

## IDENTIFIKASI TEKS DOKUMEN MENGGUNAKAN METODE PROFILE PROJECTION DAN TEMPLATE MATCHING

<sup>[1]</sup>Erwin Wahyudi, <sup>[2]</sup>Dedi Triyanto, <sup>[3]</sup>Ikhwan Ruslianto

<sup>[1][2][3]</sup>Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

Jl. Prof. Dr. H. Hadari Nawawi, Pontianak

Telp./Fax.: (0561) 577963

e-mail:

<sup>[1]</sup>erwin.wahyudi92@gmail.com, <sup>[2]</sup>dedi.triyanto@siskom.untan.ac.id,

<sup>[3]</sup>ikhwanruslianto@siskom.untan.ac.id

### Abstrak

Dokumen merupakan unit informasi terekam yang terstruktur, diterbitkan atau tidak diterbitkan dalam bentuk data mentah atau bentuk elektronik dan dikelola sebagai suatu informasi. Mengolah dokumen mentah menjadi dokumen digital diperlukan media alat pemindai atau scanner. Namun, hasil dari pemindaian dokumen mentah adalah data gambar dan bukan dalam bentuk teks digital sehingga tidak bisa diolah kembali. Oleh karena itu, penelitian ini bertujuan menghasilkan teks digital dari dokumen hasil pemindaian dengan memanfaatkan cabang ilmu pengolahan citra digital. Metode yang digunakan untuk mengenali teks dokumen adalah metode template matching dan metode median filter untuk membersihkan noise yang terdapat pada dokumen serta metode profile projection untuk segmentasi karakter. Pengujian dilakukan terhadap 26 dokumen dengan 7 jenis huruf yang berbeda dengan ukuran huruf 12pt, 14pt dan 16pt serta 1 pengujian terhadap dokumen dengan kondisi buatan seperti penambahan noise, teks yang kabur dan teks yang terpotong. Metode median filter melakukan pembersihan dengan baik yang menghasilkan presentase keberhasilan pengenalan dengan metode template matching yaitu 95,3% untuk 26 dokumen dan 92% untuk kondisi dokumen buatan.

**Kata kunci:** Pengolahan Citra, Template Matching, Median Filter, Profile Projection.

### 1. PENDAHULUAN

Dokumen adalah informasi atau objek terekam yang dapat diperlakukan sebagai suatu kesatuan. Dokumen merupakan unit informasi terekam yang terstruktur, diterbitkan atau tidak diterbitkan, dalam bentuk data mentah atau bentuk elektronik, dan dikelola sebagai unit diskrit dalam sistem informasi [1].

Dokumen yang dibuat biasanya perlu dilakukan proses pengelolaan berupa dokumentasi. Dokumentasi adalah pengumpulan, pemilihan, pengolahan, dan penyimpanan informasi dari berbagai bidang atau juga pengumpulan bukti dan keterangan-keterangan seperti gambar, kutipan, guntingan koran, bahan referensi, dan lain-lain.

Sebelum teknologi mengalami kemajuan yang pesat seperti sekarang, dokumentasi data masih dilakukan secara manual dengan menggunakan buku catatan, dan tentu saja hal itu telah membuat manusia semakin sulit dalam melakukan

proses manipulasi data, seperti pencarian data, pemasukan data, dan lain-lain.

Proses dokumentasi sekarang dipermudah dengan berkembangnya teknologi seperti dengan ditemukannya *printer scanner* yang berfungsi untuk memindai dokumen cetak menjadi dokumen digital sehingga dokumentasi dapat dilakukan tidak hanya dalam bentuk data mentah seperti kertas tetapi juga secara digital. Data hasil proses pemindaian dapat dibaca di media elektronik biasanya dengan format gambar atau format *.pdf*. Dalam proses dokumentasi menggunakan *printer scanner* memiliki permasalahan, dimana teks hasil *scan* dalam format gambar atau *.pdf* tidak dapat diolah kembali sesuai dengan kebutuhan yang ada.

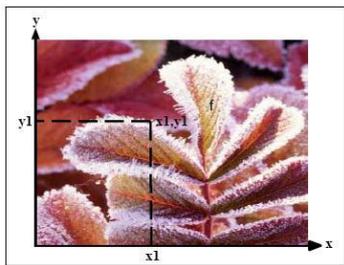
Berdasarkan permasalahan tersebut, dalam penelitian ini akan dibuat aplikasi untuk mengenali teks dokumen menggunakan metode dalam bidang pengolahan citra digital dengan harapan dokumen dapat didokumentasikan menjadi

teks dalam komputer sehingga dapat diolah kembali, baik dengan format yang sama atau dengan merubah seluruh format yang ada pada dokumen sebelumnya. Aplikasi yang dibuat, tidak hanya mengenali teks tetapi juga akan berfokus pada proses pembersihan *noise* atau derau pada dokumen sehingga setiap huruf yang ada dapat dikenali dengan baik pada saat proses pengenalan karakter.

## 2. TINJAUAN PUSTAKA

### 2.1 Representasi Citra dan Komponen Citra

Citra digital merupakan citra yang memiliki fungsi dua variabel,  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat dan nilai  $f(x,y)$  adalah intensitas citra pada koordinat tersebut. Contoh citra digital dapat dilihat pada Gambar 2.1.



Gambar 2.1 Citra Digital.

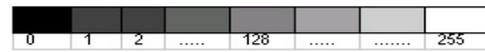
Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (pixel atau “picture element”). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi.

#### 2.1.1 Citra Warna

Citra berwarna atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang bisa disajikan mencapai  $255 \times 255 \times 255$  atau 16.581.375 warna.

#### 2.1.2 Citra Grayscale

Citra grayscale adalah sebuah citra skala keabuan dengan nilai intensitas paling besar 255 berwarna putih hingga warna hitam dengan nilai intensitas paling kecil (0) seperti yang terlihat pada gambar 2.2 :

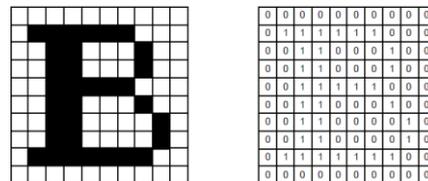


Gambar 2.2 Skala yang digunakan pada grayscale.

Citra digital dengan skala keabuan 8-bit memiliki  $2^8 = 256$  kemungkinan warna, yaitu 0 (minimal) hingga 255 (maksimal) [2].

#### 2.1.3 Citra Biner

Citra biner adalah citra yang hanya mempunyai dua nilai derajat keabuan, hanya hitam dan putih. Piksel-piksel objek bernilai 1 dan piksel-piksel latar belakang bernilai 0. Pada waktu menampilkan gambar, 1 adalah putih sedangkan 0 adalah hitam.



Gambar 2.3 Huruf “B” dan citra dalam biner.

#### 2.1.4 Konversi Citra

Mengubah citra berwarna adalah dengan mengambil masing-masing nilai  $r$ ,  $g$  dan  $b$  menjadi citra grayscale. Salah satu metode yang baik digunakan untuk mengkonversi citra berwarna ke citra abu-abu adalah dengan menggunakan persamaan (1) berikut [3] :

$$Gray = (0.2989 \times R) + (0.587 \times G) + (0.114 \times B) \quad (1)$$

Proses konversi ke citra biner bertujuan untuk memisahkan objek dengan latar belakang. Pada tahap ini, citra grayscale diproses untuk menjadi citra biner (black and white) dimana tiap piksel citra diwakili oleh dua nilai, yaitu satu dan nol. Nilai satu mewakili warna putih, sedangkan nilai nol mewakili warna hitam. Untuk mengubah citra abu-abu ke citra biner, dibutuhkan nilai thresholding atau pengambangan. Tiap piksel pada citra grayscale dibandingkan nilainya dengan nilai threshold. Apabila lebih besar, maka piksel akan diberi nilai satu, bila lebih kecil maka piksel diberi nilai nol [4].

### 2.1.5 Thresholding

*Thresholding* adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan latar belakang dari citra secara jelas. Metode yang umum digunakan dalam proses pengembangan citra adalah metode Otsu. Formulasi dari metode *Otsu* adalah Nilai ambang yang akan dicari dari suatu citra *gray level* dinyatakan dengan  $k$ . nilai  $k$  berkisar antara 1 sampai  $L$ , dengan nilai  $L=255$ . Probabilitas untuk piksel  $i$  dinyatakan dengan [5]:

$$P_i = \frac{n_i}{N} \quad (2)$$

$n_i$  = menyatakan jumlah piksel dengan tingkat keabuan  $i$

$N$  = menyatakan total jumlah piksel pada gambar

Nilai momen kumulatif ke-nol (3), nilai momen kumulatif ke-satu (4) dan nilai rata-rata (5) berturut – turut adalah :

$$\omega(k) = \sum_{i=1}^k P_i \quad (3)$$

$$\mu(k) = \sum_{i=1}^k i \cdot P_i \quad (4)$$

$$\mu_T = \sum_{i=1}^L i \cdot P_i \quad (5)$$

Nilai ambang  $k$  dapat ditentukan dengan memaksimumkan persamaan (6) :

$$\text{dengan } \sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (6)$$

## 2.2 Analisis Citra

Analisis Citra adalah kegiatan menganalisis citra sehingga dapat menghasilkan informasi. Informasi adalah keadaan dari suatu citra, berdasarkan informasi yang didapat, kemudian dapat diambil keputusan pengolahan apa yang akan dilakukan terhadap citra tersebut.

### 2.2.1 Median Filter

*Median filter* merupakan teknik pembersihan *noise*. Konsep dasarnya adalah dengan menemukan nilai piksel yang memiliki nilai intensitas dari suatu piksel yang berbeda dengan nilai piksel yang ada di daerah sekitarnya, dan menggantinya dengan nilai yang lebih cocok. Cara yang paling sederhana dalam mencapainya

adalah dengan melakukan pencegahan atau pembatasan nilai piksel, sehingga suatu piksel tidak memiliki nilai intensitas yang diluar nilai yang ada disekitarnya [6].

## 2.3 Segmentasi Citra

Salah satu metode segmentasi pada citra adalah metode *profile projection*. Metode ini sering dipakai untuk segmentasi karakter pada citra baik OCR maupun citra hasil buatan sendiri menggunakan komputer. Teknik Segmentasi wajib dilakukan ketika kita ingin mendeteksi karakter (huruf/angka) pada citra, karena algoritma yang akan digunakan untuk ekstraksi ciri dan pengenalan karakter hanya bisa mengolah per-karakter saja. Segmentasi bisa dilakukan sebelum atau sesudah pre-processing pada keseluruhan citra [7].

## 2.4 Normalisasi Citra

Hasil karakter setelah proses segmentasi memiliki ukuran yang tidak sama untuk masing-masing karakter sehingga diperlukan proses normalisasi. Normalisasi adalah proses mengubah ukuran citra, baik menambah atau mengurangi, menjadi ukuran yang ditentukan tanpa menghilangkan informasi penting dari citra tersebut. Dengan adanya proses normalisasi maka ukuran semua citra yang akan diproses menjadi seragam.

## 2.5 Pengenalan Pola

Secara umum pengenalan pola (pattern recognition) adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek [8].

*Template matching* adalah sebuah teknik pengenalan pola dalam pengolahan citra digital untuk menemukan bagian-bagian kecil dari gambar yang cocok dengan *template* gambar. *Template matching* merupakan tahap dimana citra *input* yang telah dinormalisasi dilakukan perbandingan dengan setiap *template* yang ada dan dihitung nilai korelasinya terhadap setiap *template*. Berikut persamaan (7) merupakan rumus untuk mencari nilai korelasi  $r$  antar dua buah matriks dua dimensi [9] :

$$r = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i) \cdot (x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{ik} - \bar{x}_i)^2 \cdot \sum_{k=1}^n (x_{jk} - \bar{x}_j)^2}} \quad (7)$$

Dimana :

$$\bar{x}_i = 1/n \sum_{k=1}^n x_{ik}$$
$$\bar{x}_j = 1/n \sum_{k=1}^n x_{jk}$$

Keterangan :

- $r$  : adalah nilai korelasi antara dua buah matriks (rentang nilai antara -1 dan +1)
- $x_{ik}$  : adalah nilai piksel ke- $k$  dalam matriks  $i$
- $x_{jk}$  : adalah nilai piksel ke- $k$  dalam matriks  $j$
- $\bar{x}_i$  : adalah rata-rata nilai piksel matriks  $i$
- $\bar{x}_j$  : adalah rata-rata nilai piksel matriks  $j$
- $n$  : adalah jumlah piksel dalam suatu matriks

### 3. METODOLOGI PENELITIAN

#### 3.1 Studi pustaka

Melakukan studi pustaka dengan mengkaji buku-buku yang berkaitan dengan teori-teori tentang pengolahan citra digital, pengenalan pola serta bahasa pemrograman Matlab untuk mendukung penelitian ini. Selain buku, peneliti juga mengambil referensi dari jurnal ilmiah dan berbagai sumber di internet sebagai acuan untuk melakukan penelitian.

#### 3.2 Analisa Kebutuhan

Tahap ini dilakukan agar apa yang akan dibuat atau dirancang sesuai dengan kebutuhan sistem berdasarkan data-data pada tahap-tahap perancangan. Analisa kebutuhan meliputi kebutuhan perangkat keras dan perangkat lunak. Analisa perangkat keras terdiri dari jenis alat pemindai dan spesifikasi komputer yang digunakan. Analisa perangkat lunak terdiri dari perangkat lunak yang digunakan untuk membangun aplikasi pengenalan teks dokumen.

#### 3.3 Desain Perangkat Lunak

Merancang alir kerja (workflow) dari sistem aplikasi pengenalan teks dokumen, merancang *input* output aplikasi (interface)

dan menentukan form-form dari setiap modul yang ada. Pada tahap ini dilakukan perancangan diagram alir dari aplikasi pengenalan teks dan perancangan antarmuka sistem. Diagram alir terdiri dari dua yaitu diagram untuk pelatihan data dan diagram alir bagaimana aplikasi bekerja dan antarmuka sistem dibangun.

#### 3.4 Implementasi

Pada tahap implementasi, hasil dari perencanaan dan analisa kebutuhan diproses untuk dijadikan sebuah sistem secara keseluruhan. Diagram alir yang telah dirancang, yaitu diagram alir untuk pelatihan data karakter dan diagram alir aplikasi, diimplementasikan dalam barisan kode program. Barisan kode program ini akan menghasilkan sebuah aplikasi secara keseluruhan yang dapat memproses masukan data dokumen dalam bentuk gambar dan memberi keluaran berupa data karakter.

#### 3.5 Pengujian

Setelah tahap implementasi maka selanjutnya dilakukan pengujian. Pada tahap ini akan dilakukan pengujian terhadap 26 sampel terhadap 3 dokumen yang berbeda dengan 7 jenis huruf yang berbeda dengan ukuran 12pt dan 2 dokumen dengan ukuran huruf 14pt dan 16pt, serta 1 kondisi buatan dimana teks diberi noise dan dilakukan proses bluring.

#### 3.6 Analisis dan Kesimpulan

Analisis dilakukan untuk mendapatkan kebutuhan-kebutuhan (requirements) guna dalam pengembangan sistem lebih lanjut. Selain itu juga analisis dilakukan untuk mempelajari dan memahami gambaran umum tentang aplikasi yang menerapkan metode yang digunakan dalam proses pengenalan karakter. Setelah dilakukan analisis, kemudian ditarik kesimpulan terhadap keberhasilan sistem yang dibuat.

### 4. PERANCANGAN SISTEM

#### 4.1 Pengambilan Sampel

Sampel diambil dengan memindai dokumen A4 menggunakan *printer* tipe MP287. Sampel hasil *scanning* memiliki ukuran 2500x3000 piksel dengan besar *file* 1,6-3Mb. Ukuran tersebut relatif besar

sehingga dapat menghabiskan memori yang besar dan waktu yang lama dalam proses pengolahan. Semakin tinggi resolusi sebuah citra, semakin baik kualitas citra tersebut. Namun, tingginya resolusi menyebabkan semakin banyaknya jumlah bit yang diperlukan untuk menyimpan dan mentransmisikan data citra tersebut. Berdasarkan hal tersebut, sampel akan dikonversi ke skala yang lebih kecil dari ukuran aslinya dengan besar *file* 1 Mb dengan rata-rata ukuran resolusi sebesar 1250x1750 piksel. Jika resolusi piksel terlalu kecil maka akan mengganggu kualitas citra.

Sampel terdiri dari 26 sampel dengan 3 dokumen format berbeda. Setiap dokumen terdiri dari 7 sampel dengan 7 jenis huruf yang berbeda yang umum digunakan dalam dokumen. Dokumen tersebut terdiri dari dokumen Surat Perjanjian Kerja Sama, dokumen latar belakang Laporan Kerja Praktek dan dokumen materi perkuliahan. Jenis huruf tersebut yaitu Arial, Calibri, Calibri Light (Headings), Comic Sans, Lucida Sans Ms, Times New Roman dan Tahoma dengan ukuran huruf yaitu 12pt dan 2 dokumen dengan ukuran masing-masing huruf 14pt dan 16pt. Sampel terakhir merupakan sampel jenis huruf Arial yang terdiri dari 1 paragraf dengan kondisi buatan yaitu, teks yang terpotong, teks yang kabur dan noise pada dokumen.

#### 4.2 Perancangan Flowchart

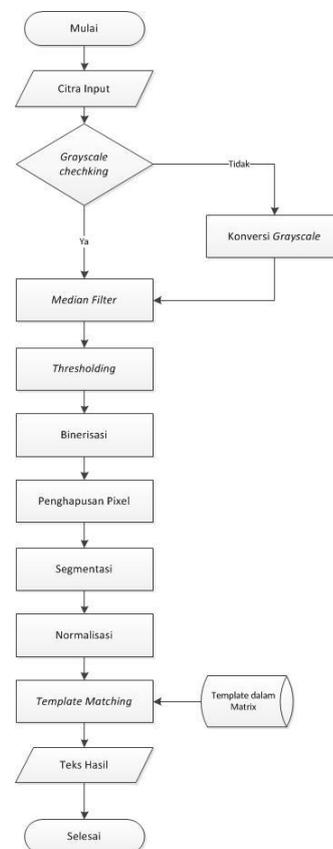
Mengenal karakter menggunakan metode *template matching*, dibutuhkan tahapan algoritma. Algoritma dari sistem pengenalan karakter yang dibangun adalah sebagai berikut :

1. Melakukan *Grayscale Checking* terhadap citra *input*, jika citra merupakan citra berwarna, maka citra terlebih dahulu dikonversi ke skala abu-abu menjadi 1 nilai dengan rentang 0-255.
2. Jika citra adalah citra abu-abu, proses selanjutnya melakukan pembersihan *noise* dengan teknik *median filter* dengan masking ukuran 3x3.
3. Melakukan proses pengambangan atau *thresholding* terhadap setiap piksel citra dengan metode Otsu kemudian

melakukan proses binerisasi yaitu mengubah citra ke skala hitam putih.

4. Selanjutnya memastikan kembali citra bersih dari noise dengan menghapus piksel yang tidak dibutuhkan.
5. Menjumlahkan setiap baris dan kolom citra untuk memisahkan karakter dengan metode *profile projection*.
6. Melakukan proses normalisasi karakter, mengubah ukuran setiap karakter menjadi 42x24 piksel sesuai dengan ukuran *database*.
7. Melakukan pengenalan karakter dengan metode *template matching* dengan membandingkan korelasi antar karakter masukan dengan karakter *database*.
8. Mencari nilai korelasi terbesar antara citra *input* karakter dengan citra *template*. Nilai korelasi antara -1 hingga +1. Jika korelasi mendekati 1 atau sama dengan 1 berarti karakter tersebut cocok dengan *database*, kemudian dicetak karakter sebagai hasil keluaran.

Flowchart dari sistem pengenalan karakter dengan metode *template matching* dapat dilihat pada gambar 4.1 berikut :



Gambar 4.1 Flowchart Aplikasi

1. *Grayscale checking*

Proses *pre-processing* adalah proses perbaikan citra yang mengkonversi citra *input* menjadi citra yang menghasilkan citra dengan kontras yang optimal sehingga mempermudah citra untuk diolah ke tahap selanjutnya. *Pre-processing* dalam pengenalan karakter terdiri dari beberapa tahap yaitu *median filter*, *threshold* dan binerisasi.

Sebelum dilakukan proses *median filter*, citra *input* terlebih dahulu dilakukan pengecekan apakah citra tersebut merupakan citra berwarna (RGB) atau citra berskala keabuan (*grayscale*) dengan mengetahui ukuran dari citra. Jika ukuran citra memiliki 3 nilai, citra terbaca sebagai citra berwarna (RGB), kemudian citra dikonversi ke skala abu-abu.

2. Pembersihan *noise* dengan *median filter*

Jika citra terbaca sebagai citra abu-abu, proses dilanjutkan ke *median filter* untuk menghilangkan *noise* melakukan pembersihan *noise* dengan matriks 3x3, mengurutkan nilai dari matriks tersebut, kemudian mencari nilai tengah dari matrix tersebut.

*Median Filter* merupakan proses untuk menghilangkan *noise* pada gambar. *Noise* yang dihilangkan yaitu jenis *salt and pepper*, ciri dari *noise* ini membentuk seperti kumpulan titik tidak beraturan.

3. *Threshold* dan Binerisasi

Proses *threshold* atau proses pengambangan merupakan proses menjadikan citra skala abu-abu menjadi citra dengan nilai biner. Pada tahap ini, citra *grayscale* diproses untuk menjadi citra hitam putih dimana tiap piksel citra diwakili oleh dua nilai, yaitu satu dan nol. Nilai satu mewakili warna putih, sedangkan nilai nol mewakili warna hitam. Nilai *threshold* dihitung dengan menggunakan metoda Otsu.

Setelah nilai *threshold* didapat, setiap piksel citra *grayscale* akan dibagi dengan nilai 255 kemudian dibandingkan dengan nilai *threshold*. Jika nilai pembagian piksel *grayscale* lebih besar dari nilai *threshold* maka piksel akan diberi nilai 1 dan sebaliknya

jika lebih kecil maka akan diberikan nilai nol, proses ini dikenal dengan proses binerisasi.

4. Penghapusan Piksel

Penghapusan piksel bertujuan untuk menghapus piksel latar belakang yang tidak dibutuhkan dalam proses pengenalan karakter. Penghapusan piksel juga melakukan pembersihan kembali *noise* yang tersisa dari proses *median filter*. Pada aplikasi pengenalan teks yang dibuat, penghapusan piksel setiap jenis huruf berbeda-beda, tergantung dari ukuran piksel sisa *noise* yang tertinggal. Penghapusan piksel terhadap masing-masing jenis huruf, dibuat 2 rentang yaitu 5 piksel terhubung dan 10 piksel terhubung. Rentang diambil berdasarkan jumlah piksel maksimal dari *noise* yang tersisa dan juga berdasarkan karakteristik ketebalan karakter. Jika penghapusan melebihi 10 piksel terhubung, beberapa citra karakter akan terhapus sehingga mengalami kegagalan dalam proses pengenalan karakter. Komponen piksel terhubung dari masing-masing jenis huruf sebagai berikut :

- Huruf Arial, Calibri dan Times New Roman yaitu 5 piksel.
- Huruf Comic Sans Ms, Lucida Sans dan Tahoma yaitu 10 piksel.

5. Segmentasi

Pada proses segmentasi ini dibagi menjadi dua bagian, yaitu segmentasi baris dan segmentasi karakter. Segmentasi baris menelusuri dokumen dan melakukan pemotongan baris secara horizontal. Sedangkan segmentasi karakter, melakukan pemotongan karakter dari hasil segmentasi baris.

Segmentasi baris melakukan pemisahan baris terhadap citra *input* yang kemudian dibagi menjadi beberapa bagian sesuai dengan jumlah baris teks yang ada pada citra.

6. Normalisasi

Normalisasi merupakan proses untuk merubah resolusi citra *input* sesuai ukuran yang diinginkan yang kemudian akan dibandingkan (pada database). Normalisasi dilakukan terhadap masing-masing karakter yang telah di-*cropping*

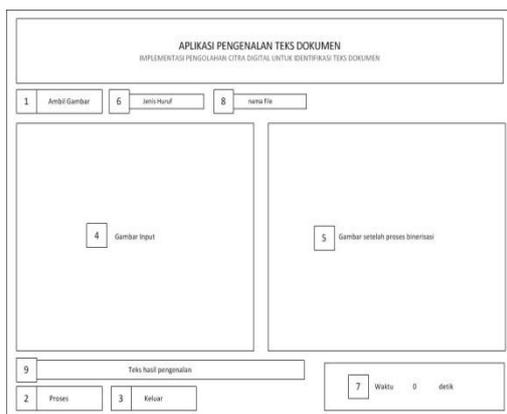
pada tahap sebelumnya. Ukuran pemotongan karakter disesuaikan dengan ukuran skala piksel dari *template* yang ada. Citra karakter hasil *cropping* yang diubah sesuai dengan ukuran *template* yaitu 42x24 piksel.

#### 7. Pengenalan Karakter

Pengenalan karakter menggunakan metode *template matching* dengan membandingkan dua variabel/matriks. Algoritma ini biasa digunakan untuk mencocokkan dua matriks citra, dengan akurasi yang cukup besar/tepat. Teknik yang dipakai untuk membandingkan karakter *input* dengan karakter *template* yaitu teknik korelasi. Nilai korelasi memiliki rentang -1 hingga +1. Nilai korelasi yang paling besar, memiliki kemungkinan terbesar dimana citra masukan memiliki kemiripan dengan citra *template*.

### 4.3 Perancangan Antar Muka

Perancangan antarmuka merupakan gambaran rancangan awal bagaimana struktur dari aplikasi pengenalan teks akan dibuat. Perancangan antar muka bertujuan untuk mempermudah menentukan modul apa yang akan digunakan dalam membangun aplikasi. Perancangan aplikasi dapat dilihat pada Gambar 4.2.



Gambar 4.2 Perancangan antarmuka aplikasi.

Fungsi dari masing-masing modul pada gambar 4.2 sesuai nomor adalah sebagai berikut :

1. Tombol Ambil Gambar : tombol ini digunakan untuk mengunggah gambar dari direktori *folder* yang akan dijadikan *input* pengolahan.

2. Tombol Proses : untuk menjalankan proses pengenalan karakter.
3. Tombol Keluar : untuk keluar dari aplikasi.
4. Gambar *Input Axes* : untuk menampilkan gambar yang dipilih dari *folder* (gambar *input*).
5. Gambar setelah proses binerisasi : axes untuk menampilkan gambar setelah proses binerisasi
6. Jenis Huruf : terdiri jenis huruf yang dapat dipilih sesuai dengan jenis karakter yang akan dikenali
7. Waktu : untuk menampilkan lama proses pengenalan teks.
8. Nama *File* : untuk menampilkan nama *file* yang diunggah.
9. Teks Hasil Pengenalan : untuk menampilkan teks hasil pengenalan dalam bentuk baris kalimat.

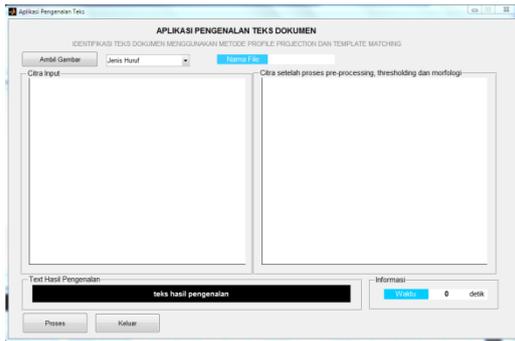
## 5. PENGUJIAN DAN ANALISIS

### 5.1 Pelatihan Data

Proses pelatihan data dilakukan membuat citra karakter yang terdiri dari huruf besar, huruf kecil, angka dan tanda baca. Pelatihan data selanjutnya dengan melakukan *cropping* karakter terhadap karakter tersebut sesuai dengan ukuran piksel 42x24 untuk masing-masing karakter. Karakter dari sampel yang tidak berhasil dikenali juga akan dijadikan data latih. Jumlah keseluruhan data latih yang dijadikan *template* berjumlah 122 data karakter. Data tersebut terdiri dari huruf, angka dan tanda baca.

### 5.2 Tampilan Antar Muka

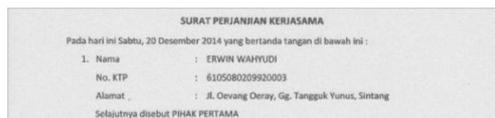
Aplikasi akan berjalan ketika tombol "Proses" ditekan. Selanjutnya aplikasi mulai mengenali satu-persatu karakter dan akan tampil di kotak berwarna hitam atau kotak "Text Hasil Pengenalan". Setelah proses pengenalan selesai, seluruh hasil teks akan tampil didalam *notepad* dengan nama *file* text.txt dan waktu di kotak informasi akan tampil sesuai dengan lamanya proses pengenalan. Tampilan antarmuka dapat dilihat pada gambar 5.1.



Gambar 5.1 Tampilan antar muka Aplikasi Pengenalan Teks.

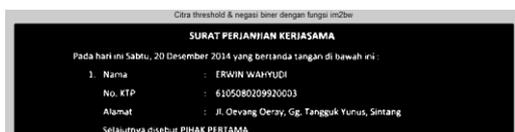
### 5.3 Hasil Pre-processing

Pada bagian ini, dilakukan pengujian terhadap proses kerja *median filter* dalam membersihkan *noise*. Citra hasil *median filter* merupakan citra dalam bentuk citra keabuan. Citra hasil *median filter* dapat dilihat pada gambar 5.2 berikut.



Gambar 5.2 Citra hasil konversi ke skala abu-abu.

Citra hasil konversi dari skala abu-abu ke skala biner, dapat dilihat pada gambar 5.3 berikut.



Gambar 5.3 Citra hasil *threshold* dan binerisasi citra setelah *median filter*.

### 5.4 Hasil Segmentasi

Segmentasi dilakukan untuk memisahkan karakter. Segmentasi pertama kali dilakukan terhadap baris teks, memisahkan menjadi masing-masing baris. Hasil segmentasi baris dapat dilihat pada gambar 5.4.



Gambar 5.4 Hasil segmentasi baris teks.

Hasil segmentasi karakter dapat dilihat pada gambar 5.5 berikut.



Gambar 5.5 Hasil Segmentasi karakter "Pada hari".

### 5.5 Hasil Normalisasi

Citra hasil segmentasi karakter memiliki ukuran yang tidak sama sehingga diperlukan proses normalisasi agar citra karakter memiliki ukuran yang sama dengan citra template yaitu 42 x 24 piksel. Hasil dari normalisasi dapat dilihat pada gambar 5.6.



Gambar 5.6 Hasil Normalisasi karakter dengan ukuran 42x24 piksel.

### 5.6 Hasil Pengujian dan Analisis

Pengujian dilakukan terhadap 4 kasus yaitu pengujian dengan ukuran huruf 12pt terhadap 21 dokumen, ukuran huruf 14pt, ukuran huruf 16pt dan kondisi dokumen dengan *noise* tambahan, teks yang dikaburkan serta terpotong. Hasil pengujian terhadap 21 dokumen dengan ukuran huruf 12pt dapat dilihat pada tabel 5.1 berikut.

Tabel 5.1 Hasil pengujian 21 sampel dokumen

Citra Masukan	Total Karakter	yang dikenali	Persentase Keberhasilan
Dokumen 1			
Arial.jpg	1129	1067	94,5%
Calibri.jpg	1129	1077	95,4%
CalibriLight.jpg	1129	1047	92,7%
ComicSansMs.jpg	1129	1054	93,3%
LucidaSans.jpg	1129	1045	92,5%
Tahoma.jpg	1129	1026	90,8%
TmsNewRoman.jpg	1129	989	87,5%
Dokumen 2			
Arial.jpg	1532	1481	96,6%
Calibri.jpg	1532	1487	97,0%
CalibriLight.jpg	1532	1375	89,7%
ComicSansMs.jpg	1532	1464	95,5%
LucidaSans.jpg	1532	1473	96,1%
Tahoma.jpg	1532	1439	93,9%
TmsNewRoman.jpg	1532	1405	91,7%
Dokumen 3			
Arial.jpg	1395	1336	95,7%
Calibri.jpg	1395	1262	90,4%
CalibriLight.jpg	1395	1224	87,7%
ComicSansMs.jpg	1395	1222	87,5%
LucidaSans.jpg	1395	1316	94,3%
Tahoma.jpg	1395	1324	94,9%
TmsNewRoman.jpg	1395	1267	90,8%
Rata-rata persentase keberhasilan			92,8%

Berdasarkan pengujian pada tabel 5.1, tingkat keberhasilan paling kecil yaitu jenis huruf *Times New Roman* pada dokumen 1 dengan tingkat keberhasilan 87,5%, *Calibri Light Heading* pada dokumen 2 yaitu 89,7%, *Calibri Light Heading* pada dokumen 3 yaitu 87,7% dan *Comic Sans Ms* yaitu 87,5%. Hal ini disebabkan karakteristik huruf *Times New Roman* dan *Calibri Light Heading* yang tipis sehingga piksel huruf tersebut hilang saat proses *thresholding* dan pembersihan *noise* yang menyebabkan kegagalan saat dikenali.

Huruf *Times New Roman* dan *Calibri Light Heading* merupakan jenis huruf berkaki, kaki pada karakter seperti huruf m, n dan h sangat berdekatan antar pikselnya sehingga terlihat seperti menyatu yang menyebabkan karakter tersebut kehilangan bentuk aslinya. Sedangkan huruf *Comic Sans Ms* mengalami kegagalan dalam membaca huruf c yang dikenali sebagai huruf c.

Hasil pengujian kedua dan ketiga terhadap dokumen dengan ukuran huruf 14pt dan 16pt dapat dilihat pada tabel 5.2 dan tabel 5.3 berikut.

Tabel 5.2 Hasil pengujian sampel dengan ukuran huruf 14pt.

Citra Masukan	Total Karakter	yang dikenali	Persentase Keberhasilan
Arial.jpg	1129	1089	96,5%
Calibri.jpg	1129	1081	95,7%
Rata-rata persentase keberhasilan			96,1%

Tabel 5.3 Hasil pengujian sampel dengan ukuran huruf 14pt.

Citra Masukan	Total Karakter	yang dikenali	Persentase Keberhasilan
Arial.jpg	1129	1098	97,2%
Calibri.jpg	1129	1092	96,6%
Rata-rata persentase keberhasilan			96,9%

Pengujian terhadap dokumen dengan ukuran huruf yang lebih besar yaitu 14pt dan 16pt menghasilkan persentase keberhasilan yang lebih tinggi dari hasil pengenalan dokumen dengan ukuran huruf 12pt. Persentase keberhasilan citra dokumen dengan ukuran huruf 12pt dengan jenis huruf *Arial* dan *Calibri* masing-masing yaitu 94,5% dan 95,4%. Persentase keberhasilan pengenalan karakter dengan ukuran huruf 14pt dengan jenis huruf *Arial*

dan *Calibri* masing-masing yaitu 96,5% dan 95,7%. Persentase keberhasilan pengenalan karakter dengan ukuran huruf 16pt masing-masing yaitu 97,2% dan 96,6%.

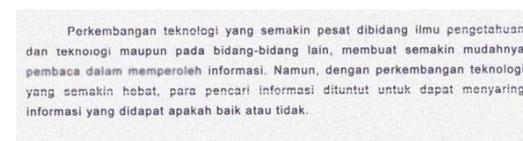
Rata-rata keberhasilan pengenalan karakter dari total 26 dokumen berdasarkan tabel 5.1, tabel 5.2 dan tabel 5.3 dapat dilihat pada tabel 5.4 berikut :

Tabel 5.4 Hasil rata-rata keberhasilan pengenalan dari 26 dokumen.

Citra Masukan	Persentase Keberhasilan
21 dokumen	92,8%
Dokumen huruf 14pt	96,1%
Dokumen huruf 116pt	96,9%
Rata-rata Persentase Keberhasilan	95,3%

Berdasarkan perbandingan 3 jenis ukuran huruf tersebut, terlihat peningkatan persentase keberhasilan sehingga dapat disimpulkan bahwa semakin besar ukuran huruf semakin baik kualitas citra karakter tersebut dan semakin tinggi tingkat keberhasilan pengenalan karakter dengan metode *template matching*.

Pengujian selanjutnya dilakukan terhadap satu paragraf teks dokumen *Arial* berjumlah 291 karakter dengan kondisi hasil buatan sendiri yaitu *noise* 10%, beberapa karakter dalam kalimat yang terpotong dan kalimat yang kabur (*blur*). Sampel dapat dilihat pada gambar 5.7.



Gambar 5.7 Teks sampel dengan kondisi buatan.

Teks yang berhasil dibaca yaitu 290 dan yang dikenali sebanyak 268 dengan persentase keberhasilan 92%.

*Median Filter* melakukan pembersihan noise terhadap citra dengan kondisi noise buatan pada gambar 5.7, hal ini terlihat dari total karakter sebelum *median filter* yaitu 291 karakter dan setelah setelah *median filter* yaitu 290 karakter. Total karakter yang tidak bertambah menunjukkan bahwa tidak ada karakter tambahan pada hasil pengenalan, hanya saja karakter titik (.) terhapus karena proses *median filter*. Teks

dengan kondisi kabur masih dapat dikenali yaitu pada kalimat “pembaca diillem mempefOleh”, beberapa karakter tidak dapat dibaca karena huruf tersebut kehilangan sifat aslinya saat dilakukan proses *bluring*. Kalimat yang mengalami pemotongan juga berhasil disegmentasi dan dikenali yaitu pada kalimat “yang somakin hCb3t. para penC3ri inform3si dituntUt untuk d3p3t menyaring”. Kalimat yang terpotong kehilangan sifat asli karakter, namun masih bisa dikenali oleh *template*.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

1. Median filter juga melakukan pembersihan dengan baik pada citra dengan kondisi noise buatan, terlihat dari jumlah karakter yang tidak bertambah yaitu total karakter awal 291 dan total karakter hasil pengenalan yaitu 290. Karakter yang tidak bertambah menunjukkan tidak adanya noise yang terbaca sebagai karakter.
2. Metode *template matching* melakukan pengenalan karakter terhadap 21 sampel dokumen dengan ukuran huruf 12pt serta jenis huruf yang berbeda dengan tingkat keberhasilan rata-rata yaitu 92,8%. Tingkat keberhasilan pengenalan karakter dokumen dengan ukuran huruf 14pt dan 16pt masing-masing yaitu 96,1% dan 96,9%. Rata-rata persentase keberhasilan pengenalan karakter terhadap 26 dokumen adalah 95,3%.
3. Metode profile projection dapat melakukan segmentasi baris dan karakter sehingga karakter dapat dipisahkan dan dikenali dengan metode *template matching*.
5. Dokumen dengan adanya noise dapat dikenali dan dikonversi dalam bentuk teks digital. Beberapa faktor penyebab kegagalan dalam proses pengenalan karakter yaitu hasil cetak dokumen yang kurang baik seperti karakter yang terputus atau kecerahan dokumen serta karakteristik huruf yang sama seperti angka 1 dan huruf i, huruf i dan huruf dan karakter lainya dengan karakteristik sama. Jarak karakter yang berdekatan juga menjadi penyebab kegagalan

dalam proses segmentasi sehingga mempengaruhi proses pengenalan karakter.

### 6.2 Saran

1. Adanya pengembangan metode segmentasi khususnya segmentasi karakter untuk melakukan segmentasi terhadap citra karakter yang berdekatan.
2. Pengenalan dapat difokuskan pada beberapa karakter yang memiliki karakteristik sama seperti huruf B dan angka 8, huruf O dan angka 0, huruf z dan angka 2 dan karakter lainya dengan karakteristik sama.

## DAFTAR PUSTAKA

- [1] Kennedy, Jay., and Schauder, Cherryl. 1998. *Records Management, A Guide to Corporate Record Keeping*. Melbourne: Longman.
- [2] Mitha.N, Wara. 2014. Analisis Citra Sinar-X Tulang Tangan Menggunakan Metode Thresholding Otsu Untuk Identifikasi Osteoporosis. *Skripsi*. Pontianak: Universitas Tanjungpura.
- [3] Kadir, Abdul., dan Susanto, Andi. 2012. *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andipublisher.
- [4] Santi, Candra Noor. 2011. Mengubah Citra Berwarna Menjadi Grayscale dan Binner. *Jurnal Teknologi Informasi DINAMIK*. Vol 16. No.1.
- [5] Gonzalez, R.C., Woods, R.E. 2002. *Digital Image Processing Second Edition*. Prentice Hall. New Jersey.
- [6] Gordon B. Davis. 1990. *Management Information System Conceptual Foundatios. Structure Development*.
- [7] Hendry, Jans. 2011. *Using Profile Projection to Segment Character in Image (MATLAB)*. EE&IT UGM. Indonesia.
- [8] Putra, Darma. 2009. *Sistem Biometrika Konsep Dasar, Teknik Analisis Citra, dan Tahapan Membangun Aplikasi Sistem Biometrika*. Yogyakarta: Andi.
- [9] Fatta, Hanif Al. 2009. *Rekayasa Sistem Pengenalan Wajah*. Yograkarta: Penerbit Andi.