ISSN: 2548-9704

PERANCANGAN APLIKASI KONVERSI FILE IMAGE HASIL SCAN MENJADI FILE TEXT DENGAN METODE FEATURE EXTRACTION

Kadri Yusuf

Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Medan
Jl. Almamater No. 1 Kampus USU 20155, Medan
E-mail :kadriyusuf80@gmail.com

ABSTRAK

Dalam proses konversi file image dengan cara proses scan menjadi file text tentunya adalah salah satu dari teknik untuk memudahkan pengguna dalam mengedit file image yang diubah dalam bentuk file text tanpa harus menyalin ulang dokumen secara manual.Feature extraction merupakan salah satu cara untuk mengenali suatu objek dengan melihat ciri-ciri khusus yang dimiliki objek tersebut. Tujuan dari feature extraction adalah melakukan perhitungan dan perbandingan yang bisa digunakan untuk mengklasifikasikan ciri-ciri yang dimiliki oleh suatu citra. Pemetaan ciri-ciri khusus yang dilakukan terhadap citra karakter adalah keterbukaan citra, jumlah garis vertikal dan horizontal, jumlah perpotongan pixel hitam di bagian tengah citra secara vertikal dan horizontal, dan histogram pixel hitam pada sembilan bagian citra yang dibandingkan dengan resolusi citra karakter.Dalam sebuah perancangan sistem yang dibuat, masukkan yang dilakukan ke dalam sistem adalah berupa file image yang berada didalam file dokumen atau hasil image pada proses scanning. Dan perangkat lunak untuk konversi file image hasil scan menjadi file text ini dibangun dengan menggunakan bahasa pemrograman microsoft visual basic 2008.

Kata Kunci : Konversi, Image, aplikasi

ABSTRACT

In the process of image file conversion by way of scanning into text files of course is one of the techniques to facilitate users in editing image files that are changed in the form of text files without having to re-copy the document manually.Feature extraction is one way to recognize an object with see the special features that the object has. The purpose of feature extraction is to do calculations and comparisons that can be used to classify the characteristics possessed by an image. The mapping of special features performed on the character image is the openness of the image, the number of vertical and horizontal lines, the number of black pixel intersections in the center of the image vertically and horizontally, and the black pixel histogram in nine parts of the image compared to the image resolution character. system design is made, insert done into the system is a file image residing in the document file or image results in the scanning process. And the software to convert the image file into text file is built using Microsoft Visual Basic 2008 programming language.

Keywords: Conversions, Image, apps

1. PENDAHULUAN

Dalam proses konversi *file image* dengan cara proses *scan* menjadi *file text* tentunya adalah salah satu dariteknik untuk memudahkan pengguna dalam mengedit *file image* yang diubah dalam bentuk *file text* tanpa harus menyalin ulang dokumen secara manual. Dalam pengerjaannya didapat dari

ISSN: 2548-9704

berbagai informasi baik berupa buku, dokumen-dokumen ataupun jurnal-jurnal yang biasa digunakan untuk keperluan mencari informasi maupun sebagai referensi dalam pembuatan sebuah karya ilmiah.Pada kenyataan saat ini berbagai sumber informasi seperti buku atau dokumen belumsemuanya tersedia dalam edisi elektronik.Hal ini menjadi kendala dalam pengaksesan,pengelolaan dan pendayagunaan data tersebut. Agar dapat dilakukan tindakan tersebutpada data maka diperlukan adanya proses *entry* data dari *hardcopy* ke dalam *softcopy*pada komputer. Mengingat pentingnya sebuah informasi dalam edisi elektronik danproses *entry* manual yang memerlukan waktu yang tidak sedikit, untuk itu akandikembangkan suatu sistem pengenalan karakter cetak secara otomatis yang dilakukanpada citra digital berupa image dari hasil *scan*.

Proses *entry* data dari *hardcopy* kedalam *softcopy* dengan kata lain mengubah atau mengkonversi sebuah *file* gambar atau dokumen yang discan menjadi sebuah *file text* yang dapat di *edit* kembali. Misalnya saja sebuah dokumen penting dan dokumen tersebut sudah berbentuk cetakan atau *hardcopy* dan hanya itu satu-satunya dokumen yang tersisa dan pemilik dokumen tersebut ingin mengedit kembali dengan waktu yang relatif singkat, bisa saja dokumen tersebut di *scan*, namun hasil scan dokumen tersebut tentunya berbentuk *file image* seperti *.Jpg, .Png, .bmp* dan lainnya. Secara sederhana sangat sulit *file image* tersebut diedit kembali dalam bentuk *file teks* dan Jika pemilik dokumen mengetik ulang kembali akan memerlukan waktu lama, selain memakan waktu dalam pengetikan, juga akan memerlukan waktu lama untuk mencari tanda tangan dan stempel perusahaan yang tertera pada dokumen aslinya. Berdasarkan permasalahan tersebut dibutuhkan sebuah sistem atau aplikasi yang dapat menerjemahkan karakter pada citra digital menjadi format teks atau biasa disebut sistem OCR (*Optical Character Recognition*). Feature Extraction merupakan metode yang cukup tepat untuk membangun aplikasi tersebut, digunakan untuk mengenali pola dengan cara memetakan ciri-ciri objek citra yang akan dikenali dan diklasifikasikan terhadap ciri-ciri citra template yang disimpan pada basis data.

2. METODOLOGI

Konversi *File*

Konversi *file* adalah mengubah bentuk dari format asal keformat yang lain seperti format *DOC, XLS, PPT, HTML* dan lain – lain kedalam bentuk format *PDF*. Hal ini juga berlaku pada *file* yang berbentuk *file image* yang dapat dirubah kedalam format *file text* berikut akan dijelaskan tentang *file image* dan *file text*.

Pengenalan Pola

Pengenalan pola dapat dikatakan sebagai kemampuan mengenali objek berdasarkan ciri-ciri dan pengetahuan yang pernah diamatinya dari objek-objek tersebut.Tujuan dari pengenalan pola adalah mengklasifikasi dan mendeskripsikan pola atau objek kompleks melalui pengetahuan sifat-sifat atau ciri-ciri objek tersebut.

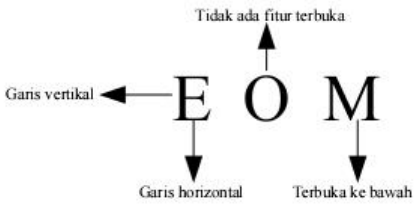
Ada tiga pendekatan dalam pengenalan pola yaitu secara sintaks, statistik, dan semantik.Pengenalan pola secara sintaks dilakukan berdasarkan ciri-ciri objek.Pengenalan pola secara statistik dilakukan berdasarkan komputasi matematis.Pendekatan dengan semantik berarti pola dikenali dalam tataran yang lebih abstrak.

Feature Extraction

Feature extraction merupakan salah satu cara untuk mengenali suatu objek dengan melihat ciri-ciri khusus yang

ISSN: 2548-9704

dimiliki objek tersebut. Tujuan dari *feature extraction* adalah melakukan perhitungandan perbandingan yang bisa digunakan untuk mengklasifikasikan ciri-ciri yang dimiliki oleh suatu citra.



Gambar 1. Ilustrasi Citra

.

Gambar 1.diatas merupakan ilustrasi citra karakter dengan ciri-cirinya. Ciri-ciri dari masing-masing citra *template* akan di simpan. Citra masukan yang akan dibandingkan akan dianalisis berdasarkan ciri- ciri citra. Ciri-ciri yang dimiliki citra masukan akan diklasifikasikan terhadap ciri-ciri citra *template*.

Pemetaan ciri-ciri khusus yang dilakukan terhadap citra karakter adalah keterbukaan citra, jumlah garis *vertikal* dan *horizontal*, jumlah perpotongan pixel hitam di bagian tengah citra secara *vertikal* dan *horizontal*, rasio citra, danhistogram pixel hitam pada sembilan bagian citra yang dibandingkan dengan resolusi citra karakter.
algoritma *feature extraction* adalah sebagai berikut:

1. Pemetaan fitur citra karakter. Pemetaan inidilakukan satu kali saat citra karakter akan dikenali.
2. Perhitungan jarak dengan menggunakan persamaan. Perhitungan yang dilakukan sebanyak citra *template* dinotasikan dengan n.
3. Pencarian jarak*minimum* dilakukan pada seluruh citra template. Jumlah pencarian yang dilakukan terhadap seluruh citra template dinotasikan dengan n.
4. Tabeldibawah ini adalah perhitungan jumlah operasi dasar yang dilakukan pada algoritma *feature extraction*.

Tabel 2.1. Algoritma *Feature Extraction*

No.	Operasi Dasar	Jumlah Iterasi
1	Pemetaan Fitur Citra Karakter	1 Kali
2	Perhitungan Jarak	n Kali
3	Pencarian Jarak Terkecil	n Kali
Total		**2n+1**

Sumber :Bahri, Raden Sofian., dan Maliki, Irfan. 2012. Perbandingan Algoritma Template Maching Dan Feature Extraction Pada Optical Character Recognition.Jurnal Komputer dan Informatika (KOMPUTA).Vol.1.No.29.

5. Berdasarkan hasil perhitungan jumlah operasi dasar *feature extraction* pada tabel diatas maka kompleksitas algoritma*feature extraction* adalah 2n+1.

Dari lima proses pemetaan fitur pada algoritma *feature extraction*, masih ada kemungkinan penambahan fitur-fitur khusus citra karakter. Salah satu contohnya adalah fitur jumlah *stroke* (jumlah garis yang membentuk karakter).Berdasarkan hasil pengenalan, algoritma *feature extraction* dapat mengenali citra karakter lebih baik dan algoritma *feature extraction* membutuhkan waktuyang lebih singkat.

Sebelum mengambil ciri dari suatu citra karakter, maka citra karakter tersebut harus dicari batas kanan, kiri, atas, dan bawahnya terlebih dahulu.Ciri-ciri yang diambil dari citra karakter *template* maupun citra karakter masukan adalah jumlah garis horizontal dan garis vertikal, perpotongan *pixel* hitam terhadap garis vertikal dan garis

ISSN: 2548-9704

horizontal di tengah citra.Citra karakter dianalisis apakah terbuka ke kanan, kiri, atas, atau bawah. Karakter citra akan dibagi menjadi sembilan bidang simetris kemudian *pixel* yang berwarna hitam pada masing-masing bidang akan dibagi dengan jumlah *pixel* keseluruhan.

1. Jumlah Garis Horizontal

Gambar 2.Citra Karakter dengan Garis Horizontal

Setiap karakter diperiksa apakah memiliki ciri berupa garis horizontaldengan cara berikut ini:

1. Telusuri citra mulai dari kiri atas ke kanan bawah.
2. Jika ditemukan *pixel* berwarna hitam, maka *pixel* tersebut akandiwakili dengan angka 1. Sedangkan jika ditemukan *pixel* berwarnaputih, maka *pixel* tersebut akan diwakili dengan angka 0.
3. Di akhir baris setiap *pixel*, jumlahkan angka yang mewakili *pixel*.Jumlah pixel akan menjadi 0 jika ditemukan pixel putih sebelumjumlah pixel mencapai 80% lebar citra karakter.
4. Jika posisi *scan* sudah mencapai kanan bawah, maka periksa jumlah*pixel* di masing-masing baris.
5. Jika jumlah *pixel* lebih dari atau sama dengan 80% lebar citra, makacitra karakter tersebut memiliki ciri berupa garis horizontal.
6. Hitung jumlah garis horizontal yang ada pada citra karakter.

2. Jumlah Garis Vertikal

Gambar 3.Citra Karakter dengan Garis Vertikal.

Langkah-langkah berikut ini digunakan untuk mencari jumlah garisvertikal suatu citra karakter.

1. *Scan* citra mulai dari kiri atas ke kanan bawah.
2. Jika ditemukan *pixel* berwarna hitam, maka *pixel* tersebut akandiwakili dengan angka 1. Sedangkan jika ditemukan *pixel* berwarnaputih, maka *pixel* tersebut akan diwakili dengan angka 0.
3. Di akhir kolom setiap *pixel*, jumlahkan angka yang mewakili *pixel*.Jumlah pixel akan menjadi 0 jika ditemukan pixel putih sebelumjumlah pixel mencapai 80% lebar citra karakter.
4. Jika posisi *scan* sudah mencapai kanan bawah, maka periksa jumlah*pixel* di masing-masing kolom.
5. Jika jumlah *pixel* lebih dari atau sama dengan 80% tinggi citra, makacitra karakter tersebut memiliki ciri berupa garis vertikal.
6. Hitung jumlah garis vertikal yang ada pada citra karakter.

3. Ciri Citra Karakter Terbuka

Suatu citra karakter diperiksa apakah terbuka ke kiri, kanan, atas, atau bawah.Gambar 2.10 adalah citra karakter yang memiliki ciri-ciri berupa terbuka kekiri, kanan, atas, atau bawah.

Gambar 4..Citra Karakter dengan Ciri-Ciri Terbuka.

Langkah-langkah berikut ini digunakan untuk mendeteksi apakah

ISSN: 2548-9704

suatucitra karakter terbuka ke atas atau ke bawah.

1. *Scan* citra dari atas hingga mencapai 40% tinggi citra. Nilai 40%didapatkan dari hasil penelitian.
2. Jika ditemukan *pixel* berwarna hitam, maka *pixel* tersebut akandiwakili dengan angka 1. Sedangkan jika ditemukan *pixel* berwarnaputih, maka *pixel* tersebut akan diwakili dengan angka 0.
3. Jumlahkan angka *pixel* pada masing-masing kolom dari kiri ke kanan.
4. Periksa jumlah pada masing-masing kolom. Jika ditemukan polajumlah *pixel* pada suatau kolom sama dengan 0 dan jumlah *pixel* padakolom di sebelahnya lebih dari 0 sebanyak dua kali, maka citrakarakter tersebut dinyatakan terbuka ke atas.
5. Hal yang sama dilakukan untuk mendeteksi citra karakter terbuka kebawah, namun citra di-*scan* mulai dari 70% tinggi citra hinggamencapai 100% tinggi citra. Nilai 70% didapatkan dari hasilpenelitian.

Langkah-langkah berikut ini digunakan untuk mendeteksi apakah suatucitra karakter terbuka ke kiri atau ke kanan.

1. *Scan* citra dari kiri hingga mencapai 30% lebar citra. Nilai 30%didapatkan dari hasil penelitian.
2. Jika ditemukan *pixel* berwarna hitam, maka *pixel* tersebut akandiwakili dengan angka 1. Sedangkan jika ditemukan *pixel* berwarnaputih, maka *pixel* tersebut akan diwakili dengan angka 0.
3. Jumlahkan angka *pixel* pada masing-masing baris dari atas ke bawah.
4. Periksa jumlah *pixel* pada masing-masing baris. Jika ditemukan polajumlah *pixel* pada suatu baris sama dengan 0 dan jumlah *pixel* padabaris di bawahnya lebih dari 0 sebanyak dua kali, maka citra karaktertersebut terbuka ke kiri.
5. Hal yang sama dilakukan untuk mendeteksi citra karakter terbuka kekanan, namun citra di-*scan* mulai dari 70% lebar citra hingga mencapai100% lebar citra. Nilai 70% didapatkan dari hasil penelitian.

5. Jumlah Perpotongan Garis Vertikal Di Tengah Citra

   Pada citra karakter ditarik suatu garis vertikal yang tepat memotong bagiantengah citra.Garis vertikal tersebut ditelusuri dan apabila menemukan *pixel*berwarna hitam dan *pixel* di bawahnya berwarna putih, maka dihitung satu perpotongan.Gambar 2.11 adalah citra karakter dengan tiga perpotongan terhadapgaris vertikal.

Gambar 5.Perpotongan Garis Vertikal di Bagian Tengah Citra.

6. Jumlah Perpotongan Garis Horizontal Di Tengah Citra

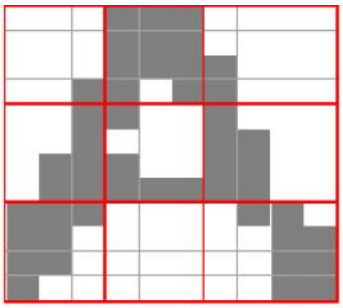
   Pada citra karakter ditarik suatu garis horizontal yang tepat memotongbagian tengah citra.Garis horizontal tersebut ditelusuri dan apabila menemukan*pixel* berwarna hitam dan *pixel* di sebelahnya berwarna putih, maka dihitung satu perpotongan.Gambar 2.12 adalah citra karakter dengan dua perpotongan terhadapgaris horizontal.

Gambar 6.Perpotongan Garis Horizontal di Bagian Tengah Citra.

7. Perbandingan *Pixel* Hitam Di Setiap Blok Citra

ISSN: 2548-9704

Citra karakter dibagi menjadi sembilan bagian simetris dan dihitungperbandingan jumlah *pixel* hitam pada masing-masing bagian tersebut denganjumlah seluruh *pixel*. Gambar 2.13 adalah ilustrasi pembagian blok citra karakter.



Gambar 7. Citra karakter yang dibagi menjadi sembilan blok.

Bagian yang pertama dihitung adalah blok pada bagian kiri atas citrakarakter. Jika ditemukan *pixel* berwarna hitam, maka *pixel* tersebut akan diwakilidengan angka 1. Sedangkan jika ditemukan *pixel* berwarna putih, maka *pixel*tersebut akan diwakili dengan angka 0. Jumlahkan angka-angka *pixel* yangterdapat di blok pertama dan dibagi dengan resolusi citra karakter.Hal yang sama dilakukan pada seluruh blok. Blok kedua terletak di bawahblok pertama. Blok ketiga terletak di bawah bagian blok kedua. Blok keempatterletak di sebelah kanan blok pertama dan begitu seterusnya hingga mencapaiblok terakhir yaitu blok kesembilan yang berada di bagian kanan bawah citra.

8. Klasifikasi

k-nearest neighbor (k-NN atau KNN) bekerja dengan cara membandingkan data uji dan data training/*template*. k-NN mencari pola data *template* yang paling mendekati dengan data uji. Dekat atau jauhnya tetangga dihitung berdasarkan jarak *Euclidean*. Dengan kata lain, kedua data dibandingkan berdasarkan atribut-atributnya.

k-NN memiliki beberapa kelebihan yaitu tangguh terhadap data *template* yang *noisy* dan efektif apabila jumlah data *template* besar. Sedangkan kelemahan dari k-NN adalah perlunya menentukan nilai parameter k (jumlah dari tetangga terdekat), pembelajaran berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample.*

Persamaan adalah persamaan yang digunakan untuk menghitung jarak.

$$d(p,q)^2 = d(p,q)^2 = \sum_{i-1}^{n} (qi - pi)^2$$

Nilai *k* yang terbaik untuk algoritma ini tergantung pada data. Pada umumnya, nilai *k* yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Langkah-langkah yang digunakan pada metode k-NN adalah sebagai berikut:

1. Tentukan parameter k. k adalah jumlah tetangga terdekat. Nilai k yang digunakan pada penelitian ini adalah 1.
2. Hitung jarak antara ciri-ciri citra *template* dan citra masukan dengan menggunakan persamaan.
3. Urutkan jarak dari kecil ke besar.
4. Gunakan parameter k sebagai acuan dalam mengambil sejumlah databerdasarkan urutan pada nomor 3.

Ketepatan algoritma k-NN ini sangat dipengaruhi oleh fitur-fitur unik setiap citra yang akan dikenali. Riset terhadap algoritma ini sebagian besar membahas

ISSN: 2548-9704

bagaimana memilih dan memberi bobot terhadap fitur, agar performa klasifikasi menjadi lebih baik.

Jika seluruh ciri-ciri citra karakter sudah didapatkan, maka ciri-ciritersebut diklasifikasikan dengan ciri-ciri yang ada pada basis data.Metodeklasifikasi yang digunakan pada penelitian ini adalah metode k-NN atau k *NearestNeighbour*.Algoritma k-*Nearest Neighbour* mengenali citra berdasarkan ciri-ciri yang paling mendekati antara citra masukan dengan citra *template*.Tujuannya adalah mengklasifikasikan citra karakter berdasarkan atribut yang dimilikinya.Berikut ini adalah contoh perhitungan k-NN untuk mengenali karakter.

3. ANALISA DAN PERANCANGAN

Analisa

Dalam proses pembuatan suatu sistem mutlak dilakukan analisis terhadap sistem yang akan dibangun, analisis yang dilakukan untuk membangun aplikasi perbandingan algoritma *feature extraction* pada OCR

Analisa terhadap suatu permasalahan yang diikuti dengan rancang bangun sebuah konsep didalam sistem merupakan sesuatu hal yang wajib dilakukan dalam pembuatan suatu perangkat lunak.Dan mempunyai tujuan untuk menghasilkan suatu sistem yang tepat dan efektif sesuai dengan permasalahan dan perbaikan perangkat lunak.

Perangkat lunak yang akan dihasilkan tentunya harus dapat melakukan proses konversi dalam mengubah suatu bentuk format *file* tanpa merubah isi data *file* yang ada. Sehingga tujuan dari proses konversi itu sendiri adalah mengetahui, menerapkan dan mengimplementasikan suatu rancangan perangkat lunak dengan alur pengenalan karakter yang ada di dalam *file image*.

Perancangan

Dalam sebuah perancangan sistem yang dibuat, masukkan yang dilakukan ke dalam sistem adalah berupa *file image* yang berada didalam *file* dokumen atau hasil *image* pada proses *scanning*. Awalnya, dilakukan proses penginputan *file* dari dokumen untuk dikonversikan menjadi *file text* yang mempunyai struktur kata dari *file image* yang dikonversikan menjadi struktur yang sama dengan format *.DOC* pada *Microsoft word*.

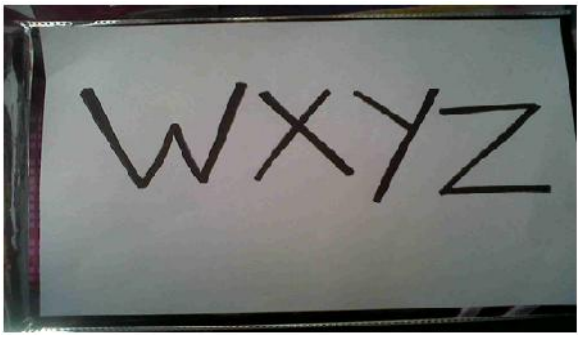
Proses *Feature extraction* ini dilakukan dengan cara mengambil ciri khusus dari tulisan pada hasil proses *scanning* yang akan dikonversikan dan bertujuan untuk menghilangkan faktor - faktor yang dapat menyebabkan program pembaca karakter pada proses konversi *file image* menjadi *file text* tidak akurat dan sulit dikenali karakternya dan juga sulit untuk diedit kembali dan hasil dari konversi disimpan dalam dokumen dengan fo 35 *.DOC*.

Analisa Dan Logika Metode

Pada kasus ini ada beberapa langkah dalam mengkonversi *file image* hasil *scan* menjadi *file text* ini maka diperlukan proses sebuah perhitungan dalam mengenali karakter dengan menggunakan metode *feature extraction* di sini akan dijelaskan beberapa langkah – langkah sebagai berikut :

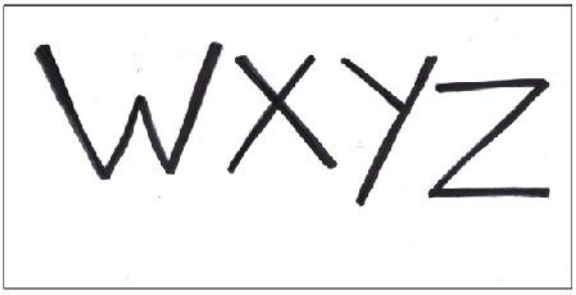
1. Membuat suatu input seperti contoh dari sebuah tulisan yang ditulis ke dalam sebuah kertas berukuran A4 seperti pada gambar dibawah ini :

ISSN: 2548-9704



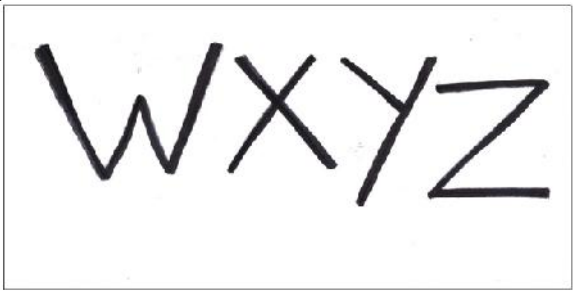
Gambar 8. Contoh Tulisan Pada Kertas A4

2. Kemudian, kertas A4 yang telah ditulis discan dengan menggunakan *Scanner* Canon LED110, maka akan menghasilkan sebuah *file image* dengan Format .JPG, seperti gambar dibawah ini:



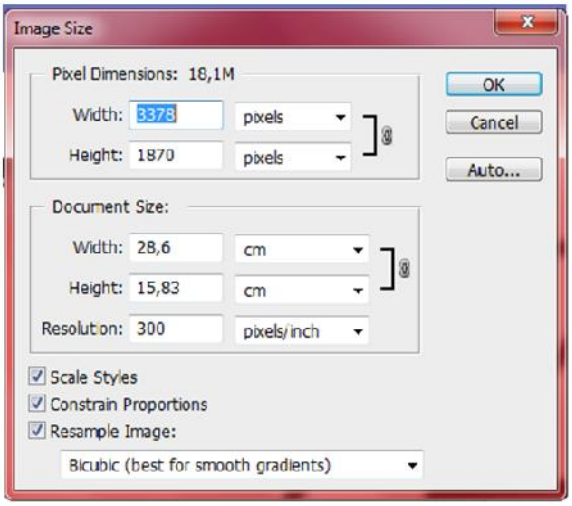
Gambar 9. Hasil Dari Menggunakan Scanner Canon LED110

3. Setelah selesai discan maka lakukan proses konversi dengan menggunakan metode *feature extraction* untuk mengenali karakter yang akan disalin kedalam *file* berbentuk *file text.*
4. Menganalisa berapa resolusi yang ada pada citra



Gambar 10. Citra Yang Akan Dianalisa Resolusinya

Pada umumnya gambar hasil scan ini mempunyai ukuran yang terlihat seperti pada gambar dibawah ini :



Gambar 11. Resolusi Citra Dari Citra Image Yang Dianalisa

5. Selanjutnya mencari Jumlah garis horizontal yang ada pada karakter.

Tabel 1.Menentukan jumlah Garis Horizontal Dari Karakter File Image

Karakter	Jumlah garis horizontal
W	0
X	0
Y	0
Z	0

6. Cari Jumlah garis vertikal yang ada pada karakter.
7.

Tabel 2.Menetukan Jumlah Garis Vertikal Dari Karakter File Image

Karakter	Jumlah garis vertikal
W	0
X	0
Y	0
Z	2

8. Di Dalam proses konversi harus ditentukan ciri dari sebuah karakter

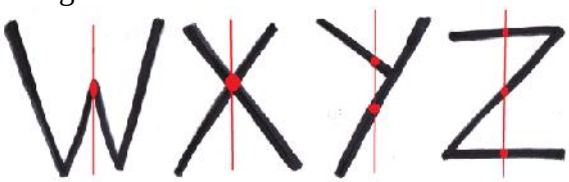
ISSN: 2548-9704

apakah karakter terbuka ke atas, ke bawah, ke kanan, atau ke kiri.

Tabel 3. Ciri – Ciri Dari Karakter Terbuka Pada File Image

Karakter Yang dideteksi	Terbuka atas	Terbuka bawah	Terbuka kanan	Terbuka kiri
W	2	1	0	0
X	1	1	1	1
Y	1	0	0	1
Z	0	0	1	1

9. Jumlah perpotongan garis vertikal ditengah citra karakter

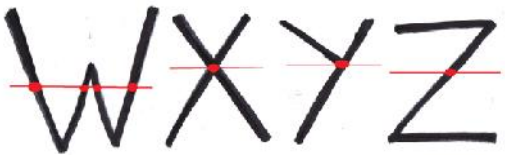


Gambar 12. Jumlah Titik Perpotongan Garis Vertikal

Tabel 4.Menetukan Jumlah Titik Potong Garis Vertikal

Karakter	Jumlah garis perpotongan vertikal
W	1
X	1
Y	2
Z	3

10. Jumlah perpotongan garis horizontal ditengah citra karakter

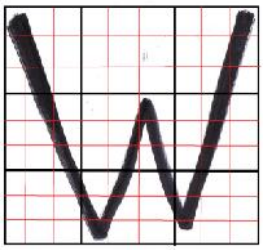


Gambar 13. Jumlah Titik Perpotongan Garis Horizontal

Tabel 5.Menetukan Jumlah Titik Potong Garis Horizontal

Karakter	Jumlah garis perpotongan horizontal
W	4
X	1
Y	1
Z	1

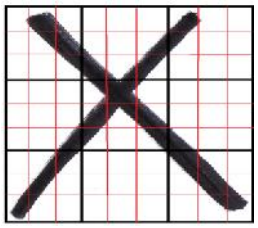
11. Perbandingan pixel hitam disetiap blok citra



Gambar 14. Perbandingan Pixel Hitam Pada Blok Citra karakter W

$$Blok = \frac{blok\ 1+blok\ 2+blok\ 3+blok\ 4+blok\ 5+blok\ 6+blok\ 7+blok8+blok\ 9}{resolusi\ citra}$$

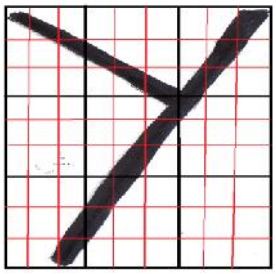
$$= \frac{6+5+2+0+6+8+5+5+4}{300} = \frac{41}{300} = 0{,}13666666666666667$$



Gambar 15. Perbandingan Pixel Hitam Pada Blok Citra karakter X

$$Blok = \frac{blok\ 1+blok\ 2+blok\ 3+blok\ 4+blok\ 5+blok\ 6+blok\ 7+blok8+blok\ 9}{resolusi\ citra}$$

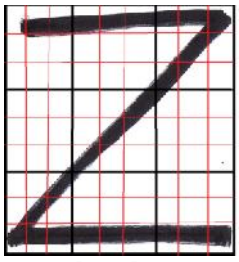
$$= \frac{6+2+6+5+8+0+3+2+7}{300} = \frac{39}{300} = 0{,}13000000000000$$



ISSN: 2548-9704

Gambar 16. Perbandingan Pixel Hitam Pada Blok Citra karakter Y

$$Blok = \frac{blok\ 1+blok\ 2+blok\ 3+blok\ 4+blok\ 5 \\ +blok\ 6+blok\ 7+blok8+blok\ 9}{resolusi\ citra} = \frac{5+0+3+5+6+4+7+1+0}{300} = \frac{31}{300} = 0{,}1033333333333333$$



Gambar 17.. Perbandingan Pixel Hitam Pada Blok Citra karakter Z

$$Blok = \frac{blok\ 1+blok\ 2+blok\ 3+blok\ 4+ \\ blok\ 5+blok\ 6+blok\ 7+blok8+blok\ 9}{resolusi\ citra} = \frac{5+1+8+3+7+4+8+1+3}{300} = \frac{40}{300} = 0{,}1333333333333333$$

Kemudian setelah selesai menentukan ciri – ciri dari karakter maka langkah selanjutnya adalah dengan melakukan klasifikasi dari ciri – ciri karakter diatas.

1. Tentukan atribut dari semua ciri – ciri karakter yang telah dijabarkan untuk proses pencocokan.

Tabel 6.Klasifikasi Dari Ciri – Ciri Karakter

No	Kar	T_kiri	T_kanan	T_atas	T_bawah	Int_v	Int_h	Garis_v	Garis_h
1	A	0	0	0	1	2	2	0	1
2	B	0	1	0	0	3	1	1	3
3	C	0	1	0	0	2	1	0	0
4	D	0	0	0	0	2	2	1	2
5	E	0	2	0	0	3	1	1	3
6	F	0	2	0	0	2	1	1	2
7	G	0	1	0	0	3	2	1	1
8	H	0	0	1	1	1	1	2	1
9	I	0	0	0	0	1	1	0	1
10	J	1	0	0	0	1	1	1	0
11	K	0	1	1	1	1	1	1	0
12	L	0	1	0	0	1	1	1	1
13	M	0	0	1	2	1	4	2	0
14	N	0	0	1	1	1	3	2	0
15	O	0	0	0	0	2	2	0	0
16	P	0	1	0	0	2	2	1	0
17	Q	0	1	0	0	2	2	0	0
18	R	0	1	0	1	2	1	1	0
19	S	1	1	0	0	3	1	0	0
20	T	1	1	0	0	1	1	1	1
21	U	0	0	1	0	1	2	0	0
22	V	0	0	1	0	1	2	0	0
23	W	0	0	2	1	1	4	0	0
24	X	1	1	1	1	1	1	0	0
25	Y	1	0	1	0	2	1	0	0
26	Z	1	1	0	0	3	1	2	0
27	a	1	0	0	0	3	1	1	0
28	b	0	1	1	0	2	2	1	0
29	c	0	1	0	0	2	1	0	0
30	d	1	0	1	0	2	2	1	0
31	e	0	1	0	0	3	1	0	1
32	f	2	2	0	0	1	1	1	1
33	g	1	1	0	0	4	1	0	0
34	h	0	1	1	1	1	2	2	0
35	i	0	0	0	0	1	1	1	0
36	j	1	0	0	0	1	1	0	0
37	k	0	1	1	1	1	1	1	0
38	l	0	0	0	0	1	1	1	0
39	m	0	0	1	2	1	3	3	0
40	n	0	0	1	1	1	2	2	0
41	o	0	0	0	0	2	2	0	0
42	p	0	1	0	1	2	2	1	0
43	q	1	0	1	0	2	2	1	0
44	r	0	1	1	0	1	1	1	0
45	s	1	1	0	0	3	1	0	0
46	t	0	1	0	0	1	1	1	1
47	u	0	0	1	0	1	2	0	0
48	v	0	0	1	0	1	2	0	0
48	w	0	0	2	1	1	4	0	0
50	x	1	1	1	1	1	1	0	0
51	y	1	0	1	0	1	2	0	0
52	z	1	1	0	0	3	1	0	2

2. Kemudian lakukan perhitungan untuk mencocokan karakter apakah ciri – ciri di yang diinputkan memang menunjukkan karakter dari klasifikasi data dari atribut yang telah disajikan diatas atau bukan, maka buat tabel yang sama dengan blok yang berbeda karena data yang diinputkan menggunakan resolusi citra 300 ppi maka tentukan ciri – ciri dari file image yang diinputkan untuk mengetahui jenis karakter yang akan dikenali, tabel di

ISSN: 2548-9704

bawah ini merupakan ciri – ciri citra karakter yang akan dikenali .

Tabel 7. Dari Klasifikasi Ciri – Ciri Karakter yang di inputkan

No	Kar	T_kiri	T_kanan	T_atas	T_bawah	Int_v	Int_h	Garis_v	Garis_h	Hlokl
1	*	0	0	2	1	1	4	0	0	0.13656
2	*	1	1	1	1	1	1	0	0	0.13000
3	*	1	0	1	0	2	1	0	0	0.10333
4	*	1	1	0	0	3	1	2	0	0.13333

3. Lakukan perhitungan dengan mencocokkan atribut diatas dari atribut diatas.

Tabel 8. Perhitungan Dari Klasifikasi Ciri – Ciri Untuk Karakter no. 1

No.	Kar	Perhitungan	Hasil perhitungan
1	W	(0-0)^2+(0-0)^2+(1-2)^2+(1-1)^2+(.-.)^2+(4-4)^2+(0-0)^2+(0-0)^2+(0 - 0.13666)^2	0.01867
2	X	(1-0)^2+(1-0)^2+(1-2)^2+(.-.)^2+(1-1)^2+(1-4)^2+(0-0)^2+(0-0)^2+(0- 0.13666)^2	12.01.867
3	Y	(1-0)^2+(0-0)^2+(1-2)^2+(0-.)^2+(2-1)^2+(1-4)^2+(0-0)^2+(0-0)^2+(0 - 0.13666)^2	13.01.867
4	Z	(1-0)^2+(1-0)^2+(0-1)^2+(0-1)^2+(3-1)^2+(1-4)^2+(2-0)^2+(0-0)^2 + (0 - 0.13666)^2	24.01.867

Tabel 9.Perhitungan Dari Klasifikasi Ciri – Ciri Untuk Karakter no. 2

No.	Kar	Perhitungan	Hasil perhitungan
1	W	(0-1)^2+(0-1)^2+(2-1)^2+(1-1)^2+(1-1)^2+(4-1)^2+(0-0)^2+(0-0)^2+(0 - 0.13000)^2	12.0169
2	X	(1-1)^2+(1-1)^2+(1-1)^2+(1-1)^2+(1-1)^2+(.-.)^2+(0-0)^2+(0-0)^2+(0 - 0.13000)^2	0.0169
3	Y	(1-1)^2+(0-1)^2+(1-1)^2+(0-1)^2+(2-1)^2+(.-.)^2+(0-0)^2+(0-0)^2+(0 - 0.13000)^2	3.0169
4	Z	(1-1)^2+(1-1)^2+(0-1)^2+(0-1)^2+(3-1)^2+(1-1)^2+(2-0)^2+(0-0)^2 + (0- 0.13000)^2	10.0169

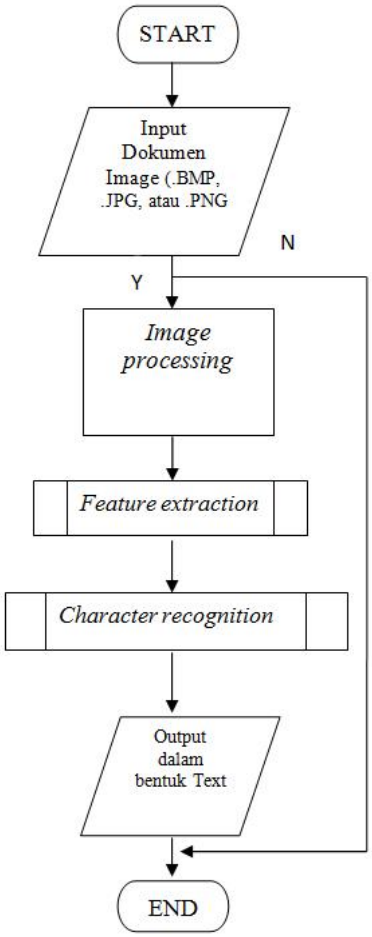
Tabel 9.Urutan Hasil Perhitungan Untuk Karakter Z

No	Kar	Hasil perhitungan
4	Z	0.01777
3	Y	7.01777
2	X	10.01777
1	W	24.01777

Berdasarkan hasil perhitungan dan pengelompokan ciri – ciri dari citra karakter maka karakter yang dikenali adalah sebuah karakter dengan huruf Z.

Dari hasil perhitungan dan pengurutan di atas didapatlah sebuah karakter dari *file image* hasil scan tadi yang telah dikonversi ke *file text* adalah karakter WXYZ.

Flowchart



Gambar 18. Rancangan *Form About*

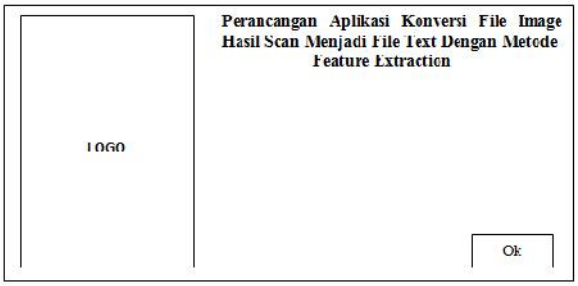
Rancangan Antarmuka

perancangan antarmuka adalah rancangan tampilan yang menghubungkan pengguna *(user)* dengan komputer melalui perantara berupa bantuan sebuah program.

ISSN: 2548-9704

Rancangan *Form* Tampilan Awal

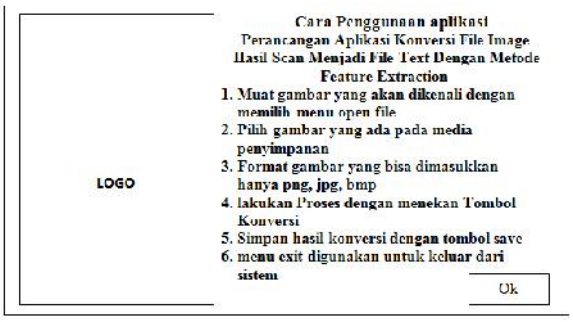
Form About merupakan tampilan yang dirancang untuk menampilkan informasi mengenai *programmer* perangkat lunak untuk melakukan sebuah konversi *file image* hasil *scan* menjadi *file text* dengan metode *feature extraction*. Adapun rancangan *form about* dapat dilihat pada gambar 3.10 di bawah ini.



Gambar 3.12. Rancangan *Form About*

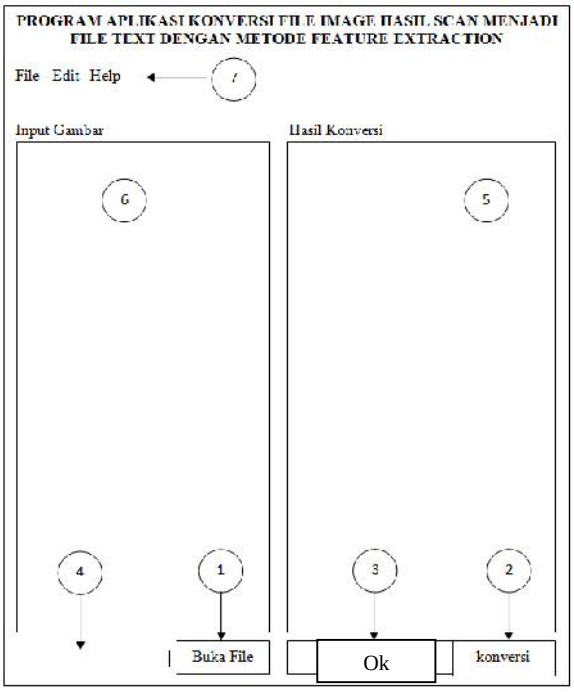
Rancangan *Form* Tampilan *Help*

Rancangan help adalah tampilan sederhana yang hanya memiliki satu tombol yaitu tombol keluar. Rancangan ini berhuna untuk menampilkan informasi tentang tata cara pengoperasian aplikasi yang dijelaskan secara bertahap.



Gambar 19. Rancangan *Form Help*

Rancangan *Form* Proses Aplikasi Utama



Gambar 20. Rancangan Aplikasi Utama

Adapun keterangan dari perancangan diatas adalah sebagai berikut :

1. *Buka File* : Digunakan untuk menampilan dari *image* yang telahdiiputkan.
2. Konversi : Digunakan untuk melakukan proses konversi pada *file* yang telah diinputkan.
3. *Progress Bar* : Digunakan untuk menampilkan lama proses yang dilakukanpada saat konversi.
4. Alamat *File* : Digunakan untuk menampilkan alamat *file* yang diambil
5. *Hasil konversi* : Digunakanuntuk menampilkan hasil dari proses konversi
6. *Input* Gambar : Digunakanuntuk menampilkan gambar yang akan dikonversi
7. *Menu* Aplikasi : Digunakan untuk menyimpan dan memberitahukan tentang informasi pembuat Program dan tata cara menggunakan aplikasi

ISSN: 2548-9704

4. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan pengujian yang dilakukan maka dapat disimpulkan bahwa :

1. Berdasarkan hasil pengujian, algoritma *feature extraction* memilikitingkat akurasi yang lebih baik. Karena algoritma *feature extraction* memiliki peluang untuk bisadikembangkan terutama pada ciri-ciri khusus citra karakter. Salah satucontohnya adalah fitur *stroke* (jumlah garis yang membentuk karakter).Berdasarkan proses yang dibutuhkan algoritma *featureextraction* membutuhkan proses yang lebih banyak dan berdasarkan hasil perhitungan, algoritma *feature extraction*memiliki kompleksitasberdasarkan dari ciri-ciri karakter.
2. Penerapan metode *feature extraction*bisa diterapkan pada aplikasi ini karena metode ini juga membahas bagaimana membaca sebuah karakter yang prosesnya hampir sama dengan aplikasi yang dirancang.
3. Perancangan aplikasi konversi ini lumayan mampu untuk membaca bagaimana karakter yang ada pada *file* berformat *file image* seperti format .**Bmp*,dan.**Jpg* walaupun masih ada banyak kekurangan pada aplikasi ini.

Saran

Penulis menyadari masih banyak kekurangan dari penelitian ini, adapun saran yang ingin disampaikan adalah :

1. Format file dapat lebih dikembangkan lagi karena aplikasi ini kedepannya dapat membaca format *file* tiff atau format *file* lainnya.
2. Gunakan citra masukan dengan karakter-karakter yang tidak saling menyambung.
3. *Scanning* citra karakter sebaiknya menggunakan resolusi 300 ppi.
4. Gunakan citra karakter dengan *font* berwarna hitam dan latar belakang berwarna putih.
5. Untuk pengembangan, gunakan metode segmentasi yang mampumemisahkan karakter-karakter yang saling menyambung. Untuk pengembangan, sebaiknya ditambahkan fitur jumlah *stroke*karakter

DAFTAR PUSTAKA

1. Agus purnama, **Defenisi Dan Pengolahan Citra Digital** (http://elektronika-dasar.com/teori-elektronika/definisi-dan-pengolahan-citra-digital/, diakses tanggal 17 April 2013).
2. Munir, Rinaldi, 2015. **Matematika Diskrit**Edisi Ketiga, Bandung : Informatika Bandung.
3. Jugiyanto HM, "***Analisis dan Desain"***, Penerbit Andi Offset, Yogyakarta, 2013.

Universitas Pamulang

Tanda Tangan Digital

Dokumen ini telah ditandatangani melalui

Aplikasi Digital Signature Universitas Pamulang