanti dkk. pada tahun 2016 yang berjudul “Implementasi *Optical Character Recognition* Berbasis *Backpropagation* untuk *Text to Speech* Perangkat Android” telah berhasil dirancang dan diimplementasikan pengenalan karakter pada citra teks berbasis algoritma *backpropagation* untuk aplikasi *text to speech* pada perangkat Android. Sistem tersebut memiliki akurasi pengenalan pada sampel uji yang digunakan sebesar 94,7% dengan kondisi jarak pengambilan gambar pada rentang jarak 3 – 8 cm dan posisi kamera tegak lurus menghadap kertas tulisan. Sistem yang dibuat bisa mengenali karakter dengan baik pada beberapa variasi warna *font* pada beberapa variasi warna kertas. Variasi warna yang diujikan yakni warna hitam, merah, dan biru, sedangkan variasi warna kertas HVS yang diujikan yakni berwarna putih, merah, biru, dan kuning. Rerata lamanya waktu proses pengenalan karakter yang dilakukan sistem yakni 86.24 ms per karakter (Apriyanti & Widodo, 2016).

Pada penelitian yang dilakukan oleh Putu Asri Sri Sutanti dkk. pada tahun 2020 yang berjudul “*Low Filtering Method for Noise Reduction at Text to Speech Application*” menunjukkan sistem tersebut dapat mengonversi teks menjadi suara dengan tingkat akurasi 70%. Penelitian tersebut telah mampu mengembangkan sistem *text-to-speech* dari penelitian sebelumnya dan sistem telah mampu mengonversi teks menjadi suara dengan dua kata bahasa Inggris (Asri et al., 2020).

Pada penelitian yang dilakukan oleh Rieke Adriati W. dkk. pada tahun 2016 yang berjudul “Pengembangan Aplikasi *Text to Speech* Bahasa Indonesia Menggunakan Metode *Finite State Automata* Berbasis Android” menghasilkan kesimpulan bahwa proses normalisasi teks dapat dilakukan dengan baik terhadap *input* teks yang mengandung angka, satuan mata uang, satuan waktu (jam dan tanggal), suhu, satuan berat dan panjang, serta singkatan umum. Metode FSA dua tingkat dapat diaplikasikan untuk memenggal kata menjadi suku kata sesuai dengan 12 pola suku kata yang ditentukan untuk Bahasa Indonesia sebesar 95.19%, baik untuk kata dasar maupun kata berimbuhan. Sedangkan hasil penulisan kembali kata

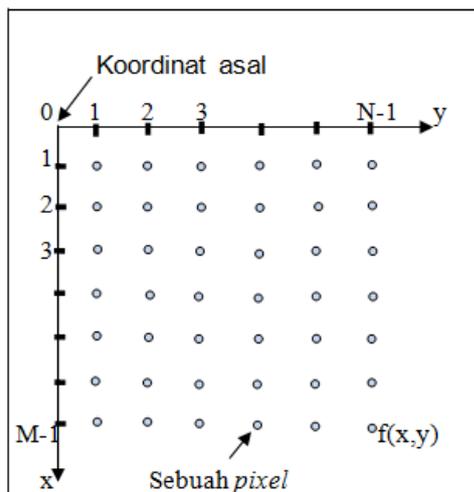
yang telah diperdengarkan kepada *user* bernilai 71% (rata-rata terdapat sebelas kata yang benar dari 20 kata yang diperdengarkan melalui aplikasi TTS) (W et al., 2016).

Pada penelitian yang dilakukan oleh Bayu Sy. Kurniawan dkk. pada tahun 2016 yang berjudul “Aplikasi Pengenal Citra Nomor Kendaraan Bermotor Menggunakan Metode *Template Matching*” menghasilkan kesimpulan bahwa Aplikasi pengenal nomor kendaraan bermotor telah berhasil dibuat dan mengenali karakter menggunakan keseluruhan 30 data sampel dengan total 238 karakter dengan tingkat keberhasilan 80,25 %. Dengan pengujian yang hanya menggunakan citra ideal berjumlah 22 data sampel dengan total 176 karakter, didapatkan hasil tingkat akurasi hingga 97,77% (Kurniawan et al., 2016).

2.2 Citra

Citra adalah suatu gambaran atau kemiripan dari suatu objek. Citra analog tidak dapat dipresentasikan dalam komputer, sehingga tidak bisa diproses oleh komputer secara langsung. Tentu agar bisa diproses di komputer, citra analog harus dikonversi menjadi citra digital. Citra digital adalah citra yang dapat diolah oleh komputer. Sedangkan citra yang dihasilkan dari peralatan digital (citra digital) langsung bisa diolah oleh komputer.

Citra digital adalah kumpulan nilai *real* atau kompleks yang diwakili oleh sekumpulan bit dalam suatu larik (*array*). Suatu citra dapat dicirikan sebagai fungsi $f(x,y)$ dengan M baris dan N kolom, di mana x dan y adalah koordinat spasial dan amplitudo f pada koordinat (x,y) adalah intensitas atau tingkat keabuan citra. Citra merupakan citra digital jika nilai keseluruhan x, y, dan nilai amplitudo f berhingga (*finite*) dan memiliki nilai diskrit. Gambar 2. 1 menggambarkan posisi koordinat citra digital. (Nafi'iyah, 2015).



Gambar 2. 1 Koordinat Citra Digital.

Sumber: (Nafi'iyah, 2015).

Ada tiga macam citra yang biasa digunakan dalam penelitian yaitu citra biner, citra *grayscale*, dan citra berwarna (W et al., 2016).

2.2.1 Citra Biner

Citra biner adalah citra di mana setiap piksel hanya dapat diwakili oleh salah satu dari dua nilai (yaitu nilai 0 dan 1). Nilai 0 menunjukkan hitam, sedangkan nilai 1 menunjukkan putih. Format gambar ini umumnya digunakan dalam pemrosesan gambar untuk memperoleh tepi bentuk suatu objek (Nafi'iyah, 2015).



Gambar 2. 2 Citra Biner.

Sumber: (Nafi'iyah, 2015).

2.2.2 Citra *Grayscale*

Citra *Grayscale* adalah gambar di mana setiap piksel hanya memiliki satu nilai saluran, yaitu $RED = GREEN = BLUE$. Angka ini digunakan untuk mewakili derajat intensitas {Formatting Citation}.



Gambar 2. 3 Citra *Grayscale*.

Sumber: (Tarigan et al., 2016).

2.2.3 Citra Warna

Citra berwarna, sering disebut sebagai citra RGB, adalah sejenis citra yang warnanya diwakili oleh komponen R (merah), G (hijau), dan B (biru) (biru). Delapan bit digunakan untuk setiap komponen warna (nilainya berkisar dari 0 hingga 255). Hasilnya, jumlah total warna yang dapat ditampilkan adalah $255 \times 255 \times 255$, atau 16.581.375 (Nafi'iyah, 2015).



Gambar 2. 4 Citra Warna.

Sumber: (Nafi'iyah, 2015).

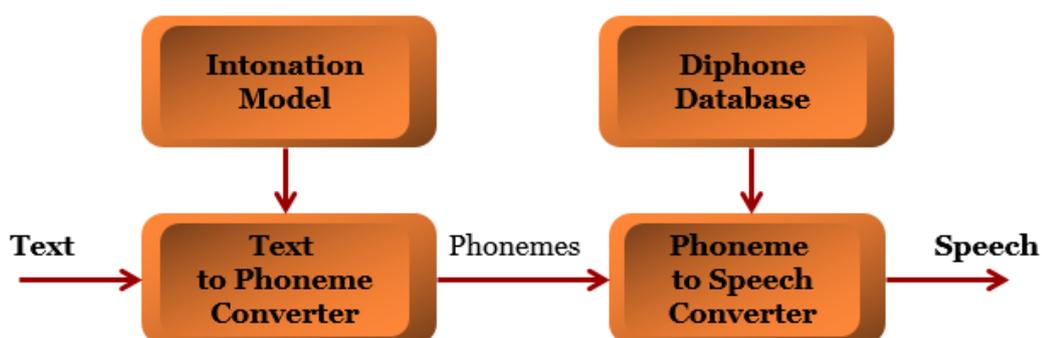
2.3 Tunanetra

Tunanetra kehilangan penglihatannya karena kedua indra penglihatannya tidak bekerja sebaik orang yang normal. Tunanetra diklasifikasikan sebagai buta (buta total) atau *low vision*. Tunanetra menghadapi tantangan serupa dalam hal memperoleh informasi. Tunanetra tidak memiliki tingkat kontrol yang sama atas lingkungan mereka dan diri mereka sendiri sebagai orang yang dapat melihat. Kendala ini dapat membuat mereka sulit untuk menyelesaikan tugas perkembangan mereka (Harimukthi & Dewi, 2014).

2.4 Text-To-Speech (TTS)

Secara umum pengertian *Text-To-Speech* adalah sebuah sistem yang mengonversi sebuah tulisan menjadi ucapan. Menurut (W et al., 2016), *Text-To-Speech* adalah sebagai produksi ucapan secara otomatis melalui transkripsi *grapheme to phoneme* dari sebuah kalimat.

TTS terdiri dari dua sub sistem: konverter teks ke fonem (teks ke fonem) dan konverter fonem ke ucapan (fonem ke ucapan). Bagian pengubah teks ke fonem mengubah kalimat masukan teks dalam bahasa tertentu menjadi rangkaian kode suara, yang biasanya diwakili oleh kode fonem, durasi, dan nada. Bagian ini sangat bergantung pada bahasa. Bagian ini harus benar-benar dikembangkan khusus untuk bahasa baru (W et al., 2016).



Gambar 2. 5 Sistem *Text-To-Speech*.
(Sumber: (W et al., 2016)).

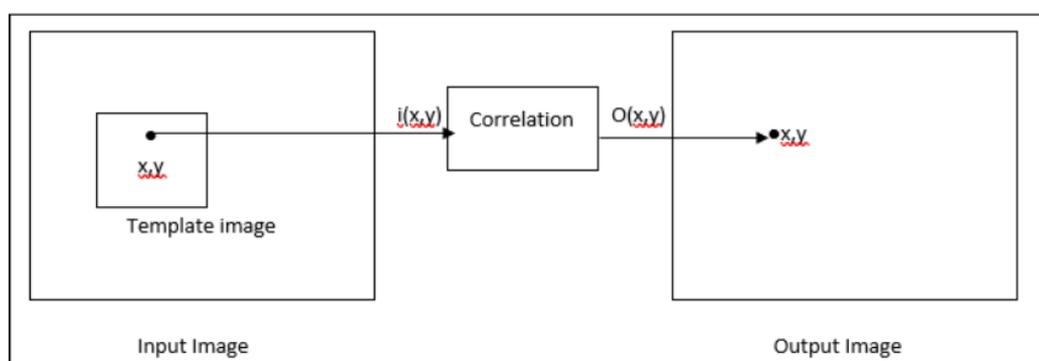
Kode fonem, serta nada dan durasi yang dihasilkan oleh bagian sebelumnya, dimasukkan ke dalam bagian konverter fonem ke suara. Bagian pengubah fonem

ke suara menggunakan kode-kode ini untuk menghasilkan suara atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan (W et al., 2016).

2.5 *Template Matching Correlation*

Template Matching adalah pendekatan pengolahan citra digital yang bekerja dengan cara mencocokkan setiap bagian dari suatu citra dengan citra *template* (referensi). Gambar *input* dibandingkan dengan gambar *template database*, dan kemudian suatu aturan digunakan untuk mencari kesamaan. Tingkat kemiripan/kesamaan yang tinggi dalam pencocokan gambar memutuskan bahwa suatu gambar diidentifikasi sebagai salah satu gambar (Hartanto et al., 2015).

Ukuran yang lebih kecil akan dibandingkan dengan semua bagian dari gambar sumber yang lebih besar selama proses pencocokan *template* (gambar masukan). Untuk mendapatkan nilai tingkat kecocokan yang paling mirip, proses ini diulang dari kiri atas ke kanan bawah. Pencocokan gambar yang menghasilkan tingkat kemiripan yang tinggi akan dikenali sebagai bagian dari gambar asli. Proses ini diulangi sampai seluruh citra telah dibandingkan dengan citra masukan (Rahmad et al., n.d.).



Gambar 2. 6 Metode *Template Matching*.

Sumber: (Rahmad et al., n.d.).

Memindahkan gambar *template* ke semua titik yang terdapat pada gambar asli yang lebih besar (gambar masukan) dan menghasilkan indeks numerik yang mewakili seberapa cocok *template* dengan gambar pada posisi tersebut dapat dilihat pada Gambar 2.6 Metode Pencocokan *Template Matching* di atas. Pencocokan piksel demi piksel ini dilakukan (Rahmad et al., n.d.).

Template Matching memiliki kelebihan dan kekurangan. Algoritma ini memiliki keuntungan karena mudah dikodekan dalam bahasa pemrograman dan mudah untuk menyiapkan data referensi. Karena datanya berbentuk matriks, maka perhitungannya tidak terlalu memakan waktu. Namun, algoritma ini memiliki sejumlah kelemahan selain kelebihannya. Kelemahannya adalah hasil yang ideal memerlukan sejumlah besar bahan referensi atau *database*. Sebuah citra atau *image* yang telah dijadikan matriks dapat digunakan sebagai *database*. Semakin banyak *font* yang ingin kita kenali, semakin banyak data referensi yang perlu kita lacak (Hartanto et al., 2015)

Kesamaan antar dua buah matriks citra dapat dihitung nilainya dengan menghitung nilai korelasinya (*correlation*). Nilai korelasi dua buah matriks dapat dihitung dengan menggunakan rumus 2.1 (Hartanto et al., 2015).

$$r = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i) \cdot (x_{jk} - \bar{x}_j)}{\sqrt{[\sum_{k=1}^n (x_{ik} - \bar{x}_i)^2 \cdot \sum_{k=1}^n (x_{jk} - \bar{x}_j)^2]}} \quad (2.1)$$

Dimana \bar{x}_i dirumuskan dengan persamaan 2.2 dan \bar{x}_j dirumuskan dengan persamaan 2.3 (Hartanto et al., 2015).

$$\bar{x}_i = \frac{1}{n} \sum_{k=1}^n x_{ik} \quad (2.2)$$

$$\bar{x}_j = \frac{1}{n} \sum_{k=1}^n x_{jk} \quad (2.3)$$

Keterangan:

r adalah nilai korelasi antara dua buah matriks (nilainya antara -1 dan +1).

x_{ik} adalah nilai *pixel* ke- k dalam matriks i .

x_{jk} adalah nilai *pixel* ke- k dalam matriks j .

x_i adalah rata-rata nilai *pixel* matriks i .

x_j adalah rata-rata nilai *pixel* matriks j .

n menyatakan jumlah *pixel* dalam suatu matriks.

Dalam penelitian ini, citra tingkat keabuan dicocokkan dengan menggunakan pendekatan *template matching*. Karena pencocokan dilakukan setelah citra melalui proses *thresholding* untuk menghasilkan keluaran berupa citra biner, maka disebut dengan tingkat keabuan. Dari kiri atas ke kanan bawah, piksel gambar biner dilacak. Nilai 1 akan diberikan ke gambar biner dengan piksel hitam. Piksel gambar putih

akan memiliki nilai 0, sedangkan piksel gambar hitam akan memiliki nilai 1 (Rahmad et al., n.d.).

0	0	0	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	1	0
1	1	1	0	0	0	0	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	0	0	0	0	1	1	1
0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1
1	1	1	0	0	0	0	1	1	1
0	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	0	0

Gambar 2. 7 Ilustrasi biner setiap pixel pada angka 9.

Sumber: (Rahmad et al., n.d.).

2.6 OpenCV

OpenCV atau disebut dengan *Open Source Computer Vision Library* adalah sebuah *library* (perpustakaan) yang digunakan untuk mengolah gambar dan video hingga kita dapat meng-ekstrak informasi di dalamnya. *OpenCV* menggunakan lisensi BSD dan bersifat gratis baik untuk penggunaan akademis maupun komersial. *OpenCV* dapat berjalan di berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan juga *support* di berbagai platform seperti Windows, Linux, Mac OS, iOS dan Android. Dalam penelitian ini, penulis menggunakan *OpenCV* versi 3.4.3