

OPTICAL CHARACTER RECOGNITION MENGGUNAKAN ALGORITMA TEMPLATE MATCHING CORRELATION

Suryo Hartanto¹, Aris Sugiharto², dan Sukmawati Nur Endah³

Jurusan Ilmu Komputer / Informatika, FSM Universitas Diponegoro

suryo.hartanto@gmail.com

Abstract

OCR (Optical Character Recognition) adalah suatu solusi yang efektif untuk proses konversi dokumen cetak ke dokumen digital. Permasalahan yang timbul dalam proses pengenalan dokumen komputer adalah bagaimana teknik pengenalan untuk mengidentifikasi berbagai jenis karakter dengan berbagai ukuran dan bentuk. Metode pengenalan yang digunakan dalam tugas akhir ini adalah metode *Template Matching Correlation*. Sebelum proses pengenalan, citra masukan dengan format * bmp atau jpg * diolah terlebih dahulu di proses preprocessing, yang meliputi binerisasi, segmentasi, dan normalisasi gambar. Rata-rata tingkat keberhasilan pengenalan yang dihasilkan oleh sistem ini adalah 92,90%. Hasil akhir menunjukkan bahwa penggunaan metode *Template Matching Correlation* cukup untuk membangun sebuah sistem OCR dengan akurasi yang baik efektif.

Keywords : OCR, Optical Character Recognition, Template Matching Correlation, preprocessing

I. PENDAHULUAN

Perkembangan teknologi saat ini telah banyak memberi pengaruh terhadap perkembangan ilmu pengetahuan, salah satunya adalah dalam hal pengenalan pola (*pattern recognition*). Pengenalan pola merupakan suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek. OCR (*Optical Character Recognition*) merupakan salah satu area studi dalam bidang pengenalan pola yang menarik untuk dieksplorasi.

Sistem pengenalan huruf atau sering disebut OCR merupakan solusi yang efektif untuk proses konversi dari dokumen cetak ke dalam bentuk dokumen digital. Permasalahan yang muncul dalam melakukan proses pengenalan huruf komputer adalah bagaimana sebuah teknik pengenalan dapat mengenali berbagai jenis huruf dengan ukuran, ketebalan, dan bentuk yang berbeda. Secara umum terdapat dua hal utama yang mempengaruhi proses OCR yaitu: mekanisme ekstraksi ciri dan mekanisme pengenalan.

Mekanisme ekstraksi ciri dilakukan untuk mendapatkan ciri atau identitas dari suatu karakter atau huruf. Proses pengenalan dilakukan setelah mekanisme ekstraksi ciri.

Proses pengenalan bertujuan untuk mencocokkan pola huruf yang berasal dari inputan dengan pola yang ada dalam basis pengetahuan. Beberapa algoritma yang dapat digunakan untuk proses pengenalan antara lain, jaringan syaraf tiruan, logika *fuzzy*, *k-Nearest Neighbor Algorithm*, *sequence alignment*, *template matching* dan lain-lain. Dari beberapa algoritma tersebut, algoritma *template matching* merupakan salah satu algoritma yang efektif untuk diterapkan dalam sistem OCR. Algoritma pengenalan ini cukup sederhana dan dapat diterapkan tanpa harus melakukan proses ekstraksi ciri terlebih dahulu, namun mempunyai hasil pengenalan yang tinggi.

Pada tugas akhir ini diteliti bagaimana pengimplementasian algoritma *template matching correlation* pada proses OCR.

II. TINJAUAN PUSTAKA

Tinjauan Pustaka berisi kumpulan studi pustaka yang berhubungan dengan topik tugas akhir. Dasar teori ini meliputi Pengertian Sistem Informasi, Konsep Rekayasa Perangkat Lunak, Metode Pengembangan Sistem, Konsep Client-Server, Sekilas Mengenai Internet, dan Bahasa Pemrograman dan Perangkat Lunak Pembangunan Sistem.

Citra Digital

Secara harfiah, *image* atau citra merupakan gambar pada bidang dwimatra (dua dimensi). Sedangkan dilihat dari sudut pandang matematis, citra merupakan fungsi kontinu atau menerus dari intensitas cahaya pada bidang dwimatra [11]. Citra terdiri dari dua macam jenis, yaitu citra kontinu dan citra diskrit. Citra kontinu berasal dari sistem optik yang menerima sinyal analog. Sedangkan citra diskrit berasal dari proses digitalisasi terhadap citra kontinu. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut sebagai digitalisasi. Citra yang dihasilkan inilah disebut citra digital.

Secara matematis fungsi intensitas cahaya pada bidang dwimatra disimbolkan dengan $f(x, y)$, yang dalam hal ini:

(x, y) : koordinat pada bidang dwimatra.

$f(x, y)$: intensitas cahaya (*brightness*) pada titik (x, y)

OCR

(*Optical Character Recognition*) OCR adalah sebuah aplikasi komputer yang digunakan untuk mengidentifikasi citra huruf maupun angka untuk dikonversi ke dalam bentuk *file* tulisan [3]. Sistem pengenalan huruf ini dapat meningkatkan fleksibilitas atau kemampuan dan kecerdasan sistem komputer. Sistem pengenalan huruf yang cerdas sangat membantu usaha besar-besaran yang saat ini dilakukan banyak pihak yakni usaha digitalisasi informasi dan pengetahuan, misalnya dalam pembuatan koleksi pustaka digital, koleksi sastra kuno digital, dan lain-lain.

Secara umum proses OCR dapat dilihat pada gambar 1 [16], dengan penjelasan sebagai berikut:

a. *File Input*

File input berupa *file* citra digital dengan format *.bmp atau *.jpg.

b. *Preprocessing*

Preprocessing merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak

diperlukan pada gambar *input* untuk proses selanjutnya.

c. *Segmentasi*

Segmentasi adalah proses memisahkan area pengamatan (*region*) pada tiap karakter yang dideteksi.

d. *Normalisasi*

Normalisasi adalah proses merubah dimensi *region* tiap karakter dan ketebalan karakter.

e. *Ekstraksi ciri*

Ekstraksi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter yang diamati.

f. *Recognition*

Recognition merupakan proses untuk mengenali karakter yang diamati dengan cara membandingkan ciri-ciri karakter yang diperoleh dengan ciri-ciri karakter yang ada pada basis data.



Gambar 1 Proses OCR secara umum.

Preprocessing Citra Digital

Preprocessing adalah tahap pertama yang harus dilakukan pada proses OCR. Tahap ini sangat penting untuk menentukan keberhasilan suatu proses pengenalan pola. Beberapa proses yang dapat dilakukan pada tahap *preprocessing* antara lain, proses binerisasi, segmentasi, dan normalisasi.

a. *Binerisasi*

Pada tahap proses binerisasi, *file* citra digital dikonversi menjadi citra biner. Citra biner (*binary image*) adalah citra yang hanya memiliki dua nilai derajat keabuan, yaitu hitam dan putih.

Pixel-pixel objek bernilai 1 dan *pixel-pixel* latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi, pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam [10].

Konversi dari citra hitam-putih ke citra biner dilakukan dengan menggunakan operasi pengambangan (*thresholding*). Operasi pengambangan mengelompokkan nilai derajat keabuan setiap *pixel* ke dalam 2 kelas, yaitu hitam dan putih.

b. Pengambangan global

Pengambangan global memetakan setiap *pixel* di dalam citra ke dalam dua nilai, 1 atau 0 dengan fungsi pengambangan di persamaan 2.3 [10].

$$f_B(i, j) = \begin{cases} 1, f_g(i, j) \leq T \\ 0, \text{lainnya} \end{cases} \dots\dots\dots (2.1)$$

Keterangan :

$f_g(i, j)$ adalah citra hitam-putih.

$f_B(i, j)$ adalah citra biner.

T adalah nilai ambang yang dispesifikasikan.

c. Pengambangan Lokal Adaptif

Pengambangan secara global tidak selalu tepat untuk seluruh macam gambar. Beberapa informasi penting di dalam gambar mungkin hilang karena pengambangan global.

Pengambangan secara lokal dilakukan terhadap daerah-daerah di dalam citra. Citra dipecah menjadi bagian-bagian kecil, kemudian proses pengambangan dilakukan secara lokal [10]. Dengan pengambangan lokal secara adaptif, secara subjektif citra biner yang dihasilkan terlihat lebih menyenangkan dan sedikit informasi yang hilang [10].

d. Pengambangan metode Otsu (Otsu Thresholding)

Pengambangan Otsu adalah sebuah teknik pengambangan yang diperkenalkan oleh Nobuyuki Otsu, yang secara otomatis mencari batas ambang terbaik untuk citra yang diolah [1]. Metode Otsu menghitung nilai ambang T secara otomatis berdasarkan citra masukan. Pendekatan

yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis Diskriminan berfungsi memaksimumkan variabel tersebut agar dapat memisahkan objek dengan latar belakang [14].

Nilai ambang yang dicari dari suatu citra hitam-putih dinyatakan dengan k . Nilai k berkisar antara 1 sampai dengan L , dengan nilai $L = 255$. Probabilitas setiap *pixel* pada level ke i dapat dinyatakan dalam persamaan 2.5 [14].

$$p_i = \frac{n_i}{N} \dots\dots\dots (2.2)$$

keterangan :

n_i menyatakan jumlah *pixel* pada level ke i

N menyatakan total jumlah *pixel* pada citra.

Nilai momen kumulatif ke nol $\omega(k)$ dapat dinyatakan dalam persamaan 2.3, momen kumulatif ke satu $\mu(k)$ dapat dinyatakan dalam persamaan 2.4, dan nilai rata-rata μ_T dapat dinyatakan dalam persamaan 2.5 [14].

$$\omega(k) = \sum_{i=1}^k p_i \dots\dots\dots (2.3)$$

$$\mu(k) = \sum_{i=1}^k i \cdot p_i \dots\dots\dots (2.4)$$

$$\mu_T = \sum_{i=1}^L i \cdot p_i \dots\dots\dots (2.5)$$

Nilai ambang k dapat ditentukan dengan memaksimumkan persamaan 2.7.

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} \sigma_B^2(k) \dots\dots\dots (2.6)$$

Dimana nilai $\sigma_B^2(k)$ dapat dihitung menggunakan persamaan

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \dots\dots\dots (2.7)$$

e. Penapis Luas

Citra biner yang dihasilkan pada proses pengambangan (*thresholding*) kadang masih mengandung beberapa daerah yang dianggap sebagai gangguan. Daerah gangguan itu

biasanya berukuran kecil. Penapis luas dapat digunakan untuk menghilangkan daerah gangguan tersebut [10].

Segmentasi Citra Biner

Segmentasi citra biner bertujuan untuk mengelompokkan *pixel-pixel* objek menjadi wilayah yang merepresentasikan suatu objek [10]. Batas antara objek dengan latar belakang terlihat jelas pada citra biner terlihat sangat jelas. *Pixel* objek berwarna hitam sedangkan *pixel* latar belakang berwarna putih. Pertemuan antara *pixel* hitam dengan *pixel* putih dimodelkan sebagai segmen garis. Penelusuran batas wilayah dianggap sebagai pembuatan rangkaian keputusan untuk bergerak lurus, belok kiri, atau belok kanan.

Normalisasi

Normalisasi merupakan salah satu tahap dalam *preprocessing* citra yang dilakukan sebelum masuk ke proses pengenalan. Proses ini dilakukan dengan tujuan untuk menyesuaikan data citra masukan dengan data citra pada basis data. Proses normalisasi disesuaikan dengan kebutuhan pada proses pengenalan yang digunakan [3]. Salah satu proses normalisasi yang paling sederhana dalam pengenalan pola adalah normalisasi ukuran citra. Sebagai contoh, pada proses pengenalan pola menggunakan algoritma *template matching*, ukuran citra input harus disesuaikan dengan ukuran citra yang ada di basis data.

Bicubic Interpolation

Bicubic interpolation merupakan suatu metode interpolasi yang menggunakan 4 x 4 *pixel* tetangga sebagai informasi untuk menentukan nilai *pixel* yang baru [13]. Interpolasi adalah sebuah metode pencitraan untuk meningkatkan atau mengurangi jumlah *pixel* dalam citra digital. Interpolasi citra mencoba untuk mencapai pendekatan yang terbaik dari sebuah *pixel* berdasarkan nilai-nilai di sekitar *pixel*.

Template Matching Correlation

Template matching adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu

citra dengan citra yang menjadi *template* (acuan) [14]. Citra masukan dibandingkan dengan citra *template* yang ada di dalam basis data, kemudian dicari kesamaannya dengan menggunakan suatu aturan tertentu [2]. Pencocokkan citra yang menghasilkan tingkat kemiripan / kesamaan yang tinggi menentukan suatu citra tersebut dikenali sebagai salah satu dari citra *template*.

Template matching memiliki kelebihan dan kekurangan. Kelebihannya adalah algoritma ini mudah untuk dituliskan ke dalam bahasa program dan mudah untuk mempersiapkan data referensinya. Komputasi tidak terlalu besar karena data yang digunakan berupa matriks. Namun, dibalik kelebihannya itu algoritma ini secara umum memiliki kekurangan. Kekurangannya adalah membutuhkan data referensi atau basis data yang banyak untuk mendapatkan hasil yang optimal. Basis data bisa berupa citra ataupun citra yang telah dijadikan matriks. Semakin banyak jenis huruf yang ingin kita deteksi, maka semakin banyak data referensi yang harus disimpan.

Kesamaan antar dua buah matriks citra dapat dihitung nilainya dengan menghitung nilai korelasinya (*correlation*). Nilai korelasi dua buah matriks dapat dihitung dengan menggunakan rumus 2.8.

$$r = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i) \cdot (x_{jk} - \bar{x}_j)}{\sqrt{[\sum_{k=1}^n (x_{ik} - \bar{x}_i)^2 \cdot \sum_{k=1}^n (x_{jk} - \bar{x}_j)^2]}} \dots (2.8)$$

Dimana \bar{x}_i dirumuskan dengan persamaan 2.9 dan \bar{x}_j dirumuskan dengan persamaan 2.10.

$$\bar{x}_i = \frac{1}{n} \sum_{k=1}^n x_{ik} \dots \dots \dots (2.9)$$

$$\bar{x}_j = \frac{1}{n} \sum_{k=1}^n x_{jk} \dots \dots \dots (2.10)$$

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*.
- \bar{x}_i adalah rata-rata nilai *pixel* matriks *i*.
- \bar{x}_j adalah rata-rata nilai *pixel* matriks *j*.

n menyatakan jumlah *pixel* dalam suatu matriks.

III. ANALISIS DAN PERANCANGAN SISTEM

Pada Analisis dan Perancangan Sistem akan dijelaskan kebutuhan untuk membangun sistem, yang meliputi analisis kebutuhan, DFD, perancangan proses OCR, dan perancangan antarmuka aplikasi.

Analisis Kebutuhan

Perangkat lunak *Optical Character Recognition* menggunakan algoritma *template matching correlation* digunakan untuk mengkonversi citra digital menjadi *file* teks (*.txt). Dalam membangun aplikasi OCR ini, diperlukan analisis kebutuhan yang jelas sebagai tujuan utamanya agar tidak keluar dari rencana yang telah ditetapkan. Beberapa kemampuan sistem OCR ini dapat didefinisikan sebagai berikut:

- Memiliki kemampuan untuk memasukkan *file* citra digital *RGB* maupun *grayscale* (*.bmp atau *.jpg) sebagai citra masukan.
- Memiliki kemampuan untuk menampilkan informasi citra masukan, yaitu meliputi nama, ukuran, lebar, tinggi, dan kedalaman *file* citra.
- Memiliki kemampuan untuk melakukan proses *preprocessing* citra digital yang meliputi proses binerisasi, segmentasi, dan normalisasi.
- Memiliki kemampuan untuk melakukan proses pengenalan pola karakter dengan menggunakan algoritma *template matching correlation*.
- Memiliki kemampuan untuk menghitung dan menampilkan lama waktu proses pengenalan karakter.
- Memiliki kemampuan untuk menulis dan menyimpan pola karakter yang sudah dikenali dalam bentuk *file* teks (*.txt) sebagai hasil akhir.

- Memiliki kemampuan untuk menampilkan hasil dari proses pengenalan melalui layar tampilan utama.

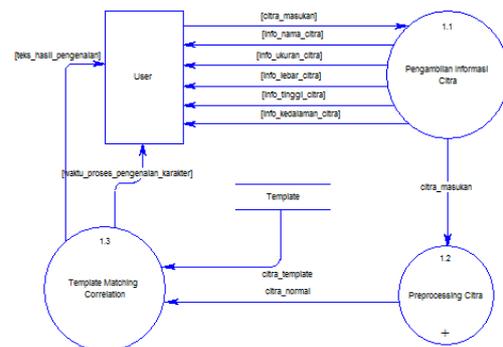
DFD

Data Flow Diagram level 0 berguna untuk menggambarkan secara umum perangkat lunak tersebut berjalan. Pendefinisian dengan menggunakan DFD level 0 memberikan gambaran data yang mengalir antara sistem dengan lingkungan yang digambarkan secara global. DFD level 0 aplikasi OCR dapat dilihat pada gambar 3.1.



Gambar 3.1 DFD level 0 aplikasi OCR

Dari *Data Flow Diagram* level 0 pada gambar 3.1, maka dapat dibuat sebuah *Data Flow Diagram* yang lebih rinci lagi, seperti pada gambar 3.2.

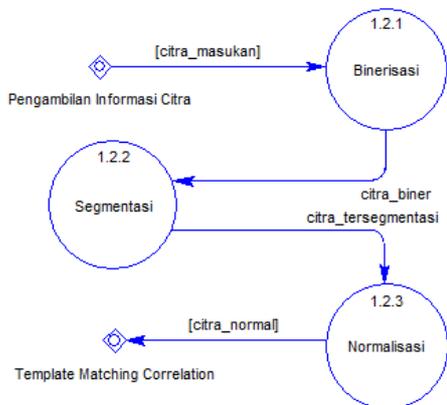


Gambar 3.2 DFD level 1 aplikasi OCR

Pada DFD level 1, dipecah menjadi tiga proses yang lebih rinci lagi yaitu :

- Pengambilan informasi citra
- Preprocessing citra
- Template Matching Correlation*

Gambaran mengenai Proses Preprocessing Citra dapat dilihat pada Gambar 3.3.



Gambar 3.3 DFD Level 2, Proses *Preprocessing* Citra

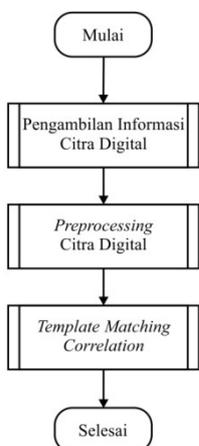
Pada proses 1.2 DFD level 1 dipecah menjadi proses yang lebih spesifik yaitu:

- a. Binerisasi
- b. Segmentasi
- c. Normalisasi

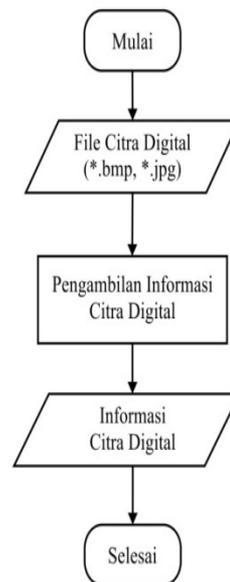
Perancangan Proses OCR

Terdapat tiga proses utama yang ada dalam proses OCR, yaitu:

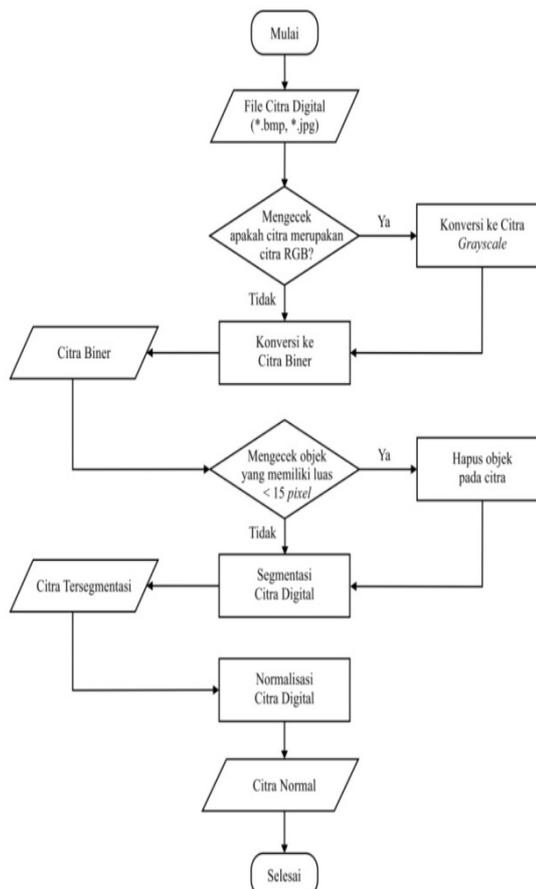
- a. Proses pengambilan informasi citra digital, dapat dilihat pada gambar 3.4.
- b. Proses *preprocessing* citra digital, dapat dilihat pada gambar 3.5.
- c. Proses *template matching correlation*, dapat dilihat pada gambar 3.6.
- d. Sedangkan *flowchart* secara global dari aplikasi OCR dapat dilihat pada gambar 3.4.



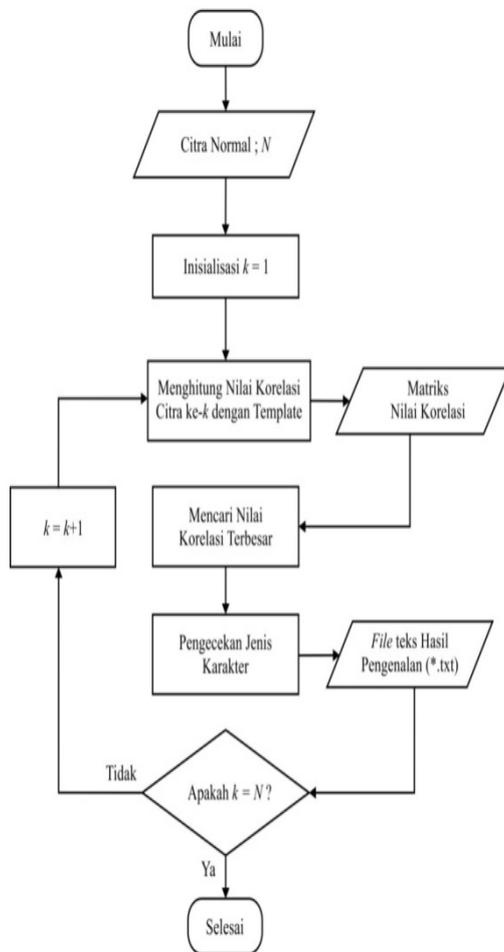
Gambar 3.4. *Global flowchart* proses OCR.



Gambar 3.5. *Flowchart* proses pengambilan informasi citra digital.



Gambar 3.6. *Flowchart* proses *preprocessing* citra digital.



Gambar 3.7. Flowchart proses *template matching correlation*.

Perancangan Antarmuka Aplikasi

Rancangan antarmuka dibuat sesederhana mungkin agar mempermudah dalam penggunaannya. Ada 3 jenis rancangan antarmuka yaitu rancangan *form* pembuka, rancangan *form* utama dan rancangan menu bantuan.

Rancangan Form Pembuka

Rancangan *form* pembuka meliputi judul tugas akhir, logo universitas, nama pembuat, nama program studi dan sebuah tombol (*pushbutton*). Gambar rancangan form pembuka yang dibuat seperti pada gambar 3.8.

Gambar 3.8 Rancangan *Form* Pembuka.

Rancangan Form Utama

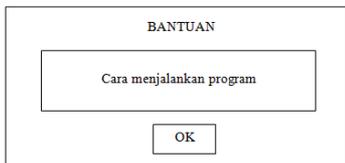
Rancangan *form* utama meliputi logo universitas, judul tugas akhir, sumbu citra masukan, informasi citra masukan, tombol buka citra, waktu proses pengenalan citra, tombol keluar, hasil pengenalan, nama pembuat, nomor induk mahasiswa dan menu bantuan. Gambar rancangan *form* utama yang dibuat seperti pada gambar 3.9.

Gambar 3.9 Rancangan *Form* Utama

Rancangan Menu Bantuan

Rancangan menu bantuan meliputi cara menjalankan program dan sebuah tombol masuk.

Gambar rancangan menu bantuan yang dibuat seperti pada gambar 3.10.



Gambar 3.10 Rancangan Menu Bantuan.

IV. PENGUJIAN, DAN ANALISIS HASIL

Pengujian

Pengujian adalah tahap yang penting dalam mengerjakan aplikasi perangkat lunak. Hasil yang diperoleh dapat dijadikan sebagai parameter yang penting dalam proses pengembangan selanjutnya jika masih dibutuhkan suatu penyempurnaan. Pengujian ini dilakukan untuk mengetahui hasil dari perangkat lunak yang telah dibuat.

Lingkungan Pengujian

Lingkungan pengujian terbagi dua macam yaitu perangkat keras dan perangkat lunak. Berikut spesifikasi masing-masing lingkungan pengujian.

Perangkat Keras

Komputer PC dengan spesifikasi sebagai berikut :

- a. Processor : *Intel Core 2 Duo* 1,86 GHz
- b. RAM : 4 GB DDR2
- c. VGA : *ATI Radeon HD5670* 512 MB DDR5

Perangkat Lunak

Aplikasi ini diujikan dalam lingkungan perangkat lunak sebagai berikut :

- a. Sistem operasi menggunakan *Microsoft Windows 7 Professional*.
- b. *Software* MATLAB 7.12 (R2011a)

Pelaksanaan Pengujian

Selama proses pelaksanaan pengujian perangkat lunak ini, terdapat serangkaian langkah-langkah

yang harus ditempuh. Proses pengujian dilakukan sebanyak tiga kali. Tiap pengujian menggunakan ukuran huruf yang berbeda, yaitu 12pt, 20pt, dan 36pt.

Adapun langkah-langkah yang harus ditempuh adalah sebagai berikut:

- a. Menyiapkan citra uji untuk pengujian pertama dengan ketentuan sebagai berikut:
- b. Citra uji terdiri dari 5 buah citra dengan jenis huruf yang berbeda, yaitu *Arial*, *Times New Roman*, *Comic Sans MS*, *Cambria*, dan *Courier New*. Untuk lebih jelasnya bisa dilihat pada gambar 4.10 sampai 4.15.
- c. Citra uji berisi sebuah paragraf yang terdiri dari 263 karakter (tidak termasuk tanda baca dan spasi).
- d. Ukuran huruf yang digunakan adalah 12pt dan *style* huruf yang digunakan adalah normal.

Hasil Pengujian

Data hasil pengenalan aplikasi OCR pada pengujian pertama (menggunakan huruf ukuran 12pt) dapat dilihat pada tabel 4.1.

Tabel 4.1. Persentase hasil pengenalan pada pengujian pertama (huruf 12pt)

| Citra masukan | Jumlah karakter yang dikenali | Total karakter pada citra | Persentase keberhasilan |
|--|-------------------------------|---------------------------|-------------------------|
| Citra_Uji_Arial_12pt.jpg | 249 | 263 | 94,68% |
| Citra_Uji_Times_12pt.jpg | 223 | 263 | 84,79% |
| Citra_Uji_Comic_12pt.jpg | 257 | 263 | 97,72% |
| Citra_Uji_Cambria_12pt.jpg | 221 | 263 | 84,03% |
| Citra_Uji_Courier_12pt.jpg | 263 | 263 | 100% |
| Rata-rata persentase keberhasilan | | | 92,24% |

Data hasil pengenalan aplikasi OCR pada pengujian kedua (menggunakan huruf ukuran 20pt) dapat dilihat pada tabel 4.2.

Tabel 4.2. Persentase hasil pengenalan pada pengujian kedua (huruf 20pt)

| Citra masukan | Jumlah karakter yang dikenali | Total karakter pada citra | Persentase Keberhasilan |
|--|-------------------------------|---------------------------|-------------------------|
| Citra_Uji_Arial_20pt.jpg | 252 | 263 | 95,82% |
| Citra_Uji_Times_20pt.jpg | 232 | 263 | 88,21% |
| Citra_Uji_Comic_20pt.jpg | 260 | 263 | 98,86% |
| Citra_Uji_Cambria_20pt.jpg | 232 | 263 | 88,21% |
| Citra_Uji_Courier_20pt.jpg | 254 | 263 | 96,58% |
| Rata-rata persentase keberhasilan | | | 93,54% |

Data hasil pengenalan aplikasi OCR pada pengujian ketiga (menggunakan huruf ukuran 36pt) dapat dilihat pada tabel 4.3.

Tabel 4.3. Persentase hasil pengenalan pada pengujian ketiga (huruf 36pt)

| Citra masukan | Jumlah karakter yang dikenali | Total karakter pada citra | Persentase Keberhasilan |
|--|-------------------------------|---------------------------|-------------------------|
| Citra_Uji_Arial_36pt.jpg | 251 | 263 | 95,44% |
| Citra_Uji_Times_36pt.jpg | 232 | 263 | 88,21% |
| Citra_Uji_Comic_36pt.jpg | 257 | 263 | 97,72% |
| Citra_Uji_Cambria_36pt.jpg | 222 | 263 | 84,41% |
| Citra_Uji_Courier_36pt.jpg | 260 | 263 | 98,86% |
| Rata-rata persentase keberhasilan | | | 92,93% |

Berdasarkan data hasil pengujian pada tabel 4.1 sampai dengan tabel 4.3, dapat disusun suatu tabel rata-rata persentase hasil pengenalan seperti pada tabel 4.4.

Tabel 4.4. Rata-rata persentase keberhasilan aplikasi OCR

| Pengujian | Rata-rata Persentase Keberhasilan |
|--|-----------------------------------|
| Pertama (Ukuran huruf 12pt) | 92,24% |
| Kedua (Ukuran huruf 20pt) | 93,54% |
| Ketiga (Ukuran huruf 36pt) | 92,93% |
| Total rata-rata persentase keberhasilan | 92,90% |

Analisis Hasil

Dari tabel 4.4 dapat diketahui bahwa rata-rata persentase keberhasilan pengenalan pada pengujian pertama sebesar 92,24%, pada pengujian kedua sebesar 93,54%, dan pada pengujian ketiga sebesar 92,93%. Sedangkan total rata-rata persentase keberhasilan pengenalan secara menyeluruh dari pengujian pertama sampai dengan pengujian ketiga sebesar 92,90%.

Tingkat keberhasilan pengenalan yang dihasilkan cukup tinggi meskipun jenis dan ukuran huruf yang digunakan sebagai masukan berbeda dengan *template*.

V. PENUTUP

Kesimpulan

Kesimpulan yang dapat diambil dalam pembuatan tugas akhir ini adalah:

1. Dihasilkan sebuah aplikasi *Optical Character Recognition* menggunakan algoritma *Template Matching Correlation*.

2. Algoritma *Template Matching Correlation* cukup efektif untuk pengenalan karakter huruf cetak. Rata-rata tingkat keberhasilan pengenalan yang dihasilkan sebesar 92,90%.

Saran

Penelitian ini masih dapat dikembangkan lebih lanjut. Beberapa saran untuk mengembangkan penelitian ini adalah sebagai berikut :

1. Menambahkan proses ekstraksi ciri yang sesuai sebelum proses *template matching correlation*, mungkin akan menghasilkan pengenalan yang lebih akurat.
2. Citra huruf *template* dapat dikembangkan dengan penggunaan huruf tulisan tangan, jadi tidak hanya terbatas pada huruf cetak saja.

VI. DAFTAR PUSTAKA

- [1] Balza, Achmad, _____, "Otsu Thresholding", diakses dari <http://balzach.staff.ugm.ac.id/vis/index.php/image-processing/segmentation/1-otsu-thresholding>, pada tanggal 20 Juli 2012, pukul 16.17 WIB.
- [2] Brunelli Roberto, 2009, "Template Matching Techniques In Computer Vision", John Willey & Sons. Inc.
- [3] Cheriet M., Kharna N., Liu C., Suen C.Y., 2006, "Character Recognition System A Guide for Student and Practioners", John Willey & Sons. Inc.
- [4] Eikvil Line, 1993, "Optical Character Recognition", Nork Regnesentral.
- [5] Gonzales Rafael C. dan Woods Richard E., 2001, "Digital Image Processing", Prentice Hall.
- [6] Hendry J., 2011, "Deteksi Kkarakter Pada Citra Digital (OCR) dengan Teknik Korelasi (Template Matching)", <http://www.scribd.com/doc/68866773/Project-Deteksi-Karakter-Pada-Citra-Digital-Ocr-Dengan-Teknik-Korelasi-Template-Matching>, pada tanggal 30 Juli 2012, pukul 08.20 WIB.
- [7] Keys, Robert G., 1981, "Cubic Convolution Interpolation for Digital Image Processing", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-29.
- [8] MathWorks, _____, "Convert RGB image or colormap to grayscale", diakses dari <http://www.mathworks.com/help/images/ref/rgb2gray.html>, pada tanggal 19 September 2012, pukul 11.45 WIB.
- [9] Moler, 2004, "MATLAB About", diakses dari <http://www.mathworks.com>, pada tanggal 30 Juli 2012, pukul 14.50 WIB.
- [10] Munir, Rinaldi, 2004, "Pengolahan Citra Digital dengan Pendekatan Algoritmik", Informatika, Bandung.
- [11] Nurwanto T. B., 2007, "Pengenalan Huruf Tulisan Tangan Menggunakan Logika Fuzzy Dengan Pendekatan Neural Networks Back Propagation", STT Telkom, Bandung.
- [12] Parno, _____, "DFD", diakses dari http://parno.staff.gunadarma.ac.id/Downloads/files/4395SI_03_DFD.pdf, pada tanggal 11 Juli 2012, pukul 12.17 WIB.
- [13] Purnomo, Seno, 2009, "Interpolasi", diakses dari <http://senosuke.wordpress.com/2009/12/2/interpolasi/>, pada tanggal 21 September 2012, pukul 10.12 WIB.
- [14] Putra, Darma, 2010, "Pengolahan Citra Digital", Penerbit Andi, Yogyakarta.
- [15] Sdarsono, _____, "Flowchart", diakses dari sdarsono.staff.gunadarma.ac.id/Downloads/files/16512/Flowchart.pdf, pada tanggal 11 Juli 2012, pukul 12.17 WIB.
- [16] Sofani, Rachmah, 2009, "Sistem OCR", http://digilib.itelkom.ac.id/index.php?option=com_content&view=article&id=413:sistem-ocr&catid=15:pemrosesan-sinyal&Itemid=14, pada tanggal 30 Maret 2012, jam 08.10 wib.

- [17] Sugiharto, Aris, 2006, "*Pemrograman GUI dengan MATLAB*", Penerbit Andi, Yogyakarta.
- [18] Wardhana A. W., Prayudi Y., 2008, "*Penggunaan Metode Template Matching Untuk Identifikasi Kecacatan Pada PCB*", Universitas Islam Indonesia, Yogyakarta.
- [19] Wirayuda T. A. B., Vaulin S., dan Novi D. R., 2009, "*Pengenalan Huruf Komputer Menggunakan Algoritma Berbasis Chain Code dan Algoritma Sequence Alignment*", IT Telkom, Bandung.

