

**PENGENALAN POLA TULISAN TANGAN TEGAK BERSAMBUNG
MENGUNAKAN
METODE *PROJECTION-BASED* SEGMENTASI DAN *ARTIFICIAL
NEURAL NETWORK* (ANN)**

TUGAS AKHIR

Oleh:

JESIKA NAULY HUTAGALUNG
NIM. 151112262

FADLY PRANATA SIMANUNGKALIT
NIM. 151111992



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
MIKROSKIL
MEDAN
2019**

**PATTERN RECOGNITION HANDWRITING CURSIVE USING
PROJECTION-BASED SEGMENTATION METHOD AND ARTIFICIAL
NEURAL NETWORK (ANN)**

FINAL RESEARCH

By:

JESIKA NAULY HUTAGALUNG

NIM. 151112262

FADLY PRANATA SIMANUNGKALIT

NIM. 151111992



**STUDY PROGRAM OF INFORMATICS ENGINEERING
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
MIKROSKIL
MEDAN
2019**

LEMBAR PENGESAHAN
Pengenalan Pola Tulisan Tangan Tegak Bersambung
Menggunakan Metode Projection-Based Segmentasi
Dan Artificial Neural Network (ANN)

TUGAS AKHIR

Diajukan untuk Melengkapi Persyaratan Guna
Mendapatkan Gelar Sarjana Strata Satu
Program Studi Teknik Informatika

Oleh :

JESIKA NAULY HUTAGALUNG
NIM. 151112262

FADLY PRANATA SIMANUNGKALIT
NIM. 151111992

Disetujui Oleh :

Dosen Pembimbing I,

Dosen Pembimbing II

Ng Poi Wong, S.Kom., M.T.I

Ali Akbar Lubis, S.Kom., M.TI

Medan,.....20...

Diketahui dan Disahkan Oleh :

Ketua Program Studi
Teknik Informatika

Sunario Megawan, S.Kom., M.Kom

LEMBAR PERNYATAAN

Saya yang membuat pernyataan ini adalah mahasiswa Jurusan/program Studi S-1 Teknik Informatika STMIK Mikroskil Medan dengan identitas mahasiswa sebagai berikut :

Nama : Jesika Naully Hutagalung

Nim : 151112262

Peminatan : Komputasi Ilmiah

Saya telah melaksanakan penelitian dan penulisan Tugas Akhir dengan Judul "PENGENALAN POLA TULISAN TANGAN TEGAK BERSAMBUNG MENGGUNAKAN PROJECTION-BASED SEGMENTASI DAN ARTIFICIAL NEURAL NETWORK (ANN)", dengan ini saya menyatakan dengan sebenar-benarnya bahwa penelitian dan penulisan Tugas Akhir tersebut merupakan hasil karya saya sendiri (tidak menyuruh orang lain yang mengerjakannya) dan semua sumber, baik yang dikutip maupun dirujuk, telah saya nyatakan dengan benar. Bila di kemudian hari ternyata terbukti bahwa bukan saya yang telah mengerjakannya (membuatnya), maka saya bersedia dikenakan sanksi yang telah ditetapkan oleh STMIK Mikroskil Medan, yakni pencabutan ijazah yang telah saya terima dan ijazah tersebut dinyatakan tidak sah.

Selain itu, demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada STMIK Mikroskil Medan Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalty Free Right) atas Tugas Akhir saya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, STMIK Mikroskil Medan berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan Tugas Akhir saya, secara keseluruhan atau hanya sebagai atau hanya ringkasannya saja dalam bentuk format tercetak dan/atau elektronik, selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta. Menyatakan juga bahwa saya telah mempertahankan hak eksklusif saya untuk menggunakan seluruh atau sebagian isi Tugas Akhir saya guna pengembangan karya masa depan, misalnya dalam bentuk artikel, buku, ataupun perangkat lunak.

Demikian pernyataan ini saya perbuat dengan sungguh-sungguh, dalam keadaan sadar dan tanpa ada tekanan dari pihak manapun.

Medan, 25 April 2019

Saya yang membuat pernyataan,



Jesika Naully Hutagalung

LEMBAR PERNYATAAN

Saya yang membuat pernyataan ini adalah mahasiswa Jurusan/program Studi S-1 Teknik Informatika STMIK Mikroskil Medan dengan identitas mahasiswa sebagai berikut :

Nama : Fadly Pranata Simanungkalit

Nim : 151111992

Peminatan : Komputasi Ilmiah

Saya telah melaksanakan penelitian dan penulisan Tugas Akhir dengan Judul "PENGENALAN POLA TULISAN TANGAN TEGAK BERSAMBUNG MENGGUNAKAN PROJECTION-BASED SEGMENTASI DAN ARTIFICIAL NEURAL NETWORK (ANN)", dengan ini saya menyatakan dengan sebenar-benarnya bahwa penelitian dan penulisan Tugas Akhir tersebut merupakan hasil karya saya sendiri (tidak menyuruh orang lain yang mengerjakannya) dan semua sumber, baik yang dikutip maupun dirujuk, telah saya nyatakan dengan benar. Bila di kemudian hari ternyata terbukti bahwa bukan saya yang telah mengerjakannya (membuatnya), maka saya bersedia dikenakan sanksi yang telah ditetapkan oleh STMIK Mikroskil Medan, yakni pencabutan ijazah yang telah saya terima dan ijazah tersebut dinyatakan tidak sah.

Selain itu, demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada STMIK Mikroskil Medan Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalty Free Right) atas Tugas Akhir saya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, STMIK Mikroskil Medan berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan Tugas Akhir saya, secara keseluruhan atau hanya sebagai atau hanya ringkasannya saja dalam bentuk format tercetak dan/atau elektronik, selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta. Menyatakan juga bahwa saya telah mempertahankan hak eksklusif saya untuk menggunakan seluruh atau sebagian isi Tugas Akhir saya guna pengembangan karya masa depan, misalnya dalam bentuk artikel, buku, ataupun perangkat lunak.

Demikian pernyataan ini saya perbuat dengan sungguh-sungguh, dalam keadaan sadar dan tanpa ada tekanan dari pihak manapun.

Medan, 25 April 2019

Saya yang membuat pernyataan,



Fadly Pranata Simanungkalit

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Metodologi Penelitian	3
BAB II	5
TINJAUAN PUSTAKA	5
2.1 Citra	5
2.1.1 Citra Analog	5
2.1.2 Citra Digital	5
2.2 Pengolahan Citra	6
2.3 Metode Projection-Based Segmentasi	9
2.3.1 Profil Proyeksi	9
2.3.2 Tahapan Profil Proyeksi	10
2.4 Pengenalan Pola	12
2.5 Optical Character Recognition	13
2.6 Ekstraksi Ciri	14
2.7 Kecerdasan Buatan	16
2.8 Jaringan Syaraf Tiruan	16
2.9 Artificial Neural Network (ANN)	17
2.9.1 Jaringan Dengan Lapisan Tunggal (<i>Single Layer Neural Network</i>)	18
2.9.2 Jaringan Dengan Lapisan Jamak (<i>Multi Layer Neural Network</i>)	19
2.9.3 <i>Backpropagation</i>	19
2.9.4 Model <i>Backpropagation</i> pada <i>Artificial Neural Network</i>	20
2.9.5 Tahapan <i>Backpropagation</i>	22
2.10 Tulisan Tegak Bersambung	25

BAB III	26
ANALISIS DAN PERANCANGAN	26
3.1 Analisis Proses	26
3.1.1 Citra Awal	26
3.1.2 Analisis Proses Pra-Pengolahan	27
3.1.3 Analisi Proses Segmentasi dengan Metode <i>Projection Based Segmentation</i>	32
3.1.4 Analisis Proses Ekstraksi Ciri	37
3.1.5 Normalisasi Data Input	41
3.1.6 Analisis Proses Pengenalan Pola dengan Metode <i>Artificial Neural Network</i>	42
3.1.7 Analisis Proses Pengenalan Dengan <i>Backpropagation</i>	51
3.2 Analisis Kebutuhan Sistem	54
3.2.1 Analisis Kebutuhan Fungsional	54
3.2.2 Analisis Kebutuhan Non-Fungsional	58
3.3 Perancangan Tampilan	59
3.3.1 Perancangan Tampilan <i>Form</i> Halaman Utama	59
3.3.2 Perancangan Tampilan <i>Form</i> Pelatihan	61
3.3.3 Perancangan Tampilan Form Pengenalan	63
DAFTAR PUSTAKA	67

DAFTAR GAMBAR

Gambar 2.1 Dasar-Dasar Pengolahan Citra (Munir, 2004)	7
Gambar 2.2 Contoh Citra setelah ditajamkan (Munir, 2004)	7
Gambar 2.3 Contoh citra setelah debluring (Munir, 2004)	8
Gambar 2.4 Contoh citra hasil pendeteksian seluruh tepi (Ambarwati, Passarella, & Sutarno, 2016).....	9
Gambar 2.5 Profil Proyeksi Horizontal dan Vertikal (Widiarti, 2006)	10
Gambar 2.6 Penerapan Kecerdasan Buatan (Kvanicku, & Haykani 2011).....	16
Gambar 2.7 <i>Artificial Neural Netwok</i> dengan <i>Single Layer</i> (Siang, 2009).	19
Gambar 2.8 <i>Artificial Neural Netwok</i> dengan <i>Multi Layer</i> (Siang, 2009)	19
Gambar 2.9 Arsitektur Jaringan <i>Backpropagation</i> (Siang, 2005).	21
Gambar 2.10 Bentuk Tulisan Tangan yang Baku.....	25
Gambar 3.1 Skema Umum Pengenalan Pola Tulisan Tangan Tegak Bersambung.....	26
Gambar 3.2 Citra Awal Tulisan Tangan Tegak Bersambung	27
Gambar 3.3 <i>Flowchart</i> Pra-Pengolahan	28
Gambar 3.4 <i>Flowchart Metode Projection Based Segmentation</i>	32
Gambar 3.5 <i>Flowchart</i> Tahapan Profil Proyeksi Horizontal.....	33
Gambar 3.6 <i>Flowchart</i> Tahapan Profil Proyeksi Vertikal.....	35
Gambar 3.7 <i>Flowchart</i> Tahapan Perhitungan <i>Hu Invariant Moments</i>	37
Gambar 3.8 <i>Flowchart</i> Tahapan Algoritma <i>Backpropagation</i>	43
Gambar 3.9 Arsitektur Jaringan Saraf <i>Backpropagation</i>	44
Gambar 3.10 <i>Use Case</i> Pengenalan Pola Tulisan Tangan Tegak Bersambung	54
Gambar 3.11 Tampilan <i>Form</i> Halaman Utama	60
Gambar 3.12 Tampilan <i>Form</i> Pra-Pengolahan Pelatihan.....	62
Gambar 3.13 Tampilan <i>Form</i> Pelatihan	63
Gambar 3.14 Tampilan <i>Form</i> Pra-Pengolahan Pengenalan	64
Gambar 3.15 Tampilan <i>Form</i> Pengenalan.....	65

DAFTAR TABEL

Tabel 3.1 Nilai Piksel Pada Citra Tulisan Tangan Tegak Bersambung	27
Tabel 3.2 Nilai Piksel Citra <i>Grayscale</i>	30
Tabel 3.3 Penentuan Lokasi Entri Pada Kernel <i>Filter</i>	30
Tabel 3.4 Nilai Piksel Citra <i>Filter Gaussian</i>	31
Tabel 3.5 Nilai Piksel Citra Biner	31
Tabel 3.6 Nilai Piksel m00	38
Tabel 3.7 Nilai Piksel m10	38
Tabel 3.8 Nilai Piksel m01	38
Tabel 3.9 Nilai Piksel <i>Invarian Moment Hu</i>	41
Tabel 3.10 Contoh Identitas Pada Database	51
Tabel 3.11 Narasi <i>Use Case</i> Pelatihan	54
Tabel 3.12 Narasi <i>Use Case</i> Pengenalan	56

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknik pengenalan pola huruf tulisan tangan adalah sebuah teknik dimana input data yang berisi tulisan tangan dari lembaran kertas hasil scan menggunakan scanner dan menghasilkan gambar pada komputer yang dikenali sebagai titik-titik (bitmap), bitmap inilah yang kemudian diproses lebih lanjut menggunakan algoritma tertentu menjadi karakter, sehingga dapat dikenali dan diolah menjadi informasi (Kumar, S., 2009). Bagi manusia tentunya tidaklah sulit untuk mengenali sebuah huruf dalam tulisan tangan walaupun berbeda-beda bentuk antara penulis satu dengan penulis lain. Namun hal itu menjadi sulit jika mesin yang berusaha untuk mengenali tulisan tangan dari manusia yang berbeda-beda antara satu dan yang lainnya. Dalam kasus ini lebih sulit lagi jika tulisan tangan yang akan dikenali yaitu tulisan tangan tegak bersambung karena seringkali tulisan tangan tegak bersambung memiliki tulisan yang rapat yang mengakibatkan proses segmentasi untuk mendeteksi huruf pada citra menjadi lebih sulit.

Dalam berbagai tinjauan referensi yang telah dilakukan, penelitian untuk mengenali huruf dan angka tulisan tangan seperti yang dilakukan oleh (Sam'ani, & Haris, 2017) dengan mengimplementasikan metode *Convolution Neural Network (CNN)* memberikan hasil akurasi uji coba pada huruf dan angka tunggal sebanyak 184 citra uji diperoleh jawaban benar sebanyak 153 dan jawaban salah sebanyak 31. Sedangkan hasil uji coba dengan input berupa kata dan kalimat yang terdiri dari 191 huruf dan angka dengan jumlah jawaban benar sebanyak 158 dan jawaban salah sebanyak 33 huruf yang tidak dapat dikenali dengan benar. Berdasarkan penelitian tersebut untuk menguji tingkat akurasi dilakukan dengan metode *Convolution Neural Network (CNN)* namun untuk proses segmentasi tidak menggunakan metode khusus dan hanya dilakukan dengan teknik cropping. Pada penelitian yang lain dilakukan oleh (Rubul, & RuhUllah, 2018) untuk mengenali tulisan tangan Assamese dengan menggunakan *Texture* dan *Diagonal Orientation features* dengan *Artificial Neural Network*. Pada penelitian ini akuransi yang di dapatkan untuk proses segmentasi garis dan kata sekitar 98% dan akurasi untuk mengenali karakter sekitar 90,34%. Berdasarkan penelitian tersebut belum teruji untuk pengenalan pola tulisan tangan latin (bersambung) yang memiliki tingkat kesulitan yang lebih tinggi dalam proses pemecahan kata.

Segmentasi merupakan suatu proses untuk mendapatkan area atau obyek yang diinginkan pada suatu citra dengan memisahkan area atau objek dari latar belakangnya. Segmentasi bertujuan untuk mengumpulkan pixel-pixel objek menjadi wilayah yang merepresentasikan suatu objek (Hartanto et al., 2012). Pada penelitian ini dilakukan dengan menggabungkan metode *Projection-Based* Segmentasi dan *Artificial Neural Network* (ANN). Metode *Projection-Based* Segmentasi merupakan salah satu teknik segmentasi berdasarkan *Profil Proyeksi* yang diimplementasikan untuk menyelesaikan permasalahan segmentasi karakter pada citra baik *Optical Character Recognition* (OCR) maupun citra hasil buatan sendiri menggunakan komputer. Metode *Profil Proyeksi* banyak digunakan untuk segmentasi baris untuk dokumen cetak (*printed document*) dimana tulisan antar baris terpisah dengan jelas. Namun, metode ini juga dapat diadaptasi untuk dokumen tulisan tangan (Likforman-Sulem, Zahour, & Taconet, 2006). Kemudian metode tersebut dikombinasikan dengan *Artificial Neural Network* (ANN) yang merupakan salah satu bentuk kecerdasan buatan yang mempunyai kemampuan untuk belajar dari data dan tidak membutuhkan waktu lama dalam pembuatan model (Setiawan, & Rudiyanto, 2004). *Artificial Neural Network* (ANN) digunakan untuk meminimalkan error keluaran pada pengenalan pola untuk mencapai nilai kesalahan (Herman, & Harjoko, 2015). Namun, metode ini mampu mengenali pola yang akan digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa tapi tidak sama dengan pola yang dipakai selama pelatihan.

Berdasarkan uraian diatas, kami tertarik untuk melakukan penelitian terhadap tulisan tangan tegak bersambung dengan mengkombinasikan antara metode *Projection-Based* Segmentasi dan *Artificial Neural Network* (ANN) dengan judul **“Pengenalan Pola Tulisan Tangan Tegak Bersambung menggunakan Metode *Projection-Based* Segmentasi dan *Artificial Neural Network* (ANN)”**

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka yang menjadi rumusan masalah dalam tugas akhir ini adalah :

- a. Proses segmentasi pada citra tulisan tangan tegak bersambung untuk memisahkan karakter pada tiap baris dan tiap kolom secara otomatis.
- b. Mengenali pola karakter huruf dari citra tulisan tangan tegak bersambung yang seringkali memiliki tulisan yang rapat dan sulit dikenali oleh mesin.

1.3 Tujuan

Tujuan yang akan dicapai dalam penelitian tugas akhir ini adalah mengembangkan sistem pengenalan pola tulisan tangan tegak bersambung dengan implementasi metode *Projection-Based Segmentasi* dan *Artificial Neural Network* (ANN).

1.4 Manfaat

Manfaat dari Tugas Akhir ini adalah :

- a. Sistem diharapkan dapat mempermudah untuk mengubah tulisan tangan yang ditulis pada media kertas menjadi huruf digital sehingga efek baik dari menulis dengan tangan tetap diperoleh tanpa harus mengetik kembali teks yang ditulis dengan tangan.
- b. Sistem dapat digunakan sebagai referensi untuk penelitian dan pengembangan lebih lanjut di bidang pengolahan citra.

1.5 Batasan Masalah

Dalam upaya mencapai visi dalam penelitian tugas akhir ini, maka ruang lingkup pembahasan mencakup :

- a. Inputan berupa citra digital tulisan tangan tegak bersambung dengan format *input* .jpeg, .bmp, dan .png .
- b. Sistem yang digunakan hanya mengenali 26 huruf alphabet tegak bersambung termasuk huruf besar (A-Z) dan huruf kecil (a-z).

1.6 Metodologi Penelitian

Metodologi yang akan dilakukan dalam penyusunan tugas akhir ini adalah:

1. Studi Literatur

Studi literatur mencakup analisis terhadap hasil penelitian dunia dalam topik pengenalan pola, mengumpulkan bahan referensi yang berhubungan dengan penelitian, seperti : segmentasi, ekstraksi fitur dan analisis, klasifikasi, pengolahan citra digital, metode *Projection-Based Segmentasi* dan *Artificial Neural Network* (ANN), dan bahan referensi lain yang terkait dengan Tugas Akhir mengenai aplikasi pengenalan pola tulisan tangan tegak bersambung.

2. Pengembangan Sistem

Proses pengembangan sistem mengacu pada metode *Waterfall* dengan tahapan sebagai berikut :

a. Analisis

Proses analisis ini mencakup analisis proses dan kebutuhan. Diagram alir (*flowchart*) untuk mendeskripsikan rincian jalannya sistem dalam tiap tahap proses pada analisis proses dan tool UML Use Case Diagram digunakan untuk memodelkan kebutuhan fungsional sistem dan tool PIECES digunakan untuk memodelkan kebutuhan non-fungsional sistem.

b. Desain Sistem (*Design*)

Proses desain akan menerjemahkan hasil perancangan arsitektur algoritma ke sebuah perancangan aplikasi perangkat lunak yang akan dibuat. Proses ini akan berfokus pada tool Balsamiq.

c. Pengkodean (*Coding*)

Pengkodean merupakan proses menerjemahkan desain ke dalam suatu bahasa pemrograman yang bisa dimengerti oleh komputer. Proses *coding* dalam penelitian ini menggunakan bahasa pemrograman C# dari *Microsoft Visual Studio 2012*.

d. Pengujian (*Testing*)

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibuat dari hasil penggabungan metode *Projection-Based* Segmentasi dan *Artificial Neural Network* (ANN). Adapun hal-hal yang diuji pada saat program dijalankan adalah pengujian dilakukan dengan menginput citra tulisan tangan tegak bersambung dengan data sampel sebanyak 208 huruf citra tulisan tangan tegak bersambung yang diambil dari 4 orang berbeda. Pengujian difokuskan pada 52 huruf citra tulisan tangan tegak bersambung dari 1 orang yang telah di scan terlebih dahulu menggunakan scanner yang bertujuan untuk menguji apakah citra dapat menghasilkan output karakter huruf berupa teks yang sesuai dengan citra yang diinput atau tidak.

BAB II

TINJAUAN PUSTAKA

2.1 Citra

Citra (*image*) – istilah lain dari gambar, yang merupakan suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek (Sutojo, 2004). Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindra optik, misalnya mata manusia, kamera, scanner, dan sebagainya (Sapermana, 2016). Citra terbagi atas 2 jenis, yaitu citra yang bersifat analog dan citra yang bersifat digital (Sutojo, 2004).

Citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom dengan x dan y adalah koordinat spasial dan amplitudo f di titik koordinat (x, y) dinamakan intensitas atau tingkat keabuan dari citra (Putra, 2010).

2.1.1 Citra Analog

Citra analog adalah citra yang bersifat continue, seperti citra pada monitor, televisi, foto sinar X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan, citra - citra yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer, sehingga tidak bisa diproses di komputer secara langsung (Munir, 2004). Oleh sebab itu, agar ini dapat diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, CT scan, sensor rontgen untuk foto thorax, sensor gelombang pendek pada sistem radar, sensor ultrasound pada sistem USG, dan lain-lain (Munir, 2004).

2.1.2 Citra Digital

Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra atau *pixel* (Gonzales, & Woods, 2002). Citra digital adalah citra yang dapat diolah oleh komputer. Istilah citra digital sangat populer pada masa sekarang. Banyak peralatan elektronik, misalnya scanner, kamera digital, mikroskop digital, dan fingerprint reader (pembaca sidik jari), yang menghasilkan citra digital juga sangat populer digunakan oleh pengguna untuk mengolah foto atau untuk berbagai keperluan lain.

Sebagai contoh, Adobe Photoshop dan GIMP (GNU Image Manipulation Program) menyajikan berbagai fitur untuk memanipulasi citra digital (Abdul, & Adhi, 2013).

2.2 Pengolahan Citra

Pengolahan citra digital merupakan pengolahan dan analisis yang banyak melibatkan persepsi visual. Citra digital dapat diperoleh secara otomatis dari sistem penangkapan citra membentuk matrik yang elemenelemennya menyatakan nilai intensitas cahaya atau tingkat keabuan suatu piksel (Fadliansyah, 2007). Pengolahan citra adalah salah satu aplikasi yang dapat mengubah gambar menjadi suatu informasi (Marvin Wijaya, 2007). Meskipun citra kaya informasi, namun seringkali citra yang dimiliki mengalami penurunan mutu (degradasi). Seperti mengandung cacat atau derau (noise), warnanya terlalu kontras, kurang tajam, kabur (blurring) dan sebagainya. Agar citra yang mengalami gangguan mulai diinterpretasi (baik manusia maupun mesin) maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Operasi-operasi pada pengolahan citra diterapkan pada citra bila :

1. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung dalam citra.
2. Elemen di dalam citra perlu dikelompokkan dan diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra yang lain. Jadi masukannya adalah citra dan keluarannya juga citra. Namun citra keluaran mempunyai kualitas lebih baik dari pada citra masukan.

Secara umum operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut (Munir, 2004):



Gambar 2.1 Dasar-Dasar Pengolahan Citra (Munir, 2004)

1. Perbaikan kualitas citra (*image enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter parameter citra. Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan. Contoh-contoh operasi perbaikan citra:

- a. Perbaikan kontra gelap/terang.
- b. Perbaikan tepian objek (*edge enhancement*).
- c. Penajaman (*sharpening*).
- d. Pemberian warna semu (*pseudocoloring*).
- e. *Noise filtering*.



Gambar 2.2 Contoh Citra setelah ditajamkan (Munir, 2004)

2. Pemugaran citra (*image restoration*)

Pemugaran citra merupakan proses merekonstruksi atau mendapatkan kembali citra asli dari citra yang cacat atau terdegradasi agar dapat menyerupai citra aslinya. Pemugaran citra berkaitan dengan penghilang atau pengurangan degradasi pada citra yang terjadi karena proses akusisi. Citra degradasi yang dimaksud termasuk derau (yang merupakan *error* dalam nilai *pixel*) atau efek optik misalnya *blur* (citra kabur) akibat kamera yang tidak fokus atau karena gerakan kamera (Marvin Wijaya, 2007). Operasi pemugaran citra bertujuan untuk menghilangkan atau meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar dapat diketahui. Contoh-contoh operasi pemugaran citra:

- a. Penghilangan kesamaran (*deblurring*).
- b. Penghilau derau (*noise*).



Gambar 2.3 Contoh citra setelah debluring (Munir, 2004)

3. Pemampatan citra (*image compression*)

Jenis operasi ini dilakukan agar citra dapat dipresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG.

4. Segmentasi citra (*image segmentation*)

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.

5. Analisis citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra masuk untuk menghasilkan informasi citra. Teknik pengolahan citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Contoh-contoh operasi analisis citra:

- a. Pendeteksian tepi objek (*edge detection*).
- b. Ekstraksi batas (*boundary*).
- c. Representasi daerah (*region*).



Gambar 2.4 Contoh citra hasil pendeteksian seluruh tepi (Ambarwati, Passarella, & Sutarno, 2016)

6. Rekonstruksi citra

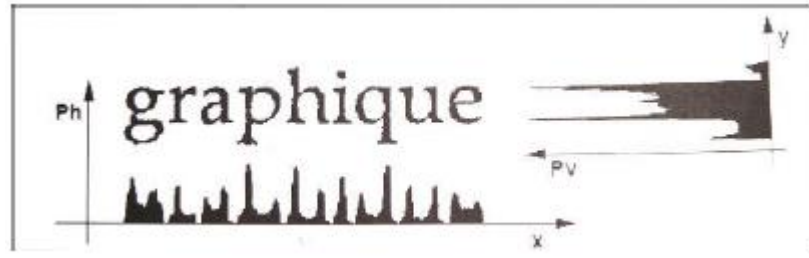
Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa foto rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

2.3 Metode Projection-Based Segmentasi

Metode berbasis proyeksi. Metode segmentasi yang umum dan menjadi dasar pada pengembangan metode segmentasi selanjutnya adalah metode profil proyeksi (dos Santos dkk, 2009). Umumnya, metode profil proyeksi dimanfaatkan untuk memisahkan baris pada dokumen cetak dimana spasi antar baris terkelompokkan dengan jelas. Tetapi, ada juga hasil penelitian yang menerapkan metode profil proyeksi untuk dokumen tulisan tangan (Likforman-Sulem, Zahour, & Taconet, 2006).

2.3.1 Profil Proyeksi

Proyeksi citra adalah bentuk satu dimensi isi citra yang dihitung berdasarkan sumbu koordinat. Proses segmentasi dengan menggunakan profil proyeksi dilakukan dengan mengekstraksi ciri berdasarkan profil proyeksi dari garis pada teks. Profil proyeksi akan mencari garis pada teks secara vertikal dan horizontal, dimana proyeksi horizontal akan mendapatkan baris teks kemudian proyeksi vertikal akan memisahkan setiap kolom teks. Dimana $S(N, M)$ merupakan citra biner dengan N baris dan M kolom.



Gambar 2.5 Profil Proyeksi Horizontal dan Vertikal (Widiarti, 2006)

Profil proyeksi terdapat 2 jenis yaitu :

1. Proyeksi Horizontal

Menjumlahkan pixel putih yang tegak lurus dengan sumbu x, yang diwakili vektor P_{hor} dengan ukuran M yang didefinisikan sebagai :

$$P_{hor}(b) = \sum_{j=1}^N I[b, k_j] \quad (1)$$

2. Proyeksi Vertikal

Menjumlahkan pixel putih yang tegak lurus dengan sumbu y, yang diwakili vektor P_{ver} dengan ukuran N yang didefinisikan sebagai :

$$P_{ver}(k) = \sum_{j=1}^M I[b_k, k] \quad (2)$$

Dalam hal ini M adalah tinggi citra dan N adalah lebar citra. Definisi diatas menyatakan bahwa proyeksi vertikal $P_{ver}(k)$ adalah jumlah piksel pada kolom k citra, sedangkan $P_{hor}(b)$ adalah jumlah piksel pada baris b citra (Kadir dkk, 2013).

2.3.2 Tahapan Profil Proyeksi

Pre-processing adalah tahap pertama dari analisis dokumen. Tujuannya adalah untuk meningkatkan kualitas gambar diproses. Sebuah metode secara berarti meningkatkan visibilitas beberapa objek hampir tidak dikenali digunakan oleh (Fabija'nska, 2014). Namun, teks pada gambar tulisan tangan juga terlihat dan mereka hanya perlu pembersihan noise. Dengan demikian, *pre-processing* yang digunakan dalam metode yang diusulkan terdiri dari langkah-langkah berikut:

1. Dasar dari metode yang diusulkan, serta algoritma yang diterapkan, adalah proyeksi profile dari *grayscale* atau citra biner. Gambar grayscale ditentukan dengan menghitung jumlah R, G dan B komponen untuk setiap pixel dari gambar warna :

$$S = 0,2989 * R + 0,5870 * G + 0,1140 * B \quad (3)$$

Keterangan :

Bobot jumlah tersebut berasal dari model Y'UV dan Y'IQ yang digunakan oleh PAL dan NTSC.

2. *Pre-processing* binarisasi. Sebuah gambar grayscale diubah menjadi satu biner menggunakan thresholding. Piksel luminance lebih besar dari ambang batas diganti dengan putih dan lainnya piksel dengan warna hitam. Secara matematis dapat ditulis sebagai berikut.

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T \\ 0, & \text{if } f(x, y) < T \end{cases} \quad (4)$$

Dengan $g(x, y)$ adalah citra biner dari citra grayscale $f(x, y)$, dan T menyatakan nilai threshold (Purba, D., 2010). Persamaan untuk menentukan nilai thresholding dapat dirumuskan sebagai berikut :

$$T = \frac{\max(f(x, y), (y, x) \in W) + \min(f(y, x), (y, x) \in W)}{2} \quad (5)$$

3. Setelah binarisasi, pengurangan noise dilakukan dengan proses *filtering*. Beberapa proyeksi berbasis metode menggunakan grafik *filtering*. *Filtering* merupakan suatu proses pengambilan sebagian sinyal dari frekwensi tertentu dan menempatkan sinyal tersebut pada frekwensi yang lain. *Filtering* pada citra juga menggunakan prinsip yang sama, yaitu mengambil fungsi citra pada piksel-piksel tertentu dan menempatkan atau mengantikan fungsi citra tersebut pada piksel-piksel tertentu. *Filter Gaussian* sangat baik untuk menghilangkan noise yang bersifat sebaran nomal, yang banyak di jumpai pada sebaran citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri (Usman, 2005). *Filter Gaussian* juga termasuk salah satu metode linear filter yang banyak digunakan untuk mengaburkan gambar atau mengurangi noise (Chandel, & Gupta, 2013). Bentuk fungsi *Gaussian* dapat dilihat pada persamaan berikut :

$$G(i, j) = c \cdot e^{-\frac{(i - u)^2 + (j - v)^2}{2\sigma^2}} \quad (6)$$

Keterangan :

c dan σ = konstanta

$G(i, j)$ = elemen matriks kernel *gauss* pada posisi (i, j)

u, v = indeks tengah dari matriks kernel *gauss*

Dalam hal ini, σ adalah deviasi standar dan piksel pada pusat (x, y) mendapatkan bobot terbesar berupa 1. Data yang diproses dapat dilakukan algoritma segmentasi.

4. Implementasi dalam aplikasi ini profil proyeksi dibagi menjadi dua tipe yaitu *Horizontal Projection Profile* (HPP) dan *Vertical Projection Profile* (VPP). *Horizontal projection profile* digunakan untuk mencari baris dari dokumen tersebut. Sedangkan *vertical projection profile* digunakan untuk mensegmentasi karakter dari tiap-tiap baris. Berikut ini adalah algoritma untuk menghitung *horizontal projection profile*:

- a. Lakukan perulangan untuk setiap pixel dalam gambar.
- b. Untuk tiap baris, hitung nilai dengan persamaan rumus (1).

Algoritma untuk menghitung *vertical projection profile*:

- a. Lakukan perulangan untuk setiap baris yang sudah ditemukan dengan *Horizontal Projection Profile*.
- b. Dalam satu baris lakukan perulangan sebesar baris yang sudah disegmentasi tersebut.
- c. Untuk tiap kolom, hitung nilai dengan persamaan rumus (2).

2.4 Pengenalan Pola

Pola adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Pola bisa merupakan kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor atau matriks. Secara umum pengenalan pola (Pattern Recognition) adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu obyek (Putra, 2010). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Tujuan dari pengenalan pola ini untuk mengklasifikasi dan mendeskripsikan pola atau objek kompleks melalui ciri-ciri objek tersebut.

Pengenalan pola pada dasarnya terdiri dari 3 langkah utama, yaitu *preprocessing*, ekstraksi ciri dan pengenalan. *Preprocessing* merupakan langkah awal dimana dilakukan dari keseluruhan data objek yang ada agar dapat menghasilkan ciri yang lebih baik pada tahap berikutnya. Pada tahap ini informasi yang dianggap penting akan lebih ditonjolkan. Tahap selanjutnya adalah ekstraksi ciri, tahap ini berfungsi untuk menemukan karakteristik pembeda yang mewakili sifat utama suatu data objek, sekaligus mengurangi jumlah data tersebut menjadi lebih sedikit tetapi representatif. Tahap akhir yaitu pengenalan, pada tahap ini data

yang akan dikelompokkan menjadi kelas yang sesuai. Salah satu contoh implementasi dari pengenalan pola adalah pengenalan tulisan tangan (Hastiana, 2010).

2.5 Optical Character Recognition

Optical Character Recognition (OCR) adalah Sebuah aplikasi komputer yang digunakan untuk mengidentifikasi citra huruf maupun angka untuk dikonversi kedalam bentuk file tulisan (Cheriet dkk, 2006). Adanya sistem pengenalan huruf ini akan meningkatkan fleksibilitas ataupun kemampuan dan kecerdasan sistem komputer. *Optical Character Recognition* (OCR) dapat dipandang sebagai bagian dari pengenalan otomatis yang lebih luas yakni pengenalan pola otomatis (automatic pattern recognition). Ada banyak pendekatan yang dapat dipakai untuk mengembangkan pembuatan pengenalan pola otomatis antara lain memakai pendekatan numerik, statistik, sintaktik, neural dan, aturan produksi (rule-based). Secara umum metode-metode tersebut dapat digolongkan menjadi dua kelompok metode yakni metode berbasis statistik dan metode berbasis struktur (Sukmawan, 2008).

Sebelum menjadi teks mentah, karakter sebuah citra melalui serangkaian proses sehingga dapat dikenali satu-persatu (Pangestu, 2015).

1. Preprocessing

Tahap ini merupakan tahap awal penerjemahan. Dalam preprocessing, dilakukan penghilangan noise agar pola-pola yang tidak dibutuhkan dihilangkan.

2. Segmentasi

Tahap ini merupakan tahap dimana aplikasi menentukan letak teks dalam citra sehingga memudahkan pemilihan wilayah pemindaian. Proses ini akan membagi citra menjadi dua wilayah, yaitu wilayah latar dan wilayah teks. Setelah melakukan pembagian, OCR hanya akan melakukan proses selanjutnya pada wilayah teks yang sudah tersegmentasi.

3. Normalisasi

Tahap ini merupakan tahap pengecekan bentuk, pola, maupun ketebalan dari setiap karakter yang terdeteksi. Dimisalkan OCR menerima masukan berupa sebuah kata yang terdiri dari berbagai ukuran. Pada proses normalisasi akan dideteksi perbedaan tersebut agar membentuk sebuah keseragaman sehingga mempermudah OCR untuk melakukan ekstraksi di tahap selanjutnya.

4. Ekstraksi

Tahap ini merupakan pengambilan ciri khas dari karakter yang terdeteksi dan ternormalisasi. Hal ini menyebabkan OCR dapat mengetahui typeface dari tulisan yang digunakan.

5. Recognition

Tahap ini merupakan tahap akhir penerjemahan. informasi-informasi yang sudah diambil dari ekstraksi. Karakter yang sudah dipisah satu-persatu tersebut kemudian akan dibandingkan dengan pola-pola karakter yang bersesuaian dengan pola yang sudah ada, biasanya disimpan di dalam sebuah basis data.

2.6 Ekstraksi Ciri

Feature Extraction atau dalam bahasa Indonesia disebut juga Ekstraksi Ciri. Feature Extraction merupakan proses terpenting dalam pengenalan objek. Fitur adalah karakteristik atau hal-hal khusus yang mencirikan satu objek. Ekstraksi fitur atau Feature Extraction berarti mengambil dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra. Fitur-fitur inilah yang kemudian akan digunakan sebagai pembanding untuk mengenali objek tertentu pada suatu citra. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan sebagai berikut (Arfian, 2012) :

- a. Dapat membedakan suatu objek dengan yang lainnya.
- b. Kompleksitas komputasi yang tidak terlalu rumit.
- c. Tidak terikat (invarian) terhadap transformasi.
- d. Jumlahnya sedikit.

Moment Invariant merupakan sebuah metode pengambilan ciri dari sebuah objek. Ciri yang diambil dapat berupa posisi, area, orientasi dan ciri lainnya. Metode ini dikenalkan oleh Hu pada tahun 1962 (Rizanti dkk, 2013). Diberikan sebuah fungsi $I(x,y)$, moment didefinisikan sebagai berikut :

Momen spasial orde (m,n) dapat didefinisikan sebagai berikut (Kadir dkk, 2013):

$$M_{i,j} = \sum_{x=0}^M \sum_{y=0}^N x^i y^j I(x,y) \quad (7)$$

Dengan ketentuan sebagai berikut :

$i, j = 0,1,2,\dots$, dengan I_j menyatakan orde momen

M = jumlah kolom dalam baris

N = jumlah baris pada citra

x = ordinat piksel

y = absis piksel

$I(x, y)$ = intensitas piksel pada posisi (x, y)

Untuk menormalisasi invariant translasi dalam bidang citra, centroid citra digunakan menentukan moment pusat. Koordinat pusat gravitasi dari citra dihitung dengan menggunakan persamaan (2) dan diberikan oleh:

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ dan } \bar{y} = \frac{m_{01}}{m_{00}} \quad (8)$$

Selanjutnya, moment pusat dapat ditentukan secara diskrit seperti berikut:

$$\mu_{i,j} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^i (y - \bar{y})^j I(x, y) \quad (9)$$

Moment selanjutnya dinormalisasi untuk efek perubahan skala dengan menggunakan rumus sebagai berikut :

$$\eta_{i,j} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (10)$$

Dimana faktor normalisasi $\gamma = \left(\frac{p+q}{2}\right) + 1$. Hasil dari momen yang sudah ternormalisasi tersebut, dihasilkan 7 Hasil dari momen yang sudah ternormalisasi tersebut, maka dihasilkan 7 Hu moment invariants yang dapat dilihat pada rumus (9) sampai dengan (15) (Huang & Leng, 2010) :

$$\phi_1 = \eta_{20} + \eta_{02} \quad (11)$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + (4\eta_{11})^2 \quad (12)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (13)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (14)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{(3\eta_{21} - \eta_{03})^2 - (\eta_{21} + \eta_{03})^2\} \end{aligned} \quad (15)$$

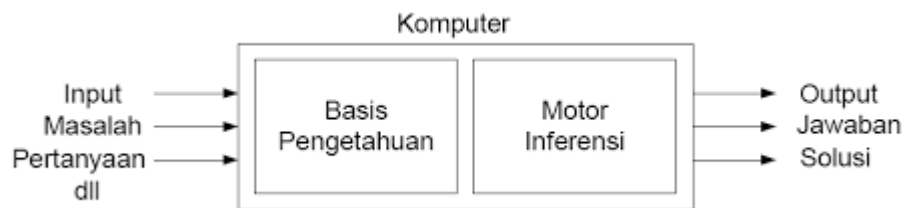
$$\begin{aligned} \phi_6 = & (\eta_{20} + \eta_{02})\{(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\} \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \quad (16)$$

$$\begin{aligned}\phi_7 = & (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{30})^2\} \\ & + (3\eta_{21} - \eta_{03})(3\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2\}\end{aligned}\quad (17)$$

2.7 Kecerdasan Buatan

Kecerdasan buatan (*Artificial Intelligence*) adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia (Kvanicku, & Haykani 2011). Penerapan kecerdasan buatan digambarkan pada Gambar Dua bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan adalah (Kvanicku, & Haykani 2011):

- Basis pengetahuan (knowledge base): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- Motor inferensi (inference engine) : kemampuan menarik kesimpulan berdasarkan pengetahuan.



Gambar 2.6 Penerapan Kecerdasan Buatan (Kvanicku, & Haykani 2011)

2.8 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba mensimulasikan proses pembelajaran pada otak manusia tersebut, istilah buatan digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Inggit Prahesti, 2013). Jaringan syaraf tiruan merupakan generalisasi model matematis yang disusun dengan asumsi yang sama seperti jaringan syaraf biologis (Siang, 2009) :

- Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- Sinyal dikirimkan di antara neuron-neuron melalui penghubung-penghubung.
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.

2.9 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) adalah suatu metode pengelompokan dan pemisahan data yang prinsip kerjanya sama seperti *neural network* pada manusia. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. *Artificial neural network* dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (Puspitaningrum, 2006). *Artificial Neural Network* dapat memperkirakan rentang yang cukup luas suatu model statistika dan fleksibel dalam menggambarkan model (linier maupun nonlinier) (Bar-Yam, 2008). *Artificial Neural Network* dipakai untuk mengenali pola (misalnya: huruf, angka, suara dan tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpai (Siang, 2009).

Kegunaan dari *Artificial Neural Network* menurut (Siang, 2009) adalah sebagai berikut:

a. Pengenalan pola (*Pattern Recognition*)

Artificial Neural Network dipakai untuk mengenali pola (misalnya: huruf, angka, suara dan tanda tangan) yang sudah sedikit berubah. Hal ini mirip dengan otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpainya (mungkin wajah/bentuk tubuhnya sudah sedikit berubah).

b. Signal Processing

Artificial Neural Network (model ADALINE) dapat dipakai untuk menekan *noise* dalam saluran telepon.

c. Peramalan

Artificial Neural Network juga dapat dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang ada di masa lampau. Ini dapat dilakukan mengingat kemampuan *Artificial Neural Network* dan membuat generalisasi dari apa yang sudah ada sebelumnya.

Artificial Neural Network memiliki fungsi aktivasi yang digunakan untuk keluaran suatu *neuron*. Argumen fungsi aktivasi adalah *net* masukan (kombinasi linier masukan dan bobotnya) (Siang, 2009) yang dinyatakan sebagai berikut :

$$net = \sum x_i w_i \quad (18)$$

Dengan fungsi aktivasi

$$F(net) = f\left(\sum x_i w_i\right) \quad (19)$$

Fungsi aktivasi yang sering digunakan dalam *Artificial neural network*, yaitu :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (20)$$

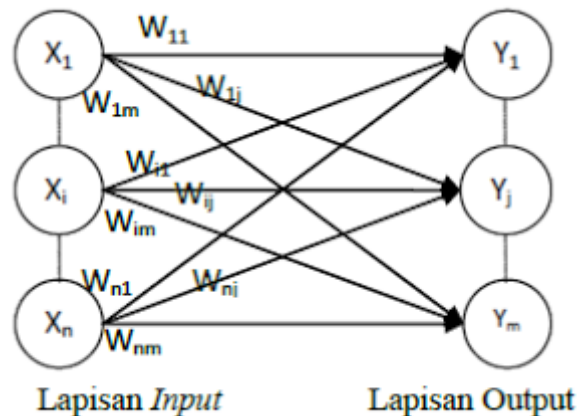
Proses dalam *Artificial Neural Network* dibedakan menjadi tiga tahap utama yaitu *feed forward*, *backpropagation* dan *update* nilai bobot. Pada tahap *feed forward* dilakukan proses dari input sampai dengan diperoleh hasil output, sedangkan pada tahap *backpropagation* dilakukan proses perbandingan nilai output dari tahap *feed forward* dengan nilai target yang telah ditentukan, kemudian dilanjutkan ke depan sampai *input layer* sehingga diperoleh nilai *error*. Pada tahap *update* nilai bobot dilakukan peng-*update*-an nilai bobot sampai diperoleh *error* yang minimal.

2.9.1 Jaringan Dengan Lapisan Tunggal (*Single Layer Neural Network*)

Jaringan dengan lapisan tunggal terdiri dari 1 lapisan *input* dan 1 lapisan *output* (Sutojo, 2010) Pada jaringan ini, sekumpulan masukan *neuron* dihubungkan langsung dengan sekumpulan keluarannya. Sinyal mengalir searah dari lapisan masukan sampai lapisan keluaran.

Setiap simpul dihubungkan dengan simpul lainnya yang berada di atasnya dan di bawahnya, tetapi tidak dengan simpul yang berbeda pada lapisan yang sama. Model yang masuk kategori ini antara lain: *ADALINE*, *Hopefield*, *Perceptron*, *LVQ*, dan lain-lain (Siang, 2009).

Pada gambar berikut diperlihatkan jaringan dengan n unit masukan (x_1, x_2, \dots, x_n) dan m keluaran (y_1, y_2, \dots, y_m).

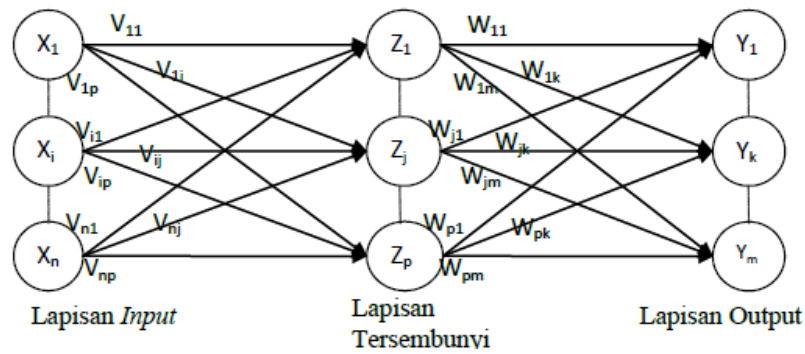


Gambar 2.7 *Artificial Neural Network* dengan *Single Layer* (Siang, 2009).

2.9.2 Jaringan Dengan Lapisan Jamak (*Multi Layer Neural Network*)

Jaringan lapisan banyak mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output* (Sutojo, 2010). Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih baik kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama tentu dengan pembelajaran yang lebih rumit. Akan tetapi, pada beberapa kasus pembelajaran pada jaringan yang mempunyai banyak lapisan ini lebih sukses dalam menyelesaikan masalah. Model yang termasuk kategori ini antara lain: *MADALINE* dan *Backpropagation* (Siang, 2009).

Pada gambar berikut diperlihatkan jaringan dengan n unit masukan (x_1, x_2, \dots, x_n) dan layar tersembunyi yang terdiri dari m (y_1, y_2, \dots, y_m) dan 1 unit keluaran.



Gambar 2.8 *Artificial Neural Network* dengan *Multi Layer* (Siang, 2009)

2.9.3 *Backpropagation*

Proses dalam tahap ini merupakan proses koreksi terhadap nilai *output* yang dihasilkan oleh tahap sebelumnya. Setiap nilai *output* akan dibandingkan dengan nilai target, dan dicari besarnya *error* untuk tiap nilai tersebut. Dimana nilai *error* tersebut akan digunakan sebagai dasar untuk meng-*update* nilai bobot yang menghubungkan antara *hidden layer* dan *output layer*. Pada tahap *backpropagation* proses dilakukan dari *output layer* ke *input layer*.

Dalam proses *backpropagation*, bentuk fungsi *error* yang dieliminasi adalah:

$$E = \frac{1}{2} \sum_{k=1}^r (t_k - y_k^{(T)})^2 \quad (21)$$

Dimana $y^{(T)}$ merupakan nilai *output* pada *layer* terakhir.

Nilai vektor selisih yang dievaluasi dengan perulangan mundur $\tau = \tau - 1, \tau - 2, \dots, 1$ adalah:

$$\delta_k^\tau(i) = \left[\sum_{l=1}^{\tau+1} \delta_k^{\tau+1}(l) W^{\tau+1}(li) \right] \frac{\partial f^\tau}{\partial n_k^\tau(i)} \quad (22)$$

Secara umum, algoritma dalam proses *backpropagation* adalah sebagai berikut:

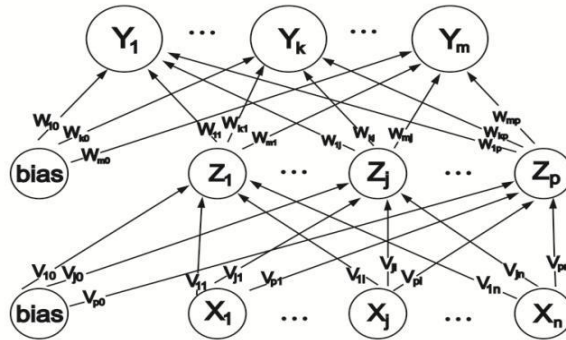
1. Bandingkan setiap nilai *output* yang dihasilkan ($y_k, k = 1, 2, \dots, m$) dengan nilai target (t_k).
2. Hitung nilai koreksi untuk nilai bobot yang menghubungkan *hidden node* dan *output node*, serta *output node* dan *bias*.
3. Nilai *error* yang diperoleh dari *output layer* dilanjutkan keseluruhan *node* yang ada pada *hidden layer*, kemudian dikalikan dengan nilai bobot yang ada pada jalur yang dilaluinya. Setiap nilai yang diterima oleh *hidden layer node* akan dijumlahkan.
4. Hitung nilai koreksi untuk nilai bobot antara *input node* dan *hidden node* serta antara *hidden node* dan *bias*.

Proses estimasi dengan menggunakan *backpropagation*, terdapat beberapa permasalahan yang sering dijumpai antara lain pemilihan nilai awal, lokal minimum, jumlah *neuron* pada lapisan *hidden* dan lain-lain. Meskipun telah ada beberapa metode untuk memperkecil peluang munculnya permasalahan tersebut, akan tetapi tidak menjamin bahwa estimasi *backpropagation* akan memberikan hasil yang lebih baik (Fauset, 1994).

2.9.4 Model *Backpropagation* pada *Artificial Neural Network*

Backpropagation atau propagasi balik merupakan suatu teknik pembelajaran atau pelatihan *supervised learning* yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks. Di dalam jaringan propagasi balik ini, setiap unit yang berbeda dilapisan *input* terhubung dengan setiap unit yang ada di lapisan *output*. Jaringan ini terdiri dari banyak lapisan. Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit-unit dilapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respon sebagai keluaran *Artificial neural network*. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan (Puspitaningrum, 2006).

Algoritma *Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur. Untuk mendapat *error* ini, tahap perambatan maju harus dikerjakan terlebih dahulu. Pada saat perambatan maju, *neuron-neuron* diaktifkan dengan menggunakan fungsi aktivasi *sigmoid* atau *sigmoid biner* (Siang, 2005).



Gambar 2.9 Arsitektur Jaringan *Backpropagation* (Siang, 2005).

Metode pelatihan *backpropagation* terdiri dari 3 tahapan yaitu: Tahap perambatan maju (*forward propagation*), tahap perambatan mundur (*backward propagation*), tahap pengupdatean bobot dan bias. Berikut adalah penjelasan dari ketiga tahapan *backpropagation*:

1. Tahap I: Perambatan Maju (*forward propagation*)

Selama perambatan maju, sinyal masukan ($=X_i$) dirambatkan ke lapisan tersembunyi. Keluaran dari setiap unit lapisan tersembunyi (Z_j) tersebut selanjutnya dirambatkan maju lagi ke lapisan tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ($=y_k$). Berikutnya, keluaran jaringan ($=y_k$) dibandingkan dengan target yang harus dicapai ($=t_k$). Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

2. Tahap II: Perambatan Mundur (*backward propagation*)

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_k ($k = 1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . Δk juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor δ_j di setiap unit di lapisan tersembunyi sebagai dasar perubahan bobot semua garis yang berasal

dari unit tersembunyi di lapisan di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

3. Tahap III: Perubahan Bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layer atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke lapisan keluaran didasarkan atas δ_k yang ada di unit keluaran. Ketiga tahap tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan. Algoritma propagasi balik menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward propagation*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi *sigmoid bipolar*, yaitu:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

2.9.5 Tahapan *Backpropagation*

Algoritma pelatihan jaringan *backpropagation* dengan satu layer tersembunyi (dengan fungsi aktivasi *Sigmoid Biner*), adalah sebagai berikut:

- Langkah 0 : Membuat inisialisasi semua bobot dengan kecil (Siang, 2009).
- Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9.
- Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8
- Fase I : Propagasi maju (Forward Propagation)
- Langkah 3 : Tiap unit masukan $X_i = (i = 1, 2, 3, \dots, n)$ menerima sinyal X_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya.
- Langkah 4 : Tiap-tiap unit lapisan tersembunyi $Z_j = (j = 1, 2, 3, \dots, p)$ menjumlahkan sinyal-sinyal *input* terbobot:

$$Z_{in_j} = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (23)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi:

$$Z_j = f(Z_{in_j}) = \frac{1}{1 + e^{-Z_{in_j}}} \quad (24)$$

Dan dikirimkan sinyal tersebut ke semua unit di lapisan atasnya.

Langkah 5 : Hasil fungsi tersebut dikirim ke semua unit pada lapisan di atasnya. Tiap-tiap unit keluaran $Y_k = (k = 1, 2, 3, \dots, m)$ menjumlahkan sinyal-sinyal *input* terbobot.

$$Y_{in_k} = W_{ok} + \sum_{i=1}^p Z_j W_{jk} \quad (25)$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi:

$$Y_k = f(Y_{in_k}) = \frac{1}{1 + e^{-Y_{in_k}}} \quad (26)$$

Dan dikirimkan sinyal tersebut ke semua unit di lapisan atasnya.

Fase II : Propagasi Mundur (*Backpropagation*)

Langkah 6 : Tiap-tiap unit keluaran $Y_k (k = 1, 2, 3, \dots, m)$ menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*:

$$\delta_k = (t_k - Y_k) f'(Y_{in_k}) \quad (27)$$

$$= (t_k - Y_k) y_k (1 - Y_k) \quad (28)$$

Kemudian dihitung koreksi nilai bobot yang kemudian akan digunakan untuk memperbaharui nilai W_{jk} :

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (29)$$

Hitung koreksi nilai *bias* yang kemudian akan digunakan untuk memperbaharui nilai W_{ok} :

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (30)$$

Kirimkan δ_k ke unit-unit yang ada dilapisan bawahnya.

Langkah 7 : Tiap - tiap unit *hidden* $Z_j = (j = 1,2,3, \dots, p)$
menjumlahkan *delta input* (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_j W_{jk} \quad (31)$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung informasi kesalahan:

$$\delta_j = \delta_{in_j} f'(Z_{in_j}) \quad (32)$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui V_{Ij} :

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (33)$$

Hitung juga koreksi *bias* (yang nantinya akan digunakan untuk memperbaiki nilai V_{0j}):

$$\Delta V_{jk} = \alpha \delta_j \quad (34)$$

Fase III : Perubahan bobot

Langkah 8 : Tiap-tiap unit *output* $Y_k (k = 1,2,3, \dots, m)$ memperbaiki *bias* dan bobotnya ($j = 1,2,3, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (35)$$

Tiap-tiap unit *hidden* $Z_j (j = 1,2,3, \dots, p)$ memperbaiki *bias* dan bobotnya ($i = 1,2,3, \dots, n$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (36)$$

Langkah 9 : Menghitung *mean square error*, jika nilai *mean square error* belum lebih kecil daripada target *error*, maka langkah 2-8 terus dilakukan (Kusumadewi, 2004).

Setelah pelatihan selesai, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, langkah 4 dan 5 saja yang dipakai untuk menentukan keluaran. Apabila fungsi yang dipakai bukan fungsi *sigmoid biner*, maka langkah 4 dan 5 harus disesuaikan, begitu juga turunannya pada langkah 6 dan 7.

Pada perakteknya, perancangan arsitektur *Artificial Neural Network Backpropagation* sangat tergantung pada masalah yang akan diselesaikan. Himpunan masukan berdimensi besar

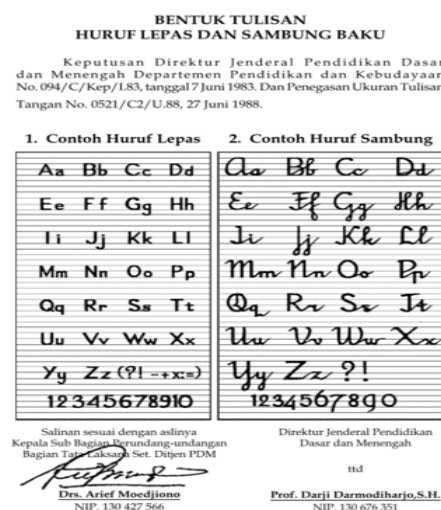
atau jumlah kelas keluaran yang diinginkan besar, maka diperlukan jumlah *node* pada lapisan tersembunyi yang lebih banyak, atau diperlukan lebih dari satu lapisan tersembunyi, tetapi tentu saja ada batas optimumnya untuk kedua parameter tersebut (Siang, 2009).

2.10 Tulisan Tegak Bersambung

Menulis tegak bersambung adalah menulis dengan menyambungkan huruf-huruf sesuai dengan aturan-aturan yang berlaku. Menulis tegak bersambung merupakan salah satu keterampilan yang harus dikuasai siswasejak tingkat dasar. Dalam menulis sambung, siswa tidak hanyamenyambungkan huruf demi huruf melainkan juga harus memperhatikan hal-halsebagai berikut, yaitu (Ridwanuddin, 2015) :

- Ketepatan bentuk setiap huruf harus benar.
- Ketepatan ukuran setiap huruf (ke atas dan ke bawah garis) harus tepat.
- Ketepatan huruf harus tegak lurus.

Secara lebih lengkap, menulis tegak bersambung adalah proses membentuk huruf yang saling berkait satu sama lain membentuk sebuah kata, pada bidang datar dengan menggunakan alat tulis (Erdogan, 2012). Berikut ini merupakan bentuk tulisan tangan yang baku berdasarkan keputusan Direktur Jendral Pendidikan Dasar dan Menengah Departemen Pendidikan dan Kebudayaan No.094/C/Kep/I.83 tanggal 7 Juni 1983 dan Penegasan Ukuran Tulisan Tangan No. 052/C2/U.88 tanggal 27 Juni 1988.



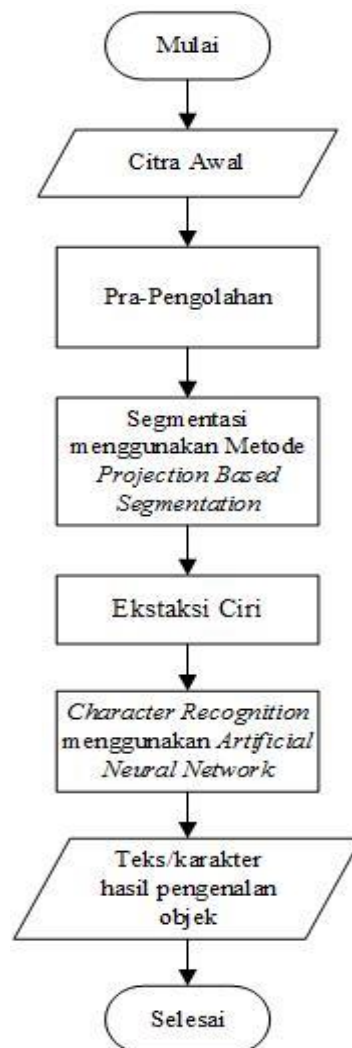
Gambar 2.10 Bentuk Tulisan Tangan yang Baku

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Proses

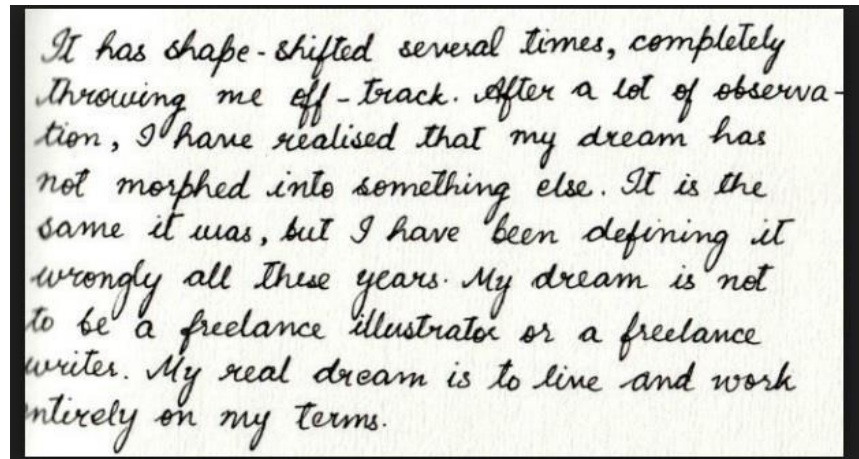
Secara garis besar, sistem pengenalan pola yang dikembangkan dalam penelitian ini memiliki skema yang dapat dimodelkan dengan *flowchart* dibawah ini :



Gambar 3.1 Skema Umum Pengenalan Pola Tulisan Tangan Tegak Bersambung

3.1.1 Citra Awal

Sebelum melakukan tahapan pengenalan pola tulisan tangan tegak bersambung, terlebih dahulu user memasukan citra awal. Adapun citra awal di dalam analisis proses ini diasumsikan seperti gambar dibawah ini :



Gambar 3.2 Citra Awal Tulisan Tangan Tegak Bersambung

Citra tulisan tangan tegak bersambung tersebut akan di representasikan ke dalam matriks berukuran 5×5 piksel dengan nilai RGB (*Red, Green, Blue*) sebagai berikut:

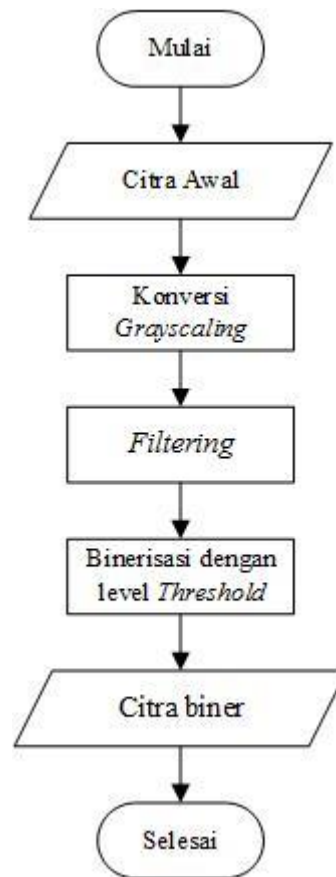
Tabel 3.1 Nilai Piksel Pada Citra Tulisan Tangan Tegak Bersambung

R = 119	R = 129	R = 134	R = 124	R = 114
G = 119	G = 129	G = 134	G = 124	G = 115
B = 119	B = 129	B = 133	B = 122	B = 111
R = 107	R = 121	R = 129	R = 121	R = 110
G = 107	G = 121	G = 130	G = 121	G = 110
B = 107	B = 121	B = 128	B = 119	B = 107
R = 124	R = 134	R = 143	R = 135	R = 126
G = 124	G = 135	G = 143	G = 136	G = 127
B = 123	B = 134	B = 141	B = 133	B = 123
R = 147	R = 152	R = 156	R = 152	R = 148
G = 148	G = 152	G = 156	G = 152	G = 148
B = 146	B = 151	B = 154	B = 149	B = 145
R = 168	R = 165	R = 163	R = 164	R = 170
G = 169	G = 166	G = 164	G = 165	G = 170
B = 167	B = 164	B = 161	B = 162	B = 166

3.1.2 Analisis Proses Pra-Pengolahan

Dibawah ini, akan ditampilkan mengenai langkah - langkah melakukan proses pra-pengolahan sebelum melakukan proses segmentasi citra tulisan tangan tegak bersambung

dengan metode *Projection Based Segmentation* yang akan dijelaskan dalam bentuk *flowchart* pada gambar dibawah ini:



Gambar 3.3 *Flowchart* Pra-Pengolahan

Flowchart untuk tahapan pra-pengolahan terbagi atas beberapa tahapan yaitu tahap pertama adalah *grayscale* yang berguna untuk merubah citra berwarna menjadi citra keabuan. Tahap kedua yakni *filtering*, proses ini memodifikasi citra dengan meningkatkan kualitas citra dan menghilangkan derau yang ada. Tahap selanjutnya binerisasi dengan *threshold*, proses mengubah citra keabuan menjadi citra biner dapat dijelaskan sebagai berikut :

1. Citra Awal

Pada tahapan ini, akan dilakukan tahapan memasukan citra awal. Dimana tahapan ini hanya memasukan citra awal seperti pada gambar 3.2.

2. Tahapan Konversi Citra Berwarna Menjadi Citra Grayscale

Pada tahap ini, akan dilakukan tahapan konversi citra berwarna menjadi citra *grayscale*. Dimana tahapan ini bertujuan untuk mengubah setiap piksel RGB (*red, green, blue*) masing-masing menjadi skala keabuan yang bernilai 0 sebagai warna hitam sampai 255 sebagai warna putih dimana nilai diantaranya merepresentasikan warna keabuan. Adapun

penjelasan selengkapnya dapat dilihat pada proses dibawah ini. Meratakan nilai warna red, green, dan blue, menjadi nilai warna keabuan berdasarkan rumus (3) dan membulatkan nilai desimal :

$$S = 0,2989 * 119 + 0,5870 * 119 + 0,1140 * 119 = 119$$

$$S = 0,2989 * 129 + 0,5870 * 129 + 0,1140 * 129 = 129$$

$$S = 0,2989 * 134 + 0,5870 * 134 + 0,1140 * 133 = 134$$

$$S = 0,2989 * 124 + 0,5870 * 124 + 0,1140 * 122 = 124$$

$$S = 0,2989 * 114 + 0,5870 * 115 + 0,1140 * 111 = 114$$

$$S = 0,2989 * 107 + 0,5870 * 107 + 0,1140 * 107 = 107$$

$$S = 0,2989 * 121 + 0,5870 * 121 + 0,1140 * 121 = 121$$

$$S = 0,2989 * 129 + 0,5870 * 130 + 0,1140 * 128 = 129$$

$$S = 0,2989 * 121 + 0,5870 * 121 + 0,1140 * 119 = 121$$

$$S = 0,2989 * 110 + 0,5870 * 110 + 0,1140 * 107 = 110$$

$$S = 0,2989 * 124 + 0,5870 * 124 + 0,1140 * 123 = 124$$

$$S = 0,2989 * 134 + 0,5870 * 135 + 0,1140 * 134 = 135$$

$$S = 0,2989 * 143 + 0,5870 * 143 + 0,1140 * 141 = 143$$

$$S = 0,2989 * 135 + 0,5870 * 136 + 0,1140 * 133 = 135$$

$$S = 0,2989 * 126 + 0,5870 * 127 + 0,1140 * 123 = 126$$

$$S = 0,2989 * 147 + 0,5870 * 148 + 0,1140 * 146 = 147$$

$$S = 0,2989 * 152 + 0,5870 * 152 + 0,1140 * 151 = 152$$

$$S = 0,2989 * 156 + 0,5870 * 156 + 0,1140 * 154 = 156$$

$$S = 0,2989 * 152 + 0,5870 * 152 + 0,1140 * 149 = 152$$

$$S = 0,2989 * 148 + 0,5870 * 148 + 0,1140 * 145 = 148$$

$$S = 0,2989 * 168 + 0,5870 * 169 + 0,1140 * 167 = 168$$

$$S = 0,2989 * 165 + 0,5870 * 166 + 0,1140 * 164 = 165$$

$$S = 0,2989 * 163 + 0,5870 * 164 + 0,1140 * 161 = 163$$

$$S = 0,2989 * 164 + 0,5870 * 165 + 0,1140 * 162 = 164$$

$$S = 0,2989 * 170 + 0,5870 * 170 + 0,1140 * 166 = 170$$

Perhitungan tersebut dilakukan untuk semua piksel, sehingga matriks citra hasil konversi menjadi seperti berikut :

Tabel 3.2 Nilai Piksel Citra *Grayscale*

119	129	134	124	114
107	121	129	121	110
124	135	143	135	126
142	152	156	152	148
168	165	163	164	170

3. Tahapan *Filtering*

Tahapan *filtering* digunakan untuk menghilangkan noise yang bersifat sebaran normal, yang banyak di jumpai pada sebaran citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. *Filter Gaussian* dapat digunakan untuk tahapan *filtering* ini karena dapat mengaburkan gambar atau mengurangi noise pada citra. Sebagai contoh, pada Tabel 3.2 nilai piksel citra biner akan dilakukan penentuan lokasi entri pada kernel *filter* seperti pada tabel dibawah ini :

Tabel 3.3 Penentuan Lokasi Entri Pada Kernel *Filter*

-2	119	129	134	124	114
-1	107	121	129	121	110
0	124	135	143	135	126
1	142	152	156	152	148
2	168	165	163	164	170
	-2	-1	0	1	2

Filter Gaussian paling tidak berukuran 5×5. Sebagai contoh, bobot-bobotnya dapat diperoleh dengan membuat σ^2 bernilai 1. Adapun penjelasan selengkapnya dapat dilihat pada proses dibawah ini dengan menggunakan rumus (4):

$$G(0,0) = e^{-\frac{(0-0)^2+(0-0)^2}{2 \times 1^2}} = e^{-0} = 1$$

$$G(1,0) = G(0,1) = G(-1,0) = G(0,-1) = e^{-\frac{1}{2}} = 0,6065$$

$$G(1,1) = G(1,-1) = G(-1,1) = G(-1,-1) = e^{-1} = 0,3679$$

$$G(2,0) = G(0,2) = G(-2,0) = G(0,-2) = e^{-2} = 0,1353$$

$$G(2,1) = G(1,2) = G(-2,1) = G(-2,-1) = e^{-\frac{5}{2}} = 0,0821$$

$$G(2,2) = G(-2,-2) = G(-2,2) = G(2,-2) = e^{-4} = 0,0183$$

Dengan mengatur nilai terkecil menjadi 1, maka setiap nilai diatas perlu dikalikan dengan 55 (diperoleh dari $1/0,0183$ dan kemudian hasilnya dibulatkan keatas). Dengan demikian, diperoleh hasil seperti pada tabel dibawah ini :

Tabel 3.4 Nilai Piksel Citra *Filter Gaussian*

1	5	7	5	1
5	20	33	20	5
7	33	55	33	7
5	20	33	20	5
1	5	7	5	1

4. Tahapan Binerisasi Dengan Level *Threshold*

Proses selanjutnya adalah binerisasi citra. Proses binerisasi citra digunakan untuk mengubah citra grayscale menjadi citra biner, artinya mengubah warna tiap-tiap piksel pada citra bernilai 0 dan 255 ke dalam piksel bernilai 0 dan 1. Sehingga citra hanya berwarna hitam dan putih. Pada proses binerisasi, menggunakan nilai ambang untuk menentukan nilai grayscale tertentu yang diubah menjadi piksel bernilai 0 atau 1. Dimana nilai piksel pada Tabel 3.4 akan direpresentasikan kedalam citra biner dengan menggunakan rumus (5) menghasilkan nilai thresholding yang digunakan pada proses ini adalah 28. Kemudian pengambangan *threshold* dilakukan dengan menggunakan persamaan(4) dan mendapatkan hasil sebagai berikut:

- i. Jika piksel bernilai < 28 maka piksel = 0
- ii. Jika piksel bernilai ≥ 28 maka piksel = 1

Dengan demikian maka semua piksel tersebut akan diubah menjadi piksel citra biner pada tabel dibawah ini :

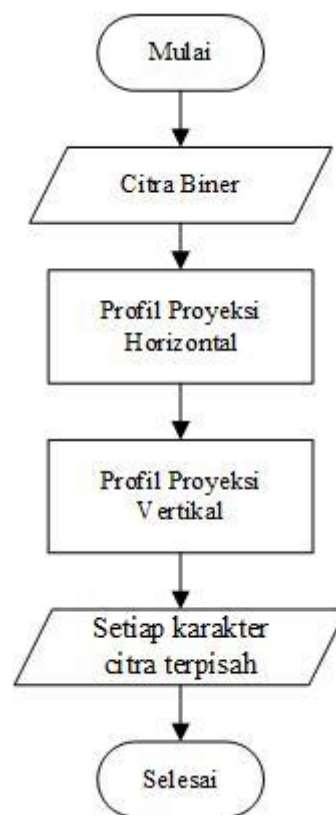
Tabel 3.5 Nilai Piksel Citra Biner

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

0	0	1	0	0
0	0	0	0	0

3.1.3 Analisa Proses Segmentasi dengan Metode *Projection Based Segmentation*

Dibawah ini, akan ditampilkan mengenai langkah – langkah melakukan proses segmentasi karakter citra tulisan tangan tegak bersambung dengan menggunakan metode *Projection Based Segmentation* yang akan dijelaskan dalam bentuk *flowchart* pada gambar dibawah ini:

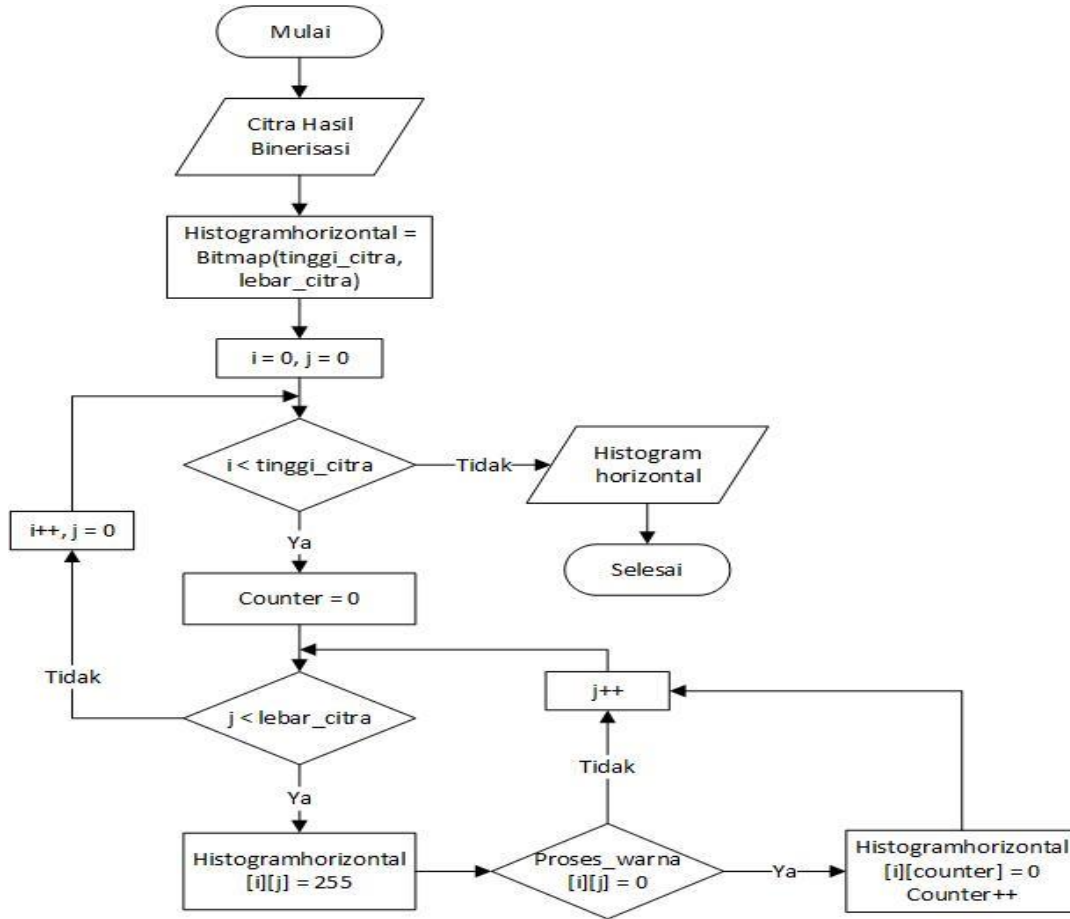


Gambar 3.4 *Flowchart Metode Projection Based Segmentation*

Flowchart tahapan metode *Projection Based Segmentation* dilakukan setelah proses pra-pengolahan, *user* dapat memilih secara manual batas objek yang ingin dikenali. Batasan objek kemudian di segmentasi dengan tujuan agar karakter pada citra dapat dikenali satu persatu. Profil proyeksi dianalisis sebagai struktur data, yang digunakan untuk menyimpan jumlah piksel non-latar belakang, ketika gambar diproyeksikan di atas sumbu X-Y normal. Metode profil proyeksi diklasifikasikan sebagai proyeksi horizontal dan proyeksi vertikal dapat dijelaskan sebagai berikut :

1. Tahapan Profil Proyeksi Horizontal

Tahapan profil proyeksi horizontal digunakan untuk melakukan segmentasi garis. Adapun penjelasan selengkapnya dapat dilihat pada gambar dibawah ini:



Gambar 3.5 Flowchart Tahapan Profil Proyeksi Horizontal

Tahapan profil proyeksi horizontal didapatkan dengan menjumlahkan nilai piksel sepanjang garis horizontal (sumbu x) untuk setiap baris (sumbu y). Dimana nilai piksel pada Tabel 3.5 akan direpresentasikan untuk melakukan proses segmentasi baris dengan profil proyeksi horizontal dapat dilihat pada proses dibawah ini dengan menggunakan rumus (1):

$$P_{hor}(b1) = \sum_{j=1}^N I[b, k_j] = 0 + 0 + 0 + 0 + 0 = 0$$

b1	0	0	0	0	0
----	---	---	---	---	---

$$P_{hor}(b2) = \sum_{j=1}^N I[b, k_j] = 0 + 0 + 1 + 0 + 0 = 1$$

b2	0	0	1	0	0
----	---	---	---	---	---

$$P_{hor}(b3) = \sum_{j=1}^N I[b, k_j] = 0 + 1 + 1 + 1 + 0 = 3$$

b3	0	1	1	1	0
----	---	---	---	---	---

$$P_{hor}(b4) = \sum_{j=1}^N I[b, k_j] = 0 + 0 + 1 + 0 + 0 = 1$$

b4	0	0	1	0	0
----	---	---	---	---	---

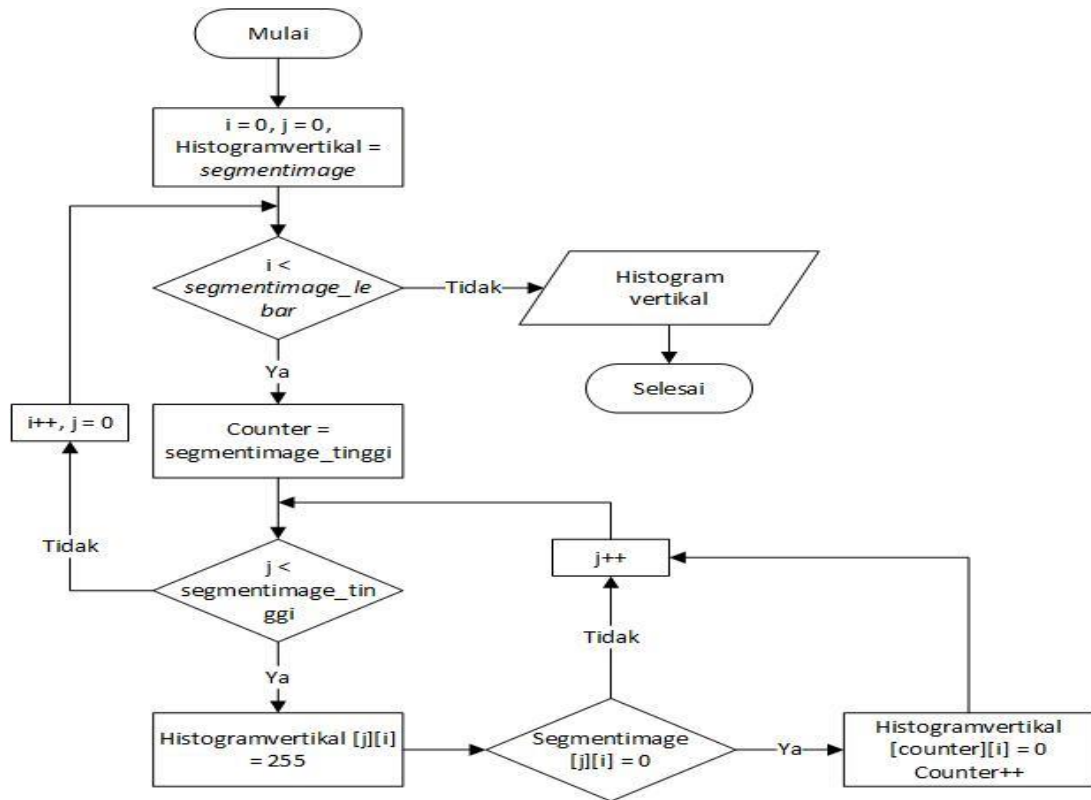
$$P_{hor}(b5) = \sum_{j=1}^N I[b, k_j] = 0 + 0 + 0 + 0 + 0 = 0$$

b5	0	0	0	0	0
----	---	---	---	---	---

Setelah berhasil dihitung dari hasil $P_{hor}(b)$ yaitu jumlah piksel pada baris b citra akan diidentifikasi titik atas dari sebuah baris, dan titik bawah dari sebuah baris. Proses ini disebut Proses horizontal histogram, output dari proses ini adalah jumlah baris yang berhasil dideteksi.

2. Tahapan Profil Proyeksi Vertikal

Tahapan profil proyeksi vertikal digunakan untuk mensegmentasi karakter dari tiap-tiap baris. Adapun penjelasan selengkapnya dapat dilihat pada gambar dibawah ini:



Gambar 3.6 Flowchart Tahapan Profil Proyeksi Vertikal

Setelah baris tersebut dideteksi, maka akan dihitung nilai $P_{ver}(k)$ dengan menggunakan rumus(2). Penghitungan nilai $P_{ver}(k)$ dihitung untuk tiap-tiap baris. Kemudian, dari $P_{ver}(k)$ baris akan dilakukan segmentasi dengan melakukan perhitungan titik kiri dan titik kanan untuk tiap-tiap tulisan. Dimana nilai piksel pada Tabel 3.5 akan direpresentasikan untuk melakukan proses segmentasi karakter dengan profil proyeksi vertikal dapat dilihat pada proses dibawah ini:

$$P_{ver}(k1) = \sum_{j=1}^M I[b_k, k] = 0 + 0 + 0 + 0 + 0 = 0$$

k1
0
0
0
0

0

$$P_{ver}(k2) = \sum_{j=1}^M I[b_k, k] = 0 + 0 + 1 + 0 + 0 = 1$$

k2
0
0
1
0
0

$$P_{ver}(k3) = \sum_{j=1}^M I[b_k, k] = 0 + 1 + 1 + 1 + 0 = 3$$

k3
0
1
1
1
0

$$P_{ver}(k4) = \sum_{j=1}^M I[b_k, k] = 0 + 0 + 1 + 0 + 0 = 1$$

k4
0
0
1
0

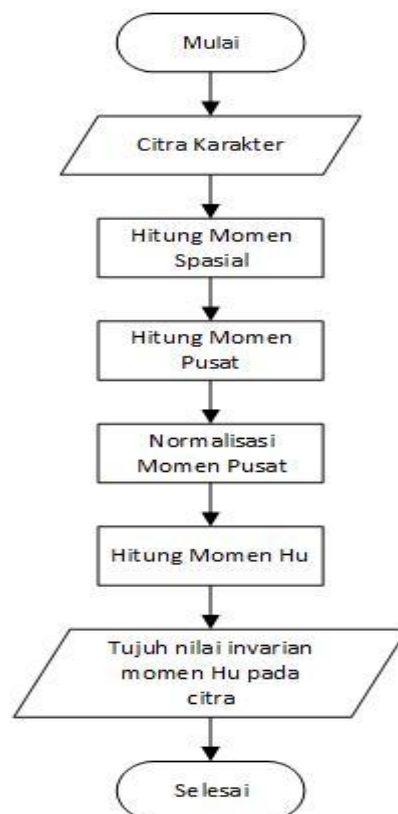
0

$$P_{ver}(k5) = \sum_{j=1}^M I[b_k, k] = 0 + 0 + 0 + 0 + 0 = 0$$

k5
0
0
0
0
0

3.1.4 Analisis Proses Ekstraksi Ciri

Setelah dilakukan segmentasi karakter terhadap citra maka tahapan selanjutnya dihitung ketujuh nilai momentnya guna mendeskripsikan ciri-ciri objek citra. Tahapan proses ini akan dijelaskan dalam bentuk *flowchart* pada gambar dibawah ini:



Gambar 3.7 *Flowchart* Tahapan Perhitungan *Hu Invariant Moments*

Langkah pertama objek citra dihitung momen (m_{00} , m_{01} , m_{10}) pada sistem diskrit, menggunakan persamaan rumus(7). Dimana H dan W masing-masing merupakan tinggi dan lebar citra dan $p = 0,1,2,\dots$ dan $q = 0,1,2,\dots$ adalah integer.

Tabel 3.6 Nilai Piksel m_{00}

$\begin{matrix} x \\ y \end{matrix}$	1	2	3	4
1	$1^0 * 1^0 * 0 = 0$	$2^0 * 1^0 * 0 = 0$	$3^0 * 1^0 * 0 = 0$	$4^0 * 1^0 * 0 = 0$
2	$1^0 * 2^0 * 0 = 0$	$2^0 * 2^0 * 0 = 0$	$3^0 * 2^0 * 1 = 1$	$4^0 * 2^0 * 0 = 0$
3	$1^0 * 3^0 * 0 = 0$	$2^0 * 3^0 * 1 = 1$	$3^0 * 3^0 * 1 = 1$	$4^0 * 3^0 * 1 = 1$
4	$1^0 * 4^0 * 0 = 0$	$2^0 * 4^0 * 0 = 0$	$3^0 * 4^0 * 1 = 1$	$4^0 * 4^0 * 0 = 0$
Hasil	0	1	3	1

Tabel 3.7 Nilai Piksel m_{10}

$\begin{matrix} x \\ y \end{matrix}$	1	2	3	4
1	$1^1 * 1^0 * 0 = 0$	$2^1 * 1^0 * 0 = 0$	$3^1 * 1^0 * 0 = 0$	$4^1 * 1^0 * 0 = 0$
2	$1^1 * 2^0 * 0 = 0$	$2^1 * 2^0 * 0 = 0$	$3^1 * 2^0 * 1 = 3$	$4^1 * 2^0 * 0 = 0$
3	$1^1 * 3^0 * 0 = 0$	$2^1 * 3^0 * 1 = 2$	$3^1 * 3^0 * 1 = 3$	$4^1 * 3^0 * 1 = 4$
4	$1^1 * 4^0 * 0 = 0$	$2^1 * 4^0 * 0 = 0$	$3^1 * 4^0 * 1 = 3$	$4^1 * 4^0 * 0 = 0$
Hasil	0	2	9	4

Tabel 3.8 Nilai Piksel m_{01}

$\begin{matrix} x \\ y \end{matrix}$	1	2	3	4
1	$1^0 * 1^1 * 0 = 0$	$2^0 * 1^1 * 0 = 0$	$3^0 * 1^1 * 0 = 0$	$4^0 * 1^1 * 0 = 0$
2	$1^0 * 2^1 * 0 = 0$	$2^0 * 2^1 * 0 = 0$	$3^0 * 2^1 * 1 = 2$	$4^0 * 2^1 * 0 = 0$
3	$1^0 * 3^1 * 0 = 0$	$2^0 * 3^1 * 1 = 3$	$3^0 * 3^1 * 1 = 3$	$4^0 * 3^1 * 1 = 3$
4	$1^0 * 4^1 * 0 = 0$	$2^0 * 4^1 * 0 = 0$	$3^0 * 4^1 * 1 = 4$	$4^0 * 4^1 * 0 = 0$
Hasil	0	3	9	3

$$m_{00} = 5$$

$$m_{10} = 15$$

$$m_{01} = 15$$

Selanjutnya, hitung momen pusat untuk suatu citra dengan persamaan rumus (9) sebagai berikut:

Dimana :

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ dan } \bar{y} = \frac{m_{01}}{m_{00}}$$

Maka :

$$\bar{x} = \frac{15}{5} = 3$$

$$\bar{y} = \frac{15}{5} = 3$$

Sehingga :

μ_{11}	$(1-3)^1(1-3)^1 = 4$	$(2-3)^1(1-3)^1 = 2$	$(3-3)^1(1-3)^1 = 0$	$(4-3)^1(1-3)^1 = -2$
	$(1-3)^1(2-3)^1 = 2$	$(2-3)^1(2-3)^1 = 1$	$(3-3)^1(2-3)^1 = 0$	$(4-3)^1(2-3)^1 = -1$
	$(1-3)^1(3-3)^1 = 0$	$(2-3)^1(3-3)^1 = 0$	$(3-3)^1(3-3)^1 = 0$	$(4-3)^1(3-3)^1 = 0$
	$(1-3)^1(4-3)^1 = -2$	$(2-3)^1(4-3)^1 = -1$	$(3-3)^1(4-3)^1 = 0$	$(4-3)^1(4-3)^1 = 1$
Hasil	4	2	0	-2

μ_{20}	$(1-3)^2(1-3)^0 = 4$	$(2-3)^2(1-3)^0 = 1$	$(3-3)^2(1-3)^0 = 0$	$(4-3)^2(1-3)^0 = 1$
	$(1-3)^2(2-3)^0 = 4$	$(2-3)^2(2-3)^0 = 1$	$(3-3)^2(2-3)^0 = 0$	$(4-3)^2(2-3)^0 = 1$
	$(1-3)^2(3-3)^0 = 4$	$(2-3)^2(3-3)^0 = 1$	$(3-3)^2(3-3)^0 = 0$	$(4-3)^2(3-3)^0 = 1$
	$(1-3)^2(4-3)^0 = 4$	$(2-3)^2(4-3)^0 = 1$	$(3-3)^2(4-3)^0 = 0$	$(4-3)^2(4-3)^0 = 1$
Hasil	16	4	0	4

μ_{02}	$(1-3)^0(1-3)^2 = 4$	$(2-3)^0(1-3)^2 = 4$	$(3-3)^0(1-3)^2 = 4$	$(4-3)^0(1-3)^2 = 4$
	$(1-3)^0(2-3)^2 = 1$	$(2-3)^0(2-3)^2 = 1$	$(3-3)^0(2-3)^2 = 1$	$(4-3)^0(2-3)^2 = 1$
	$(1-3)^0(3-3)^2 = 0$	$(2-3)^0(3-3)^2 = 0$	$(3-3)^0(3-3)^2 = 0$	$(4-3)^0(3-3)^2 = 0$
	$(1-3)^0(4-3)^2 = 1$	$(2-3)^0(4-3)^2 = 1$	$(3-3)^0(4-3)^2 = 1$	$(4-3)^0(4-3)^2 = 1$
Hasil	6	6	6	6

μ_{30}	$(1-3)^3(1-3)^0 = -8$	$(2-3)^3(1-3)^0 = -1$	$(3-3)^3(1-3)^0 = 0$	$(4-3)^3(1-3)^0 = 1$
	$(1-3)^3(2-3)^0 = -8$	$(2-3)^3(2-3)^0 = -1$	$(3-3)^3(2-3)^0 = 0$	$(4-3)^3(2-3)^0 = 1$
	$(1-3)^3(3-3)^0 = -8$	$(2-3)^3(3-3)^0 = -1$	$(3-3)^3(3-3)^0 = 0$	$(4-3)^3(3-3)^0 = 1$
	$(1-3)^3(4-3)^0 = -8$	$(2-3)^3(4-3)^0 = -1$	$(3-3)^3(4-3)^0 = 0$	$(4-3)^3(4-3)^0 = 1$
Hasil	-32	-4	0	4

μ_{03}	$(1-3)^0(1-3)^3 = -8$	$(2-3)^0(1-3)^3 = -8$	$(3-3)^0(1-3)^3 = -8$	$(4-3)^0(1-3)^3 = -8$
	$(1-3)^0(2-3)^3 = -1$	$(2-3)^0(2-3)^3 = -1$	$(3-3)^0(2-3)^3 = -1$	$(4-3)^0(2-3)^3 = -1$
	$(1-3)^0(3-3)^3 = 0$	$(2-3)^0(3-3)^3 = 0$	$(3-3)^0(3-3)^3 = 0$	$(4-3)^0(3-3)^3 = 0$
	$(1-3)^0(4-3)^3 = 1$	$(2-3)^0(4-3)^3 = 1$	$(3-3)^0(4-3)^3 = 1$	$(4-3)^0(4-3)^3 = 1$
Hasil	-8	-8	-8	-8

μ_{12}	$(1-3)^1(1-3)^2 = -8$	$(2-3)^1(1-3)^2 = -4$	$(3-3)^1(1-3)^2 = 0$	$(4-3)^1(1-3)^2 = 4$
------------	-----------------------	-----------------------	----------------------	----------------------

	$(1-3)^1(2-3)^2 = -2$	$(2-3)^1(2-3)^2 = -1$	$(3-3)^1(2-3)^2 = 0$	$(4-3)^1(2-3)^2 = 1$
	$(1-3)^1(3-3)^2 = 0$	$(2-3)^1(3-3)^2 = 0$	$(3-3)^1(3-3)^2 = 0$	$(4-3)^1(3-3)^2 = 0$
	$(1-3)^1(4-3)^2 = -2$	$(2-3)^1(4-3)^2 = -1$	$(3-3)^1(4-3)^2 = 0$	$(4-3)^1(4-3)^2 = 1$
Hasil	-12	-6	0	6

μ_{21}	$(1-3)^2(1-3)^1 = -8$	$(2-3)^2(1-3)^1 = 2$	$(3-3)^2(1-3)^1 = 0$	$(4-3)^2(1-3)^1 = -2$
	$(1-3)^2(2-3)^1 = -4$	$(2-3)^2(2-3)^1 = -1$	$(3-3)^2(2-3)^1 = 0$	$(4-3)^2(2-3)^1 = -1$
	$(1-3)^2(3-3)^1 = 0$	$(2-3)^2(3-3)^1 = 0$	$(3-3)^2(3-3)^1 = 0$	$(4-3)^2(3-3)^1 = 0$
	$(1-3)^2(4-3)^1 = 4$	$(2-3)^2(4-3)^1 = 1$	$(3-3)^2(4-3)^1 = 0$	$(4-3)^2(4-3)^1 = 1$
Hasil	-8	-2	0	2

Jadi hasil dari masing-masing momen pusat sebagai berikut:

$$\mu_{11} = 4 + 2 - 2 = 4$$

$$\mu_{20} = 16 + 4 + 4 = 24$$

$$\mu_{02} = 6 + 6 + 6 + 6 = 24$$

$$\mu_{30} = -32 + (-4) + 4 = -32$$

$$\mu_{03} = -8 + (-8) + (-8) + (-8) = -32$$

$$\mu_{12} = -12 + (-6) + 6 = -12$$

$$\mu_{21} = -8 + (-2) + 2 = -12$$

Kemudian normalisasi pusat momen sebagai berikut:

$$\eta_{i,j} = \frac{\mu_{pq}}{\mu_{00}^\gamma}$$

Dimana faktor normalisasi $\gamma = \left(\frac{p+q}{2}\right) + 1$.

$$\eta_{11} = \frac{4}{5^2} = 0,16$$

$$\eta_{20} = \frac{24}{5^2} = 0,96$$

$$\eta_{02} = \frac{24}{5^2} = 0,96$$

$$\eta_{30} = \frac{-32}{5^{2,5}} = -0,5724$$

$$\eta_{03} = \frac{-32}{5^{2,5}} = -0,5724$$

$$\eta_{12} = \frac{-12}{5^{2,5}} = -0,2147$$

$$\eta_{21} = \frac{-12}{5^{2,5}} = -0,2147$$

Sehingga ketujuh nilai invariant moments dapat diturunkan dari moment kedua dan ketiga berdasarkan rumus(11) hingga (17) sebagai berikut.

$$\phi_1 = 0,96 + 0,96 = 1,92$$

$$\phi_2 = (0,96 + 0,96)^2 + (4 \times 0,16)^2 = 0,0026$$

$$\phi_3 = ((-0,5724) - 3(-0,2147))^2 + (3(-0,2147) + 0,5724)^2 = 1,4848$$

$$\phi_4 = ((-0,5724) + (-0,2147))^2 + (-0,2147 + (-0,5724))^2 = 1,2390$$

$$\begin{aligned} \phi_5 &= ((-0,5724) - 3(-0,2147))((-0,5724) + (-0,2147))\{((-0,5724) + (-0,2147))^2 \\ &\quad - 3(-0,2147 + (-0,5724))^2\} \\ &\quad + (3(-0,2147) + 0,5724)(-0,2147 \\ &\quad + (-0,5724))\{3(-0,2147 + 0,5724)^2 - (-0,2147 + (-0,5724))^2\} \\ &= 0,2238 \end{aligned}$$

$$\begin{aligned} \phi_6 &= (0,96 + 0,96)\{(-0,5724 + (-0,2147))^2 - (-0,2147 + (-0,5724))^2\} \\ &\quad + 4(0,16)(-0,5724 + (-0,2147))(-0,2147 + (-0,5724)) = 0,1802 \end{aligned}$$

$$\begin{aligned} \phi_7 &= (3(-0,2147) + 0,5724)(-0,5724 + (-0,2147))\{(-0,5724 + (-0,2147))^2 \\ &\quad - 3(-0,2147 + (-0,5724))^2\} \\ &\quad + (3(-0,2147) + 0,5724)(3\eta_{21} + (-0,5724))\{3(-0,5724 + (-0,2147))^2 \\ &\quad - 3(-0,2147 + (-0,5724))^2\} = 0,6978 \end{aligned}$$

Sehingga didapat nilai invariant moment hu pada Tabel 3.9 sebagai berikut:

Tabel 3.9 Nilai Piksel *Invarian Moment Hu*

	Invarian Moment Hu
ϕ_1	1,92
ϕ_2	0,0026
ϕ_3	1,4848
ϕ_4	1,2390
ϕ_5	0,2238
ϕ_6	0,1802
ϕ_7	0,6978

3.1.5 Normalisasi Data Input

Nilai-nilai yang didapatkan dari Invarian Moment hu belum memenuhi persyaratan data masukan neural network, oleh sebab itu setiap nilai ciri objek perlu dinormalisasi. Normalisasi nilai ciri objek menggunakan persamaan :

$$\gamma_i = \frac{X_i - \bar{X}}{\sigma}$$

Dimana :

X_i = data ke i

\bar{X} = rata-rata data

σ = standar deviasi data

Berdasarkan data yang telah didapat dan dapat dilihat pada Tabel 3.9. Normalisasi dapat dilakukan sebagai berikut :

$$\gamma_1 = \frac{1,92 - 0,82117}{0,73885} = 1,48722$$

$$\gamma_2 = \frac{0,0026 - 0,82117}{0,73885} = -1,10790$$

$$\gamma_3 = \frac{1,4848 - 0,82117}{0,73885} = 0,89819$$

$$\gamma_4 = \frac{1,2390 - 0,82117}{0,73885} = 0,56551$$

$$\gamma_5 = \frac{0,2238 - 0,82117}{0,73885} = -0,80852$$

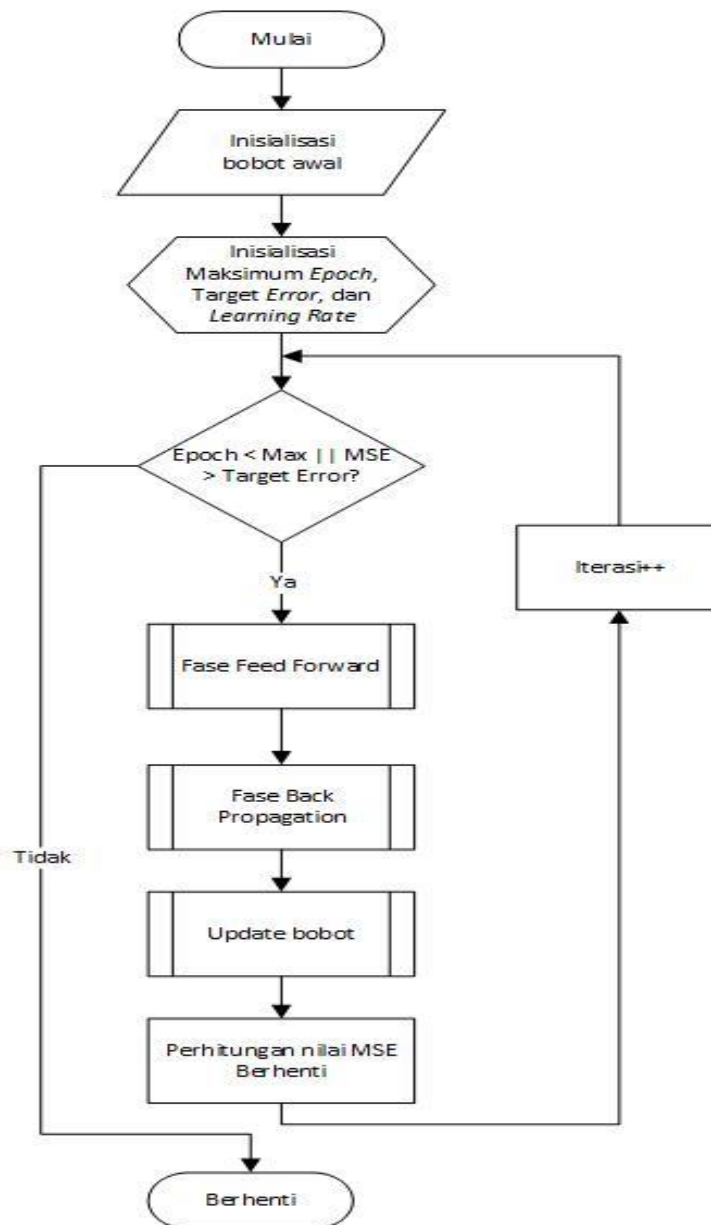
$$\gamma_6 = \frac{0,1802 - 0,82117}{0,73885} = -0,86753$$

$$\gamma_7 = \frac{0,6978 - 0,82117}{0,73885} = -0,16698$$

Setelah semua nilai dinormalisasi, semua nilai telah memenuhi syarat sebagai masukan dalam *Artificial Neuran Network*.

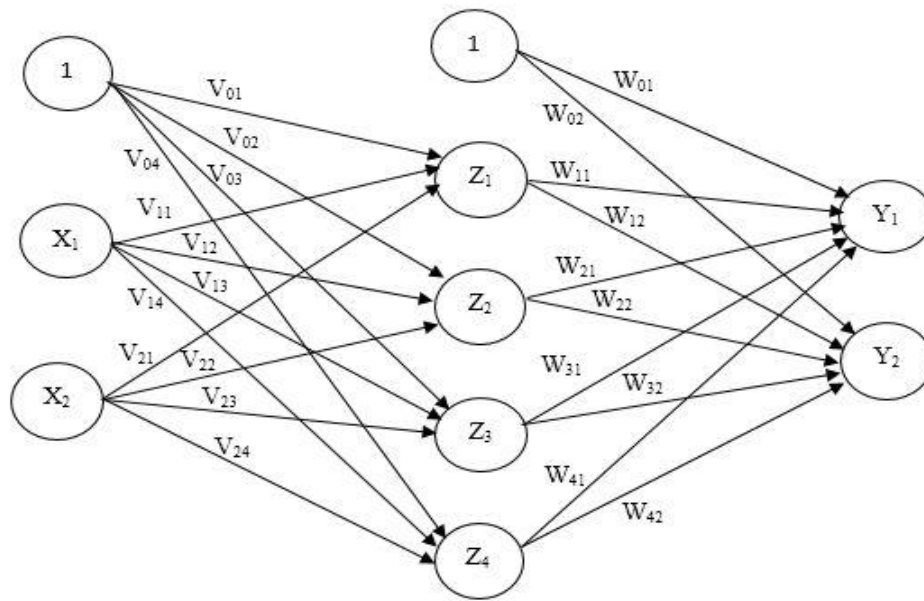
3.1.6 Analisis Proses Pengenalan Pola dengan Metode *Artificial Neural Network*

Pada tahapan ini hasil dari ekstraksi ciri citra tulisan tangan tegak bersambung akan dilakukan proses pelatihan dengan menggunakan jaringan syaraf tiruan *backpropagation* pada metode *Artificial Neural Network*. Tahapan algoritma pelatihan *Backpropagation* digunakan untuk proses koreksi terhadap nilai *output* yang dihasilkan oleh tahap sebelumnya. Setiap nilai *output* akan dibandingkan dengan nilai target, dan dicari besarnya *error* untuk tiap nilai tersebut. Adapun penjelasan selengkapnya dapat dilihat pada gambar dibawah ini:



Gambar 3.8 *Flowchart Tahapan Algoritma Backpropagation*

Flowchart untuk pelatihan tahapan algoritma *backpropagation* terdiri dari 3 tahapan yaitu tahapan perambatan maju (*forward propagation*), tahap perambatan mundur (*backward propagation*), tahap pengupdatetan bobot dan bias. Secara rinci algoritma pelatihan *backpropagation* dapat diuraikan sebagai berikut:



Gambar 3.9 Arsitektur Jaringan Saraf *Backpropagation*

Langkah 0 : Membuat inisialisasi semua bobot dengan bilangan acak kecil
 Nilai bobot dan bias awal diacak sebagai berikut.

Bobot awal bias *hidden layer*

$V_{01}=0,2$ $V_{02}=0,3$ $V_{03}=0,4$ $V_{04}=0,5$

Bobot awal *input ke hidden layer*

$V_{11}=0,1$ $V_{12}=0,2$ $V_{13}=0,3$ $V_{04}=0,4$

$V_{21}=0,1$ $V_{22}=0,2$ $V_{23}=0,3$ $V_{04}=0,4$

Bobot awal bias ke *output layer*

$W_{01}=0,5$ $W_{02}=0,5$

Bobot awal *hidden layer ke output layer*

$W_{11}=0,1$ $W_{12}=0,2$

$W_{21}=0,4$ $W_{22}=0,5$

$W_{31}=0,1$ $W_{32}=0,2$

$W_{41}=0,4$ $W_{42}=0,5$

Untuk kebutuhan pelatihan jaringan ditentukan pula:

Learning rate (α) = 0,2

Maksimum *Epoch* = 200

Target *Error* = 0,003

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9

Epoch ke-1

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Data ke = 1

Input = “00”

Target = “01”

Fase I : Propagasi Maju

Langkah 3 : Tiap unit masukan $X_i = (i = 1, 2, 3, \dots, n)$ menerima sinyal X_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya.

Input = “00”, maka $X_1 = 0, X_2 = 0$

Langkah 4 : Tiap-tiap unit lapisan tersembunyi $Z_j = (j = 1, 2, 3, \dots, p)$ menjumlahkan sinyal-sinyal input terbobot:

$$Z_{in_j} = V_{oj} + \sum_{i=1}^n X_i V_{ij}$$

Operasi pada hidden layer

$$Z_{in_j} = V_{oj} + \sum_{i=1}^n X_i V_{ij}$$

$$Z_{in1} = 0,2 + 0(0,1) + 0(0,1) = 0,2$$

$$Z_{in2} = 0,3 + 0(0,2) + 0(0,2) = 0,3$$

$$Z_{in3} = 0,4 + 0(0,3) + 0(0,3) = 0,4$$

$$Z_{in3} = 0,5 + 0(0,4) + 0(0,4) = 0,5$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi:

$$Z_j = f(Z_{in_j}) = \frac{1}{1 + e^{-Z_{in_j}}}$$

$$Z_1 = \frac{1}{1 + e^{-0,2}} = 0,5498$$

$$Z_2 = \frac{1}{1 + e^{-0,3}} = 0,5744$$

$$Z_3 = \frac{1}{1 + e^{-0,4}} = 0,5986$$

$$Z_4 = \frac{1}{1 + e^{-0,5}} = 0,6224$$

Langkah 5 : Hasil fungsi tersebut dikirim ke semua unit pada lapisan di atasnya. Tiap-tiap unit keluaran Y_k ($k = 1, 2, 3, \dots, m$) menjumlahkan sinyal-sinyal input terbobot.

$$Y_{in_k} = W_{ok} + \sum_{i=1}^p Z_i W_{ik}$$

$$Y_{in1} = 0,5 + 0,5498(0,1) + 0,5744(0,4) + 0,5986(0,1) + 0,6224(0,4) \\ = 1,0935$$

$$Y_{in2} = 0,5 + 0,5498(0,2) + 0,5744(0,5) + 0,5986(0,2) + 0,6224(0,5) \\ = 1,3281$$

Kemudian dihitung nilai keluaran dengan menggunakan fungsi aktivasi:

$$Y_k = f(Y_{in_k}) = \frac{1}{1 + e^{-Y_{in_k}}}$$

$$Y_1 = \frac{1}{1 + e^{-1,0935}} = 0,7490$$

$$Y_2 = \frac{1}{1 + e^{-1,3281}} = 0,7905$$

Fase II : Propagasi Mundur (*Backpropagation*)

Langkah 6 : Tiap-tiap unit keluaran Y_k ($k = 1, 2, 3, \dots, m$) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*:

$$\delta_k = (t_k - Y_k)f'(Y_{in_k}) = (t_k - Y_k)y_k(1 - Y_k)$$

Hitung koreksi nilai bobot

$$\Delta W_{jk} = \alpha \delta_k Z_j$$

$$\delta_1 = (t_1 - Y_1)y_1(1 - Y_1) \\ = (0 - 0,7490)(0,7490)(1 - 0,7490) \\ = -0,1408$$

$$\Delta W_{01} = \alpha \delta_k \\ = 0,2(-0,1408) = -0,0281$$

$$\Delta W_{11} = \alpha \delta_1 Z_1 \\ = 0,2(-0,1408)(0,5498) = -0,0154$$

$$\begin{aligned}\Delta W_{21} &= \alpha \delta_1 Z_2 \\ &= 0,2(-0,1408)(0,5744) = -0,0170\end{aligned}$$

$$\begin{aligned}\Delta W_{31} &= \alpha \delta_1 Z_3 \\ &= 0,2(-0,1408)(0,5986) = -0,0168\end{aligned}$$

$$\begin{aligned}\Delta W_{41} &= \alpha \delta_1 Z_4 \\ &= 0,2(-0,1408)(0,6224) = -0,0184\end{aligned}$$

$$\begin{aligned}\delta_2 &= (t_2 - Y_2)y_2(1 - Y_2) \\ &= (1 - 0,7905)(0,7905)(1 - 0,7905) \\ &= 0,0346\end{aligned}$$

$$\begin{aligned}\Delta W_{02} &= \alpha \delta_k \\ &= 0,2(0,0346) = 0,0069\end{aligned}$$

$$\begin{aligned}\Delta W_{12} &= \alpha \delta_2 Z_1 \\ &= 0,2(0,0346)(0,5498) = 0,0038\end{aligned}$$

$$\begin{aligned}\Delta W_{22} &= \alpha \delta_2 Z_2 \\ &= 0,2(0,0346)(0,5744) = 0,0039\end{aligned}$$

$$\begin{aligned}\Delta W_{32} &= \alpha \delta_2 Z_3 \\ &= 0,2(0,0346)(0,5986) = 0,0041\end{aligned}$$

$$\begin{aligned}\Delta W_{42} &= \alpha \delta_2 Z_4 \\ &= 0,2(0,0346)(0,6224) = 0,0043\end{aligned}$$

Langkah 7 : Tiap - tiap unit *hidden* $Z_j = (j = 1,2,3, \dots, p)$ menjumlahkan *delta input* (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_j W_{jk}$$

$$\begin{aligned}\delta_{in_1} &= \delta_1 * w_{11} + \delta_2 * w_{12} \\ &= (-0,1408) * 0,1 + (0,0346) * 0,2 \\ &= -0,1408 + (0,0069) \\ &= -0,1338\end{aligned}$$

$$\delta_{in_2} = \delta_1 * w_{21} + \delta_2 * w_{22}$$

$$\begin{aligned}
&= (-0,1408) * 0,4 + (0,0346) * 0,5 \\
&= -0,0563 + (0,0173) \\
&= -0,039
\end{aligned}$$

$$\begin{aligned}
\delta_{in_3} &= \delta_1 * w_{31} + \delta_2 * w_{32} \\
&= (-0,1408) * 0,1 + (0,0346) * 0,2 \\
&= -0,1408 + (0,0069) \\
&= -0,1338
\end{aligned}$$

$$\begin{aligned}
\delta_{in_4} &= \delta_1 * w_{41} + \delta_2 * w_{42} \\
&= (-0,1408) * 0,4 + (0,0346) * 0,5 \\
&= -0,0563 + (0,0173) \\
&= -0,039
\end{aligned}$$

$$\begin{aligned}
\delta_1 &= \delta_{in_1} \left(\frac{1}{1 + e^{-Z_{in_1}}} \right) \left[1 - \left(\frac{1}{1 + e^{-Z_{in_1}}} \right) \right] \\
&= -0,1338 \left(\frac{1}{1 + e^{-0,2}} \right) \left[1 - \left(\frac{1}{1 + e^{-0,2}} \right) \right] \\
&= -0,1338 * 0,5498 * 0,4502 = -0,0331
\end{aligned}$$

$$\begin{aligned}
\delta_2 &= \delta_{in_2} \left(\frac{1}{1 + e^{-Z_{in_2}}} \right) \left[1 - \left(\frac{1}{1 + e^{-Z_{in_2}}} \right) \right] \\
&= -0,039 \left(\frac{1}{1 + e^{-0,3}} \right) \left[1 - \left(\frac{1}{1 + e^{-0,3}} \right) \right] \\
&= -0,039 * 0,5744 * 0,4256 = -0,0095
\end{aligned}$$

$$\begin{aligned}
\delta_3 &= \delta_{in_3} \left(\frac{1}{1 + e^{-Z_{in_3}}} \right) \left[1 - \left(\frac{1}{1 + e^{-Z_{in_3}}} \right) \right] \\
&= -0,1338 \left(\frac{1}{1 + e^{-0,4}} \right) \left[1 - \left(\frac{1}{1 + e^{-0,4}} \right) \right] \\
&= -0,1338 * 0,5986 * 0,4014 = -0,0321
\end{aligned}$$

$$\begin{aligned}
\delta_4 &= \delta_{in_4} \left(\frac{1}{1 + e^{-Z_{in_4}}} \right) \left[1 - \left(\frac{1}{1 + e^{-Z_{in_4}}} \right) \right] \\
&= -0,039 \left(\frac{1}{1 + e^{-0,5}} \right) \left[1 - \left(\frac{1}{1 + e^{-0,5}} \right) \right] \\
&= -0,039 * 0,6224 * 0,3776 = -0,0091
\end{aligned}$$

$$\Delta v_{01} = \alpha * \delta_1 = 0,2 * (-0,0331) = -0,0066$$

$$\Delta v_{02} = \alpha * \delta_2 = 0,2 * (-0,0095) = -0,0019$$

$$\Delta v_{03} = \alpha * \delta_3 = 0,2 * (-0,0321) = -0,00642$$

$$\Delta v_{04} = \alpha * \delta_4 = 0,2 * (-0,0091) = -0,0018$$

$$\Delta v_{11} = \alpha * \delta_1 * X_1 = 0,2 * (-0,0331) * 0 = 0$$

$$\Delta v_{12} = \alpha * \delta_2 * X_1 = 0,2 * (-0,0095) * 0 = 0$$

$$\Delta v_{13} = \alpha * \delta_3 * X_1 = 0,2 * (-0,0321) * 0 = 0$$

$$\Delta v_{14} = \alpha * \delta_4 * X_1 = 0,2 * (-0,0091) * 0 = 0$$

$$\Delta v_{21} = \alpha * \delta_1 * X_2 = 0,2 * (-0,0331) * 0 = 0$$

$$\Delta v_{22} = \alpha * \delta_2 * X_2 = 0,2 * (-0,0095) * 0 = 0$$

$$\Delta v_{23} = \alpha * \delta_3 * X_2 = 0,2 * (-0,0321) * 0 = 0$$

$$\Delta v_{24} = \alpha * \delta_4 * X_2 = 0,2 * (-0,0091) * 0 = 0$$

Fase III : Perubahan bobot

Langkah 8 : Tiap-tiap unit output $Y_k (k = 1,2,3, \dots, m)$ memperbaiki bias dan bobotnya ($j = 1,2,3, \dots, p$)

$$w_{01}(baru) = w_{01}(lama) + \Delta w_{01} = 0,5 + (-0,0281) = 0,4719$$

$$w_{02}(baru) = w_{02}(lama) + \Delta w_{02} = 0,5 + (0,0069) = 0,5069$$

$$w_{11}(baru) = w_{11}(lama) + \Delta w_{11} = 0,1 + (-0,0154) = 0,1154$$

$$w_{12}(baru) = w_{12}(lama) + \Delta w_{12} = 0,2 + (0,0038) = 0,2038$$

$$w_{21}(baru) = w_{21}(lama) + \Delta w_{21} = 0,4 + (-0,0170) = 0,383$$

$$w_{22}(baru) = w_{22}(lama) + \Delta w_{22} = 0,5 + (0,0039) = 0,5039$$

$$w_{31}(baru) = w_{31}(lama) + \Delta w_{31} = 0,1 + (-0,0168) = 0,0832$$

$$w_{32}(baru) = w_{32}(lama) + \Delta w_{32} = 0,2 + (0,0041) = 0,2041$$

$$w_{41}(baru) = w_{41}(lama) + \Delta w_{41} = 0,4 + (-0,0184) = 0,3816$$

$$w_{42}(baru) = w_{42}(lama) + \Delta w_{42} = 0,5 + (0,0043) = 0,5043$$

Langkah 9 : Menghitung *mean square error*, jika nilai *mean square*

error belum lebih kecil daripada target *error*, maka langkah 2-8 terus dilakukan.

$$MSE = \frac{\sum_{i=1}^n e_i^2}{n}$$

Mean Square Error dihitung sebagai berikut:

1. Hitung keluaran jaringan syaraf untuk masukan pertama aktivasi prediksi

$$y_k = \frac{1}{1 + e^{-y_{ink}}}$$

$$y_1 = \frac{1}{1 + e^{-1,0935}} = 0,7490$$

$$y_2 = \frac{1}{1 + e^{-1,3281}} = 0,7905$$

2. Hitung selisi antara nilai target dengan nilai keluaran prediksi

$$e_1 = 0 - 0,7490 = -0,7490$$

$$e_2 = -0,7490 - 0,7905 = -1,5395$$

3. Kuadratkan setiap selisih

$$e_1^2 = (-0,7490)^2 = 0,5610$$

$$e_2^2 = (-1,5395)^2 = 2,3700$$

4. Jumlahkan semua kuadrat selisih dari tiap-tiap data pembelajaran dalam satu epoch

$$e_1^2 + e_2^2 = 0,5610 + 2,3700 = 2,931$$

5. Bagi hasil penjumlahan tersebut dengan jumlah data

$$MSE = \frac{2,931}{2} = 1,4655$$

Check Error (iterasi berhenti bila $|1,4655| \leq 0,003$)

Maka Kondisi Tidak Berhenti

Lakukan Iterasi

Misalkan setelah akhir iterasi diperoleh nilai bobot input dan bias akhir berikut.

Bobot awal bias ke *hidden layer*:

$$V_{01} = 2,4618 \quad V_{02} = -0,3884 \quad V_{03} = -1,4258 \quad V_{04} = -0,6994$$

Bobot awal *input* ke *hidden layer*

$$\begin{array}{llll} V_{11} = 5,8716 & V_{12} = 3,6067 & V_{13} = 3,4877 & V_{14} = -0,0704 \\ V_{21} = -4,8532 & V_{22} = 2,8028 & V_{23} = -5,1943 & V_{24} = -0,7636 \end{array}$$

Bobot awal bias ke *output layer*

$$W_{01}=0,6571 \quad W_{02}=0,6553$$

Bobot awal *hidden layer* ke *output layer*

$$W_{11}=-7,0997 \quad W_{12}=3,5782$$

$$W_{21}=6,9212 \quad W_{22}=-0,7503$$

$$W_{31}=-7,0855 \quad W_{32}= 3,5343$$

$$V_{41}=6,8812 \quad V_{42}=-0,7444$$

3.1.7 Analisis Proses Pengenalan Dengan *Backpropagation*

Pada proses pengenalan pola tulisan tangan tegak bersambung ini, dilakukan hal yang hampir sama seperti pelatihan. Pengenalan pola tulisan tangan tegak bersambung menggunakan *Backpropagation* dengan menerapkan fungsi aktivasi *sigmoid biner* dimana jika nilai $y \leq 0,5$ berarti 0, dan nilai $y \geq 0,5$ berarti 1. Nilai *biner* ini kemudian dicocokkan dengan citra tulisan tangan tegak bersambung pada database.

Untuk lebih jelas dalam proses pengenalan dapat dilihat pada contoh dibawah ini.

Tabel 3.10 Contoh Identitas Pada Database

No	ID	Huruf
1	01	A
2	1,0	B

Misalkan pada contoh sederhana ini terdapat 2 data identitas yang telah disimpan pada database.

a. Proses pengenalan (Identitas Dikenali)

Pengenlan terhadap data citra tulisan tangan tegak bersambung, diketahui citra masukan memiliki nilai ekstrasi 2 digit {0,0}, maka nilai X_i adalah $X_1=0, X_2=0$

Operasi pada hidden layer

$$\begin{aligned}
 Z_{in1} &= v_{01} + (v_{11} * X_1) + (v_{21} * X_2) \\
 &= 2,4618 + (5,8716 * 0) + (-4,8532 * 0) = 2,4618 \\
 Z_{in2} &= v_{02} + (v_{12} * X_1) + (v_{22} * X_2) \\
 &= -0,3884 + (3,6067 * 0) + (2,8028 * 0) = -0,3884 \\
 Z_{in3} &= v_{03} + (v_{13} * X_1) + (v_{23} * X_2) \\
 &= -1,4258 + (5,4877 * 0) + (-5,1943 * 0) = -1,4258 \\
 Z_{in4} &= v_{04} + (v_{14} * X_1) + (v_{24} * X_2)
 \end{aligned}$$

$$= -0,6994 + (-0,0704 * 0) + (0,7636 * 0) = -0,6994$$

Fungsi aktivasi pada *hidden layer* $Z_i = \frac{1}{1+e^{-Z_{in}}}$

$$Z_1 = \frac{1}{1 + e^{2,4618}} = 0,9214$$

$$Z_2 = \frac{1}{1 + e^{0,3884}} = 0,4041$$

$$Z_3 = \frac{1}{1 + e^{1,4258}} = 0,1938$$

$$Z_4 = \frac{1}{1 + e^{0,6994}} = 0,3319$$

Operasi pada *output layer*

$$\begin{aligned} y_{in1} &= w_{01} + (w_{11} * Z_1) + (w_{21} * Z_2) + (w_{31} * Z_3) + (w_{41} * Z_4) \\ &= 0,6571 + (-7,0997 * 0,9214) + (6,9212 * 0,4041) + (-7,0855 * 0,1938) \\ &\quad + (6,8812 * 0,3319) \end{aligned}$$

$$= 0,6571 + (-6,5417) + 2,7969 + (-1,3732) + 2,2839 = -2,1770$$

$$\begin{aligned} y_{in2} &= w_{02} + (w_{12} * Z_1) + (w_{22} * Z_2) + (w_{32} * Z_3) + (w_{42} * Z_4) \\ &= 0,6553 + (3,5782 * 0,9214) + (-0,7503 * 0,4041) + (3,5343 * 0,1938) \\ &\quad + (-0,7444 * 0,3319) \end{aligned}$$

$$= 0,6553 + 3,2969 + (-0,3032) + 0,6849 + (-0,2471) = 4,0868$$

Fungsi aktivasi pada *output layer* $y_k = \frac{1}{1+e^{-y_{ink}}}$

$$y_1 = \frac{1}{1 + e^{2,1770}} = 0,1018$$

$$y_2 = \frac{1}{1 + e^{-4,0868}} = 0,9835$$

Fungsi aktivasi : $\begin{cases} 0, & \text{jika } y \leq 0,5 \\ 1, & \text{jika } y \geq 0,5 \end{cases}$

$Y1 \geq 0,5 = 0,1018 \geq 0,5 \rightarrow$ dibulatkan menjadi 0

$Y2 \geq 0,5 = 0,9835 \geq 0,5 \rightarrow$ dibulatkan menjadi 1

Maka hasil *output layer* 01 dikenali pada basis data sebagai Z

b. Proses Pengenalan (Identitas Tidak Dikenali)

Pengenalan terhadap data citra tulisan tangan tegak bersambung, diketahui citra masukan memiliki nilai ekstrasi 2 digit {1,1}, maka nilai Xi adalah X1=1, X2=1, dan X3=1

Operasi pada *hidden layer*

$$\begin{aligned}
 Z_{in1} &= v_{01} + (v_{11} * X1) + (v_{21} * X2) \\
 &= 2,4618 + (5,8716 * 1) + (-4,8532 * 1) = 3,4802 \\
 Z_{in2} &= v_{02} + (v_{12} * X1) + (v_{22} * X2) \\
 &= -0,3884 + (3,6067 * 1) + (2,8028 * 1) = 6,0211 \\
 Z_{in3} &= v_{03} + (v_{13} * X1) + (v_{23} * X2) \\
 &= -1,4258 + (5,4877 * 1) + (-5,1943 * 1) = -3,1324 \\
 Z_{in4} &= v_{04} + (v_{14} * X1) + (v_{24} * X2) \\
 &= -0,6994 + (-0,0704 * 1) + (0,7636 * 1) = -0,0062
 \end{aligned}$$

Fungsi aktivasi pada *hidden layer* $Z_i = \frac{1}{1+e^{-Z_{in}}}$

$$Z_1 = \frac{1}{1 + e^{-3,4802}} = 0,9701$$

$$Z_2 = \frac{1}{1 + e^{-6,0211}} = 0,9976$$

$$Z_3 = \frac{1}{1 + e^{3,1324}} = 0,0481$$

$$Z_4 = \frac{1}{1 + e^{0,0062}} = 0,4985$$

Operasi pada *output layer*

$$\begin{aligned}
 y_{in1} &= w_{01} + (w_{11} * Z_1) + (w_{21} * Z_2) + (w_{31} * Z_3) + (w_{41} * Z_4) \\
 &= 0,6571 + (-7,0997 * 0,9701) + (6,9212 * 0,9976) + (-7,0855 * 0,0481) \\
 &\quad + (6,8812 * 0,4985) \\
 &= 0,6571 + (-6,8874) + 6,9046 + (-0,2962) + 3,4303 = 3,8084 \\
 y_{in2} &= w_{02} + (w_{12} * Z_1) + (w_{22} * Z_2) + (w_{32} * Z_3) + (w_{42} * Z_4) \\
 &= 0,6553 + (3,5782 * 0,9701) + (-0,7503 * 0,9976) + (3,5343 * 0,0481) \\
 &\quad + (-0,7444 * 0,4985) \\
 &= 0,6553 + 3,471 + (-0,7485) + 0,1477 + (-0,3711) = 3,1546
 \end{aligned}$$

Fungsi aktivasi pada *output layer* $y_k = \frac{1}{1+e^{-y_{ink}}}$

$$y_1 = \frac{1}{1 + e^{-3,8084}} = 0,9783$$

$$y_2 = \frac{1}{1 + e^{-3,1546}} = 0,9591$$

Fungsi aktivasi : $\begin{cases} 0, & \text{jika } y \leq 0,5 \\ 1, & \text{jika } y \geq 0,5 \end{cases}$

$Y1 \geq 0,5 = 0,9783 \geq 0,5 \rightarrow$ dibulatkan menjadi 1

$Y2 \geq 0,5 = 0,9591 \geq 0,5 \rightarrow$ dibulatkan menjadi 1

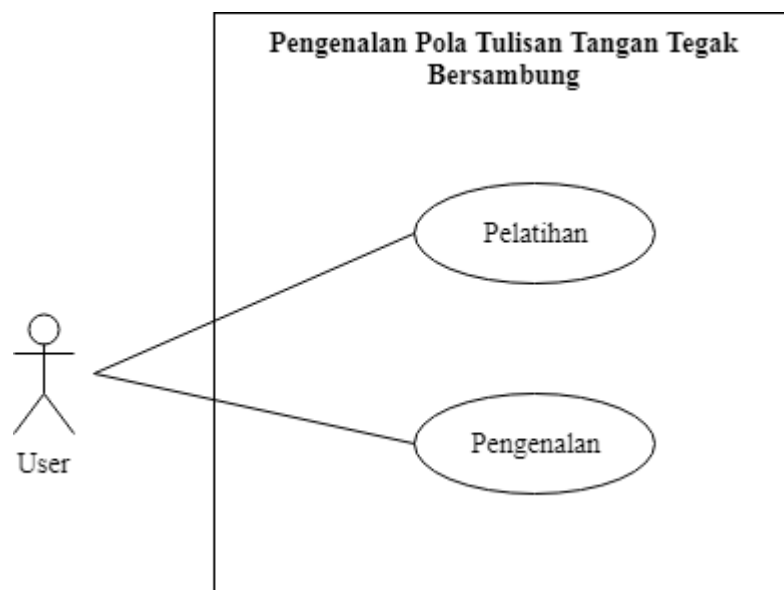
Maka hasil *output layer* 11 tidak dikenali

3.2 Analisis Kebutuhan Sistem

Sebelum memulai proses perancangan aplikasi, maka terlebih dahulu perlu dilakukan proses analisis terhadap kebutuhan sistem yang akan dirancang. Proses analisis kebutuhan sistem mencakup kebutuhan fungsional dan kebutuhan non-fungsional.

3.2.1 Analisis Kebutuhan Fungsional

Analisis kebutuhann fungsional adalah analisis kebutuhan sistem yang ditinjau dari seluruh komponen (entitas). Analisis kebutuhan fungsional dilakukan untuk memberikan gambaran mengenai proses yang dapat dilakukan sebelum merancang sebuah perangkat lunak. Adapun kebutuhan fungsional sistem dapat dilihat pada Gambar 3.10 dibawah ini.



Gambar 3.10 *Use Case* Pengenalan Pola Tulisan Tangan Tegak Bersambung

Tabel 3.11 Narasi *Use Case* Pelatihan

Nama <i>Use Case</i>	Pelatihan	
Aktor	User	
Deskripsi	Use case menjelaskan keadaan pada menu pelatihan	
	Aksi Aktor	Respon Sistem

Bidang Khas suatu <i>event</i>	1. User menekan tombol Open Image yang tersedia dalam form aplikasi untuk memasukan sampel citra tulisan tangan tegak bersambung	2. Sistem membuka <i>file explorer</i> untuk memilih gambar yang akan diproses.
	3. User memilih gambar yang akan diproses dengan mengklik salah satu gambar dan menekan tombol open .	4. Sistem memasukkan gambar ke <i>form</i> aplikasi untuk diolah.
	5. User menekan tombol proses yang tersedia dalam <i>form</i> aplikasi untuk memulai proses <i>grayscale</i> .	6. Sistem melakukan proses konversi citra warna menjadi citra <i>grayscale</i> dan menampilkan hasil proses pada <i>form</i> aplikasi
	7. User menekan tombol proses yang tersedia dalam <i>form</i> aplikasi untuk memulai proses <i>filtering</i> .	8. Sistem melakukan proses <i>filtering</i> untuk menghilangkan noise yang terdapat pada citra hasil <i>grayscale</i> .
	9. User menekan tombol proses yang tersedia dalam <i>form</i> aplikasi untuk memulai proses binerisasi.	10. Sistem melakukan proses binerisasi untuk mengubah citra <i>grayscale</i> menjadi citra biner sehingga warna tiap-tiap piksel pada citra bernilai 0 dan 255 di ubah ke dalam piksel bernilai 0 dan 1.
	11. User menekan tombol proses Hpp yang tersedia dalam <i>form</i> aplikasi untuk memulai proses segmentasi baris dan karakter dengan profil proyeksi	12. Sistem melakukan proses segmentasi baris pada citra dengan profil proyeksi horizontal dan menampilkan hasil proses pada <i>form</i> aplikasi.
	13. User menekan tombol proses vpp yang tersedia dalam <i>form</i> aplikasi untuk memulai proses segmentasi baris dan karakter dengan profil proyeksi	14. Sistem melakukan proses segmentasi karakter pada citra dengan profil proyeksi vertikal dan menampilkan hasil proses pada <i>form</i> aplikasi.

	15. User menekan tombol Ekstraksi Ciri untuk mengambil dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra.	16. Sistem melakukan proses ekstraksi ciri terhadap citra hasil segmentasi karakter dan melakukan pengambilan dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra.
	17. User menekan tombol Bantuan.	18. Sistem menampilkan informasi proses kerja dari sistem.
	19. User menekan tombol reset untuk kembali ke proses awal.	20. Sistem mengembalikan ke tampilan awal.
	21. User menekan tombol Pelatihan yang tersedia dalam <i>form</i> aplikasi.	22. Sistem menampilkan <i>form</i> pelatihan.
		23. Sistem menampilkan nilai bobot awal.
	24. User memasukkan nilai target.	
	25. User mengisi karakter huruf.	
	26. User menekan tombol Simpan.	27. Sistem melakukan proses simpan data.
	28. User menekan tombol Hapus.	29. Sistem melakukan proses hapus data.
	30. User menekan tombol Lakukan Pelatihan.	31. Sistem melakukan proses pelatihan <i>Artificial Neural Network</i> .
	32. User menekan tombol Kembali.	33. Sistem kembali ke <i>form</i> pra-pengolahan pelatihan.
Bidang Alternatif	-	

Tabel 3.12 Narasi *Use Case* Pengenalan

Nama <i>Use Case</i>	Pengenalan	
Aktor	User	
Deskripsi	Use case menjelaskan keadaan pada menu pengenalan	
	Aksi Aktor	Respon Sistem

Bidang Khas suatu <i>event</i>	1. User menekan tombol Open Image yang tersedia dalam form aplikasi untuk memasukan sampel citra tulisan tangan tegak bersambung	2. Sistem membuka <i>file explorer</i> untuk memilih gambar yang akan diproses.
	3. User memilih gambar yang akan diproses dengan mengklik salah satu gambar dan menekan tombol open .	4. Sistem memasukkan gambar ke <i>form</i> aplikasi untuk diolah.
	5. User menekan tombol proses yang tersedia dalam <i>form</i> aplikasi untuk memulai proses <i>grayscale</i> .	6. Sistem melakukan proses konversi citra warna menjadi citra <i>grayscale</i> dan menampilkan hasil proses pada <i>form</i> aplikasi
	7. User menekan tombol proses yang tersedia dalam <i>form</i> aplikasi untuk memulai proses <i>filtering</i> .	8. Sistem melakukan proses <i>filtering</i> untuk menghilangkan noise yang terdapat pada citra hasil <i>grayscale</i> .
	9. User menekan tombol proses yang tersedia dalam <i>form</i> aplikasi untuk memulai proses binerisasi.	10. Sistem melakukan proses binerisasi untuk mengubah citra <i>grayscale</i> menjadi citra biner sehingga warna tiap-tiap piksel pada citra bernilai 0 dan 255 di ubah ke dalam piksel bernilai 0 dan 1.
	11. User menekan tombol proses Hpp yang tersedia dalam <i>form</i> aplikasi untuk memulai proses segmentasi baris dan karakter dengan profil proyeksi	12. Sistem melakukan proses segmentasi baris pada citra dengan profil proyeksi horizontal dan menampilkan hasil proses pada <i>form</i> aplikasi.
	13. User menekan tombol proses vpp yang tersedia dalam <i>form</i> aplikasi untuk memulai proses segmentasi baris dan karakter dengan profil proyeksi	14. Sistem melakukan proses segmentasi karakter pada citra dengan profil proyeksi vertikal dan menampilkan hasil proses pada <i>form</i> aplikasi.

	15. User menekan tombol Ekstraksi Ciri untuk mengambil dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra.	16. Sistem melakukan proses ekstraksi ciri terhadap citra hasil segmentasi karakter dan melakukan pengambilan dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra.
	17. User menekan tombol Bantuan.	18. Sistem menampilkan informasi proses kerja dari sistem.
	19. User menekan tombol reset untuk kembali ke proses awal.	20. Sistem mengembalikan ke tampilan awal.
	21. User menekan tombol Pengenalan yang tersedia dalam <i>form</i> aplikasi.	22. Sistem menampilkan <i>form</i> pengenalan.
		23. Sistem melakukan proses pengenalan.
		24. User menampilkan nilai bobot akhir.
		25. User menampilkan hasil pengenalan.
	26. User menekan tombol Tampilkan perhitungan.	27. Sistem menampilkan hasil perhitungan <i>Artificial Neural Network</i> .
	32. User menekan tombol Kembali	33. Sistem kembali ke <i>form</i> pra-pengolahan pengenalan.
Bidang Alternatif	-	

3.2.2 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional perangkat lunak merupakan batasan-batasan seperti ciri, karakteristik, dan batasan lain yang juga menentukan tingkat kepuasan akan sistem simulasi. Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi dalam performansi, informasi, ekonomi, kontrol, efisiensi, dan servis. Klasifikasi tersebut dikenal dengan PIECES.

1. *Perfomence*, yaitu mampu mengenali pola tulisan tangan tegak bersambung dengan memiliki akurasi yang baik.

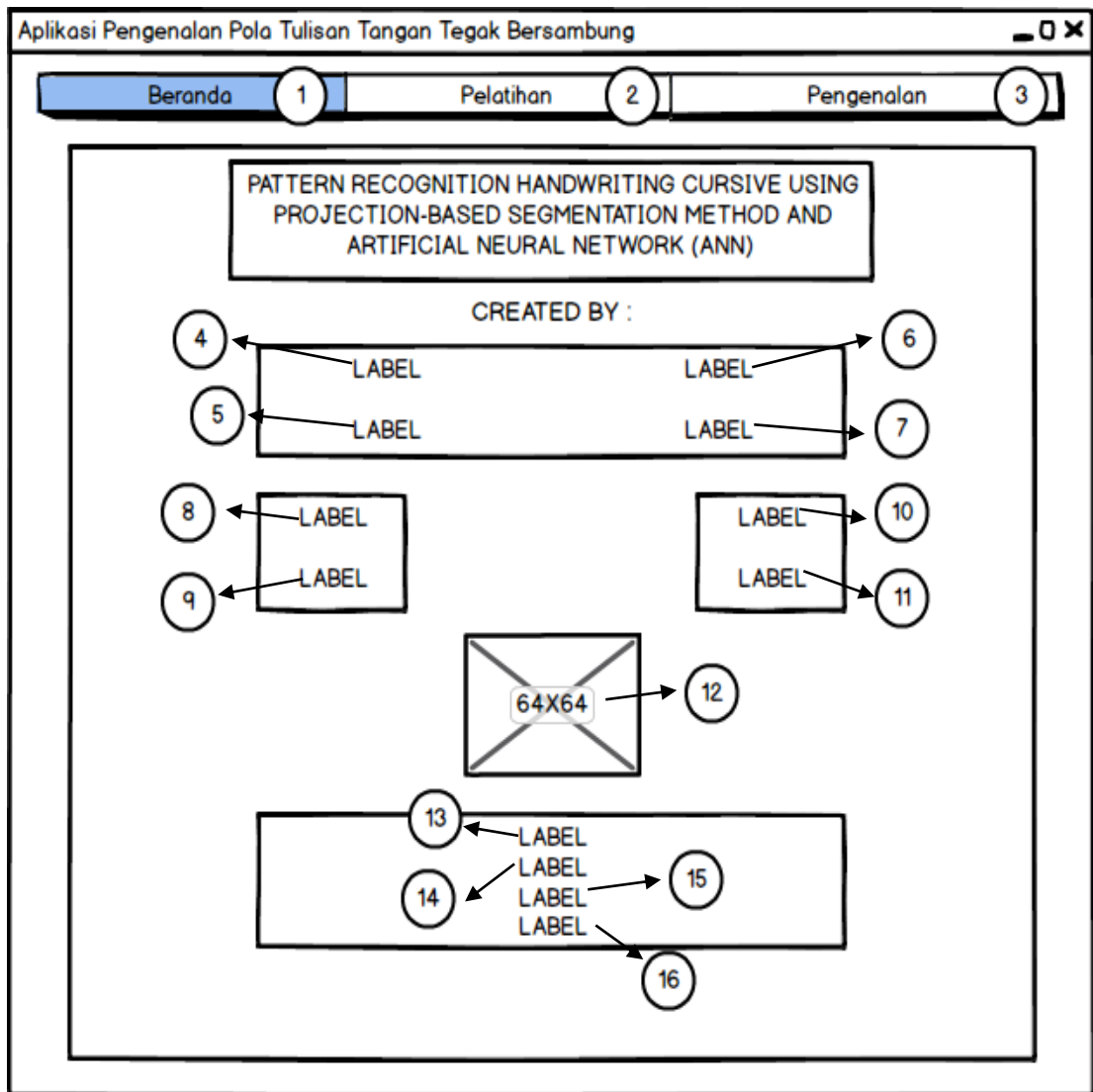
2. *Information*, yaitu perangkat lunak mampu menampilkan perubahan gambar pada setiap proses yang dijalankan.
3. *Economy*, yaitu tidak perlu mengeluarkan biaya tambahan untuk dapat menggunakan aplikasi pengenalan pola tulisan tangan tegak bersambung ini. Aplikasi hanya menggunakan *.NET Framework*.
4. *Control*, yaitu perangkat lunak akan menampilkan pesan kesalahan apabila terdapat kesalahan selama aplikasi dijalankan. Pesan tersebut akan ditampilkan dengan *message box* pada aplikasi ini.
5. *Effeciency*, yaitu waktu yang diperlukan untuk identifikasi citra tulisan tangan tegak bersambung lebih cepat.
6. *Service*, yaitu aplikasi menyediakan petunjuk (*Bantuan*) untuk mempermudah *user* dalam menggunakan aplikasi.

3.3 Perancangan Tampilan

Pada perancangan tampilan dibagi menjadi 3 tampilan, yaitu tampilan *form* menu utama, dan perancangan tampilan *form* proses pra-pengolahan, *form* proses pengenalan.

3.3.1 Perancangan Tampilan Form Halaman Utama

Pada *form* Halaman Utama dari program ini berfungsi sebagai menu awal program. Tampilan menu awal ini berguna sebagai *interface* untuk menuju pada beberapa *form* yang ada pada aplikasi. Rancangan tampilan *form* Halaman Utama dapat dilihat seperti Gambar 3.11 yang terdapat dibawah ini:



Gambar 3.11 Tampilan *Form* Halaman Utama

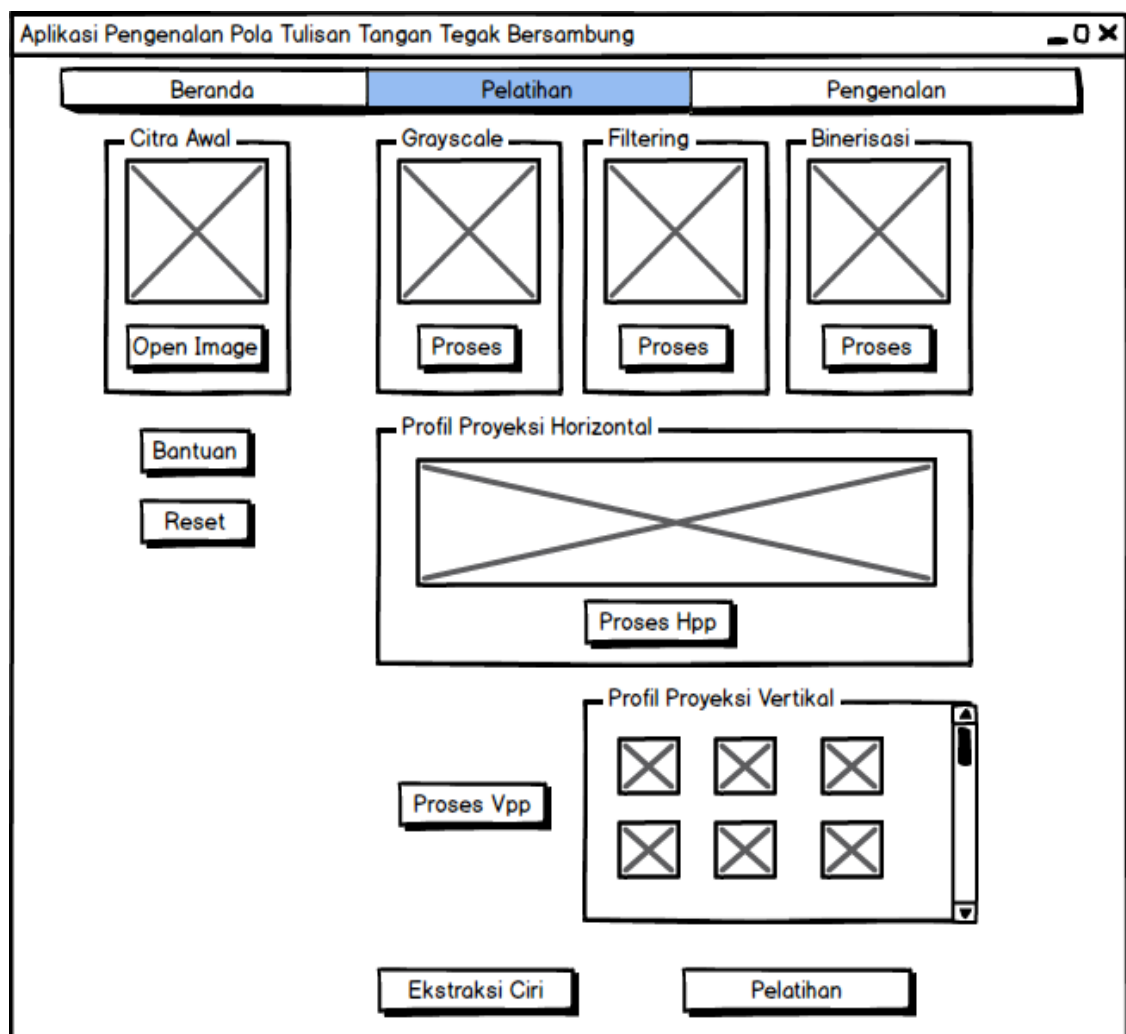
Keterangan Gambar :

1. Beranda, tombol ini berfungsi menampilkan halaman awal sistem.
2. Pelatihan, tombol ini berfungsi mengarahkan user menuju *form* pelatihan yang digunakan sebagai *form* untuk melatih citra tulisan tangan tegak bersambung.
3. Pengenalan, tombol ini berfungsi mengarahkan user menuju *form* pengenalan yang digunakan sebagai *form* untuk mengenali citra tulisan tangan tegak bersambung.
4. Nama Mahasiswa I.
5. NIM Mahasiswa I.
6. Nama Mahasiswa II.
7. NIM Mahasiswa II.
8. Pembimbing I (titel).

9. Nama Pembimbing I.
10. Pembimbing II (titel).
11. Nama Pembimbing II.
12. Logo Mikroskil.
13. Informasi dari jurusan.
14. Informasi dari sekolah tinggi.
15. Nama Kota.
16. Tahun.

3.3.2 Perancangan Tampilan *Form* Pelatihan

Pada form Pelatihan merupakan *form* yang digunakan untuk mengisi atau memasukan citra tulisan tangan tegak bersambung dan juga identitas huruf secara lengkap yang kemudian akan dilakukan proses ekstraksi untuk mendapatkan nilai dari pola tulisan tangan tegak bersambung tersebut. Rancangan tampilan *form* Pra-Pengolahan Pelatihan dapat dilihat seperti Gambar 3.12 yang terdapat dibawah ini:



Gambar 3.12 Tampilan *Form* Pra-Pengolahan Pelatihan

Keterangan Gambar :

1. *Open Image*, tombol ini berfungsi memanggil input citra masukan (Tulisan tangan tegak bersambung).
2. Proses pada *Grayscale* , tombol ini berfungsi melakukan proses *grayscale* untuk mengubah citra awal menjadi keabuan.
3. Proses pada *Filtering* , tombol ini berfungsi melakukan proses *filtering* untuk menghilangkan noise pada citra *grayscale*.
4. Proses pada *Binerisasi* , tombol ini berfungsi melakukan proses *binerisasi* untuk mengubah citra *grayscale* menjadi citra biner.
5. Proses Hpp, tombol ini berfungsi untuk melakukan proses segmentasi baris dengan profil proyeksi horizontal.
6. Proses Vpp, tombol ini berfungsi untuk melakukan proses segmentasi karakter dengan profil proyeksi vertikal.
7. Ekstraksi Ciri, tombol ini berfungsi untuk melakukan proses ekstraksi ciri terhadap citra hasil segmentasi karakter dan melakukan pengambilan dan menyimpan karakteristik - karakteristik khusus dari suatu objek pada citra.
8. Bantuan, berfungsi untuk memberikan bantuan kepada user dalam menggunakan aplikasi pada bagian menu pra-pengolahan pelatihan.
9. Reset, berfungsi untuk kembali ke proses awal.
10. Pelatihan, tombol ini berfungsi untuk menuju ke *form* pelatihan.

Rancangan tampilan *form* Pelatihan dapat dilihat seperti Gambar 3.13 yang terdapat dibawah ini:

Bobot Awal

Nilai Input Citra

Masukan Nilai Target

Generate Batal

Isi Nama Huruf

Simpan Hapus Lakukan Pelatihan

Bobot Akhir

Indeks Bobot	Bobot Akhir

Input	Target	Huruf
010011	0000	A
011001	0000	C
110011	0000	B
000111	0000	F

Kembali

Gambar 3.13 Tampilan *Form* Pelatihan

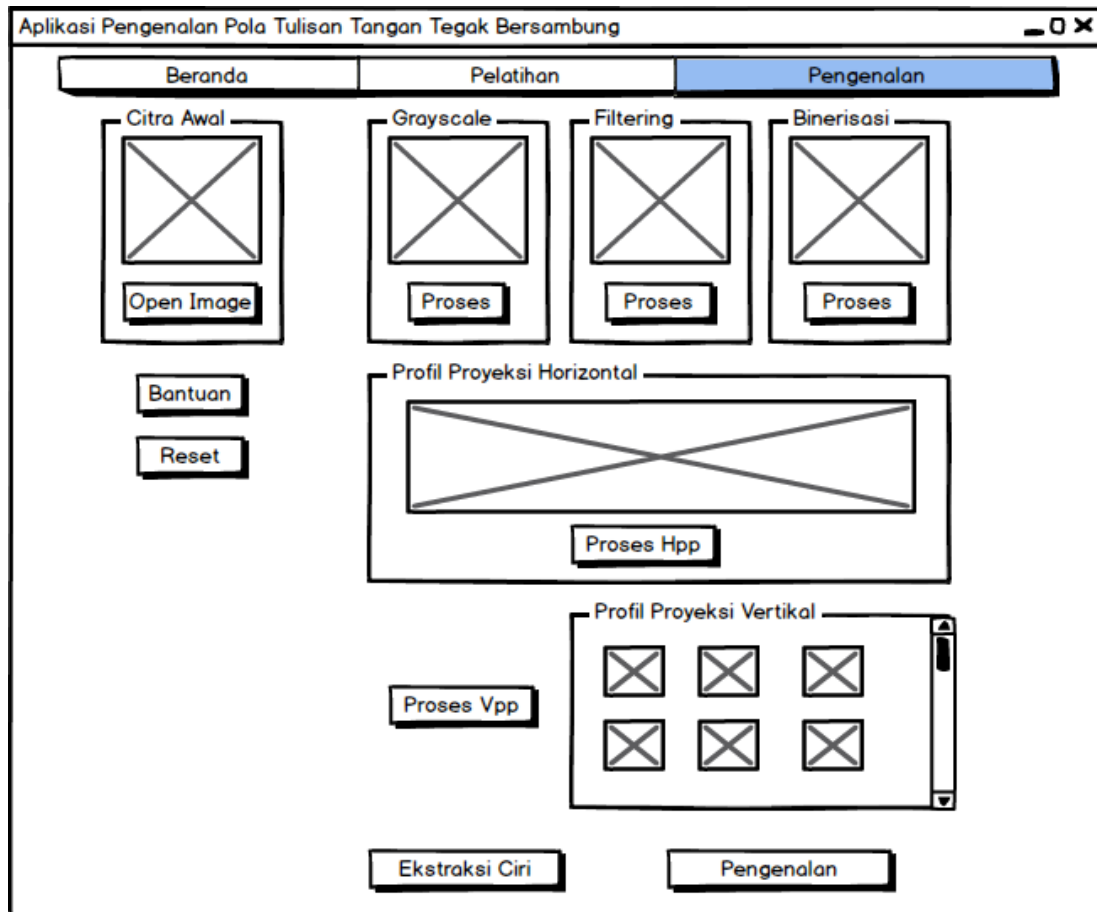
Keterangan Gambar :

1. Bobot Awal, berfungsi menampilkan nilai acak bobot awal.
2. Nilai Target, *textbox* ini berfungsi untuk memasukkan nilai target berupa angka biner.
3. *Generate*, tombol ini berfungsi untuk mengisi nilai target secara acak.
4. Batal, tombol ini berfungsi mengosongkan nilai target yang telah di acak.
5. Isi Nama Huruf, *textbox* ini berfungsi untuk memasukkan nama karakter huruf.
6. Simpan, tombol ini berfungsi untuk menyimpan data ke database.
7. Hapus, tombol ini berfungsi untuk menghapus data pada database.
8. Lakukan Pelatihan, tombol ini berfungsi untuk melakukan proses pelatihan.
9. Tabel data Bobot, tabel ini berfungsi menampilkan semua data bobot.
10. Tabel identitas, berfungsi menampilkan semua data nilai input citra.
11. Kembali, tombol ini berfungsi untuk kembali ke *form* pra-pengolahan pelatihan.

3.3.3 Perancangan Tampilan Form Pengenalan

Pada *form* Pengenalan memiliki fungsi sebagai pengenalan atau membandingkan pola citra tulisan tangan tegak bersambung dan mencocokkan apakah pola tulisan tangan tegak bersambung terdapat pada basis data, jika pola tersebut cocok atau sama maka akan tampil

informasi jenis huruf dari citra masukan tulisan tangan tegak bersambung. Rancangan tampilan *form* Pra-Pengolahan Pengenalan dapat dilihat seperti Gambar 3.14 yang terdapat dibawah ini:



Gambar 3.14 Tampilan *Form* Pra-Pengolahan Pengenalan

Keterangan Gambar :

1. *Open Image*, tombol ini berfungsi memanggil input citra masukan (Tulisan tangan tegak bersambung).
2. Proses pada *Grayscale* , tombol ini berfungsi melakukan proses *grayscale* untuk mengubah citra awal menjadi keabuan.
3. Proses pada *Filtering* , tombol ini berfungsi melakukan proses *filtering* untuk menghilangkan noise pada citra *grayscale*.
4. Proses pada *Binerisasi* , tombol ini berfungsi melakukan proses *binerisasi* untuk mengubah citra *grayscale* menjadi citra biner.
5. Proses Hpp, tombol ini berfungsi untuk melakukan proses segmentasi baris dengan profil proyeksi horizontal.

6. Proses Vpp, tombol ini berfungsi untuk melakukan proses segmentasi karakter dengan profil proyeksi vertikal.
7. Ekstraksi Ciri, tombol ini berfungsi untuk melakukan proses ekstraksi ciri terhadap citra hasil segmentasi karakter dan melakukan pengambilan dan menyimpan karakteristik-karakteristik khusus dari suatu objek pada citra.
8. Bantuan, berfungsi untuk memberikan bantuan kepada user dalam menggunakan aplikasi pada bagian menu Pelatihan.
9. Reset, berfungsi untuk kembali ke proses awal.
10. Pengenalan, tombol ini berfungsi untuk menuju ke *form* pengenalan.

Rancangan tampilan *form* Pengenalan dapat dilihat seperti Gambar 3.15 yang terdapat dibawah ini:

Indeks Bobot	Bobot Akhir

Input	Target	Huruf
010011	0000	A
011001	0000	C
110011	0000	B
000111	0000	F

Tampilkan Perhitungan

Data dikenali sebagai

Kembali

Gambar 3.15 Tampilan *Form* Pengenalan

Keterangan Gambar :

1. Bobot Akhir, tabel ini berfungsi menampilkan semua data bobot.
2. Data Jenis Huruf, tabel ini berfungsi menampilkan semua data nilai input citra.
3. Tampilkan Perhitungan, tombol ini berfungsi menampilkan hasil perhitungan *Artificial Neural Network*.

4. Data dikenali sebagai, berfungsi untuk menampilkan hasil pengenalan.
5. Kembali, tombol ini berfungsi untuk kembali ke *form* pra-pengolahan pengenalan.

DAFTAR PUSTAKA

- Abdul, K., & Adhi, S. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta, Andi Offset.
- Ahmad, Usman. (2005). *Pengolahan Citra Digital dan Teknik Pemogramannya*. Edisi Pertama. Yogyakarta : Graha Ilmu.
- Ambarwati, A., Passarella, R., Sutarno. (2016). Segmentasi Citra Digital Menggunakan Thresholding Otsu untuk Analisa Perbandingan Deteksi Tepi. *Prosiding Annual Research Seminar* 2016, 2(1). Retrieved from <http://seminar.ilkom.unsri.ac.id/index.php/ars/article/viewFile/855/764>
- Arfian, Riki. (2012). *Pengenalan Huruf Pada Form Menggunakan Metode Ekstraksi Ciri Diagonal Dan Algoritma Probabilistic Neural Network*. Bandung. Institute Teknologi Telkom.
- Chandel Ruchika, Gupta Gaurav. (2013). *Image Filtering Algorithms and Techniques: A Review*. International Journal of Advanced Research in Computer Science and Software Engineering Vol.3, Issue 10.
- Cheriet, M., Kharma, N., Liu, C., Suen, C.Y., (2006). *"Character Recognition System A Guide for Student and Practioners"*, John willey & Sons. Inc.
- dos Santos, R.P., Clemente, G.S., Ren, T.I. and Cavalcanti, G.D. (2009). Text line segmentation based on morphology and histogram projection, *10th International Conference on Document Analysis and Recognition, ICDAR'09, Barcelona, Spain*, pp. 651–655.
- Erdogan, T., & Erdogan, O. (2012). An analysis of the legibility of cursive handwriting of prospective primary school teachers. *Procedia-Social and Behavioral Science*, 46, 5214- 5218.
- Fabijańska, A., Węgliński, T., Zakrzewski, K. & Nowosławska, E. (2014). Assessment of hydrocephalus in children based on digital image processing and analysis, *International Journal of Applied Mathematics and Computer Science* 24(2): 299–312, DOI: 10.2478/amcs-2014-0022.
- Fadliansyah. (2007). *Computer Vision dan Pengolahan Citra*. Andi: Yogyakarta.
- Fausett, L. (1994). *Fundamentals of Neural Network: Architectures, Alghorithms, and Applications*, Prentice-Hall, New Jersey.
- Gonzalez, R., & Woods, R. (2002). *Digital Image Processing (2nd ed.)*. Prentice Hall: New Jersey.
- Hartanto, S., Sugiharto A., & Endah, S. N. (2012). *"Optical Character Recognition Menggunakan Algoritma Template Matching Correlation"*, Journal of Informatic and Technology, Vol. 1, pp.11-20.

- Hastiana, Rumi. (2010). *Segmentasi Citra Digital Pembulu Darah Mata Untuk Mendeteksi Tingkat Keparahan Diabetic Retinopathy*. Malang: Skripsi UIN Maulana Malik Ibrahim bidang Teknik Informatika.
- Herman & Harjoko, A. (2015). "Pengenalan Spesies Gulma Berdasarkan Bentuk dan Tekstur Daun Menggunakan Jaringan Syaraf Tiruan," *IJCCS.*, Vol. 9, No. 2, pp. 207-218.
- Huang, Z., & Leng, J. (2010). "Analysis of Hu's Moment Invariants on Image Scaling and Rotation," *2010 2nd International Conference on Computer Engineering and Technology*, pp. 476-480.
- Inggit Prahesti. (2013). "Implementasi Jaringan Syaraf Tiruan Algoritma Backpropagation Untuk Memprediksi Curah Hujan di Yogyakarta (Naskah Publikasi)." Yogyakarta: STMIK Amikom.
- Kadir, Abdul & Adhi Susanto. (2013). *Teori dan Aplikasi Pengolahan Citra*. Yogyakarta: Andi Publisher.
- Kumar, S. (2009). *A Comprehensive On On-Line Handwriting Recognition Technology And Its Real Application To The Nepalese Natural Handwriting*. Kathmandu University Journal of Science, Engineering and Technology.
- Kusumadewi, S.. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*. Yogyakarta: Graha Ilmu.
- Kvanicku, Haykina. (2011). *Feedforward Neural Network*, <http://smecnet.itchybit.org/wiki/cogsci/ui>, 17 Maret 2011 20.00
- Likforman-Sulem, L., Zahour, A., & Taconet, B. (2006). Text line segmentation of historical documents: a survey. *International Journal of Document Analysis and Recognition (IJ DAR)*, 9(2-4), 123–138. <http://doi.org/10.1007/s10032-006-0023-z>
- Marvin Wijaya. Ch. (2007). *Pengolahan Citra Digital Menggunakan MATLAB Image Processing Toolbox*. Informatika: Bandung.
- Munir, R. (2004). *Pengolahan Citra Digital dengan Pendetakan Algoritmik*. Informatika Bandung.
- Pangestu, P. (2015). *Penerapan Histogram Equalization pada Optical Character Recognition Preprocessing*.
- Puspitanigrum, D.. (2006). *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Penerbit Andi.
- Purba, D. (2010). *Pengolahan Citra Digital*. Andi, Yogyakarta.
- Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta : Andi Offset. [Online]
Available from :
<https://books.google.co.id/books?id=NectMutqXJAC&printsec=frontcover#v=onepage&q&f=false> [Accesed 18-Oktober-2017 10.00 PM]

- Ridwanuddin, Dindin. (2015). *Bahasa Indonesia*. Jakarta : UIN Press.
- Rubul, K.B., & RuhUllah khan. (2018). *Handwritten Assamese Character Recognition using Texture and Diagonal Orientation features with Artificial Neural Network*.
- Sam'ani, & Haris, M. (2017). *Pengenalan Huruf Dan Angka Tulisan Tangan Menggunakan Metode Convolution Neural Network*.
- Sapermana, R. (2016, 11 23). *Pengertian Citra dan Pengolahan Citra*. Retrieved from Romli Sapermana Blog: <http://www.romlisapermana.com/2015/07/pengertian-citra-dan-pengolahancitra>.
- Setiawan, B.I., Rudiyanto. (2004). “Aplikasi Neural Networks Untuk Prediksi Aliran Sungai”, *Prosiding Semiloka Teknologi Simulasi dan Komputasi serta Aplikasi 2004 – BPPT*, Jakarta.
- Siang, J.J.. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta : Penerbit Andi.
- Siang, J.J.. (2009). *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- Sukmawan, Arief. Tugas Akhir. (2008). *Implementasi Pembacaan Huruf Hijaiyyah dengan Pendekatan Struktur Melalui Ekstraksi Ciri Vektor dan Pengenalan Menggunakan Jaringan Syaraf Tiruan SOMs (Self-Organizing Maps)*. Jurusan Teknik Telekomunikasi IT Telkom Bandung.
- Sutojo, S. (2004). *Membangun Citra Perusahaan*. Jakarta: Damar Mulia Pustaka.
- Sutojo, T., Mulyanto, E., dan Suhartono, V. (2010). *Kecerdasan Buatan*, edisi 1, Penerbit Andi, Semarang.
- Rizanti, H. V., Suprpto & Regasari, R. (2013). “Pengenalan Citra Alphabet Berdasarkan Paramater Momen Invarian dengan Metode Case-Base Reasoning,” *Repositori Jurnal Mahasiswa PTIIK UB*, vol. 1.
- Widiarti Anastasia R. (2006), *Pengenalan Citra Dokumen Sastra Jawa Konsep dan Implementasinya*.
http://www.library.usd.ac.id/Data%20PDF/Tesis%20dan%20Disertasi%20Dosen/Tesis%20dan%20Disertasi/1365_full.pdf diakses tanggal 12 Oktober 2016.