

**SISTEM PEMBACAAN CUP COUNTER ANEMOMETER MENGGUNAKAN
WEBCAM DENGAN OPTICAL CHARACTER RECOGNITION
PADA BMKG KARANGPLOSO**

SKRIPSI



Disusun Oleh:

**MUHAMAD MA'SHUM ARIF
09.18.157**



**JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

3014

THESE ARE THE RESULTS OF THE RESEARCH
CONDUCTED BY THE RESEARCHERS
IN THE FIELD OF THE RESEARCH

RESEARCH
RESEARCHERS
RESEARCHERS

RESEARCH

RESEARCH
RESEARCHERS
RESEARCHERS

LEMBAR PERSETUJUAN DAN PENGESAHAN
SISTEM PEMBACAAN CUP COUNTER ANEMOMETER
MENGUNAKAN WEBCAM DENGAN OPTICAL CHARACTER
RECOGNITION PADA BMKG KARANGPLOSO

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna
mencapai Gelar Sarjana Komputer Strata Satu (S-1)*

Disusun Oleh :
Muhamad Ma'shum Arif
09.18.157

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Sonny Prasetyo, ST, MT
NIP.P. 1031000433

Program Studi Teknik Informatika S-1
Ketua

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : MUHAMAD MA'SHUM ARIF

Nim : 09.18.157

Program Studi : Teknik Informatika S-1

Fakultas : Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

“SISTEM PEMBACAAN CUP COUNTER ANEMOMETER MENGGUNAKAN WEBCAM DENGAN OPTICAL CHARACTER RECOGNITION PADA BMKG KARANGPLOSO” Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 8 Maret 2014

yang membuat pernyataan



Muhamad Ma'shum Arif

Sistem Pembacaan Cup Counter Anemometer Menggunakan Webcam Dengan Optical Character Recognition Pada BMKG Karangploso

Muhamad Ma'shum Arif (0918157)

Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional
Jalan Raya Karanglo Km2 Malang
E-mail: maksum.arif@gmail.com

Abstrak

Optical Character Recognition (OCR) adalah sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (image character) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam database aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar karakter dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan.

Dalam skripsi ini, dibuat sistem atau program untuk mengenali karakter alat pengukur kecepatan angin cup counter anemometer menggunakan metode OCR (optical character recognition). Citra display cup counter anemometer dijadikan sebagai masukkan sistem, citra tersebut lalu di olah dengan cara mendeteksi kontur obyek yang berbentuk karakter pada citra. Pada citra display cup counter anemometer inilah proses ekstraksi obyek karakter dilakukan kemudian dikenali dengan menggunakan metode OCR (optical character recognition) dan Backpropagation.

Hasil yang diperoleh setelah dilakukan pengujian pada beberapa sample menunjukan bahwa pengenalan karakter pada alat pengukur kecepatan angin cup counter anemometer bisa dilakukan dengan menggunakan metode OCR (optical character recognition). Dengan adanya penelitian ini diharapkan bisa bermanfaat bagi instansi dalam melakukan pencatatan kecepatan angin.

Kata kunci: *Optical Character Recognition, Jaringan Backpropagation, Cup Counter Anemometer*

KATA PENGANTAR

Puji dan Syukur penyusun panjatkan kehadiran Tuhan Yang Maha Esa, karena telah memberikan rahmat-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul **SISTEM PEMBACAAN CUP COUNTER ANEMOMETER MENGGUNAKAN WEBCAM DENGAN OPTICAL CHARACTER RECOGNITION PADA BMKG KARANGPLOSO** sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Ucapkan terima kasih yang kepada :

1. Ir. Soeparno, MT selaku Rektor ITN Malang.
2. Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Joseph Dedy Irawan, ST, MT selaku Ketua Jurusan Teknik Informatika ITN Malang.
4. Joseph Dedy Irawan, ST, MT selaku dosen pembimbing I.
5. Sonny Prasetyo, ST, MT selaku dosen pembimbing II
6. Semua pihak yang telah membantu terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca. Semoga skripsi ini bisa bermanfaat bagi para pembaca sekalian.

Malang, Februari 2014

Penyusun

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN DAN PENGESAHAN SKRIPSI	ii
SURAT PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Pembahasan	3
 BAB II LANDASAN TEORI	 5
2.1 Image	5
2.1.1. Analog Image	5
2.1.2. Digital Image.....	5
2.2 Pengolahan Citra	6
2.2.1 Pengertian Citra	6
2.2.2 RGB	8
2.2.3 Derajat Keabuan (Grayscale)	8
2.2.4 Threshold	8
2.2.5 Cropping	12

2.3	Computer Vision	13
2.3.1	Image Acquisition	13
2.3.2	Image Enhancement	14
2.3.3	Image Segmentation	16
2.3.4	Feature Extraction	21
2.4	Artificial Neural Network	25
2.4.1	Analogi Neural Network Biologis	25
2.4.2	Definisi Artificial Neural Network	25
2.4.3	Cara Kerja Artificial Neural Network	26
2.4.4	Karakteristik Artificial Neural Network	27
2.4.5	Bidang Aplikasi Artificial Neural Network	29
2.5	Bidirectional Associative Memory (BAM)	29
2.5.1	Konsep dasar BAM	30
2.5.2	Stabilitas BAM	31
2.5.3	Hamming Distance	32
2.5.4	Kapasitas Penyimpanan BAM	33
2.5.5	Sifat Penyimpanan dan Pemanggilan	33
2.5.6	Pola Orthogonal	34
2.5.7	Peningkatan Kemampuan BAM	35
2.5.7.1	Multiple Training Encoding	35
2.5.7.2	Dummy Augmentation Encoding	36
2.6	Optical Character Recognition	36
2.7	Automatic Number Plate Recognition	38
2.7.1	Definisi	38
2.7.2	Nama Lain ANPR	38
2.7.3	Komponen Sistem	39
2.7.4	Algoritma Software	39
2.7.5	Cara Kerja ANPR	40
2.7.6	Kehandalan Performa ANPR	41
2.7.7	Bidang Aplikasi	43

2.8	Jaringan Syaraf Biologi dan Tiruan	44
2.8.1	Jaringan Syaraf Biologi.....	44
2.8.2	Jaringan Syaraf Tiruan	45
2.8.3	Aplikasi Jaringan Syaraf Tiruan	47
2.9	BackPropagation	47
2.9.1	Arsitektur BackPropagation	48
2.9.2	Fungsi Aktivasi	48
2.9.3	Pelatihan Standar BackPropagation	49
2.9.4	Optimasi Arsitektur BackPropagation	52
2.9.4.1	Pemilihan Bobot	52
2.9.4.2	Jumlah Unit Tersembunyi	53
2.9.4.3	Lama Iterasi	53
2.9.5	Variasi Backpropagation.....	54
2.9.5.1	Momentum	54
BAB III PERENCANAAN SISTEM.....		55
3.1	Cara Kerja Sistem.....	55
3.2	Perancangan Penempatan Webcam.....	56
3.3	Perancangan Blog Diagram Aplikasi	57
3.4	Optical Character Recognition	58
3.5	Image Acquisition	60
3.6	Pre-Processing	61
3.6.1	Greyscale	61
3.6.2	Thresholding.....	62
3.6.3	Cropping	64
3.7	Document Page Analisis	65
3.7.1	Line Segmentation.....	66
3.8	Feature Extraction	67
3.9	Training JST Backpropagation.....	68

3.10	Perancangan Flowchart	76
3.10.1	Flowchart Aplikasi	76
3.10.2	Flowchart OCR	78
3.10.3	Flowchart Metode Learning Backpropagation.....	79
3.11	Perancangan GUI	80
BAB IV IMPLEMENTASI DAN PENGUJIAN		83
4.1	Implementasi	83
4.1.1	Pembuatan Tampilan Aplikasi	83
4.1.2	Tampilan Aplikasi	85
4.1.3	Start Timer	86
4.1.4	Preview.....	86
4.1.5	Tombol Capture.....	87
4.1.6	Tombol Cropping	87
4.1.7	Tombol Proses	89
4.1.8	Tombol Simpan	89
4.1.9	Tombol Tampilkan	89
4.1.10	Tombol Keluar	89
4.2	Pengujian Aplikasi	89
4.2.1	Prototipe Pengujian	90
4.2.2	Pengujian berdasarkan jarak pengambilan citra	91
4.2.3	Pengujian berdasarkan jenis font.....	94
4.2.3	Hasil Pengujian.....	95
BAB V PENUTUP.....		102
5.1	Kesimpulan.....	102
5.2	Saran.....	102
DAFTAR PUSTAKA.....		103

DAFTAR GAMBAR

Gambar 2.1. Pemrosesan Citra	6
Gambar 2.2. Kompresi Warna RGB	8
Gambar 2.3 Penentuan Nilai Ambang T	10
Gambar 2.4. Gangguan pada Citra Biner	11
Gambar 2.5. Kesalahan pada Citra Biner.....	12
Gambar 2.6. Koordianat Titik Cropping Citra	12
Gambar 2.7. Image Sebelum dilakukan Manipulasi	15
Gambar 2.8. Image Setelah dilakukan Manipulasi	15
Gambar 2.9. Dua hubungan antar Pixel	22
Gambar 2.10. Proyeksi Binary Image	23
Gambar 2.11. Sel Syaraf Biologis	25
Gambar 2.12. Processing Element	26
Gambar 2.13 ANN Sederhana	27
Gambar 2.14. Bidirectional Assotiative Memory	30
Gambar 2.15. Blok Diagram kerja OCR.....	36
Gambar 2.16. Struktur otak pada manusia	45
Gambar 2.17. Jaringan Syaraf Tiruan	46
Gambar 2.18. Arsitektur Backpropagation	48
Gambar 2.19. Grafik Fungsi Sigmoid Biner	49
 Gambar 3.1. Gambar cara kerja sistem	 55
Gambar 3.2. Blok diagram cara kerja sistem	56
Gambar 3.3. Gambar penempatan Webcam	57
Gambar 3.4. Gambar blok diagram Aplikasi	58
Gambar 3.5. Blok diagram Optical Character Recognition	59
Gambar 3.6. Penentuan nilai ambang T	63
Gambar 3.7. Koordinat titik Cropping	64
Gambar 3.8. Letak titik Cropping	65

Gambar 3.9. Proses deteksi tepi Citra	66
Gambar 3.10. Feature Extraction	68
Gambar 3.11. Flowchart Aplikasi	77
Gambar 3.12. Flowchart OCR	78
Gambar 3.13. Flowchart metode Learning	79
Gambar 3.14. Perancangan GUI	81
Gambar 4.1. Tampilan bacacupcounter.fig.....	83
Gambar 4.2. Desain Tampilan Utama.....	84
Gambar 4.3. Tampilan Aplikasi.....	86
Gambar 4.4. Tampilan Preview	86
Gambar 4.5. Tampilan Capture.....	87
Gambar 4.6. Tampilan Cropping	87
Gambar 4.7. Tampilan Proses	88
Gambar 4.8. Tampilan Simpan	88
Gambar 4.9. Tampilan Riwayat	89
Gambar 4.10. Tampilan Dialog Keluar.....	89
Gambar 4.11. Prototipe Pengujian Paper Number	90
Gambar 4.12. Prototipe Pengujian pada Anemometer.....	91
Gambar 4.13. Sampel citra Paper Number 5 Cm.....	92
Gambar 4.14. Sampel citra Paper Number 10 Cm.....	92
Gambar 4.15 Sampel citra Paper Number 15 Cm.....	93
Gambar 4.16. Sampel citra Paper Number 20 Cm.....	93
Gambar 4.17. Sampel citra Anemometer 10 Cm	94
Gambar 4.18. Pengujian Paper Number 10 Cm.....	99
Gambar 4.19. Pengujian Paper Number 15 Cm.....	100
Gambar 4.20. Hasil Pengujian Cup Counter 10 Cm	101

DAFTAR TABEL

Tabel 4.1	Tag Tools Tampilan Aplikasi	84
Tabel 4.2	Pengujian berdasarkan jenis Font	94
Tabel 4.3	Hasil pengujian	95
Tabel 4.4	Hasil pengujian berdasarkan jenis Font	97
Tabel 4.5	Hasil pengujian Cup Counter Anemometer 10 Cm	100

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Seiring dengan perkembangan dan kemajuan teknologi yang semakin pesat, semua dituntut untuk menjadi lebih praktis. Dari hari ke hari semakin banyak perkembangan yang dijumpai dalam berbagai bidang. Salah satunya yaitu perkembangan mengenai proses pencitraan (Image Processing).

Dengan perkembangan proses pencitraan ini, banyak yang bisa dilakukan, misalnya digunakan untuk pencatatan besarnya kecepatan angin menggunakan cup counter anemometer. Dimana pada saat ini untuk mencatat kecepatan angin dengan menggunakan alat cup counter anemometer, petugas harus mencatat secara manual dari alat cup counter anemometer yang dilakukan secara periodik. Dimana untuk mencatatnya sangat dibutuhkan kejelian. Hal ini dirasa kurang efektif. Untuk mengatasi masalah ini yaitu digunakanlah Webcam yang nantinya dapat mengambil gambar setiap angka / nomor yang ditunjukkan alat cup counter anemometer, kemudian diproses menggunakan image processing untuk membaca berapa angka yang ditunjukkan alat tersebut. Metode Optical Character Recognition yang digunakan untuk pembacaan angka alat ini yaitu menggunakan Jaringan Syaraf Tiruan (JST) Backpropagation. Diharapkan dengan adanya ini semua, pencatatan kecepatan angin secara periodik semakin efisien dan memudahkan petugas dalam mencatat kecepatan angin.

Penggunaan sistem otomatis pada pembacaan cup counter anemometer ini dapat menjadi solusi dari kekurangan-kekurangan sistem manual. Oleh karena itu, akan dilakukan penelitian mengenai teknik pengenalan karakter yang dapat mendukung terciptanya sistem yang dapat menerima masukan data dari digit angka yang ditunjukkan alat cup counter. Data tersebut akan diperoleh melalui sensor citra yang dapat menangkap gambar nomor cup counter secara otomatis. Melalui citra dari display cup counter yang ditangkap oleh sebuah sensor digital dan dilanjutkan dengan pemrosesan secara otomatis oleh komputer, diharapkan dapat meningkatkan kehandalan sistem.

1.2 RUMUSAN MASALAH

Permasalahan pada skripsi ini adalah bagaimana membuat sistem yang dapat membaca angka yang ditunjukkan alat pengukur kecepatan angin cup counter anemometer dari citra yang ditangkap menggunakan webcam secara otomatis dan cara manual.

1.3 TUJUAN

Tujuan dari pembuatan skripsi ini adalah merancang dan membuat aplikasi dengan digital image processing untuk pembacaan alat pengukur kecepatan angin (cup counter anemometer) dengan menggunakan Webcam. Metode yang digunakan untuk pembacaan alat ini yaitu menggunakan Jaringan Syaraf Tiruan (JST) Backpropagation.

1.4 BATASAN MASALAH

Dari permasalahan yang ada dapat dibatasi dengan ketentuan sebagai berikut :

1. Bahasa pemrograman yang digunakan adalah MATLAB versi 7.12.0 (Matlab R2011a).
2. Metode yang digunakan dalam Optical Character Recognition yaitu menggunakan Jaringan Syaraf Tiruan (JST) Backpropagation.
3. Posisi saat pengambilan gambar harus tepat dan jarak yang digunakan tetap.
4. Jenis dan type Cup Counter anemometer yang digunakan dalam pengujian adalah Anemometer jenis cup / mangkok kincir angin.
5. Sistem tidak membaca karakter melainkan hanya membaca angka.

1.5 METODOLOGI

Metode yang dilakukan untuk menyelesaikan skripsi ini yaitu sebagai berikut

1. Studi Literatur

Langkah untuk mempelajari tentang Image Processing (Pengolahan Citra), bahasa pemrograman MATLAB, Backpropagation, serta buku-buku literatur dan informasi dari internet yang berhubungan dengan skripsi ini.

2. Perencanaan Sistem

Pembuatan sistem pembacaan alat pengukur kecepatan angin (cup counter anemometer) menggunakan Webcam yang disambungkan dan dapat diakses sebuah computer (laptop) untuk menangkap gambar. Selanjutnya dilakukan Optical Character Recognition atau pengenalan karakter dengan menggunakan metode Jaringan Syaraf Tiruan (JST) Backpropagation.

3. Pengujian Sistem

Pengujian perangkat lunak untuk membaca alat pengukur kecepatan angin (cup counter anemometer) dengan menggunakan Webcam. Sehingga komputer dapat mengenali angka pada alat yang dibaca sebagai kecepatan angin.

4. Analisa Sistem

Melakukan uji coba program perangkat lunak untuk beberapa kondisi. Dari beberapa hasil yang didapat, bisa diketahui apakah sistem yang dibuat ini sudah sesuai dengan apa yang diharapkan.

5. Kesimpulan

Dari hasil pengujian dapat disimpulkan beberapa hal yang berhubungan dengan sistem yang dibuat.

1.6 SISTEMATIKA PEMBAHASAN

Untuk pembahasan mengenai realisasi skripsi ini dijelaskan berdasarkan sistematika sebagai berikut :

BAB I PENDAHULUAN

Bab pendahuluan yang berisi tentang latar belakang, permasalahan dan batasan masalah, tujuan, metodologi serta sistematika pembahasan penulisan buku skripsi ini.

BAB II LANDASAN TEORI

Membahas mengenai teori penunjang yang akan digunakan dalam penyelesaian tugas akhir ini yang sesuai dengan perencanaan dan pembuatan sistem pembacaan alat pengukur kecepatan angin (cup counter anemometer).

BAB III PERANCANGAN

Membahas mengenai perencanaan atau perancangan sistem dan tahap-tahap pembuatan software pada sistem.

BAB IV IMPLEMENTASI DAN

Berisi data hasil pembuatan sistem dan proses pengujian. Kemudian melakukan analisa dari hasil pengujian.

BAB V PENUTUP

Kesimpulan dari pembahasan bab-bab sebelumnya, saran dan pengembangan dan penyempurnaan untuk skripsi ini.

BAB II

LANDASAN TEORI

Bab ini membahas tentang teori apa saja yang dapat mendukung perancangan dan pembuatan sistem pembacaan alat pengukur kecepatan angin (cup counter anemometer).

2.1 Image

Data masukan yang diproses adalah suatu image. Image merupakan sebuah representasi khusus dari suatu obyek, baik obyek dua dimensi maupun tiga dimensi. Di mana representasi tersebut dinyatakan dalam bentuk dua dimensi. Image dapat berbentuk nyata, maya, ataupun dalam bentuk optik. Selain itu, image juga dapat berupa rekaman, seperti video image, digital image, atau sebuah gambar (*Haralick dan Shapiro, 1992*). Image dapat dikategorikan sebagai :

2.1.1. Analog Image

Menurut *Shapiro dan Stockman (2001)*, analog image adalah image 2D $F(x,y)$ yang memiliki ketelitian tidak terbatas dalam parameter spasial x dan y dan ketelitian tak terbatas pada intensitas tiap titik spasial (x,y) .

2.1.2. Digital Image

Menurut *Shapiro dan Stockman (2001)*, digital image adalah image 2D $I[r,c]$ yang direpresentasikan oleh array diskrit 2D dari intensitas sampel, dimana masing masing titik direpresentasikan dengan ketelitian terbatas.

Digital image juga didefinisikan sebagai representasi dari gambar dua dimensi sebagai himpunan terbatas dari nilai digital yang disebut picture elements atau pixel. Umumnya pixel disimpan dalam komputer sebagai gambar raster, yaitu array dua dimensi dari integer. Nilai ini kadang disimpan dalam bentuk terkompresi. Digital image dapat diperoleh dari berbagai macam alat dan teknik pengambil gambar, seperti kamera digital, scanner, radar, dan sebagainya. Dapat pula disintesis dari data seperti fungsi matematika dan lain lain.

Menurut *Jain, Kasturi dan Schunck (1995)*, pixel adalah sampel dari intensitas image yang terkuantisasi ke dalam nilai integer. Sementara Image merupakan array dua dimensi dari pixel-pixel tersebut. Image inilah yang akan menjadi input awal dalam Computer Vision.

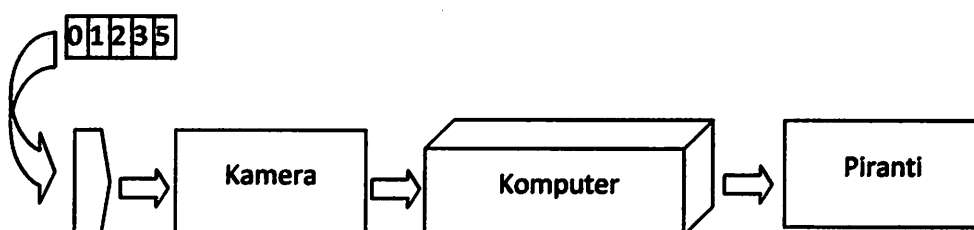
Beberapa bentuk digital image yang sering digunakan dalam Computer Vision (*Shapiro dan Stockman, 2001*):

1. Binary image, yaitu digital image dengan nilai pixel 1 atau 0.
2. Gray scale image, yaitu digital image monokrom dengan satu nilai intensitas tiap pixel.
3. Multispectral image, adalah image 2D yang memiliki vektor nilai pada tiap pixel, jika image-nya berwarna maka vektornya memiliki 3 elemen.
4. Labeled image, adalah digital image dimana nilai pixel-nya adalah simbol dari alfabet terbatas.

2.2 Pengolahan Citra

2.2.1 Pengertian Citra

Citra merupakan informasi yang secara umum tersimpan dalam bentuk pemetaan bit-bit atau sering dikenal dengan bitmap. Setiap bit-bit membentuk satu titik informasi yang dikenal dengan piksel. Satu piksel merupakan satu titik citra yang terdiri dari satu atau beberapa bit informasi. Satuan dari piksel biasanya dinyatakan dengan posisi x, posisi y dan nilai dari piksel (warna atau gray). Dalam satu bidang gambar, sepenuhnya terdiri dari piksel-piksel. Karena itu, file yang menyimpan citra biasa disimpan dengan nama BMP. Untuk mengurangi ukuran dari file, biasanya file citra dimampatkan dengan menggunakan teknik tertentu, misal yang terkenal yaitu JPEG atau GIF.



Gambar 2.1 Pemrosesan Citra

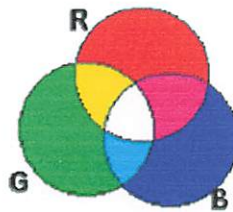
Citra adalah representasi dari dua dimensi untuk bentuk fisik nyata dari tiga dimensi. Citra dalam perwujudannya dapat bermacam-macam, mulai dari gambar berwarna yang bergerak pada pesawat televisi. Citra dapat dikatakan pula sebagai sebuah gambar, foto yang ditampilkan atau bentuk lain yang memberikan representasi visual tentang sebuah obyek atau pemandangan. Pada pemrosesan citra digital sebuah gambar bilangan array 2 dimensi, yang setiap barisnya adalah representasi piksel pada gambar setiap barisnya. Minimum nilai piksel = 0 (hitam), maksimum = 225 (putih). Komputer membutuhkan memori lebih banyak untuk data ini rata-rata 3 kali data storage. *Shapiro dan Stockman (2001)*.

Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra akan dipengaruhi oleh bermacam-macam faktor yang mengakibatkan penampilan citra suatu benda tidak sama persis dengan bentuk fisik nyatanya. Faktor-faktor tersebut merupakan efek degradasi atau penurunan kualitas yang dapat berupa rentang kontras benda yang terlalu sempit atau terlalu lebar, distorsi kekaburan (blur), kekaburan akibat objek citra yang bergerak (motion blur), noise atau gangguan yang disebabkan oleh interferensi peralatan pembuat citra berupa peralatan elektronik. Karena pengolahan citra digital dilakukan dengan komputer digital maka citra yang akan diolah terlebih dahulu ditransformasikan ke dalam bentuk besaran-besaran diskrit dari nilai tingkat keabuan pada titik-titik elemen citra.

Dalam image processing terdapat proses akuisisi citra digital yaitu proses yang sangat menentukan kualitas hasil dari proses pencitraan digital yang akan diperoleh. Agar proses akuisisi citra digital dapat terjadi diperlukan tiga komponen utama yang harus dipenuhi yaitu sumber cahaya, objek atau benda yang akan diamati, dan sensor berupa kamera itu sendiri. Prosesnya yaitu cahaya yang mengenai permukaan benda dipantulkan ke segala arah. Pantulan cahaya ini sebagian ditangkap oleh sensor peka cahaya pada kamera. Intensitas cahaya yang diterima oleh sensor merepresentasikan kondisi objek. Sehingga citra digital yang diperoleh merupakan informasi tentang objek yang terbentuk dari pantulan cahaya atau refleksi pada permukaan objek. *Shapiro dan Stockman (2001)*

2.2.2 RGB

Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 255). Nilai warna – warna pada setiap gambar merupakan gubahan warna merah, hijau dan biru seperti yang terlihat pada gambar 2.2. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar, dilakukan penggabungan antara tiga warna dasar yaitu RGB. *Jain, Kasturi dan Schunck (1995)*



Gambar 2.2 Additive warna RGB

2.2.3 Derajat Keabuan (Gray Scale)

Proses awal yang banyak dilakukan dalam image processing adalah mengubah citra berwarna menjadi citra gray scale, hal ini digunakan untuk menyederhanakan model citra. Seperti telah dijelaskan sebelumnya, citra berwarna terdiri dari 3 layer matrik yaitu, R-layer, G-layer dan B-layer. Sehingga untuk melakukan proses-proses selanjutnya tetap diperhatikan layer – layer diatas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Sehingga konsep itu diubah dengan mengubah 3 layer diatas menjadi 1 layer matrik gray scale dan hasilnya adalah citra gray scale. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. *Jain, Kasturi dan Schunck (1995)*

2.2.4 Threshold

Threshold merupakan konversi citra hitam – putih ke citra biner. Cara yang dilakukan yaitu dengan mengelompokkan nilai derajat keabuan setiap piksel ke dalam 2 kelas, yaitu hitam dan putih. Jika nilai gray scale kurang dari 128 maka ditentukan sebagai hitam (0). Jika nilai gray scale lebih besar dari 128 maka ditentukan sebagai putih (1). *Jain, Kasturi dan Schunck (1995)*

Pengkonversian citra hitam-putih (greyscale) menjadi citra biner dilakukan untuk alasan-alasan sebagai berikut:

1. Untuk mengidentifikasi keberadaan objek, yang direpresentasikan sebagai daerah (region) di dalam citra. Misalnya kita ingin memisahkan (segmentasi) objek dari gambar latar belakangnya. Pixel-pixel objek dinyatakan dengan nilai 1 sedangkan pixel lainnya dengan 0. Objek ditampilkan seperti gambar siluet. Untuk memperoleh siluet yang bagus, objek harus dapat dipisahkan dengan mudah dari gambar latar belakangnya.
2. Untuk lebih memfokuskan pada analisis bentuk morfologi, yang dalam hal ini intensitas pixel tidak terlalu penting dibandingkan bentuknya. Setelah objek dipisahkan dari latar belakangnya, properti geometri dan morfologi/topologi objek dapat dihitung dari citra biner. Hal ini berguna untuk pengambilan keputusan.
3. Untuk menampilkan citra pada piranti keluaran yang hanya mempunyai resolusi intensitas satu bit, yaitu piranti penampil dua-aras atau biner seperti pencetak (printer).
4. Mengkonversi citra yang telah ditingkatkan kualitas tepinya (edge enhancement) ke penggambaran garis-garis tepi. Ini perlu untuk membedakan tepi yang kuat yang berkoresponden dengan batas-batas objek dengan tepi lemah yang berkoresponden dengan perubahan illumination, bayangan, dll.

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

Dua pendekatan yang digunakan dalam operasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal.

a. Pengambangan secara global (global image thresholding)

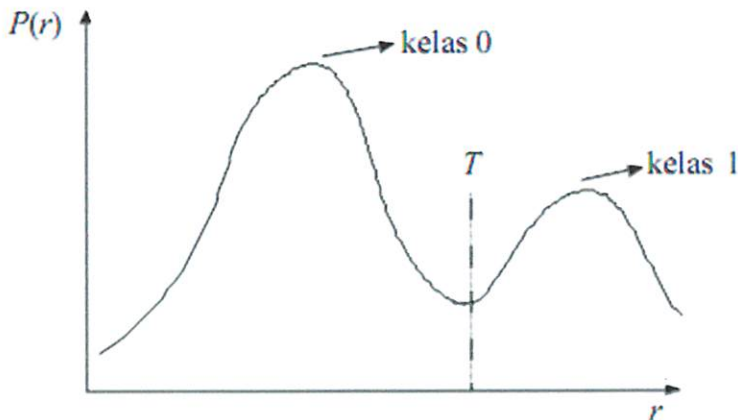
Pengambangan secara global tidak selalu tepat untuk seluruh macam gambar. Beberapa informasi penting di dalam gambar mungkin hilang karena pengambangan global ini. Lagipula, tidak ada harga nilai ambang yang berlaku secara global untuk seluruh daerah citra (misalnya pada citra kedokteran, citra pemandangan alam, dsb).

Setiap pixel di dalam citra dipetakan ke dua nilai, 1 atau 0 dengan fungsi pengambangan:

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

yang dalam hal ini, $f_g(i, j)$ adalah citra hitam-putih, $f_B(i, j)$ adalah citra biner, dan T adalah nilai ambang yang dispesifikasikan. Dengan operasi pengambangan tersebut, objek dibuat berwarna gelap (1 atau hitam) sedangkan latar belakang berwarna terang (0 atau putih).

Nilai ambang T dipilih sedemikian sehingga galat yang diperoleh sekecil mungkin. Cara yang umum menentukan nilai T adalah dengan membuat histogram citra. Jika citra mengandung satu buah objek dan latar belakang mempunyai nilai intensitas yang homogen, maka citra tersebut umumnya mempunyai histogram bimodal (mempunyai dua puncak atau dua buah maksimum lokal) seperti yang ditunjukkan pada Gambar 2.3. Nilai T dipilih pada nilai minimum lokal yang terdapat di antara dua puncak. Dengan cara seperti ini, kita tidak hanya mengkonversi citra hitam-putih ke citra biner, tetapi sekaligus melakukan segmentasi objek dari latar belakangnya.



Gambar 2.3 Penentuan nilai ambang T

Jika nilai intensitas objek diketahui dalam selang $[T_1, T_2]$, maka kita dapat menggunakan fungsi pengambangan:

$$f_B(i, j) = \begin{cases} 1, & T_1 \leq f_g(i, j) \leq T_2 \\ 0, & \text{lainnya} \end{cases} \quad (2.2)$$

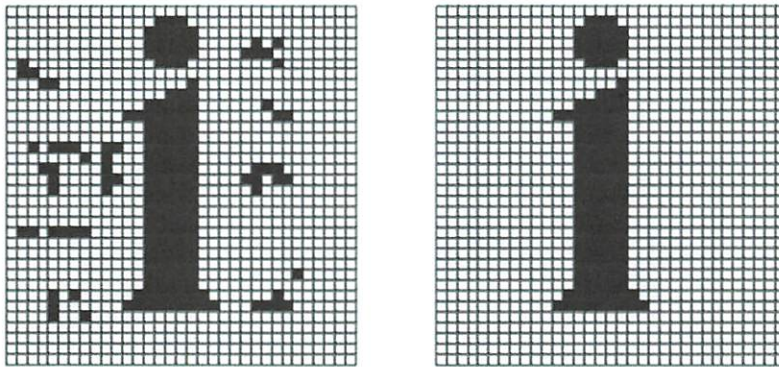
b. Pengambangan secara lokal adaptif (locally adaptive image thresholding)

Pengambangan secara lokal dilakukan terhadap daerah-daerah di dalam citra. Dalam hal ini citra di pecah menjadi bagian-bagian kecil, kemudian proses pengambangan dilakukan secara lokal. Nilai ambang untuk setiap bagian belum tentu sama dengan bagian lain. Sebagai contoh, pengambangan dilakukan terhadap daerah citra yang berukuran 3×3 atau 5×5 pixel. Nilai ambangnya ditentukan sebagai fungsi rata-rata derajat keabuan di dalam daerah citra tersebut. Intensitas pixel yang berbeda secara signifikan dari nilai rata-rata tersebut dianggap mengandung informasi kontras dan ini harus dipertahankan di dalam citra biner.

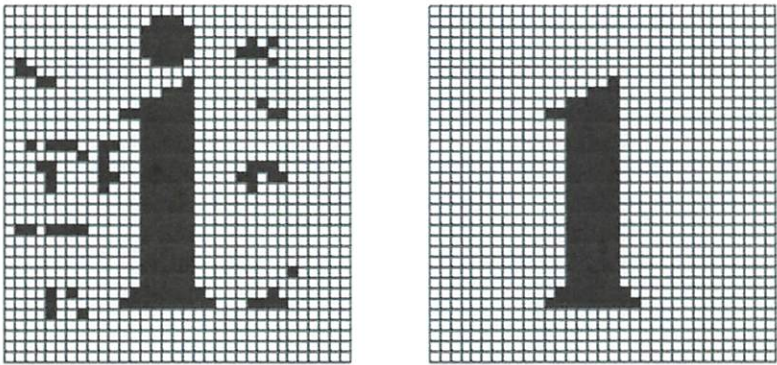
Dengan pengambangan secara lokal adaptif, secara subjektif citra biner yang dihasilkan terlihat lebih menyenangkan dan sedikit informasi yang hilang.

Penapis Luas

Proses pengambangan menghasilkan citra biner. Seringkali citra biner yang dihasilkan mengandung beberapa daerah yang dianggap sebagai gangguan. Biasanya daerah gangguan itu berukuran kecil. Penapis luas dapat digunakan untuk menghilangkan daerah gangguan tersebut. Misalkan objek yang dianalisis diketahui mempunyai luas yang lebih besar dari T . Maka, pixel-pixel dari daerah yang luasnya di bawah T dinyatakan dengan 0. Dengan cara ini, daerah yang berupa gangguan dapat dihilangkan (Gambar 2.4 dan 2.5).



Gambar 2.4 Kiri: gangguan pada citra biner yang mengandung huruf “i”; Kanan: citra yang dihasilkan setelah dilakukan penapisan ($T = 10$)



Gambar 2.5 Kesalahan yang diperoleh dari pengambilan nilai T0 yang tidak tepat (T= 25). Perhatikan bahwa “titik” di atas huruf “i” hilang karena luasnya, sehingga huruf “i” terlihat seperti angka “1”

2.2.5 Cropping

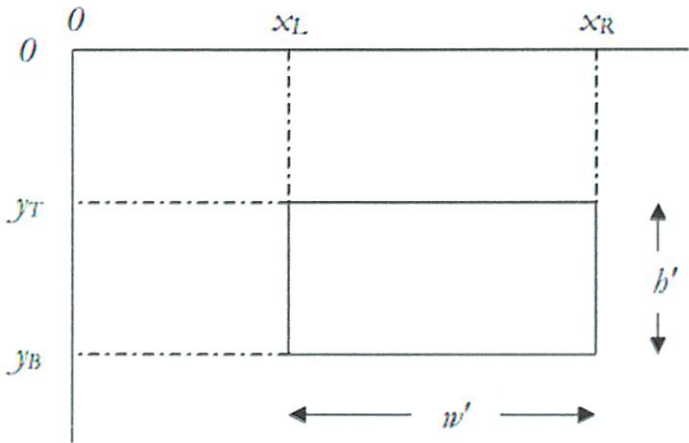
Cropping adalah memotong satu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil. Operasi ini pada dasarnya adalah operasi translasi, yaitu menggeser koordinat titik citra.

Rumus yang digunakan :

$$\begin{aligned} x' &= x - x_L && \text{untuk } x = x_L \text{ sampai } x_R \\ y' &= y - y_T && \text{untuk } y = y_T \text{ sampai } y_B \end{aligned} \tag{2.3}$$

(xL, yT) dan (xR, yB) masing-masing adalah koordinat titik pojok kiri atas dan pojok kanan bawah bagian citra yang hendak dicrop

Koordinat titik sudut bagian citra yang akan di-crop adalah seperti gambar berikut



Gambar 2.6 Koordinat titik sudut cropping citra

2.3. Computer Vision

Computer Vision (sering disebut juga dengan Machine Vision) dapat dideskripsikan sebagai ilmu yang mempelajari metode yang dapat digunakan untuk membuat komputer mengerti gambar atau data banyak dimensi umumnya

Sementara definisi Computer Vision menurut *Kulkarni (2001)*, adalah penyimpulan (deduksi) otomatis akan struktur atau properti dari dunia tiga dimensi dari satu atau lebih image dua dimensi dunia tersebut dan pengenalan objek-objek dengan bantuan properti-properti ini, atau secara singkatnya yaitu proses mengenali objek tertentu dari suatu image.

Menurut *Shapiro dan Stockman (2001)*, tujuan dari Computer Vision adalah untuk membuat keputusan yang berguna tentang objek dunia nyata dan keadaan (scene) berdasarkan image yang diambil. Untuk membuat keputusan akan objek nyata, sangat penting untuk membangun deskripsi atau model objek tersebut dari gambar. Karena itu dapat dikatakan bahwa tujuan dari Computer Vision adalah untuk membangun deskripsi keadaan dari image.

Artificial Intelligence digunakan untuk menganalisis keadaan dengan memproses representasi simbolik dari isi lingkungan setelah image telah diproses untuk diambil fiturnya. Banyak teknik dari Artificial Intelligence berperan penting didalam seluruh aspek Computer Vision. Pada dasarnya Computer Vision merupakan cabang dari Artificial Intelligence. (*Jain, Kasturi dan Schunck, 1995*)

Secara garis besar tahapan tahapan dalam pemrosesan image dalam Computer Vision terdiri dari:

2.3.1. Image Acquisition

Tahapan awal dalam Computer Vision adalah Image Acquisition (pengambilan digital image). Image Acquisition berhubungan dengan sensor yang mengambil image. Sensor yang digunakan bisa kamera atau scanner. Sifat dari sensor dan image yang dihasilkan ditentukan dari *aplikasinya (Kulkarni, 2001)*.

2.3.2. Image Enhancement

Berdasarkan *Kulkarni (2001)*, setelah digital image diperoleh, tahapan selanjutnya adalah image enhancement yang termasuk dalam tahap preprocessing, image enhancement menyangkut langkah-langkah untuk meningkatkan kualitas image untuk mendukung tahapan selanjutnya. Tujuan dari image enhancement ini secara teknis adalah untuk menghilangkan noise, memperhalus gambar, mempertajam gambar (menghilangkan blur), serta mengatur pencahayaan (brightness, contrast). Berdasarkan domainnya teknik peningkatan image dapat dibedakan menjadi 2 metode:

1. Domain spasial

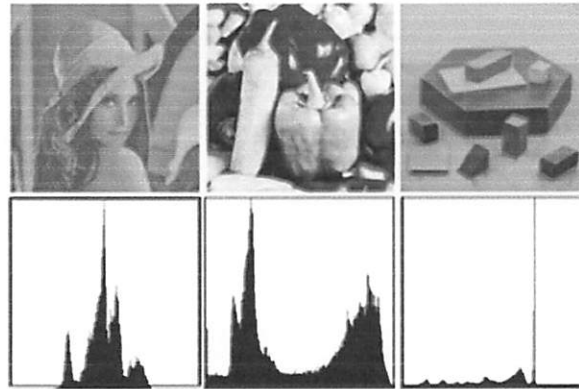
Metode ini didasarkan pada manipulasi langsung dari nilai gray (keabuan) dari pixel-pixel suatu image.

2. Domain frekuensi

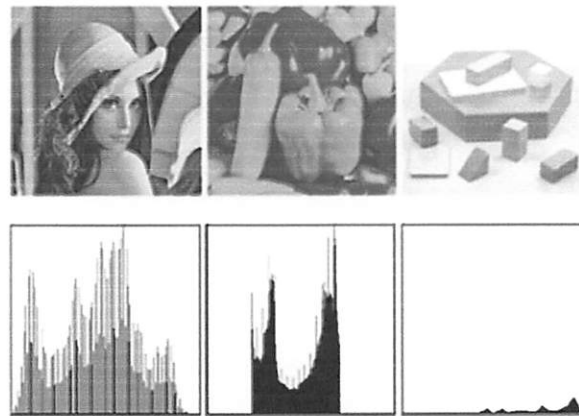
Metode ini didasarkan pada modifikasi Fourier transform dari suatu image. Fourier transform ini sendiri menurut *Nixon dan Aguado (2002)*, adalah suatu cara memetakan sinyal pada frekuensi-frekuensi komponennya. Sementara beberapa teknik image enhancement yang digunakan antara lain:

3. Gray scale manipulation (manipulasi nilai keabuan)

Merupakan teknik pemetaan intensitas dimana tiap pixel diberikan nilai keabuan yang baru untuk meningkatkan ketajaman gambar. Operasi ini tidak merubah bentuk dan geometri image, yang berubah cuma level intensitasnya. Teknik ini dilakukan dengan cara memproses histogram tingkat keabuan (gray level histogram) dari image. (*Kulkarni, 2001*) Histogram dalam computer vision adalah representasi dari image yang diperoleh dengan cara menghitung nilai tiap pixel. Singkatnya histogram merupakan grafik yang menggambarkan distribusi intensitas pixel dari suatu image atau bagian tertentu di dalam citra. Histogram menunjukkan frekuensi kemunculan intensitas pada image. Sementara gray level histogram adalah histogram yang menampilkan dalam tiap tingkat jumlah pixel dalam gambar yang memiliki tingkat keabuan (gray-level) tertentu. (*Kulkarni, 2001*) Histogram ini dimanipulasi dengan cara di stretch, shrink, atau slide.



Gambar 2.7 Image sebelum dilakukan manipulasi



Gambar 2.8 Image setelah dilakukan manipulasi (dari kiri ke kanan): stretch, shrink dan slide pada histogramnya.

4. Filtering (convolution)

Menurut *Nixon dan Aguado (2002)*, filtering yang merupakan suatu group operation pada pixel, menghitung nilai pixel baru dengan menggunakan pixel-pixel tetangganya. Filtering dijelaskan dengan istilah template convolution dimana template-nya adalah suatu matriks koefisien bobot (yang umumnya ganjil dan sama sisi, misalnya 3x3, 5x5, dan seterusnya). Nilai pixel baru dihitung dengan menempatkan template pada suatu titik, kemudian nilai-nilai pixel dikalikan dengan bobot dan ditambahkan sebagai nilai keseluruhan, jumlah tersebut menjadi nilai baru bagi pixel ditengah template, inilah yang menjadi pixel bagi image baru. Proses ini diulang pada semua pixel dalam gambar. Operator yang sering digunakan adalah averaging, gaussian, dan median filtering.

2.3.3. Image Segmentation

Menurut penjelasan *Shapiro dan Stockman (2001)*, image segmentation merupakan partisi atau pembagian suatu image menjadi sejumlah region (daerah) yang membangun image tersebut. Segmentation ini dilakukan dengan tujuan:

- untuk membagi image menjadi bagian-bagian untuk analisis lebih lanjut, artinya suatu lingkungan menjadi lebih terkendali sehingga proses segmentasi hanya mengambil bagian yang ingin dianalisis saja.
- melakukan perubahan representasi, pixel-pixel dari image harus disusun dalam unit yang lebih tinggi yang lebih berarti atau lebih efisien untuk analisis lebih lanjut.

Region suatu image berdasarkan *Jain, Kasturi dan Schunck (1995)* adalah kumpulan pixel-pixel yang saling berhubungan yang memiliki kesamaan sifat. Region penting untuk pengenalan suatu image karena berhubungan dengan objek tertentu pada lingkungan. Suatu image mungkin mengandung beberapa objek dan tiap objek mungkin mengandung beberapa region berhubungan dengan bagian objek. Supaya suatu image dapat diartikan dengan benar, maka harus dibagi menjadi region-region yang berhubungan dengan objek atau bagian dari objek.

Teknik teknik segmentasi dapat dibagi kedalam dua pendekatan:

1. Edge based segmentation

Pendekatan ini melakukan segmentasi dengan mencari pixel-pixel yang berada pada batas region. Pixel-pixel ini disebut edge (garis tepi), dapat ditemukan dengan melihat pixel-pixel tetangga. Karena pixel-pixel batas ada di tepi, dan region pada dua sisi dari batas mungkin memiliki nilai keabuan yang berbeda, batas region dapat ditemukan dengan mengukur perbedaan antara pixel-pixel yang bersebelahan. Dasar yang sering digunakan untuk menemukan edge adalah sifat intensitas, walaupun sifat turunan seperti tekstur dan motion juga bisa digunakan. (*Jain, Kasturi dan Schunck, 1995*) Menurut *Jain, Kasturi dan Schunck (1995)*, edge adalah perubahan lokal yang mencolok pada intensitas image, biasanya dihubungkan dengan diskontinuitas pada intensitas image atau pada turunan pertama intensitas image. Teknik-teknik yang digunakan dalam mendeteksi edge sering dibedakan berdasarkan operator dari edge detector yang digunakan. Edge detector adalah suatu algoritma yang menghasilkan sejumlah

edge dari suatu image. (Jain, Kasturi dan Schunck, 1995) Edge detector ini akan melakukan filtering pada image dengan operator tertentu. Berikut ini adalah operator-operator yang sering digunakan:

$$1. \text{ Roberts} \quad M_x = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \quad M_y = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \quad (2.4)$$

$$2. \text{ Sobel} \quad M_x = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad M_y = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \quad (2.5)$$

rewitt

$$3. \text{ Prewitt} \quad M_x = \begin{vmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{vmatrix} \quad M_y = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix} \quad (2.6)$$

aplacian

$$4. \text{ Laplacian} \quad \nabla^2 = \begin{vmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{vmatrix} \quad (2.7)$$

Keterangan:

M_x : mask untuk mendeteksi edge horizontal

M_y : mask untuk mendeteksi edge vertical

∇^2 : Laplacian mask

2. Region based segmentation

Dalam pendekatan berbasis region, semua pixel yang berhubungan dengan suatu objek dikelompokkan bersama dan ditandai untuk menyatakan ia tergabung dalam region mana. Pixel-pixel dikelompokkan dalam suatu region menggunakan kriteria tertentu yang membedakannya dari bagian lain dari image. Dua kriteria yang paling penting adalah kemiripan nilai (*value similarity*) dan kesamaan spasial (*spacial proximity*). Dua pixel bisa dikelompokkan dalam region yang sama bila memiliki sifat intensitas yang mirip (misalnya dengan nilai keabuan) atau jika dekat satu sama lain, misalnya dengan Euclidean Distance (*Jain, Kasturi dan Schunck, 1995*).

Teknik teknik yang digunakan dalam region based segmentation secara garis besar dibagi menjadi:

1. Metode thresholding

Thresholding adalah metode untuk merubah gray scale image menjadi binary image sehingga objek yang diinginkan terpisah dari latar belakangnya (*Jain, Kasturi dan Schunck, 1995*). Thresholding merupakan metode paling sederhana, dimana tiap objek atau region image dibedakan berdasarkan penyerapan cahaya atau reflektifitas konstan pada permukaannya. Suatu nilai threshold (nilai konstan brightness) dapat ditentukan untuk membedakan objek dengan latar belakangnya.

Tujuan dari thresholding adalah untuk memisahkan pixel yang mempunyai nilai keabuan (gray value) lebih tinggi dengan yang lebih rendah.

Misalnya pixel yang nilai keabuannya lebih tinggi diberi nilai biner 1 sedangkan pixel dengan nilai keabuan lebih rendah diberi nilai biner 0.

Berdasarkan penentuan nilai threshold-nya, metode thresholding dapat dibedakan menjadi metode manual dimana nilai threshold adalah tetap dan ditentukan secara manual, dan metode otomatis, dimana nilai threshold ditentukan oleh sistem secara otomatis berdasarkan pengetahuan sistem akan objek, lingkungan dan aplikasinya (misalnya karakteristik intensitas objek, ukuran objek, daerah image yang diduduki objek, jumlah jenis objek dalam image). Thresholding otomatis menganalisis penyebaran nilai keabuan dalam image dengan menggunakan histogram dan pengetahuan akan aplikasi tersebut

untuk menemukan threshold paling cocok.

Salah satu metode thresholding otomatis yang cukup terkenal adalah metode Otsu yang menentukan nilai threshold berdasarkan minimalisasi varian dalam kelas (within class variance) dari dua kelompok pixel yang terpisah. (*Jain, Kasturi dan Schunck, 1995*). Menurut *Morse (2002)*, nilai threshold diperoleh dari pencarian threshold secara berulang-ulang, setiap pengaturan threshold menyebabkan penyebaran kelas yang satu bertambah dan yang lainnya berkurang, nilai threshold yang dipilih adalah nilai yang meminimalisasi sebaran kombinasi kedua kelas. Within class variance dicari dengan rumus :

$$\sigma^2_{\text{Within}}(T) = n_B(T) \sigma^2_B(T) + n_O(T) \sigma^2_O(T) \quad (2.8)$$

Dimana,

$n_B(T)$ = jumlah pixel dari 0 sampai T

$n_O(T)$ = jumlah pixel dari T sampai N – 1

$\sigma^2_B(T)$ = varian pixel-pixel latar belakang (dibawah threshold)

$\sigma^2_O(T)$ = varian pixel-pixel objek (diatas threshold)

Dan [0, N-1] adalah tingkat intensitas.

Menghitung within class variance untuk masing-masing kemungkinan threshold cukup merepotkan, sehingga digunakan between class variance, yaitu pengurangan varian total dengan varian dalam kelas, yang dihitung dengan rumus :

$$\sigma^2_{\text{Between}}(T) = n_B(T) n_O(T) [\mu_B(T) - \mu_O(T)]^2 \quad (2.9)$$

Dimana μ adalah rata-rata kombinasi.

Sehingga untuk tiap kemungkinan T, dilakukan :

1. Pisahkan pixel ke dalam dua kelas berdasarkan nilai T
2. Cari rata-rata tiap kelas
3. Akarkan selisih kedua rata-rata
4. Kalikan jumlah pixel pada satu kelas dengan jumlah pixel kelas yang lain

Untuk meng-update between class variance saat iterasi pencarian T digunakan rumus :

$$\begin{aligned}
 n_B(T+1) &= n_B(T) + n_T \\
 n_O(T+1) &= n_O(T) + n_T \\
 \mu_B(T+1) &= \frac{\mu_B(T) n_B(T) + n_T T}{n_B(T+1)} \\
 \mu_O(T+1) &= \frac{\mu_O(T) n_O(T) + n_T T}{n_O(T+1)}
 \end{aligned} \tag{2.10}$$

Dan nilai threshold yang dipilih adalah nilai yang memaksimalkan between class variance ini (sebagai lawan dari meminimalkan within class variance).

2. Metode top down (region splitting and merging)

Region splitting

Adalah konsep segmentasi dimana region besar yang tidak uniform dipecahkan menjadi daerah yang lebih kecil yang mungkin sama/uniform. Dimulai dengan seluruh image yang direpresentasikan sebagai satu region yang biasanya tidak memenuhi kondisi kesamaan. Region image yang ada kemudian dipecah-pecah untuk memenuhi semua syarat kesamaan (homogenitas) yang diberikan.

Region merging

Adalah suatu konsep segmentasi dimana region tetangga dibandingkan dan di-merge jika memiliki hubungan property yang cukup dekat.

Splitting and merging

Untuk menggabungkan keuntungan dari kedua metode Top down diatas, maka kedua metode diatas dapat digabungkan (*Anonim, 2003*).

3. Metode bottom up (region growing)

Metode bottom up yang sering dikenal dengan region growing ini melakukan segmentasi dengan memulai dari satu bagian dari image (disebut seed region yang biasanya dari bagian kiri atas) dan kemudian region itu bertumbuh membentuk region yang lebih besar sampai image terbagi menjadi region-region.

Teknik region growing umumnya lebih baik untuk image yang banyak noise dimana edge-nya sangat susah dideteksi. Kesamaan region digunakan sebagai kriteria utama untuk segmentasi berbasis region growing. Kriteria untuk kesamaan region antara lain: tingkat keabuan, warna, tekstur, model bentuk.

2.3.4. Feature Extraction

Setelah berhasil mengelompokkan image ke dalam daerah-daerah tertentu, tahapan selanjutnya adalah feature extraction yang merupakan tahapan untuk menemukan (mengeksrak) fitur-fitur penting yang akan digunakan untuk proses pengambilan keputusan selanjutnya. Yang dimaksud dengan fitur disini adalah representasi informasi akan image di dalam tingkat yang lebih tinggi (yang memiliki arti dibanding hanya kumpulan pixel), dimana informasi ini yang nantinya akan digunakan untuk pengenalan objek atau keadaan yang terdapat pada image, sebagai tujuan dari keseluruhan proses Computer Vision seperti yang dijelaskan diatas.

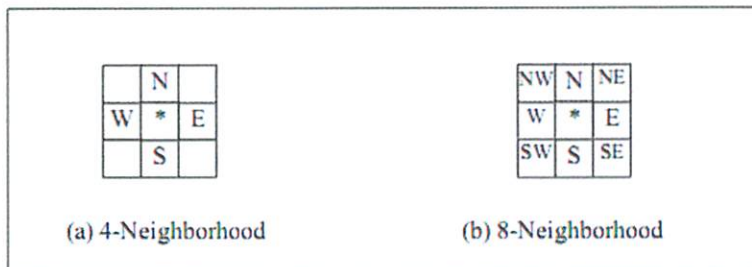
Contoh dari fitur yang akan ditemukan seperti: garis, bentuk dasar (kotak atau lingkaran), tekstur dan lain sebagainya yang merupakan sifat penting dari image tergantung keperluan masing masing aplikasi. Fitur-fitur ini juga nantinya akan direpresentasikan ke dalam bentuk data khusus untuk mempermudah proses pencarian dan pengenalan objek.

2.3.5. Operasi pada Binary Image

Seperti yang dijelaskan sebelumnya, Binary Image diperoleh dengan cara melakukan thresholding pada Grayscale Image. Pemrosesan Binary Image ini memegang peranan yang penting sebagai salah satu jenis pemrosesan dalam Computer Vision. Masing-masing tahapan proses dalam Computer Vision dapat memanfaatkan keunggulan operasi pada Binary Image. Keunggulan dari proses pada Binary Image adalah algoritma yang mudah dimengerti, lebih cepat dari pemrosesan pada gambar grayscale atau warna, dan memerlukan memory dan pemrosesan yang lebih sedikit. Beberapa teori yang berhubungan dengan proses pada Binary Image antara lain :

1. Pixel dan Neighborhood

Pixel-pixel pada binary image B adalah 0 dan 1, 1 untuk menyatakan objek dan 0 untuk menyatakan latar belakang. $B[r,c]$ menyatakan nilai pixel yang terletak pada baris ke r dan kolom ke c . banyak algoritma tidak hanya menggunakan satu pixel saja dalam melakukan pemrosesan, tetapi juga menggunakan pixel-pixel tetangganya (neighbor pixel). Dua definisi untuk neighbor yang paling umum adalah 4-neighbor dan 8-neighbors. (Gambar 2.9)



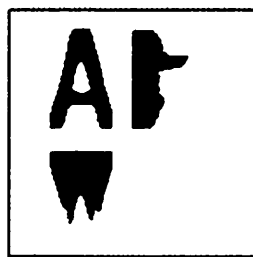
Gambar 2.9 Dua hubungan antar pixel yang sering digunakan.

2. Connected Component Labeling

Menurut *Shapiro dan Stockman (2001)*, connected component dengan nilai v adalah sejumlah pixel C , yang memiliki nilai v , dan tiap pasangan pixel di dalamnya saling berhubungan. Maka connected component labeling adalah proses memberikan label pada binary image B menghasilkan suatu labeled image LB dimana nilai tiap pixelnya adalah nilai label dari connected component-nya.

3. Proyeksi

Proyeksi dari binary image menjadi garis dapat dilakukan dengan cara membagi garis itu menjadi kolom-kolom dan menghitung jumlah pixel 1 yang ada pada tiap tiap kolom. Proyeksi merupakan representasi image yang lebih padat dimana masih banyak informasi image yang dapat diperoleh. Akan tetapi proyeksi tidak unik dimana image yang berbeda bisa memiliki proyeksi yang sama. Proyeksi bisa dilakukan secara horizontal, vertikal dan diagonal dengan menghitung jumlah pixel 1 pada tiap kolom secara horizontal, vertikal atau diagonal.



Gambar 2.10 Proyeksi Binary Image.

4. Operasi Morfologi (Morphology Operation)

Pemrosesan image secara morfologi merupakan sekumpulan teknik untuk mengolah digital image yang berdasarkan pada morfologi matematika. Morfologi matematika itu sendiri adalah model teoritis untuk digital image yang dibangun dari teori lattice dan topologi. Morfologi itu sendiri memiliki arti bentuk atau struktur.

Menurut *Shapiro dan Stockman (2001)*, operasi morfologi pada Binary Image mengambil dua input yaitu Binary Image B dan Structuring Element (SE) S , yang merupakan Binary Image dengan ukuran lebih kecil, dengan bentuk tertentu, dimana salah satu pixelnya ditetapkan sebagai pusat (biasanya titik tengah). SE ini digunakan untuk mengubah Binary Image. Dengan menggunakan titik pusat nya SE bisa diletakkan di manapun dalam image dan bisa digunakan untuk memperbesar suatu region atau untuk menguji apakah suatu bentuk cocok dalam region tersebut.

Operasi operasi dasar pada proses morfologi adalah :

1. Dilation

Dilation merupakan operasi yang akan memperbesar suatu daerah. Dilation suatu binary image B dengan SE S dapat dinyatakan sebagai berikut:

$$B \oplus S = \bigcup_{b \in B} S_b \quad (2.11)$$

Proses ini dilakukan dengan melewati S pada image B, setiap bertemu dengan pixel 1, seluruh SE di OR kan dengan image output yang awalnya diset seluruh pixelnya bernilai nol.

2. Erosion

Erosion merupakan operasi yang memperkecil suatu daerah. Erosion suatu binary image B dengan SE S dapat dinyatakan sebagai berikut:

$$B \ominus S = \{b \mid b + s \in \forall s \in S\} \quad (2.12)$$

Proses ini juga melewati SE pada seluruh image. Pada tiap posisi dimana semua pixel 1 pada SE menutupi pixel 1 pada B, pixel binary image pada posisi titik pusat SE di-OR-kan dengan image output.

3. Opening

Operasi opening dapat menghilangkan bagian kecil dari region yang keluar dari batasnya, atau untuk menghilangkan region dengan ukuran tertentu. Opening merupakan operasi yang melakukan erosion dan dilation secara berurutan dan dinyatakan dengan:

$$B \circ S = (B \ominus S) \oplus S \quad (2.13)$$

4. Closing

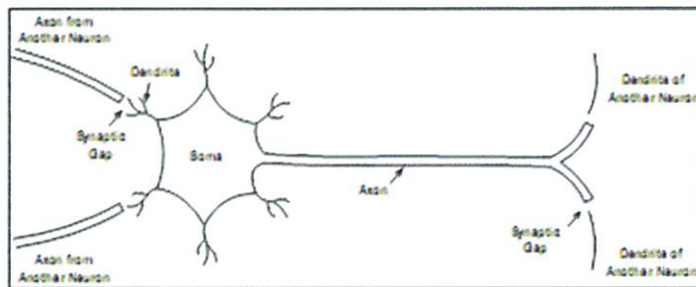
Closing dapat menutup lubang dalam region dan menghilangkan celah pada region. Closing merupakan operasi yang melakukan dilation dan erosion secara berurutan dan dinyatakan dengan:

$$B \bullet S = (B \oplus S) \ominus S \quad (2.14)$$

2.4. Artificial Neural Network

2.4.1. Analogi Neural Network Biologis

Pada jaringan saraf biologis, setiap neuron (sel saraf) menerima sinyal-sinyal dari neuron lain melalui sambungan yang disebut sinapsis. Sebagian sinyal input cenderung menyebabkan neuron tereksitasi (excitation), sementara sebagian lainnya mengalami hambatan (inhibition). Ketika efek kumulatif dari sinyal tersebut melewati suatu nilai batas (threshold), neuron yang bersangkutan akan menembakkan sinyal ke neuron lainnya.



Gambar 2.11 Sel Saraf Biologis.

2.4.2. Definisi Artificial Neural Network

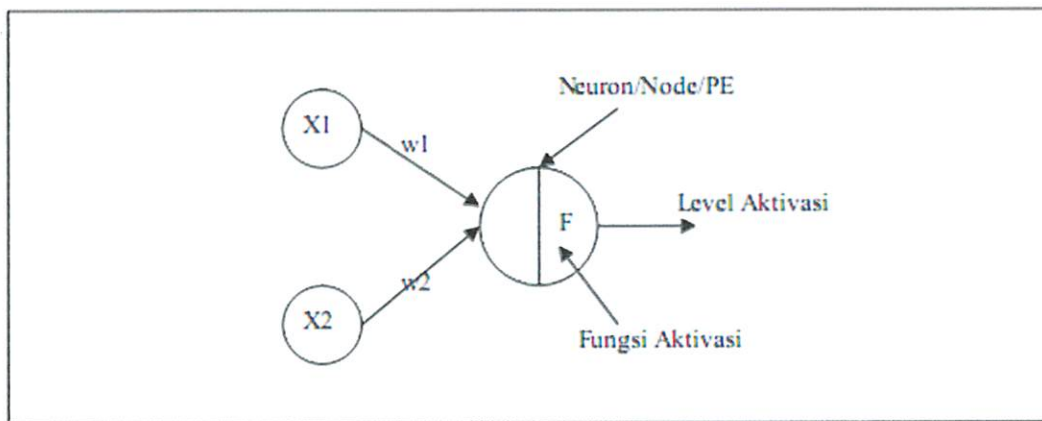
Menurut *Fausett (1994)*, Artificial Neural Network (disingkat ANN, atau dalam bahasa Indonesia disebut Jaringan Saraf Buatan adalah suatu sistem pengolahan informasi yang memiliki karakteristik tertentu yang mirip dengan jaringan saraf biologi. Jaringan saraf buatan telah dikembangkan sebagai generalisasi dari model-model matematika dari kesadaran manusia atau saraf biologi, berdasarkan anggapan bahwa:

1. Pengolahan informasi terjadi pada unsur-unsur sederhana yang dinamakan neuron.
2. Sinyal-sinyal melewati neuron-neuron dalam suatu sambungan.
3. Setiap sambungan memiliki bobot, dimana dalam suatu jaringan saraf, akan dikalikan dengan sinyal yang dikirimkan.
4. Setiap saraf menggunakan sebuah fungsi aktivasi (biasanya non-linear) untuk net inputnya (jumlah dari sinyal masukan dikalikan bobot) untuk menentukan sinyal output.

Sebuah jaringan saraf terdiri atas sejumlah besar unsur-unsur pengolahan sederhana yang dinamakan neuron, unit, cell, atau node. Setiap neuron

terhubung ke neuron-neuron lain sebagai jalur bagi hubungan komunikasi yang terarah dengan bobotnya masing-masing. Bobot tersebut mewakili informasi yang sedang digunakan untuk menyelesaikan suatu masalah. Jaringan saraf dapat diterapkan pada berbagai jenis masalah, seperti menyimpan dan memanggil data atau pola, pengelompokan pola-pola yang sejenis, atau menemukan solusi untuk masalah-masalah tertentu.

Setiap neuron memiliki keadaan internal, yang disebut level aktivasi (activation level), yang merupakan fungsi dari masukan yang diterima.



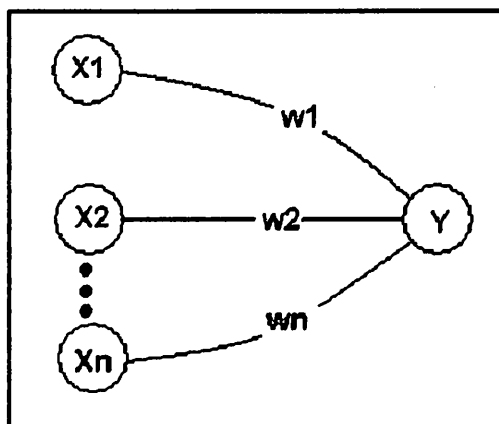
Gambar 2.12 Processing Element / Neuron pada Artificial Neural Network.

Karakteristik penting lain yang ditiru Neural Network dari neuron biologis adalah sifat fault tolerance, dimana kita dapat mengenali sinyal input yang agak berbeda dan mampu bekerja dengan sedikit kerusakan pada sistemnya sendiri.

2.4.3. Cara Kerja Artificial Neural Network

Cara kerja suatu neural network dapat digambarkan sebagai berikut. Setiap neuron menerima satu set input, kemudian setiap input dikalikan dengan sebuah bobot (weight) yang berasosiasi dengan kekuatan sinapsis seperti yang disebutkan diatas. Jumlah seluruh input berbobot tersebut menentukan apakah kemungkinan suatu neuron untuk menembakkan sinyal. Nilai ini yang disebut level aktivasi (activation level). Secara matematis untuk suatu neuron, setiap input x_i yang dimodulasikan oleh sebuah bobot w_i sehingga jumlah total input adalah $\sum x_i.w_i$. Cara perhitungan yang dimaksud dapat digambarkan sebagai berikut:

$$Y = X_1 W_1 + X_2 W_2 + \dots + X_n W_n \quad (2.15)$$



Gambar 2.13 ANN Sederhana.

Total sinyal tersebut kemudian diproses oleh sebuah fungsi aktivasi untuk menghasilkan sinyal output, yang jika tidak bernilai 0 akan ditransmisikan sebagai keluaran.

2.4.4. Karakteristik Artificial Neural Network

Dengan demikian menurut *Fausett (1994)* sebuah jaringan saraf dikategorikan berdasarkan :

1. Pola hubungan antar saraf (dinamakan arsitektur atau topology jaringan).

Arsitektur ANN dapat dibagi menjadi Single Layer Network, Multi Layer Network dan Competitive Layer. BAM termasuk dalam kategori Single Layer.

2. Metode untuk menentukan bobot dalam suatu hubungan (dinamakan algoritma training atau learning). Metode penentuan bobot ini terdiri dari:

- a. Supervised

Merupakan metode training yang dilakukan dengan memberikan sejumlah vektor atau pola training, masing-masing dengan vektor output targetnya. Bobot jaringan kemudian diatur berdasarkan algoritma learning tertentu, dan biasanya digunakan untuk klasifikasi pola dan asosiasi pola seperti yang digunakan dalam BAM.

b. Unsupervised

Merupakan metode learning dimana ANN mampu mengatur bobotnya sendiri dari sejumlah training data yang diberikan. Biasanya banyak digunakan dalam pengelompokan (clustering).

c. Fixed Weight.

Jaringan ini menggunakan metode bobot tetap dimana bobot network ditentukan dari awal tanpa menggunakan pelatihan untuk memenuhi constraint (batasan) tertentu.

3. Fungsi aktivasi yang dimiliki.

Beberapa fungsi aktivasi yang banyak digunakan antara lain :

a. Fungsi identitas, yaitu fungsi yang menghasilkan nilai berupa dirinya sendiri dan dihitung dengan :

$f(x) = x$, untuk semua x

b. Fungsi tangga biner, yaitu fungsi yang merubah variabel kontinyu menjadi output bernilai biner (0 atau 1) dan dihitung dengan :

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \theta \\ 0, & \text{jika } x < \theta \end{cases} \quad (2.16)$$

Dimana θ adalah suatu nilai threshold.

c. Fungsi tangga bipolar, yaitu fungsi yang merubah variabel kontinyu menjadi output bernilai bipolar (-1 atau 1) dan dihitung dengan :

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \theta \\ -1, & \text{jika } x < \theta \end{cases} \quad (2.17)$$

d. Fungsi sigmoid biner, yaitu fungsi yang menghasilkan output pada interval antara 0 sampai 1 dan dihitung dengan :

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \quad (2.18)$$

$$f'(x) = \sigma f(x) [1 - f(x)]$$

Dimana σ merupakan tingkat kecuraman yang diberikan.

e. Fungsi sigmoid bipolar, yaitu fungsi yang menghasilkan output pada interval yang bisa diskalakan sesuai aplikasi, biasanya antara -1 sampai 1 dan dihitung dengan :

$$g(x) = 2f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)} - 1 \quad (2.19)$$

$$= \frac{(1 - \exp(-\sigma x))}{(1 + \exp(-\sigma x))}$$

$$g'(x) = \frac{\sigma}{2} [1 + g(x)] [1 - g(x)]$$

Dimana σ merupakan tingkat kecuraman yang diberikan, dan $f(x)$ adalah fungsi sigmoid biner.

2.4.5. Bidang Aplikasi Artificial Neural Network

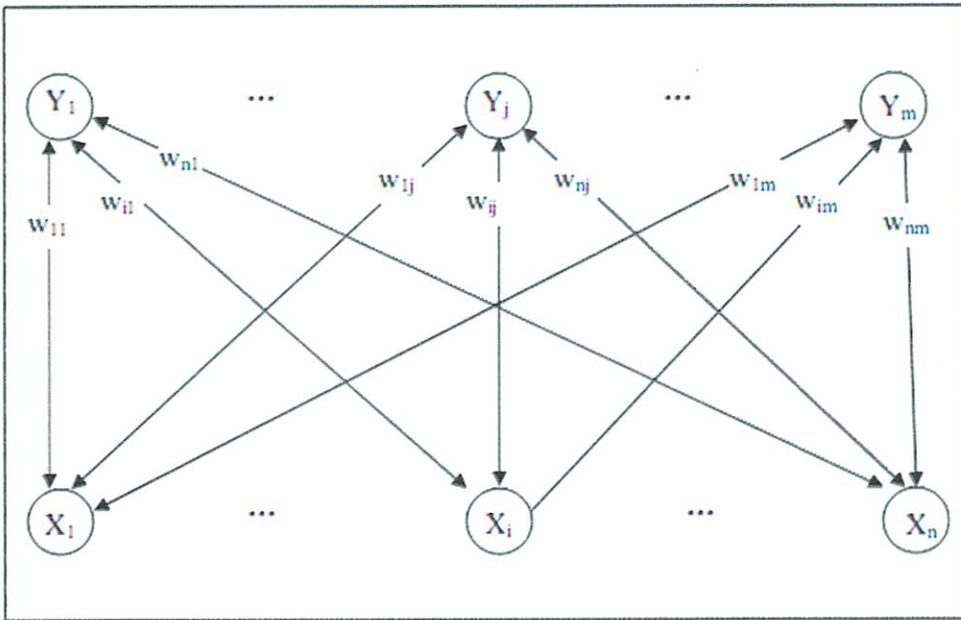
Menurut *Fausett (1994)* beberapa bidang aplikasi dimana Neural Network dapat digunakan antara lain :

1. Signal Processing (pengolahan sinyal)
2. Control
3. Pattern Recognition (pengenalan pola)
4. Medicine (kedokteran)
5. Speech Production (pembuatan kata)
6. Speech Recognition (pengenalan kata)
7. Business

2.5. Bidirectional Associative Memory (BAM)

Bidirectional associative memory merupakan salah satu versi dari heteroassociative recurrent neural network yang dikembangkan oleh *Kosko(1998)*. Dimana heteroassociative recurrent neural network itu sendiri menurut penjelasan *Fausett (1994)* adalah jaringan dimana bobotnya ditentukan dengan suatu cara sehingga jaringan itu dapat menyimpan satu set P pola asosiasi. Dengan masing-masing asosiasi adalah pasangan vektor ($s(p)$, $t(p)$) dimana $p = 1, 2, \dots, P$. Tiap vektor $s(p)$ punya n komponen dan tiap $t(p)$ punya m komponen.

BAM menyimpan sekumpulan pola yang berasosiasi dengan menjumlahkan matriks korelasi bipolar (sebuah matriks yang berukuran $n \times m$ untuk setiap pola yang akan disimpan). Arsitektur dari jaringan tersebut terdiri atas 2 lapisan neuron, yang dihubungkan oleh jalur hubungan yang memiliki bobot dan arah. Jaringan tersebut mengirimkan sinyal secara bolak balik di antara 2 lapisan tersebut sampai semua neuron mencapai keseimbangan (sampai setiap aktivasi neuron mencapai nilai yang tetap). Jaringan saraf BAM dapat merespon terhadap input yang berada di lapisan manapun.



Gambar 2.14 Bidirectional Associative Memory.

2.5.1. Konsep dasar BAM

Berikut adalah tinjauan konsep dasar BAM. Misalkan m pasangan pola $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ akan disimpan dalam BAM, dimana $X_i = (x_{i1}, x_{i2}, \dots, x_{in}) \in \{-1, 1\}^n$ dan $Y_i = (y_{i1}, y_{i2}, \dots, y_{im}) \in \{-1, 1\}^m$ adalah dua pola yang saling berasosiasi, maka m pola tersebut dapat disimpan dalam BAM dalam bentuk matrix bobot W berukuran $n \times m$ yang disusun sebagai berikut :

$$W = \sum_{i=1}^m X_i^T Y_i$$

Operasi retrieve / recall adalah sebagai berikut :

$$X_i = \Theta(Y_i W^T)$$

$$Y_i = \Theta(X_i W)$$

Dimana Θ merupakan fungsi threshold untuk X_{ik} dan Y_{jk} , fungsi threshold menggunakan fungsi tangga bipolar adalah sebagai berikut :

$$X_{ik} = \begin{cases} 1, & \text{jika } Y_i W^T > 0 \\ -1, & \text{jika } Y_i W^T \leq 0 \end{cases} \quad (2.20)$$

$$Y_{jk} = \begin{cases} 1, & \text{jika } X_i W > 0 \\ -1, & \text{jika } X_i W \leq 0 \end{cases} \quad (2.21)$$

2.5.2. Stabilitas BAM

Stabilitas BAM didefinisikan Kosko (1988) dengan mengidentifikasi fungsi Lyapunov atau fungsi energi E untuk setiap state (A_i, B_i) . Prosedur yang digunakan Kosko berdasarkan atas pendekatan dari analisis kestabilan sistem dinamik yang dikembangkan oleh A.M Lyapunov. Yang unik dari pendekatan Lyapunov ini adalah hanya bentuk persamaan differensialnya yang perlu diketahui bukan nilai hasilnya.

Fungsi Lyapunov E memetakan variabel-variabel system ke suatu bilangan real dan menurun sejalan dengan waktu. Pada BAM, E memetakan product matriks – matriks ke dalam bilangan real yang dirumuskan sebagai berikut :

$$E(A, B) = -AWBT \quad (2.22)$$

Model BAM menggunakan interlayer feedback, dengan melewati data melalui W didapatkan satu arah dan melalui tranposenya W^T didapatkan arah yang lainnya. Jika pasangan pattern (A, B) dilewatkan dalam BAM, maka akan dilakukan beberapa kali iterasi yang dapat digambarkan sebagai berikut :

$$A \rightarrow W \rightarrow B \quad (2.23)$$

$$\begin{array}{l} B \rightarrow W^T \rightarrow A' \\ A' \rightarrow W \rightarrow B' \\ B' \rightarrow W^T \rightarrow A'' \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ \cdot \quad \cdot \quad \cdot \\ A_f \rightarrow W \rightarrow Y_f \\ Y_f \rightarrow W^T \rightarrow A_f \end{array}$$

Setelah beberapa kali iterasi, (A,B) akan terkonvergensi secara tetap menjadi (Af,Bf) dengan nilai energi minimum. Hal inilah yang menjadi keunggulan BAM, yaitu bersifat stabil sehingga dapat mengenali proses yang tidak sempurna.

2.5.3. Hamming Distance

Hamming Distance (HD) dari dua vektor yang panjangnya sama adalah bilangan scalar yang menunjukkan perbedaan antara dua vektor tersebut. Misalnya vektor $X = (x_1, x_2, \dots, x_n)$ dan vektor $Y = (y_1, y_2, \dots, y_n)$. Jika x_1 dan y_1 merupakan nilai Boolean atau $x_1, y_1 \in \{0,1\}$ maka HD dapat diekspresikan dalam operasi logika sebagai berikut :

$$H(x,y) = \text{bitcount} \{ (\sim x_i \wedge y_i) \vee (x_i \wedge \sim y_i) / i = 1,2,3, \dots, n \} \tag{2.24}$$

Dimana $\wedge$ adalah operator logika AND dan $\vee$ adalah OR, $\sim$ adalah negasi (ingkaran). Fungsi bitcount {S} untuk mendapatkan jumlah / banyaknya elemen S yang bernilai 1.

Jika x_i dan y_i merupakan bentuk bipolar atau $x_i, y_i \in \{-1,1\}$ maka HD dapat dicari dengan cara berikut :

$$H(X,Y) = (n - XYT) / 2 \tag{2.25}$$

Dimana n adalah panjang vektor dan XYT adalah perkalian antara matriks X dengan matriks Y yang menghasilkan bilangan skalar.

Contoh :

$$\begin{array}{l} X = (1 \quad -1 \quad 1 \quad 1 \quad 1 \quad 1) \\ Y = (1 \quad 1 \quad -1 \quad 1 \quad 1 \quad 1) \end{array}$$

Panjang vektor : $n = 6$

Perkalian scalar antara matriks X dengan Y :

$$XY^T = \begin{pmatrix} 1 & -1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = 1 - 1 - 1 - 1 + 1 + 1 \tag{2.26}$$

$= 0$

Jadi, HD antara X dan Y : $H(X,Y) = (6-0) / 2 = 3$

Dengan kata lain, HD menunjukkan banyaknya bit yang berbeda pada posisi yang sama.

2.5.4. Kapasitas Penyimpanan BAM

Kapasitas berkaitan langsung dengan unjuk kerja. Sebagai memori, idealnya BAM memiliki kapasitas sebesar mungkin yaitu kemampuan untuk menyimpan dan memanggilnya kembali, Tetapi inilah yang menjadi kelemahan BAM yaitu kapasitas yang sangat rendah. Misalkan S sebagai himpunan dari m pasangan pola yaitu $S = \{(X_1,Y_1), (X_2,Y_2), \dots, (X_m,Y_m)\}$. X berdimensi n dan Y berdimensi p. S tersimpan dalam BAM W. Kapasitas BAM yang dilambangkan dengan C adalah jumlah maksimum pasangan pola (X_i,Y_i) yang dapat disimpan dan dipanggil dengan sempurna.

Menurut Kosko (1988), kapasitas BAM adalah $C < \min (n,p)$. Kemudian peneliti-peneliti lain memberikan estimasi yang lebih kecil, seperti :

- 1. Haines (1988), $C = R / (2 \text{ Log } R)$ dimana $R = \min(n,p)$
- 2. Wang (1991), $C \leq \sqrt{\min(n,p)}$; dan
- 3. Zhang (1991), $C \leq \min((n + \alpha) / \alpha, (p + \beta) / \beta)$ dimana α dan β masing-masing merupakan nilai maksimum ‘korelasi’ antara X_i dan Y_i .

2.5.5. Sifat Penyimpanan Dan Pemanggilan

Secara umum telah diketahui bahwa representasi data dalam bentuk bipolar dalam BAM lebih sesuai dibandingkan dengan dalam bentuk biner.

Dalam representasi biner, kemungkinan suatu pola dapat dipanggil dengan sempurna sangat kecil.

Semakin banyak pola disimpan dalam memory, semakin kecil peluang dapat dipanggil dengan sempurna (Hebbian Learning). Dalam representasi bipolar, agak sedikit lebih baik kondisinya. Suatu pola yang tersimpan dalam memori misalkan X_i akan dipengaruhi oleh sisa pasangan pola lainnya yaitu $Y_i Y_j^T$ yang biasa disebut Cross-Talks (Kohonen 1972)

Nilai-nilai Cross-Talks bisa positif atau negatif sebagai penguat atau pelemah. Jumlah nilai-nilai tersebut menggambarkan interaksi yang kuat antara pola-pola yang berasosiasi.

Hasil pemanggilan sempurna untuk pola-pola Orthogonal dua buah pola yang bersifat orthogonal dapat dipandang sebagai 2 buah vektor yang bersifat orthogonal. Dalam representasi bipolar, ke-orthogonalan dua buah pola x_1 dan x_2 dapat diekspresikan juga dalam bentuk Hamming Distance, yaitu :

$$x_1 x_2^T = 0, \text{ jika dan hanya jika } H(x_1, x_2) = n / 2 \quad (2.27)$$

dimana n adalah dimensi dari pola x_i . Perhatikan operasi retrieval berikut :

$$\begin{aligned} y_i &= \theta(X_i W) \\ &= \theta\left(X_i \left(\sum_{j=1}^m X_j^T Y_j\right)\right) \\ &= \theta\left(\left(X_i^T Y_i\right) Y_i + \left(\sum_{j \neq i} X_j^T Y_j\right) Y_j\right) \\ &= \theta\left(n Y_i + \left(\sum_{j \neq i} X_i X_j^T Y_j\right)\right), \text{ jika semua orthogonal} = 0 \\ y_i &= \theta(n Y_i) \end{aligned} \quad (2.28)$$

2.5.6. Pola Orthogonal

Pada berbagai aplikasi yang menggunakan BAM, kondisi pola-pola input dan output yang saling berasosiasi pada umumnya tidak saling orthogonal. Namun demikian, sebaliknya terdapat kemungkinan berbagai aplikasi untuk dapat memanfaatkan sifat-sifat orthogonal dari pola-pola input maupun output. Untuk itu, akan dipelajari lebih lanjut cara membentuk pola-pola yang saling orthogonal.

Pada kasus khusus, dapat di-generate himpunan pola bipolar – orthogonal yang berbasis 4 building blocks pola bipolar orthogonal sebagai berikut :

$$\begin{aligned}
 Z_A^4 = \begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix} & \quad Z_A^{e4} = \begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \\
 Z_B^4 = \begin{pmatrix} -1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 \end{pmatrix} & \quad Z_B^{e4} = \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{pmatrix}
 \end{aligned} \tag{2.29}$$

Dengan menggunakan 4 building blocks tersebut, dapat dibentuk himpunan pola bipolar orthogonal yang lebih besar dengan ukuran $2i$, dimana i adalah bilangan bulat positif secara recursive :

$$\begin{aligned}
 Z_A^c Z_A^c & \quad Z_A^c Z_B^c & \quad Z_B^c Z_B^c \\
 Z_A Z_A & \quad Z_A Z_B & \quad Z_B Z_B
 \end{aligned} \tag{2.30}$$

2.5.7. Peningkatan Kemampuan BAM

Menurut *Wang et al. (1990)*, salah satu kemampuan utama BAM adalah kemampuan untuk me-recall pola yang disimpan tanpa dipengaruhi oleh noise, tetapi BAM memiliki kapasitas yang sangat rendah. Untuk meningkatkan kemampuan BAM di gunakan dua strategi yaitu multiple training dan dummy augmentation.

2.5.7.1. Multiple Training Encoding

Adalah pelatihan BAM dengan memasukkan pasangan pola training (X-Y) sebanyak lebih dari sekali untuk menjamin salah satu pola selalu berhasil di-recall. Rumus untuk menghitung W adalah :

$$W = \sum_{i=1}^m Q_i X_i^T Y_i \tag{2.31}$$

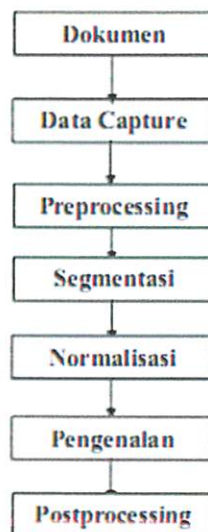
Dimana Q adalah vektor yang menyatakan jumlah suatu pasangan pola akan dilatih.

2.5.7.2. Dummy Augmentation Encoding

Merupakan metode melatih BAM dengan menambahkan vektor-vektor dummy pada setiap pola yang akan dilatih. Vektor-vektor dummy ini merupakan vektor yang bersifat strictly noise free, artinya dapat direcall dengan sempurna atau bersifat orthogonal. *Wang (1990)* menyatakan bahwa metode ini menjamin semua pola dapat di-recall.

2.6. Optical Character Recognition

Optical character recognition (OCR) adalah sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari sebuah pencetak (printer atau mesin ketik) maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (image character) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan dalam database aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar output scanner dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan. Secara umum blok diagram kerja OCR dapat dilihat pada gambar 2.15.



Gambar 2.15. Blok diagram kerja OCR

Dalam sistem OCR, input yang dimasukkan adalah gambar yang berisi karakter yang ingin dikenali. Sebelum karakter pada gambar dapat dikenali, gambar tersebut harus terlebih dahulu dilakukan preprocessing. Tujuan dari proses ini ialah menghilangkan bagian-bagian dari gambar yang tidak diinginkan (bukan bagian dari karakter yang ingin dikenali) dan untuk memperbaiki kualitas gambar sehingga objek pada gambar lebih mudah untuk dikenali. Preprocessing pada gambar biasanya meliputi grayscaling, noise removal, dan thresholding. Grayscale merupakan tahap awal dalam image preprocessing, yaitu mengubah gambar berwarna menjadi gambar yang hanya memiliki derajat keabuan saja. Selanjutnya dilakukan proses noise filtering, yaitu proses mereduksi atau mengurangi noise. Proses akhir dari image preprocessing adalah thresholding, yaitu suatu proses untuk memisahkan background dengan objek yang ingin diamati dengan mengubah gambar menjadi hitam putih.

Setelah preprocessing selesai dilakukan, maka tahap selanjutnya ialah segmentasi. Proses ini digunakan untuk memisahkan area-area pengamatan dari setiap karakter yang ingin dikenali, seperti pemisahan kalimat ke dalam kata-kata dan pemisahan kata ke dalam karakter-karakter. Tahap selanjutnya ialah normalisasi karakter hasil segmentasi, yaitu proses untuk mengubah dimensi region dari setiap karakter, seperti ketebalan karakter. Hal yang biasa dilakukan untuk proses normalisasi pada OCR adalah scaling dan thinning. Setelah normalisasi dilakukan, kemudian akan dilakukan ekstraksi fitur untuk mendapatkan karakteristik dari masing-masing karakter yang membedakannya dengan karakter lain. Tahap akhir dari proses OCR adalah pengenalan atau recognising, dalam tahap ini algoritma akan membandingkan ciri-ciri fitur yang ingin dikenali dengan data yang tersimpan sebelumnya. Hasil pengenalan berupa teks dengan kemiripan paling besar antara fitur karakter yang ingin dikenali dengan informasi yang tersimpan.

2.7. Automatic Number Plate Recognition

2.7.1. Definisi

Automatic Number Plate Recognition (ANPR), atau dalam bahasa Indonesianya pengenalan plat nomor otomatis, merupakan suatu metode pengamatan masal yang menggunakan optical character recognition (OCR) pada image untuk mengenali plat nomor pada kendaraan bermotor. ANPR dapat digunakan untuk menyimpan gambar yang diambil oleh kamera dan teks dari plat nomornya, bahkan dengan kemampuan menyimpan foto pengemudinya. Teknologi ANPR umumnya bersifat khusus untuk masing-masing daerah yang variasi plat nomornya berbeda-beda. Seiring dengan perkembangannya, sistem ini semakin bertambah akurat dan dapat dipercaya (reliable).

Bagian perangkat lunak dari sistem ini berjalan pada PC standar dan bisa digabungkan dengan aplikasi-aplikasi atau database lain. Menggunakan sejumlah teknik manipulasi image untuk mendeteksi, normalisasi, dan meningkatkan kualitas image plat nomor, dan terakhir menggunakan OCR untuk mengenali huruf dan angka (alphanumeric) dari plat nomor tersebut.

Sedangkan menurut *Hofman (2004)*, ANPR adalah teknologi image processing yang digunakan untuk mengidentifikasi kendaraan bermotor melalui plat nomornya.

Definisi dari OCR atau pengenalan karakter sendiri adalah bidang aplikasi dari pengenalan pola yang melibatkan representasi gambar garis dua dimensi dari sebuah karakter (*Fairhurst, 1988*). Pengenalan karakter adalah proses pengubahan image yang telah di-scan (scanned image) baik berupa image yang dicetak oleh mesin atau image berupa tulisan tangan menjadi format yang dapat diproses oleh komputer (sebagai contoh dalam format ASCII). Isi dari sebuah image dapat terdiri dari karakter alphabet (a,b,c,...), karakter numerik (0,1,2,...), karakter khusus (\$,%,&,...), atau obyek lain yang tidak dapat diidentifikasi.

2.7.2. Nama Lain

ANPR sering juga dikenal sebagai:

1. Automatic Vehicle Identification (AVI)
2. Car Plate Recognition (CPR)
3. Licence Plate Recognition (LPR)
4. Car Plate Reader (CPR)

2.7.3. Komponen Sistem

Menurut *Hofman (2004)*, suatu sistem ANPR pada umumnya terdiri dari :

1. Camera – yang mengambil image mobil (dari depan atau belakang)
2. Illumination – cahaya yang terkendali yang dapat menerangi plat, dan memungkinkan operasi pada siang dan malam hari. Umumnya illumination menggunakan sinar infra merah yang tidak kelihatan oleh pengemudi.
3. Frame grabber – suatu interface antara kamera dan PC, yang memungkinkan software membaca informasi image.
4. Computer – biasanya sebuah PC yang berjalan di Windows atau Linux. PC ini menjalankan aplikasi ANPR yang mengendalikan sistem, membaca image-image, menganalisis dan mengidentifikasi plat, serta menghubungkan dengan aplikasi atau sistem lainnya.
5. Software – bagian aplikasi dan paket pengenalan, biasanya paket pengenalan adalah dalam bentuk DLL (Dynamic Link Library).
6. Hardware – berbagai macam bagian input/output yang menghubungkan dengan dunia luar (seperti bagian kontrol atau jaringan).
7. Database – event yang terjadi akan disimpan dalam database lokal atau dikirimkan ke jaringan. Data yang disimpan adalah hasil akhir pengenalan (berupa teks dalam format ASCII), dan bisa juga image kendaraan dan wajah pengemudi.

2.7.4. Algoritma Software

Ada lima algoritma utama yang diperlukan oleh perangkat lunak untuk mengenali plat nomor :

1. Lokalisasi plat, digunakan untuk menemukan dan memisahkan lokasi plat nomor di gambar.

2. Orientasi dan penyekalaan plat, untuk mengubah ukuran dan sudut plat ada ukuran yang sesuai.
3. Normalisasi, untuk mengatur brightness dan contrast dari image.
4. Segmentasi karakter, memisahkan karakter pada plat nomor.
5. Optical Character Recognition (OCR), yaitu pengenalan karakter plat nomor

Kerumitan dari masing masing sub-program ini akan mempengaruhi ketepatan sistem.

2.7.5. Cara Kerja ANPR

Cara kerja ANPR pada umumnya di suatu sistem kontrol akses menurut *Hofman (2004)* dapat dijelaskan sebagai berikut :

- Kendaraan mendekati suatu secured area, dan memulai proses dengan menyentuh suatu detektor magnetik (yang merupakan sensor kendaraan yang sering digunakan). Detektor mendeteksi kendaraan tersebut dan memberi sinyal pada unit ANPR.
- Kemudian unit ANPR akan mengaktifkan illumination dan mengambil gambar depan atau belakang dengan kamera. Image kendaraan yang diambil, termasuk plat dan informasi pixel-nya dibaca oleh frame grabber.
- Unit ANPR akan menganalisis image yang diterimanya menggunakan algoritma perangkat lunak image processing yang berbeda, meningkatkan kualitas image, melacak posisi plat, memotong karakter-karakternya dan mengidentifikasi karakter tersebut menggunakan metode kecerdasan buatan tertentu (seperti Neural Network). Kebanyakan unit ANPR berdasarkan aplikasi yang berjalan pada PC dibawah Windows, sementara sistem lain menggunakan platform sendiri tanpa perlu suatu PC.
- Sistem akan memeriksa apakah kendaraan tersebut memiliki wewenang untuk memasuki daerah ini, dari daftar kendaraan yang berhak dalam database. Jika ditemukan maka sistem akan memberi sinyal pada gerbang untuk buka.
- Kendaraan yang berhak akan masuk ke dalam area, setelah masuk gerbang akan tertutup dan menunggu kendaraan selanjutnya. Aplikasi lain akan

menggunakan informasi yang diperoleh dari image, untuk tujuan yang berbeda. Misalnya untuk penilangan pelanggaran kecepatan atau lampu merah. Semua aplikasi ini tergantung pada proses pengenalan image secara otomatis yang dilakukan oleh unit ANPR, yang meniru cara kerja otak manusia.

2.7.6. Keandalan / Performa ANPR

Awalnya sistem ANPR mengalami banyak masalah pada tingkat pengenalan yang rendah, lebih rendah dari yang diperlukan sistem berjalan. Hal ini diakibatkan efek luar (cahaya yang berlebihan, plat yang buruk, tipe plat yang bermacam-macam) dan software pengenal serta hardware penglihatan yang terbatas menghasilkan sistem dengan kualitas sangat rendah.

Akan tetapi perkembangan software dan hardware saat ini menyebabkan sistem ANPR lebih reliable dan lebih meluas. Makin banyak sistem yang telah digunakan di dunia nyata dan mengotomatisasi berbagai masalah. Penggunaan yang umum ini karena pengenalan oleh sistem tidak selalu mutlak, aplikasi yang bergantung pada hasil dari sistem ini dapat mentolerir kesalahan yang ditimbulkan sehingga menghasilkan suatu sistem yang kelihatannya tanpa kesalahan (*virtually flawless system*). Misalnya dalam aplikasi perparkiran yang membandingkan hasil pengenalan saat masuk dan keluar untuk menghitung waktu, mengijinkan adanya sedikit kesalahan untuk tetap berjalan dengan baik. Hal inilah yang memungkinkan penanganan kesalahan dari suatu ANPR untuk tetap memberikan suatu sistem yang dapat dipercaya dan terotomatisasi sepenuhnya.

Menurut *Nelson (1999)*, pengujian kemampuan suatu sistem ANPR dapat dilakukan dengan menguji kemampuan engine pengenalannya pada suatu kelompok karakter pada situasi yang hampir ideal. Tetapi sangat penting juga untuk menentukan faktor-faktor yang dapat mempengaruhi operasi sistem, dan mempelajari efek variabel-variabel yang tidak konstan, yang meliputi :

1. Kecepatan kendaraan
2. Banyaknya kendaraan yang lewat
3. Iluminasi (siang, malam, matahari, bayangan)
4. Cuaca

5. Tipe Kendaraan
6. Penempatan plat
7. Variasi plat
8. Format plat
9. Jarak kamera ke plat
10. Kemiringan plat
11. Jarak antar kendaraan.
12. Kehadiran objek-objek lain pada kendaraan selain plat.

Aplikasi ANPR memiliki kebutuhan yang berbeda-beda. Ada yang langsung memakai informasi plat yang diperoleh, sedangkan yang lain memerlukan Input / Output tambahan, seperti hubungan dengan pengendali gerbang, database, atau fasilitas pengambilan jarak jauh. Dalam kasus-kasus ini perlu dipertimbangkan kemampuan komputer dan engine pengenalan untuk melakukan perhitungan tambahan, dan juga perlu dipertimbangkan peningkatan kemampuan sistem di masa depan apakah bisa tanpa konfigurasi ulang lebih jauh.

Menentukan ketepatan suatu sistem ANPR sangat rumit dan tergantung pada aplikasi, kondisi operasi dan asumsi yang dibuat pada saat pengujian. Karena itu pengukuran kemampuan sistem sangat sulit. Memang banyak yang mengharapkan pengenalan sempurna dan mengasumsikan ketepatan 100 persen. Tetapi beberapa plat memang sama sekali tidak terbaca, bahkan oleh mata manusia, seperti karena kotoran, cahaya, kerusakan, dan lain lain. Sistem yang terotomatisasi tentu saja tidak bisa diharapkan untuk mencapai kesempurnaan, bahkan dalam kondisi ideal sekalipun.

Salah satu cara adalah dengan menghitung persentasi keberhasilan sistem dalam mengenali plat nomor secara benar dalam lingkup dimana manusia dapat mengenali plat nomor itu juga. Dan tentu saja sistem hanya dapat mengenali plat nomor setelah posisi plat nomor berhasil dilacak.

Toleransi pada kesalahan pengenalan tergantung pada aplikasinya. Pada aplikasi kendali akses, kesalahan sistem sama sekali tidak diperbolehkan, karena bisa memberikan akses pada yang tidak berhak dan sebaliknya. Akan tetapi postprocessing bisa mentoleril sedikit kesalahan, misalnya jika setelah pengenalan, tiap karakter akan dicocokkan dengan database maka kesalahan satu

karakter bisa diabaikan, dan sistem dapat memberikan hak akses dengan kondisi tertentu.

2.7.7. Bidang Aplikasi

Bidang aplikasi ANPR memiliki lingkup yang luas, yang menggunakan nomor plat yang telah dikenali dan optional image untuk memberikan solusi otomatis untuk berbagai masalah (*Hofman, 2004*).

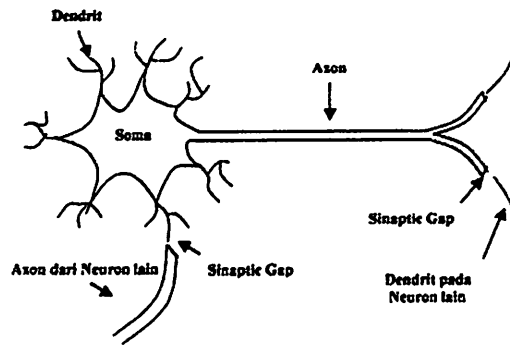
1. Perparkiran – Nomor plat digunakan untuk memasukkan pelanggan pra bayar secara otomatis dan menghitung biaya bukan pelanggan (dengan membandingkan waktu masuk dan waktu keluar). Image sampingan seperti wajah pengemudi dapat digunakan untuk mencegah pencurian.
2. Kontrol Akses – suatu gerbang akan terbuka secara otomatis untuk anggota yang berhak untuk suatu secured area, sehingga menghilangkan penggunaan penjaga gerbang. Kejadian ini akan di-log ke dalam database dan dapat digunakan untuk pencarian selanjutnya.
3. Jalan Tol - nomor plat digunakan untuk menghitung biaya jalan tol, atau digunakan untuk memeriksa ulang tiket.
4. Kontrol Batas Negara – nomor plat didatakan di pintu masuk atau keluar suatu negara, dan digunakan untuk memantau batas negara. Penggunaan sistem ini dapat mempercepat waktu penyeberangan wilayah.
5. Pencurian mobil – daftar dari mobil yang dicuri atau belum dibayar digunakan untuk memperingati mobil yang lewat yang terdaftar. Daftar “black list” dapat diupdate secara real time dan menyediakan alarm langsung untuk polisi. Sistem ANPR dipasang di jalan dan melakukan pencocokan secara real time antara mobil yang lewat dengan yang ada dalam daftar, ketika ditemukan sirene akan dibunyikan dan petugas akan diberitahu.
6. Enforcement - nomor plat dapat digunakan untuk menilang kendaraan yang melebihi batas kecepatan atau lampu merah. Proses manual yang biasa dilakukan dapat digantikan dengan proses otomatis yang menghemat waktu dan tenaga. Biaya tilang dapat dibayar secara online.

7. Pengendalian Lalu Lintas – kendaraan dapat diarahkan ke jalur lain berhubungan dengan ijin masuk, sistem ini akan mengurangi kemacetan secara efektif.
8. Alat bantu pemasaran – nomor plat dapat digunakan untuk membuat daftar pengunjung yang sering untuk keperluan pemasaran, atau untuk membangun profil lalu lintas (seperti frekuensi masuk per jam atau hari).
9. Perjalanan – Beberapa ANPR yang diletakkan di beberapa titik di jalan dapat digunakan untuk menghitung kecepatan dan waktu rata-rata antara dua titik jalan dan digunakan untuk memantau kepadatan jalan raya.
10. Perparkiran bandara – Untuk mengurangi kesalahan tiket, unit ANPR digunakan untuk menangkap nomor plat dan gambar mobil. Informasi ini akan digunakan untuk menghitung biaya parkir atau memberikan bukti saat kehilangan tiket.

2.8 Jaringan Syaraf Biologi dan Tiruan

2.8.1 Jaringan Syaraf Biologi

Otak manusia memiliki struktur yang sangat kompleks dan memiliki kemampuan yang luar biasa. Otak manusia terdiri dari neuron – neuron dan penghubung yang disebut sinapsis. Neuron bekerja berdasarkan impuls / sinyal yang diberikan pada neuron. Neuron meneruskannya pada neuron lain. Diperkirakan manusia memiliki 1012 neuron dan 6.108 sinapsis. Dengan jumlah yang begitu banyak, otak mampu mengenali pola, melakukan perhitungan, dan mengontrol organ – organ tubuh dengan kecepatan yang lebih tinggi dibandingkan komputer digital. Sebagai perbandingan, pengenalan wajah seseorang yang sedikit berubah (misal memakai topi, memiliki kumis, dan lain sebagainya) akan lebih cepat dilakukan manusia dibandingkan dengan komputer. Struktur otak manusia adalah seperti pada gambar 2.16



Gambar 2.16 Struktur otak pada manusia

Pada waktu lahir, otak mempunyai struktur yang menakjubkan karena kemampuannya membentuk sendiri aturan – aturan (pola) berdasarkan pengalaman yang diterima. Jumlah dan kemampuan neuron berkembang seiring dengan pertumbuhan fisik manusia terutama pada umur 0 – 2 tahun. Pada manusia umur 2 tahun pertama, terbentuk 1 juta sinapsis per detik.

Neuron memiliki 3 komponen penting yaitu dendrit, soma dan axon. Dendrit menerima sinyal dari neuron lain. Sinyal tersebut berupa impuls elektrik yang dikirim melalui proses kimiawi. Sinyal tersebut dimodifikasi (diperkuat / diperlemah) di celah sinaptik. Berikutnya soma menjumlahkan semua sinyal – sinyal yang masuk. Kalau jumlahan tersebut cukup kuat dan melebihi batas ambang (threshold), maka sinyal tersebut akan diteruskan ke sel lain melalui axon. Frekuensi penerusan sinyal berbeda – beda antara satu sel dengan yang lain.

Neuron biologi merupakan sistem yang “fault toleran” dalam 2 hal. Pertama, manusia dapat mengenali sinyal input yang agak berbeda dari yang pernah diterima sebelumnya. Kedua, otak manusia tetap mampu bekerja meskipun beberapa neuronnya tidak mampu bekerja dengan baik. Jika sebuah neuron rusak, neuron lain kadang – kadang dapat dilatih untuk menggantikan fungsi sel lain yang rusak tersebut.

2.8.2 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi.

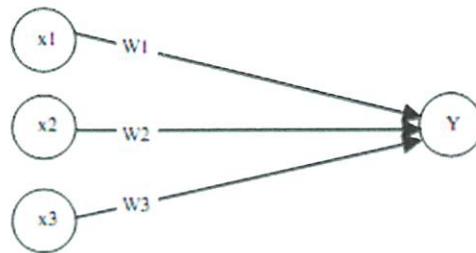
Jaringan syaraf tiruan (JST) dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

1. Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron)
2. Sinyal dikirim diantara neuron – neuron melalui penghubung – penghubung
3. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal
4. Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikarenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

Jaringan syaraf tiruan (JST) ini ditentukan oleh 3 hal :

1. Pola hubungan antar neuron (disebut arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (disebut metode training / learning / algoritma)
3. Fungsi aktivasi

Contoh dari jaringan syaraf tiruan (JST) terdapat pada gambar 2.17



Gambar 2.17 Jaringan Syaraf Tiruan

Y menerima input dari neuron x1, x2 dan x3 dengan bobot hubungan masing – masing adalah w1, w2 dan w3. Ketiga impuls neuron yang ada dijumlahkan, seperti pada persamaan 2.1

$$\text{net} = x_1w_1 + x_2w_2 + x_3w_3 \dots\dots\dots (2.32)$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y=f(\text{net})$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot.

2.8.3 Aplikasi Jaringan Syaraf Tiruan (JST)

Beberapa aplikasi jaringan syaraf tiruan antara lain :

a. Pengenalan Pola (Pattern Recognition)

Jaringan syaraf tiruan (JST) dapat dipakai untuk mengenali pola (misalkan huruf, angka, suara atau tanda tangan) yang sudah sedikit berubah. Hal ini mirip otak manusia yang masih mampu mengenali orang yang sudah beberapa waktu tidak dijumpai (mungkin wajah / bentuk tubuhnya yang sudah sedikit berubah)

b. Signal Processing

Jaringan syaraf tiruan (JST) (model ADALINE) dapat dipakai untuk menekan noise dalam saluran telepon.

c. Peramalan

Jaringan syaraf tiruan (JST) juga bisa dipakai untuk meramalkan apa yang akan terjadi di masa yang akan datang berdasarkan pola kejadian yang sudah ada di masa lampau. Ini dapat dilakukan mengingat kemampuan jaringan syaraf tiruan untuk mengingat dan membuat generalisasi dari apa yang sudah ada sebelumnya.

Selain itu, jaringan syaraf tiruan juga dilaporkan dapat menyelesaikan masalah di bidang kontrol, kedokteran dan lain sebagainya.

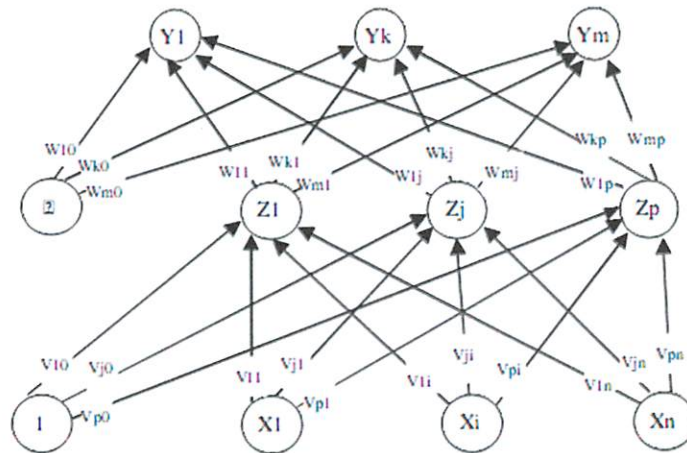
Meskipun banyak aplikasi menjanjikan yang dapat dilakukan oleh jaringan syaraf tiruan, namun jaringan syaraf tiruan juga memiliki beberapa keterbatasan umum. Pertama adalah ketidak akuratan hasil yang diperoleh. Jaringan syaraf tiruan bekerja berdasarkan pola yang terbentuk pada inputnya.

2.9 Backpropagation

Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang 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.

2.9.1 Arsitektur Backpropagation

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Arsitektur backpropagation dengan n buah masukan (ditambah satu bias), dan sebuah layar tersembunyi yang terdiri dari p unit (ditambah satu bias), dan juga m buah unit keluaran, terdapat pada gambar 2.18

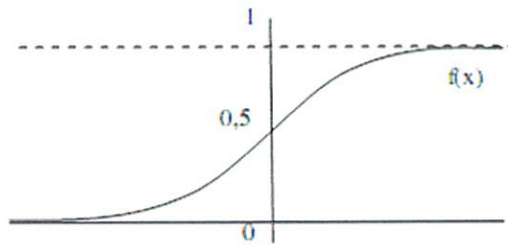


Gambar 2.18 Arsitektur Backpropagation

Dari gambar 2.18, dapat dijelaskan bahwa V_{ji} adalah bobot garis dari unit masukan X_i ke unit layar tersembunyi Z_j (V_{j0} adalah bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi Z_j). Sedangkan W_{kj} adalah bobot dari unit layar tersembunyi Z_j ke unit keluaran Y_k (W_{k0} adalah bobot dari bias di layar tersembunyi ke unit keluaran Y_k).

2.9.2 Fungsi Aktivasi

Dalam backpropagation, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu : kontinyu, terdeferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah sigmoid biner yang memiliki range (0 - 1), Grafik untuk fungsi sigmoid biner yaitu pada gambar 2.19



Gambar 2.19 Grafik Fungsi Sigmoid Biner

2.9.3 Pelatihan Standar Backpropagation

Pelatihan backpropagation meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit – unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

a. Fase 1: Propagasi maju

Selama propagation maju, sinyal masukan (X_i) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi (Z_j) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ($\mathcal{Y}_k$).

Berikutnya, keluaran jaringan ($\mathcal{Y}_k$) dibandingkan dengan target yang harus dicapai (t_k). Selisih $\mathcal{Y}_k - t_k$ 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 toleransi, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

b. Fase 2: Propagasi mundur

Berdasarkan kesalahan $\mathcal{Y}_k - t_k$, dihitung faktor δ_k ($k=1,2,\dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit $\mathcal{Y}_k$ ke semua unit

$$\delta_k$$

tersembunyi yang terhubung langsung dengan $\mathcal{V}_k$. juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama dihitung faktor δ di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Fase 3: Propagasi bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghitungan yang sering dipakai adalah jumlah iterasi dan 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 pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut :

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

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

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j=1,2,\dots,p$).

Terdapat pada persamaan 2.33 dan 2.34

$$z_j = f(z_{netj}) = \frac{1}{1 + e^{-z_{netj}}} \quad (2.33)$$

$$z_{netj} = V_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.34)$$

Langkah 5 : Hitung semua keluaran jaringan di unit $\mathcal{V}_k$ ($k=1,2,\dots,m$).

terdapat pada persamaan 2.35 dan 2.36

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (2.35)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (2.36)$$

Fase II : Propagasi mundur

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k=1,2,\dots,m$). Terdapat pada persamaan 2.37

$$\delta_k = (t_k - y_k)f'(y_{net_k}) = (t_k - y_k)y_k(1 - y_k) \quad (2.37)$$

δ_k merupakan unit kasalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk mengubah bobot w_{kj}) dengan laju pemahanan (*learning rate*) α . Terdapat pada persamaan 2.38

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.38)$$

Dimana : $k = 1,2,\dots,m$ dan $j = 0,1,\dots,p$

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ($j=1,2,\dots,p$). Terdapat pada persamaan 2.39

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (2.39)$$

Faktor δ unit tersembunyi terdapat pada persamaan 2.40

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad (2.40)$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk mengubah bobot v_{ji}). Terdapat pada persamaan 2.41

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.41)$$

Dimana : $j=1,2,\dots,p$ dan $i = 0,1,\dots,n$

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit keluaran dengan persamaan 2.42

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.42)$$

Dimana : $k=1,2,\dots,m$ dan $j=0,1,\dots,p$

Perubahan bobot garis yang menuju ke unit tersembunyi dengan persamaan 2.43

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.43)$$

Dimana : $k=1,2,\dots,m$ dan $j=0,1,\dots,n$

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

2.9.4 Optimasi Arsitektur Backpropagation

Masalah utama yang dihadapi dalam Backpropagation adalah lamanya iterasi yang harus dilakukan. Backpropagation tidak dapat memberikan kepastian tentang berapa epoch yang harus dilalui untuk mencapai kondisi yang diinginkan.

2.9.4.1 Pemilihan Bobot

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya.

Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu dalam standar Backpropagation, bobot diisi dengan bilangan acak kecil.

Nguyen dan Widrow (1990) mengusulkan cara membuat inisialisasi bobot ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat, yaitu dengan cara menginisialisasi semua bobot (w_{ij} (lama)) dengan bilangan acak dalam interval $[-0.5, 0.5]$.

2.9.4.2 Jumlah Unit Tersembunyi

Hasil teoritis yang didapat menunjukkan bahwa jaringan dengan sebuah layer tersembunyi sudah cukup bagi Backpropagation untuk mengenali sembarang perkawanan antara masukan dan target dengan tingkat ketelitian yang ditentukan. Akan tetapi penambahan jumlah layer tersembunyi kadangkala membuat pelatihan lebih mudah.

Jika jaringan memiliki lebih dari satu layer tersembunyi, maka algoritma pelatihan yang dijabarkan sebelumnya perlu direvisi. Dalam propagasi maju, keluaran harus dihitung untuk tiap layer, dimulai dari layer tersembunyi paling bawah (terdekat dengan masukan). Sebaliknya dalam propagasi mundur, faktor δ perlu dihitung untuk tiap layer tersembunyi, dimulai dari layer keluaran.

2.9.4.3 Lama Iterasi

Tujuan utama penggunaan Backpropagation adalah mendapatkan keseimbangan antara pengenalan pola pelatihan secara benar dan respon yang baik untuk pola lain yang sejenis (disebut data pengujian). Jaringan dapat dilatih terus menerus hingga semua pola pelatihan dikenali dengan benar. Akan tetapi hal ini tidak menjamin jaringan akan mampu mengenali pola pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi hingga semua kesalahan pola pelatihan = 0.

Umumnya data dibagi menjadi 2 bagian saling asing, yaitu pola data yang dipakai sebagai pelatihan dan data yang dipakai untuk pengujian. Perubahan bobot dilakukan berdasarkan pola pelatihan. Akan tetapi selama pelatihan, kesalahan yang terjadi dihitung berdasarkan semua data (pelatihan dan pengujian). Selama kesalahan ini menurun, pelatihan terus dijalankan. Akan tetapi jika kesalahan sudah meningkat, pelatihan sudah tidak ada gunanya untuk diteruskan lagi. Jaringan sudah mulai mengambil sifat yang hanya dimiliki secara spesifik oleh

data pelatihan (tapi tidak dimiliki oleh data pengujian) dan sudah mulai kehilangan kemampuan melakukan generalisasi.

2.9.5 Variasi Backpropagation

2.9.5.1 Momentum

Pada standar Backpropagation, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa (berarti arah gradien sudah benar), maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Dengan penambahan momentum, bobot baru pada waktu ke $(t+1)$ didasarkan atas bobot pada waktu t dan $(t-1)$. Disini harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika μ adalah konstanta ($0 < \mu < 1$) yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan 2.44 dan 2.45

$$W_{kj}(t+1) = W_{kj}(t) + \alpha \delta_k Z_j + \mu (W_{kj}(t) - W_{kj}(t-1)) \quad (2.44)$$

$$V_{kj}(t+1) = V_{kj}(t) + \alpha \delta_k x_j + \mu (V_{kj}(t) - V_{kj}(t-1)) \quad (2.45)$$

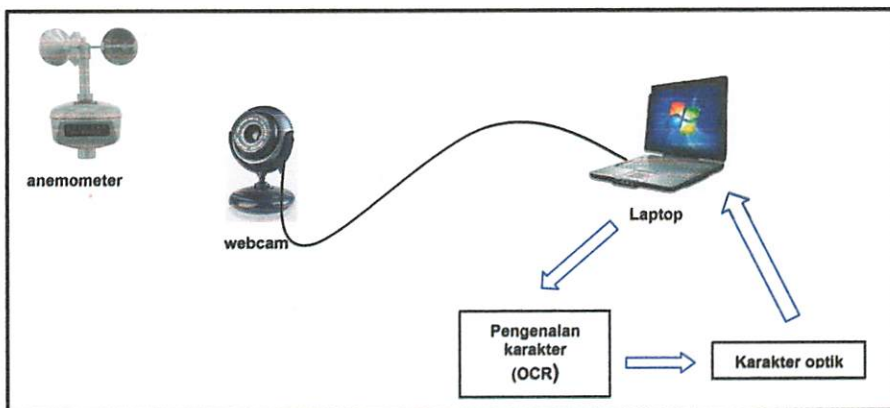
BAB III

PERANCANGAN

Pada bab ini akan membahas tentang langkah-langkah perancangan perangkat lunak secara keseluruhan yang merupakan pokok bahasan dalam skripsi.

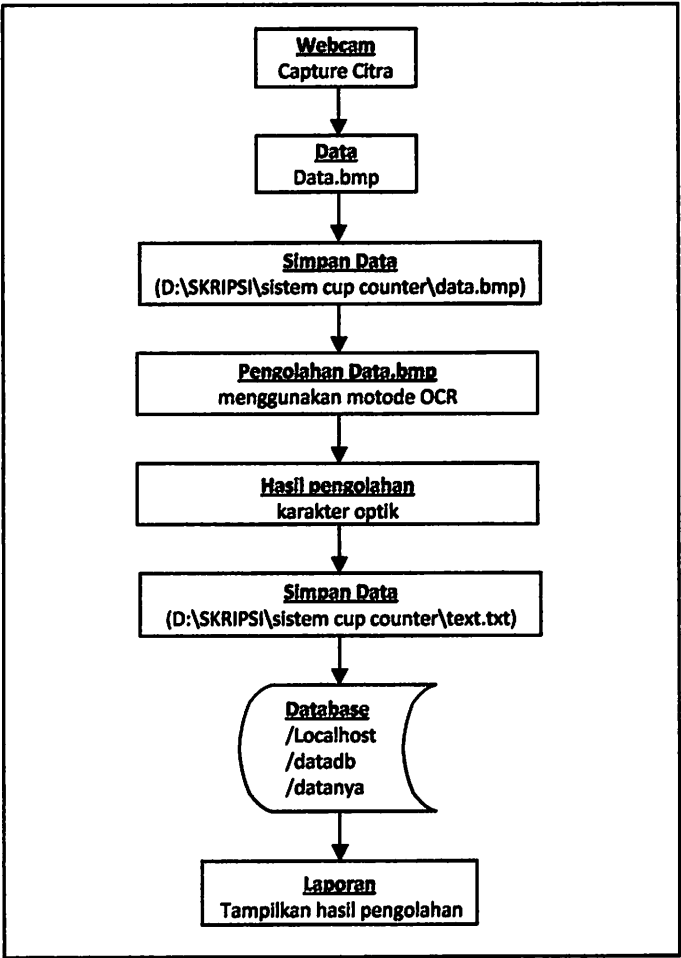
3.1 Cara Kerja Sistem

Perancangan sistem pembacaan alat pengukur kecepatan angin cup counter anemometer dijelaskan pada gambar 3.1



Gambar 3.1 Gambar Cara Kerja sistem

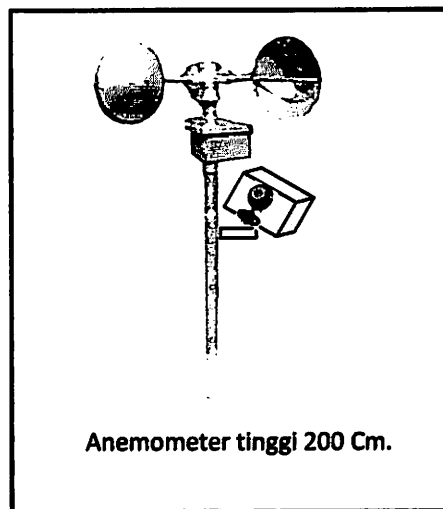
Berdasarkan gambar 3.1, sistem pembacaan alat pengukur kecepatan angin cup counter anemometer dilakukan yaitu dengan mengambil gambar dari display alat dengan menggunakan Webcam. Sistem melakukan video streaming, Kemudian dilakukan capturing image dengan mengkonversi citra dari video streaming ke format BMP. komputer akan mengolah citra ber-ekstensi .BMP dengan metode Jaringan Syaraf Tiruan Backpropagation sebagai Optical Character Recognition. Setelah dilakukan pengolahan citra hasil capture .BMP yang telah di proses akan berupa karakter optik yang dapat diolah dan dapat disimpan kedalam database.



Gambar 3.2 Blok diagram Cara Kerja Sistem

3.2 Perancangan Penempatan Webcam

Agar hasil yang didapat identik, maka penempatan Webcam dengan Alat Anemometer harus dengan jarak dan posisi yang tetap. Maka dari itu, posisinya harus lurus. Selain itu agar didapat hasil gambar yang identik, maka pencahayaan yang digunakan harus homogen. Pada kasus ini objek (Anemometer) sudah terpasang pada BMKG karangploso. Webcam dipasang ditiang yang jarak dan posisinya sudah diatur. Jarak antara Webcam dengan Anemometer diatur sejauh 10 cm. Jarak ini sudah tidak bisa digeser – geser lagi. Gambar penempatan Webcam dengan Anemometer dapat dilihat pada gambar 3.3



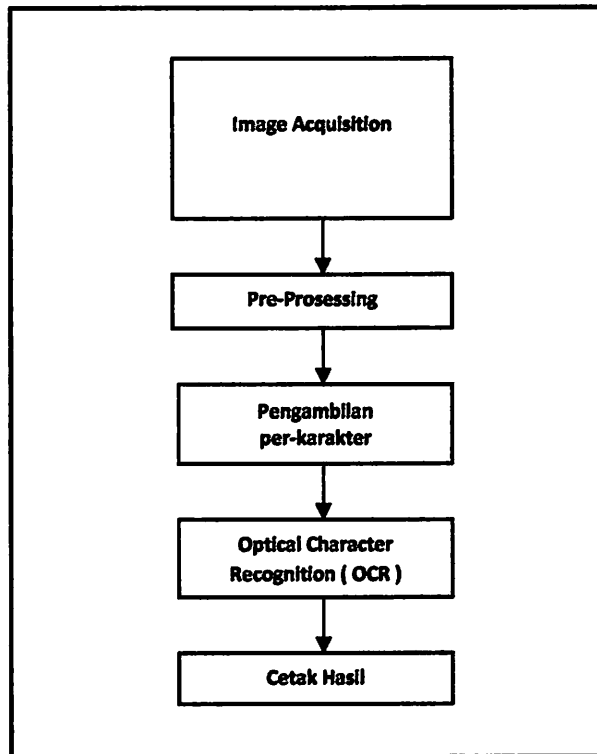
Gambar 3.3 Gambar Penempatan Webcam

3.3 Perancangan Blok Diagram Pada Aplikasi

Pada aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer terdapat alur yang akan memproses aplikasi, agar aplikasi dapat memproses penerapan optical character recognition pada sistem pembacaan alat pengukur kecepatan angin cup counter anemometer. Pada blok diagram aplikasi dapat dijelaskan sebagai berikut :

1. Proses video streaming dari webcam.
2. Proses pengambilan gambar dengan cara meng-konvert citra dari video streaming kedalam bentuk .bmp.
3. Setelah citra menjadi .bmp maka dilakukan proses tresholding. Proses ini berfungsi untuk merubah citra menjadi biner.
4. Proses pengambilan per-karakter atau segmentasi, proses memisahkan area pengamatan (region) pada tiap karakter yang akan dideteksi.
5. Proses backpropagation . pada proses ini terdapat diantaranya mengatur hasil segmentasi kedalam satu ukuran yang sama, menentukan bobot tiap tiap karakter.
6. Membandingkan dengan karakter
7. Cetak hasil angka setelah proses backpropagation dan membandingkannya dengan karakter.

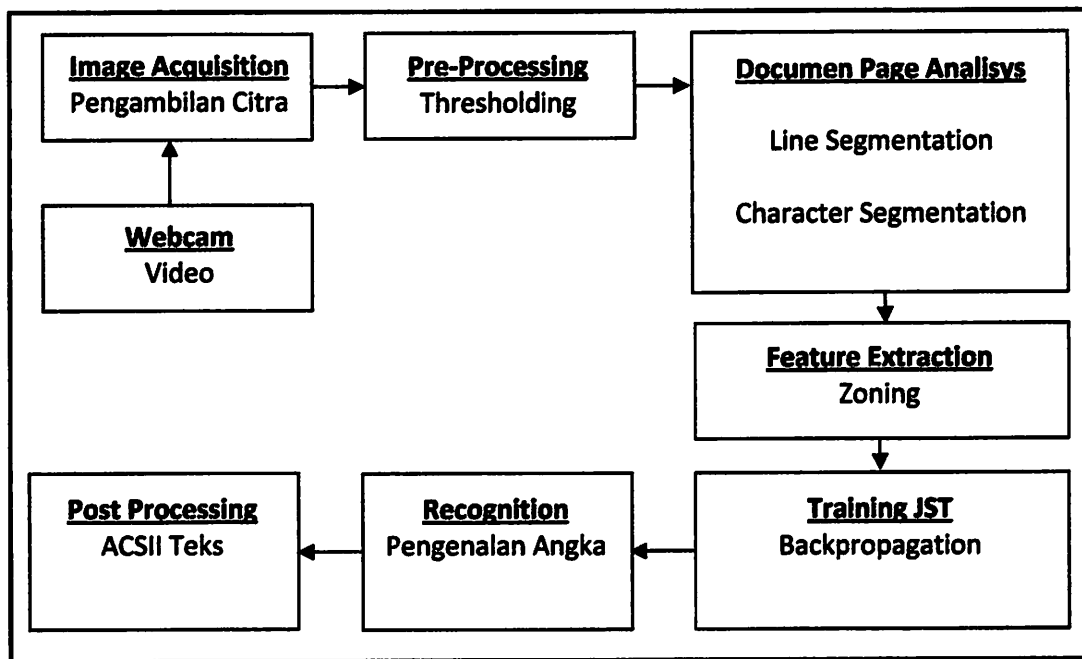
Langkah yang ada pada aplikasi penerapan optical character recognition diatas dijelaskan pada gambar 3.4.



Gambar 3.4 Gambar Blok Diagram Aplikasi

3.4 Optical Character Recognition

Pada penerapan optical character recognition untuk aplikasi sistem pembacaan cup counter anemometer, terdapat adanya proses image acquisition, pre-processing, document and analysis, feature extraction, training and recognition dan post processing pada citra. Diagram alir dalam penerapan optical character recognition untuk aplikasi sistem pembacaan cup counter anemometer dapat dilihat pada gambar 3.5



Gambar 3.5 Blog Diagram Optical Character Recognition

Pada blok diagram proses OCR di atas dapat dijelaskan sebagai berikut :

1. Image acquisition yaitu proses untuk merubah citra asli menjadi citra digital dengan cara capturing pada webcam.
2. Dalam image acquisition terdapat proses image capturing, image capturing adalah merubah format .avi dari webcam menjadi .bmp agar bisa diproses.
3. Kemudian proses selanjutnya citra yang sudah menjadi citra digital diperbaiki dengan proses pre processing, pada pre processing terdapat threshold, noise removal, sharpening, skew.
4. Proses selanjutnya yaitu Document And Analysis yang mendeteksi karakter citra agar dapat terdeteksi dengan benar, sehingga citra dapat dikenali.
5. Line segmentation adalah untuk mendeteksi perbaris citra yang sudah terdeteksi.
6. Character recognition untuk mendeteksi citra pada setiap angka yang terdapat pada karakter citra yang terdeteksi.
7. Pada proses feature extraction citra yang akan terekstrak akan dikumpulkan kembali, sehingga tersusun karakter yang sudah terdeteksi proses tersebut juga disebut zoning.

8. Lanjut pada proses training, pada proses ini karakter akan diproses dengan mencocokkan citra karakter yang sudah terdeteksi.
9. Bila pencocokan citra karakter sudah benar, maka proses selanjutnya mengklasifikasi citra dengan cara proses recognition.
10. Setelah citra karakter cocok setelah melalui proses pengenalan angka, maka citra akan ditampilkan menggunakan proses post-processing.

3.5 Image Acquisition

Image Acquisition adalah proses mendapatkan data image. Pengambilan data dilakukan dengan cara capture dari citra webcam. Dalam image acquisition pada aplikasi ini terdapat dua tahapan, capture gambar secara terus menerus lalu save sebagai file video maupun secara langsung melakukan capture video. Untuk cara yang pertama melakukan perulangan untuk pengambilan gambar, perintah yang digunakan :

```
aviObject = avifile('myVideo.avi');
for iFrame = 1:100
  F = im2frame(i);
  aviObject = addframe(aviObject,F);
end
aviObject = close(aviObject);
```

Fungsi *avifile* digunakan untuk membuat file .avi dimana untuk program ini terdiri dari capture 6 frame (gambar). cara yang digunakan untuk sistem otomatis capture yaitu langsung merekam video tanpa mengcapture dalam bentuk gambar terlebih dahulu. Pertama yang dilakukan adalah melakukan setting secara default seperti tempat menyimpan file, nama file, panjang file, adaptor kamera, dll. Kemudian setting untuk file video yang akan dihasilkan seperti frame per second, tipe kompresi, kualitas video yang dihasilkan. Setelah itu mulai melakukan proses perekaman video.

Untuk dapat mengambil data input menggunakan matlab setting yang harus diketahui terlebih dahulu.

```
>> imaqhwinfo
```

Perintah di atas digunakan untuk mengetahui tipe adaptor kamera yang digunakan,

```
>> vid=videoinput('winvideo')
```

Perintah di atas untuk menampilkan default dari kamera yang digunakan.

Untuk melakukan image acquisition pada aplikasi ini dengan manual capture adalah menggunakan :

```
>> vid=videoinput('winvideo');
```

```
>> preview (vid)
```

Untuk melakukan image acquisition pada aplikasi ini secara otomatis adalah menggunakan :

```
for ctr = 1:5
    vid=videoinput('winvideo',1,'RGB24_320x240');
    img = getsnapshot(vid);
    fname = ['Image' num2str(ctr)];
    imwrite(img,[fname '.jpeg']);
    pause(1);
end
```

3.6 Pre-Processing

Pre-Processing bertujuan untuk penghilangan noise/derau, penajaman image, pemotongan image, dll. Keluaran dari tahap ini adalah image tersegmentasi yang akan digunakan untuk proses selanjutnya. Pre-Processing yang dilakukan pada pembuatan aplikasi ini meliputi Greyscale, Threshoding dan Cropping.

3.6.1 Grayscale

Proses awal yang sering dilakukan pada image processing adalah mengubah citra berwarna menjadi citra gray-scale. Hal ini dilakukan untuk menyederhanakan model citra. Di dalam suatu gambar true color (RGB) terdapat tiga layer matriks, yaitu R-layer, G-layer, dan B-layer. Pada image processing dilakukan proses-proses terhadap ketiga layer tersebut, berarti dilakukan perhitungan yang sama pada setiap layer. Dengan demikian konsep grey-scale

adalah mengubah tiga layer tersebut menjadi satu layer matriks grey-scale, yang menghasilkan satu citra grey-scale. Di dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Secara umum untuk menghasilkan citra grey-scale, konversi dilakukan dengan mengambil rata-rata dari nilai r , g , dan b , sehingga menghasilkan nilai s sebagai nilai grey-scale nya. Secara matematis dituliskan sbb⁵:

$$s = \frac{r + g + b}{3} \quad (1)$$

Pada penjelasan di atas pengubahan citra berwarna menjadi citra grey-scale adalah dengan mencari nilai rata-rata grey-scale dari setiap layer r , g , dan b . Beberapa image belum optimal jika diberikan cara di atas, untuk keoptimalan citra grey-scale diperlukan perubahan komposisi pada layer r , g , dan b .

Pada aplikasi MATLAB telah disediakan tool untuk merubah citra rgb menjadi grey-scale. Perintah yang digunakan dalam aplikasi ini adalah

```
image=imread('datacrop.bmp');  
imagen=rgb2gray(image);
```

3.6.2 Threshoding

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

Dua pendekatan yang digunakan dalam operasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal. Operasi pengambangan (thresholding) yang dilakukan dalam pembuatan aplikasi ini adalah operasi pengambangan (thresholding) secara global.

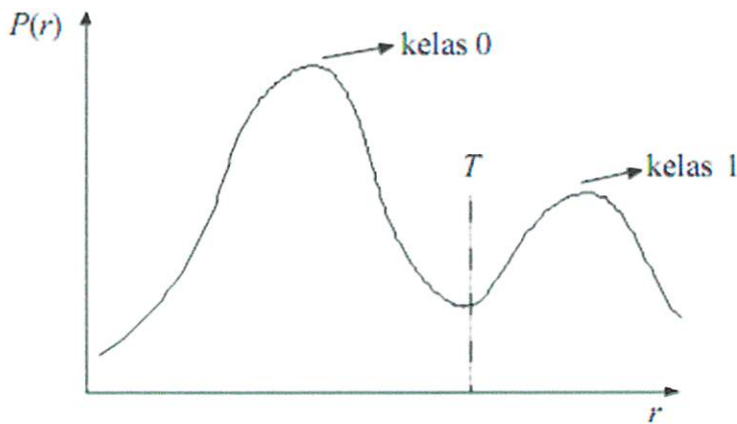
Pada pengambangan secara global Setiap pixel di dalam citra dipetakan ke dua nilai, 1 atau 0 dengan fungsi pengambangan:

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

yang dalam hal ini, $f_g(i, j)$ adalah citra hitam-putih, $f_B(i, j)$ adalah citra biner, dan T adalah nilai ambang yang dispesifikasikan. Dengan operasi

pengambangan tersebut, objek dibuat berwarna gelap (1 atau hitam) sedangkan latar belakang berwarna terang (0 atau putih).

Nilai ambang T dipilih sedemikian sehingga galat yang diperoleh sekecil mungkin. Cara yang umum menentukan nilai T adalah dengan membuat histogram citra. Jika citra mengandung satu buah objek dan latar belakang mempunyai nilai intensitas yang homogen, maka citra tersebut umumnya mempunyai histogram bimodal (mempunyai dua puncak atau dua buah maksimum lokal) seperti yang ditunjukkan pada Gambar 3.6. Nilai T dipilih pada nilai minimum lokal yang terdapat di antara dua puncak. Dengan cara seperti ini, kita tidak hanya mengkonversi citra hitam-putih ke citra biner, tetapi sekaligus melakukan segmentasi objek dari latar belakangnya.



Gambar 3.6 Penentuan nilai ambang T

Jika nilai intensitas objek diketahui dalam selang $[T_1, T_2]$, maka kita dapat menggunakan fungsi pengambangan:

$$f_B(i, j) = \begin{cases} 1, & T_1 \leq f_g(i, j) \leq T_2 \\ 0, & \text{lainnya} \end{cases}$$

Pada aplikasi MATLAB telah disediakan tool untuk melakukan perintah thresholding. Perintah yang digunakan dalam aplikasi ini adalah

```
threshold = graythresh(imagen);
imagen = im2bw(imagen, threshold);
```

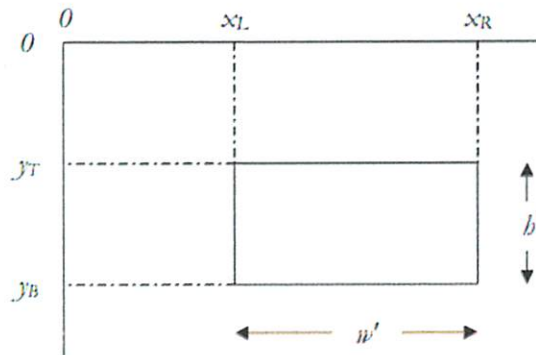
3.6.3 Cropping

Cropping adalah memotong satu bagian dari citra sehingga diperoleh citra yang berukuran lebih kecil. Operasi ini pada dasarnya adalah operasi translasi, yaitu menggeser koordinat titik citra.

$$x' = x - x_L \quad \text{untuk } x = x_L \text{ sampai } x_R$$

$$y' = y - y_T \quad \text{untuk } y = y_T \text{ sampai } y_B$$

(x_L, y_T) dan (x_R, y_B) masing-masing adalah koordinat titik pojok kiri atas dan pojok kanan bawah bagian citra yang hendak dicrop. Koordinat titik sudut bagian citra yang akan di-crop adalah seperti gambar berikut



Gambar 3.7 Koordinat titik Cropping

Ukuran citra berubah menjadi

$$w' = x_R - x_L \quad b' = y_B - y_T$$

dan transformasi baliknya adalah

$$x = x' + x_L \quad \text{untuk } x' = 0 \text{ sampai } w'-1$$

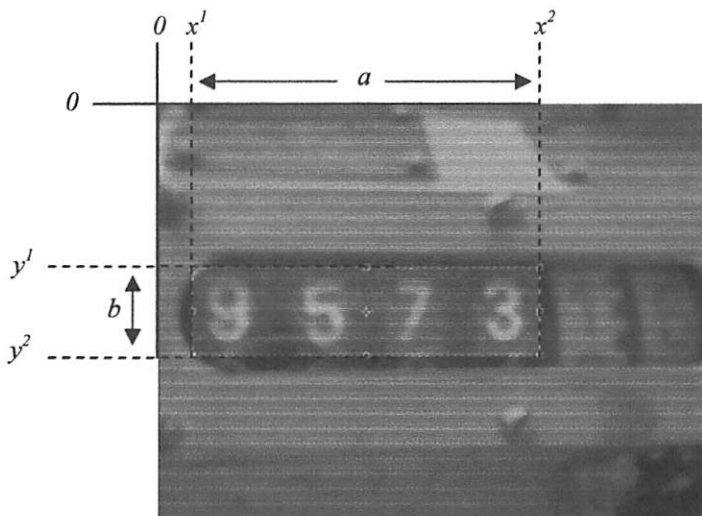
$$y = y' + y_T \quad \text{untuk } y' = 0 \text{ sampai } b'-1$$

Koordinat titik untuk melakukan cropping pada aplikasi ini adalah menggunakan koordinat titik $X = 40$ $Y = 210$ $a = 400$ $b = 80$ dengan persamaan :

```
crop=imcrop(I,[X Y a b])
crop=imcrop(I,[40 210 400 80])
```

perintah yang digunakan adalah :

```
I=imread('data.bmp');
crop=imcrop(I,[40 210 400 80])
imwrite(crop, 'datacrop.bmp');
```



Gambar 3.8 Letak Koordinat titik Cropping

3.7 Document Page Analisis

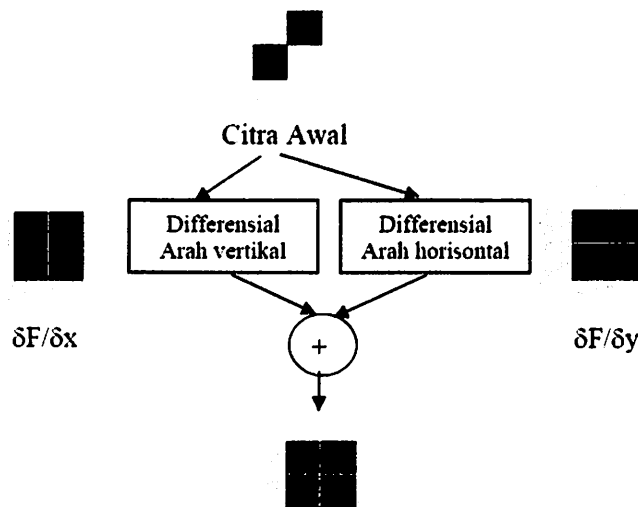
Pada tahapan ini yaitu dengan melakukan langkah Image Segmentation, menurut penjelasan *Shapiro dan Stockman (2001)*, image segmentation merupakan partisi atau pembagian suatu image menjadi sejumlah region (daerah) yang membangun image tersebut. Segmentation ini dilakukan dengan tujuan:

1. untuk membagi image menjadi bagian-bagian untuk analisis lebih lanjut, artinya suatu lingkungan menjadi lebih terkendali sehingga proses segmentasi hanya mengambil bagian yang ingin dianalisis saja.
2. melakukan perubahan representasi, pixel-pixel dari image harus disusun dalam unit yang lebih tinggi yang lebih berarti atau lebih efisien untuk analisis lebih lanjut.

Region suatu image berdasarkan *Jain, Kasturi dan Schunck (1995)* adalah kumpulan pixel-pixel yang saling berhubungan yang memiliki kesamaan sifat. Region penting untuk pengenalan suatu image karena berhubungan dengan objek tertentu pada lingkungan. Suatu image mungkin mengandung beberapa objek dan tiap objek mungkin mengandung beberapa region berhubungan dengan bagian objek. Supaya suatu image dapat diartikan dengan benar, maka harus dibagi menjadi region-region yang berhubungan dengan objek atau bagian dari objek.

3.7.1 Line Segmentation

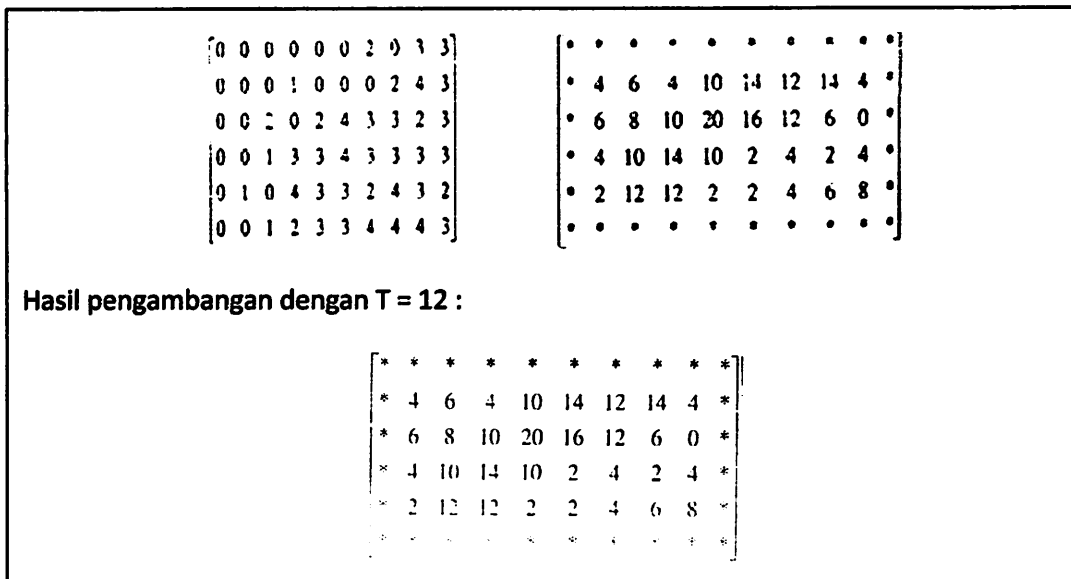
Tepi (edge) adalah perubahan nilai intensitas derajat keabuan yang cepat/tiba-tiba (besar) dalam jarak yang singkat. Sedangkan deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah Untuk menandai bagian yang menjadi detail citra Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Pada gambar 3.9 di bawah ini dapat dilihat proses yang dilakukan untuk memperoleh tepi gambar dari suatu citra yang ada.



Gambar 3.9. Proses deteksi tepi citra

Pada gambar 3.9. terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi. Berikut ini adalah operator yang digunakan adalah operator sobel deteksi tepi dengan operator Sobel, di mana hasil konvolusi diibandingkan dengan $T = 12$.

Citra : $| \text{gradien} - x | + | \text{gradien} - y | :$



Gambar 3.10. deteksi tepi dengan operator sobel

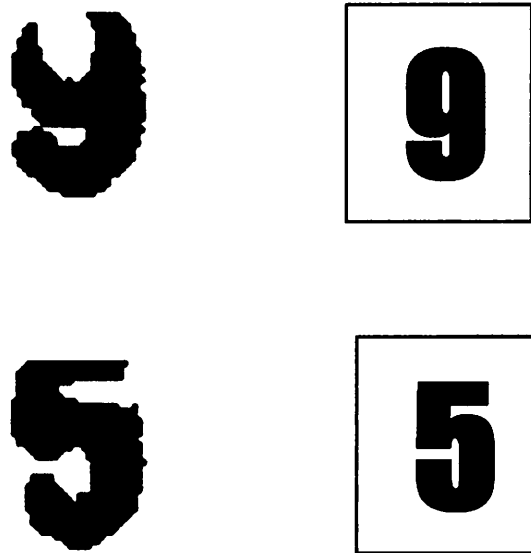
3.8 Feature Extraction

Setelah berhasil mengelompokkan image ke dalam daerah-daerah tertentu, tahapan selanjutnya adalah feature extraction yang merupakan tahapan untuk menemukan (mengeksrak) fitur-fitur penting yang akan digunakan untuk proses pengambilan keputusan selanjutnya. Yang dimaksud dengan fitur disini adalah representasi informasi akan image di dalam tingkat yang lebih tinggi (yang memiliki arti dibanding hanya kumpulan pixel), dimana informasi ini yang nantinya akan digunakan untuk pengenalan objek atau keadaan yang terdapat pada image.

Pada pembuatan aplikasi ini fitur yang akan ditemukan seperti: garis, bentuk dasar kotak persegi panjang, tekstur dan lain sebagainya yang merupakan sifat penting dari image karakter. Fitur-fitur ini juga nantinya akan

direpresentasikan ke dalam bentuk data khusus untuk mempermudah proses pencarian dan pengenalan karakter

Setiap karakter mempunyai karakteristik tersendiri, sehingga fitur tidak dapat bersifat general tetapi sangat tergantung pada model dan obyek karakter yang digunakan. Fitur dasar yang bisa diambil adalah bentuk dan tekstur. Fitur yang lebih kompleks menggunakan segmentasi.



Gambar 3.10. Feature Extraction

3.9 Training JST Backpropagation

Pada pembuatan aplikasi ini metode yang digunakan adalah dengan menggunakan pelatihan backpropagation meliputi 3 fase. Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit – unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi.

a. Fase 1: Propagasi maju

Selama propagation maju, sinyal masukan (X_i) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi (Z_j) tersebut selanjutnya dipropagasikan maju lagi ke layar 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 $Y_k - t_k$ 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 toleransi, maka bobot setiap garis dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi.

b. Fase 2: Propagasi mundur

Berdasarkan kesalahan $Y_k - t_k$, 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 . δ_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

Dengan cara yang sama dihitung faktor di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung.

c. Fase 3: Propagasi bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di layar atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke layar keluaran didasarkan atas δ_k yang ada di unit keluaran.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghitungan yang sering dipakai adalah jumlah iterasi dan 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 pelatihan untuk jaringan dengan satu layer tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut :

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

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

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j=1,2,\dots,p$).

Terdapat pada persamaan 2.33 dan 2.34

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (2.33)$$

$$z_{net_j} = V_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.34)$$

Langkah 5 : Hitung semua keluaran jaringan di unit y_k ($k=1,2,\dots,m$).

terdapat pada persamaan 2.35 dan 2.36

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (2.35)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (2.36)$$

Fase II : Propagasi mundur

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k=1,2,\dots,m$). Terdapat pada persamaan 2.37

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (2.37)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk

mengubah bobot w_{kj}) dengan laju pemahanan (*learning rate*) α . Terdapat pada persamaan 2.38

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.38)$$

Dimana : $k = 1, 2, \dots, m$ dan $j = 0, 1, \dots, p$

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ($j=1, 2, \dots, p$). Terdapat pada persamaan 2.39

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj} \quad (2.39)$$

Faktor δ unit tersembunyi terdapat pada persamaan 2.40

$$\delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j) \quad (2.40)$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk mengubah bobot v_{ji}). Terdapat pada persamaan 2.41

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.41)$$

Dimana : $j=1, 2, \dots, p$ dan $i = 0, 1, \dots, n$

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit keluaran dengan persamaan 2.42

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.42)$$

Dimana : $k=1, 2, \dots, m$ dan $j=0, 1, \dots, p$

Perubahan bobot garis yang menuju ke unit tersembunyi dengan persamaan 2.43

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.43)$$

Dimana : $k=1, 2, \dots, m$ dan $j=0, 1, \dots, n$

Berikut tahapan yang dilakukan dalam pembuatan aplikasi ini :

Proses Pelatihan (Training)

1. Sebelum melakukan simulasi (running) pada command window matlab, terlebih dahulu dibuat instruksi-instruksi (syntax) dalam M-file editor. Dan untuk menghapus semua data dan syntax pada command window dituliskan instruksi `clear`, yang diakhiri dengan tanda titik koma (seperti: `clear`).
2. Masukkan data input dan target. Pada skripsi ini, JST didesain dan dilatih untuk mengenali 9 angka. Masing – masing pola diwakili oleh nilai Boolean kisi 24x42. Yang terdiri dari 768 vektor input dan 12 vektor target. Masing-masing vektor output mewakili karakteristik sebuah sampel.. Sebagai contoh, sampel 1 diwakili oleh target 1 (untuk alasan kemudahan, karena sampel 1 adalah sampel yang diambil pertama kali).
3. Data-data yang diperoleh disimpan dalam file `.dat` lalu dimuat ke M-File.

% Data input & target

P = load ('sampel semua.dat');

T = load ('target.dat');

4. Proses awal sebelum dilakukan pelatihan, data input dan target harus dinormalisasi. Proses normalisasi dapat dilakukan dengan bantuan mean dan deviasi standar. Fungsi `prestd` akan membawa data ke bentuk normal dengan mean = 0 dan deviasi standar = 1.

% Preprocessing

[pn,meanp,stdp,tn,meant,stdt] = prestd(P,T);

5. Membangun jaringan dengan algoritma propagasi umpan balik berbasis jaringan syaraf tiruan, instruksi yang digunakan `newff`.

% Membangun jaringan dengan propagasi umpan balik berbasis jaringan syaraf tiruan

net = newff(minmax(pn),[10 5 1],{'tansig' 'logsig' 'purelin'},'traingdm')

6. Melihat bobot awal input, lapisan dan bias.

% Melihat bobot-bobot awal input, lapisan, dan bias

```

BobotAwal_Input      = net.IW{1,1}
BobotAwal_Bias_Input  = net.b{1,1}
BobotAwal_Lapisan1    = net.LW{2,1}
BobotAwal_Bias_Lapisan1 = net.b{2,1}
BobotAwal_Lapisan2    = net.LW{3,2}
BobotAwal_Bias_Lapisan2 = net.b{3,1}

```

7. Seting maksimum epoh, target error, learning rate, momentum dan epoh show.

```

% Set max epoh, target error, learning rate, momentum dan epoh show
net.trainParam.epochs = 30000;
net.trainParam.goal    = 1e-3;
net.trainParam.lr      = 0.5;
net.trainParam.mc      = 0.7;
net.trainParam.show    = 250;

```

8. Melakukan pembelajaran/pelatihan terhadap data input dan target.

```

% Melakukan pembelajaran
[net,tr] = train(net,pn,tn); pause

```

9. Melihat bobot akhir input, lapisan.

```

% Melihat bobot akhir input, lapisan, dan bias
BobotAkhir_Input      = net.IW{1,1}
BobotAkhir_Bias_Input  = net.b{1,1}
BobotAkhir_Lapisan1    = net.LW{2,1}
BobotAkhir_Bias_Lapisan1 = net.b{2,1}
BobotAkhir_Lapisan2    = net.LW{3,2}
BobotAkhir_Bias_Lapisan2 = net.b{3,1}

```

10. Melakukan simulasi dan hasilnya didenormalisasi pada data aslinya.

```

% Melakukan simulasi
an = sim(net,pn)
a = poststd(an,meant,stdt)

```

```
H = [(1:size(P,2))' a']
sprintf('%2d %5.2f\n',H')
```

11. Melakukan evaluasi terhadap output jaringan, yaitu data hasil pelatihan dengan target

```
%Melakukan evaluasi terhadap output jaringan
[m1,a1,r1]=postreg(a,T)
```

12. Menampilkan hasil pelatihan dalam bentuk grafik.

```
%Menampilkan hasil pelatihan dalam bentuk grafik.
plot([1:size(P,2)]',T,'bo',[1:size(P,2)]',a','r*')
title('Pengujian dengan data training: Target(o), output (*)')
xlabel('data ke-');ylabel('Target');pause
```

Untuk membangun jaringan dengan propagasi umpan balik berbasis jaringan syaraf tiruan pada aplikasi Matlab dapat menggunakan perintah berikut :

```
% Membangun jaringan dengan propagasi umpan balik berbasis jaringan
syaraf tiruan
net = newff(minmax(pn),[10 5 1],{'tansig' 'logsig' 'purelin'},'traingdm')

% melihat bobot awal jaringan
BobotAwal_Input = net.IW{1,1}; BobotAwal_Bias_Input = net.b{1,1};
BobotAwal_Lapisan1 = net.LW {2,1}; BobotAwal_Bias_Lapisan = net.b
{2,1};

% Set parameter pelatihan
net.trainParam.epochs = 1000; net.trainParam.goal = 1e-04;
net.trainParam.lr = 0.6; net.trainParam.showCommandLine = true;
net.trainParam.show = 100; net.trainParam.mc = 0.7;
```

% pelatihan

net = train(net,pn,tn);pause

% melihat bobot akhir jaringan

BobotAkhir_Input = net.IW{1,1}; BobotAkhir_Bias_Input = net.b{1,1};

BobotAkhir_Lapisan1 = net.LW {2,1}; BobotAkhir_Bias_Lapisan = net.b {2,1};

%simulasi terhadap data pelatihan

an = sim(net,pn); a = poststd (an,meant,stdt); H = [(1:size(P,2))' T' a' (T'-a')]; sprintf('%2d %9.2f %7.2f %5.2f\n',H')

%Evaluasi output jaringan (data pelatihan dengan target)

[m1,a1,r1] = postreg (a,T); pause plot

(1:size(P,2),T,'bo',1:size(P,2),a,'r'); title('Hasil pengujian dengan data pelatihan Target (o), Output (*)'); xlabel('Data ke-');*

ylabel('Target/Output'); pause

Proses Tes (Testing)

1. Masukan data input dan target. Data-data yang diperoleh disimpan dalam file .dat lalu dimuat ke M-File.

% Data input & target

P = load ('sampel1(input).dat');

T = load ('sampel1(target).dat');

2. Melakukan simulasi jaringan.

% Simulasi jaringan

Qn = trastd(Q,meanp,stdp)

bn = sim(net,Qn)

b = poststd(bn,meant,stdt)

3. Menampilkan hasil perbandingan antara target yang didapat melalui metode propagasi umpan balik dengan target yang diinginkan dalam bentuk grafik

%grafik

[b' TQ'];

plot(b,'b-)*

hold

plot(TQ,'r-o')

3.10 Perancangan Flowchart

Perancangan suatu aplikasi harus berdasarkan pada perancangan flowchart yang akan menjadi acuan agar suatu aplikasi dapat dibuat dengan konsep yang sudah ada. Agar perancangan sistem yang dibuat berhasil dan berjalan sesuai dengan keinginan yang ada pada sistem yang sudah dirancang.

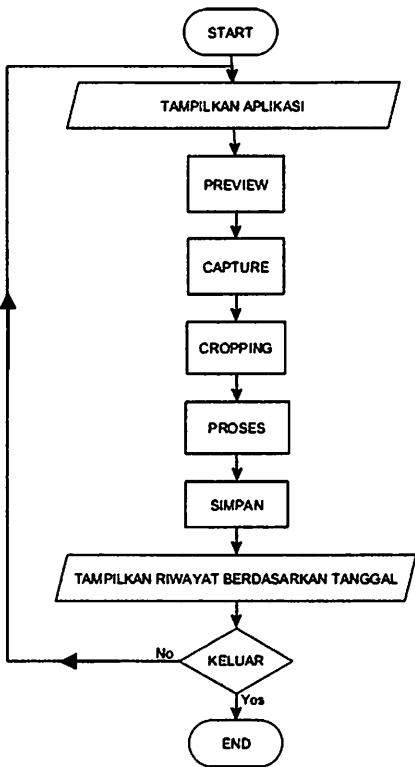
3.10.1 Flowchart Aplikasi

Pada tahap perancangan aplikasi penerapan optical character recognition untuk aplikasi sistem pembacaan cup counter anemometer dapat dijelaskan sebagai berikut :

1. Mulai dari tampilan aplikasi yang terdapat berbagai tombol proses.
2. Memilih tombol Preview, maka akan keluar hasil streaming dari webcam yang akan dilakukan capture.
3. Memilih tombol Capture, untuk menangkap citra webcam. Pada proses ini, system melakukan convert dari video ke gambar dengan format .bmp
4. Jika telah berhasil melakukan proses capture, maka data telah disimpan dan selanjutnya dilakukan proses Cropping dengan memilih tombol Cropping.
5. Memilih tombol Cropping, yaitu pemotongan bagian tertentu dari gambar menjadi matrik baru yang independen dengan tujuan untuk memudahkan dalam pengolahan pengolahan data yang memisahkan citra karakter dengan objek lain.

- 6. Memilih tombol Proses. Yang dilakukan pada tahapan ini adalah melakukan recognition pada citra dengan metode OCR dimana sebelum proses ini dilakukan akan dilakukan proses Trhesholding dapa data yaitu proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan background dari citra secara jelas.
- 7. Memilih tombol Simpan, untuk menyimpan hasil dari citra yang telah di proses.
- 8. User dapat melihat riwayat kapan saja dilakukan pengambilan gambar dengan memilih menu tampilkan berdasarkan tanggal.
- 9. Tombol keluar, untuk keluar dari Aplikasi.
- 10. Aplikasi selesai.

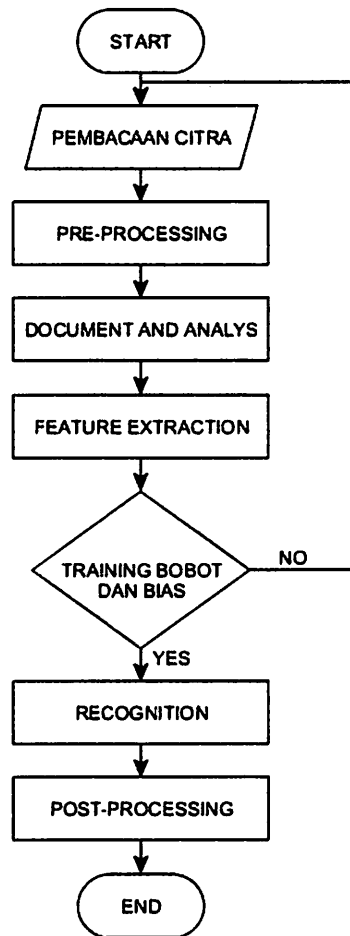
Langkah yang ada pada flowchart aplikasi sistem pembacaan cup counter anemometer dapat dijelaskan pada gambar 3.11



Gambar 3.11 Flowchart Aplikasi

3.10.2 Flowchart OCR

Pada tahap perancangan OCR pada penerapan aplikasi sistem pembacaan cup counter anemometer. dijelaskan pada gambar 3.12.



Gambar 3.12 Flowchart OCR

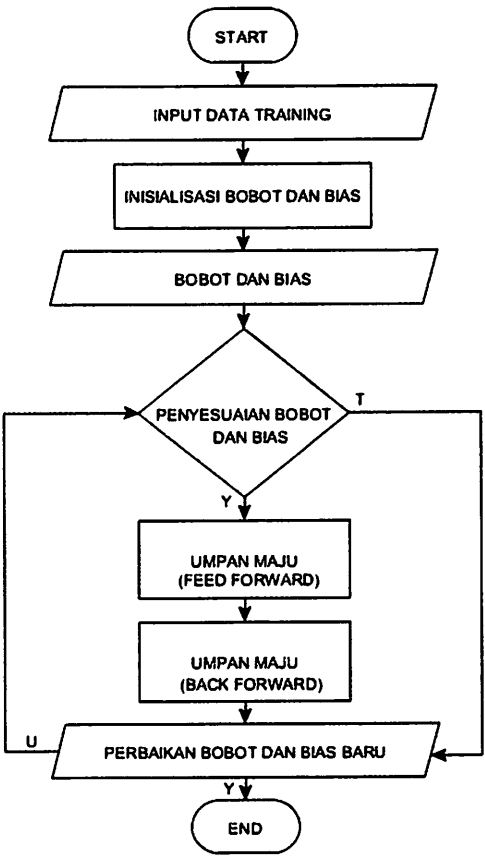
Secara umum blog diagram Optical Character Recognition (OCR) dapat dilihat seperti pada gambar 3.12 yang dilakukan proses sebagai berikut :

1. Start untuk pembacaan karakter pada citra.
2. Kemudian proses preprocessing, agar citra karakter dapat diperbaiki kualitas citra tersebut.
3. Setelah itu proses document and analysis yang akan men-segmentasi karakter perbaris dan perkolomnya, agar karakter dapat terbaca secara jelas.
4. Pada proses selanjutnya training, pada proses ini karakter yang sudah ter-segmentasi akan dibaca setiap huruf pada karakter tersebut.

- 5. Jika pembacaan selesai dan benar, maka dikelompokkan dengan proses recognition agar huruf yang sudah terbaca dapat menjadi karakter dengan benar.
- 6. Setelah melalui proses training and recognition, kemudian karakter akan dicetak sebagai teks biasa dengan proses post processing.

3.10.3 Flowchart Metode Learning Backpropagation

Pada tahap perancangan aplikasi penerapan optical character recognition untuk aplikasi, Perancangan menggunakan metode learning backpropagation untuk training pada sistem aplikasi OCR. Dapat dilihat pada gambar 3.13



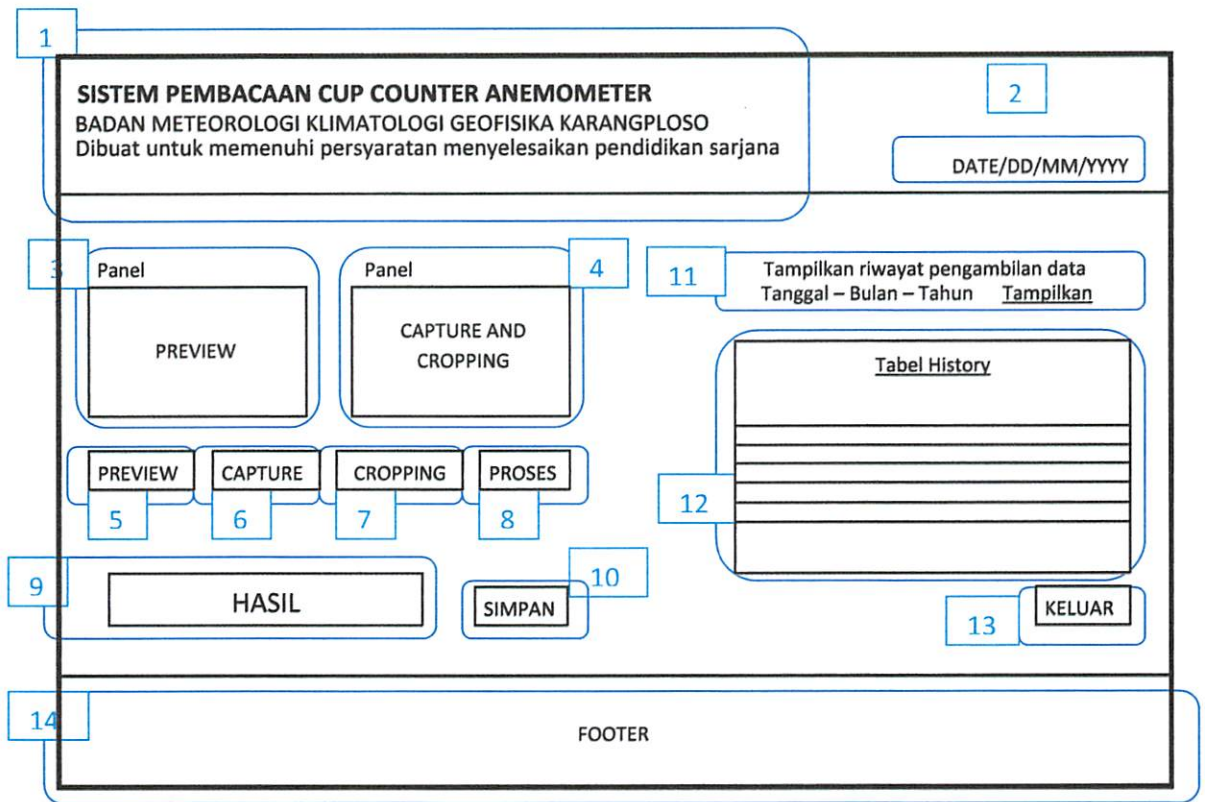
Gambar 3.13 Flowchart Metode Learning

Pada flowchart metode learning backpropagation di atas dapat dijelaskan sebagai berikut :

1. Mulai untuk sistem training pada pembacaan karakter.
2. Input data training adalah memasukkan data karakter yang sudah terdeteksi agar dapat terbaca.
3. Kemudian diproses pada inisialisasi bobot dan bias.
4. Setelah itu meng-input bobot dan bias karakter yang akan dibaca.
5. Penyesuaian pada bobot dan bias akan menyesuaikan bobot dan bias pada karakter.
6. Jika penyesuaian bobot dan bias tidak dilaksanakan, maka akan langsung berhenti.
7. Jika penyesuaian bobot dan bias dilaksanakan, maka pembacaan karakter akan dilanjutkan pada proses selanjutnya.
8. Dilanjutkan pada proses umpan maju (feed forward) yang akan membaca karakter dengan cara kedepan.
9. Kemudian dilanjutkan umpan mundur (back forward) yang akan membaca kembali karakter kebelakang.
10. Selanjutnya proses untuk perbaikan bobot dan bias pada karakter.
11. Setelah perbaikan bobot dan bias karakter, maka karakter yang baru akan menjadi bobot dan bias baru.
12. Pada proses bobot dan bias karakter baru ini akan terjadi proses perulangan yang akan membaca karakter terus menerus sehingga karakter yang terbaca sesuai dengan karakter yang terdeteksi.
13. Setelah pembacaan karakter sudah selesai dan sesuai dengan karakter yang dimasukkan, maka akan proses akan selesai.

3.11 Perencanaan GUI

GUI menggunakan Matlab ini bertujuan agar lebih mudah dalam menggunakan sistem yang dibuat. Tampilan awal GUI pada gambar 3.14



Gambar 3.14 Perancangan GUI

Pada perancangan tampilan aplikasi di atas dapat dijelaskan sebagai berikut :

1. Header Memberikan keterangan tentang aplikasi Sistem Pembacaan Cup Counter Anemometer dan lembaga instansi Badan Meteorologi Klimatologi dan Geofisika Karangploso
2. date /dd/mm/yyyy Memberikan informasi hari , tanggal , bulan dan tahun pada saat aplikasi dijalankan
3. Panel preview merupakan letak untuk menempatkan citra asli dari video streaming Webcam yang akan dilakukan capturing yang kemudian akan dilakukan processing.
4. Panel Capturing and Cropping merupakan merupakan letak untuk menempatkan hasil dari citra asli yang telah dilakukan proses trreshold.
5. Preview adalah tombol untuk menampilkan citra streaming dari webcam
6. Capture adalah tombol untuk menyimpan gambar. Pada proses ini dilakukan penyimpanan kedalam format .bmp
7. Cropping, yaitu pemotongan bagian tertentu dari gambar menjadi matrik baru yang independen dengan tujuan untuk memudahkan dalam

pengolahan pengolahan data yang memisahkan citra karakter dengan objek lain.

8. Proses adalah tombol untuk melakukan pengenalan karakter berdasarkan dari gambar yang disimpan berbentuk .bmp dengan menggunakan metode OCR.
9. Hasil adalah dynamic teks yang digunakan untuk menempatkan hasil dari proses pengenalan terhadap data yang diproses.
10. Simpan adalah tombol yang digunakan untuk menyimpan hasil pembacaan data kedalam database MySQL.
11. Tampilkan history merupakan menu untuk menampilkan data yang akan ditampilkan kedalam table history berdasarkan tanggal, bulan, dan tahun.
12. Tabel History memberikan informasi hasil riwayat pembacaan yang telah dilakukan sebelumnya. menampilkan sesuai tanggal, bulan, dan tahun yang dimasukkan.
13. Keluar adalah tombol untuk keluar dari aplikasi.
14. Footer berisikan informasi penulis.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

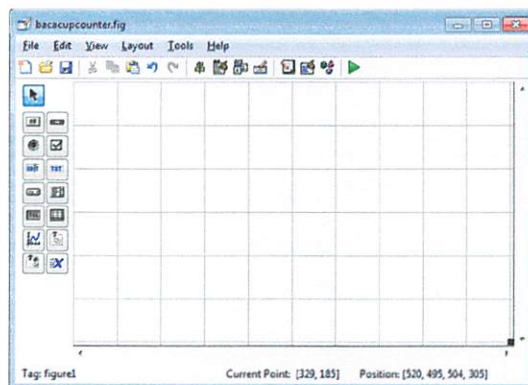
Implementasi merupakan tahap lanjutan dari proses perancangan. Tahap ini merupakan tahap dimana sistem pembacaan alat pengukur kecepatan angin cup counter anemometer dioperasikan, yang terdiri dari penjelasan implementasi program.

4.1.1 Pembuatan Tampilan Aplikasi

Tahapan-tahapan yang diperlukan dalam pembuatan tampilan aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer, antara lain :

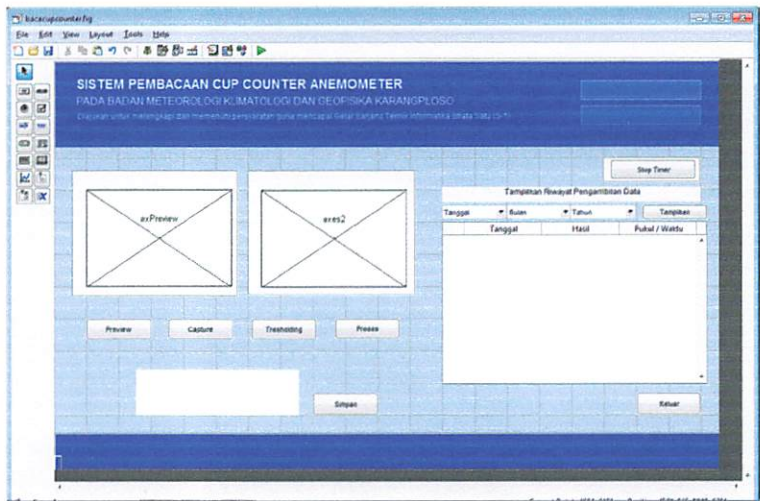
1. Menjalankan perangkat lunak Matlab R2011a (versi 7.12.0).
2. Membuat figure baru dengan nama “bacacupcounter.fig” pada Aplikasi pemrograman Matlab.
3. Melakukan konfigurasi dan design pada figure yang telah dibuat “bacacupcounter.fig” sesuai perancangan yang telah dijelaskan pada bab sebelumnya.

Pembuatan figure tampilan sistem pembacaan alat pengukur kecepatan angin cup counter anemometer, dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan awal bacacupcounter.fig

Setelah membuat tampilan awal pembuatan aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer kemudian membuat tampilan yang sudah dibahas pada bab sebelumnya tentang perancangan. Tampilan yang sudah dilakukan design dapat dilihat pada gambar 4.2



Gambar 4.2 Desain Tampilan Utama

Desain yang ada pada aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer, terdapat berbagai jenis tools yang terdapat pada figure. Tool ini berfungsi untuk memudahkan terhadap pengguna aplikasi matlab untuk membuat desain dan yang berhubungan dengan proses. Tools pada tampilan sistem pembacaan alat pengukur kecepatan angin cup counter anemometer dapat dilihat pada tabel 4.1.

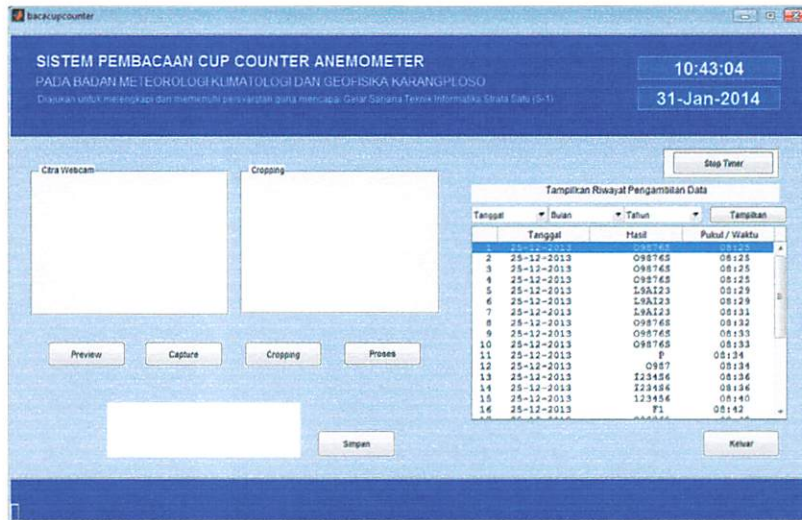
Tabel 4.1 Tag Tools Tampilan Aplikasi

Nama Desain	Nama Tag Tools
Background	figure “bacacupcounter.fig”
Panel Header	uipanel3
Panel Footer	uipanel3
Judul Aplikasi	text 1, text 2, text 3
Timer	edit2

Date	edit3
Tombol Stop Timer	pushbutton10
Panel Citra Webcam	uipanel1
Panel Cropping	uipanel2
Axes Citra Webcam	axPreview
Axes Citra Cropping	axes2
Tombol Preview	pbPreview
Tombol Capture	pbCapture
Tombol Cropping	pbCropping
Tombol Proses	pbProses
Text Hasil	hasil
Tombol Simpan	pushbutton8
Text Riwayat	text4
Menu Tanggal	popupmenu4
Menu Bulan	popupmenu2
Menu Tahun	popupmenu1
Tombol Tampilkan	tampil
Tabel Riwayat	uitable3
Tombol Keluar	keluar

4.1.2 Tampilan Aplikasi

Tampilan aplikasi untuk proses aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer ini dibuat berdasarkan perancangan yang sudah ada dan yang sudah dijelaskan pada bab sebelumnya. Tampilan aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer ini terbuat dari figure yang terdapat pada matlab. Figure aplikasi ini disebut juga GUI (graphical user interface) yang menjadi interface untuk aplikasi sistem pembacaan alat pengukur kecepatan angin cup counter anemometer. Tampilan ditunjukkan pada gambar 4.3.



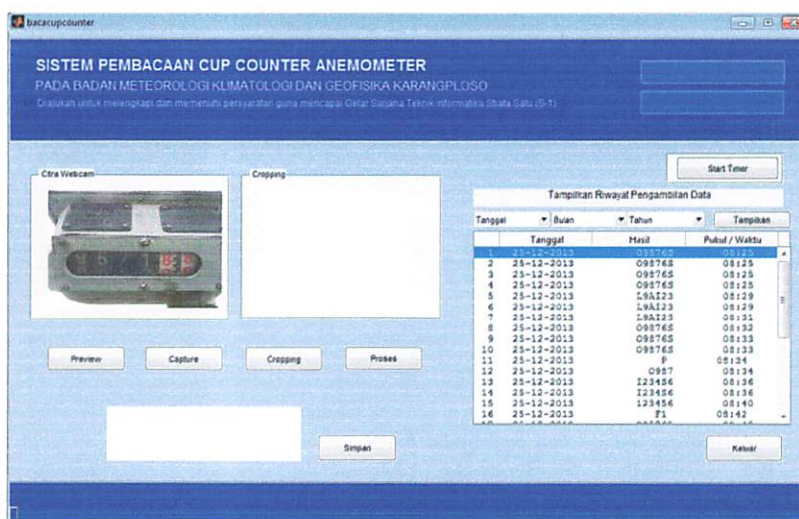
Gambar 4.3 Tampilan Aplikasi

4.1.3 Start Timer

Untuk mengetahui waktu pada saat program aplikasi ini dijalankan. Menampilkan jam, menit dan detik serta tanggal, bulan dan tahun

4.1.4 Preview

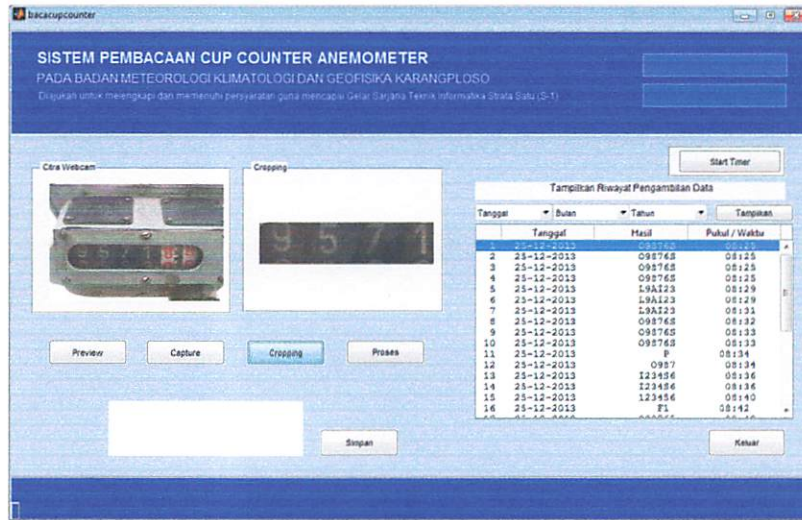
Tombol ini untuk mengaktifkan webcam dan melihat citra yang akan diolah secara streaming.



Gambar 4.4 Tampilan Preview

4.1.5 Tombol Capture

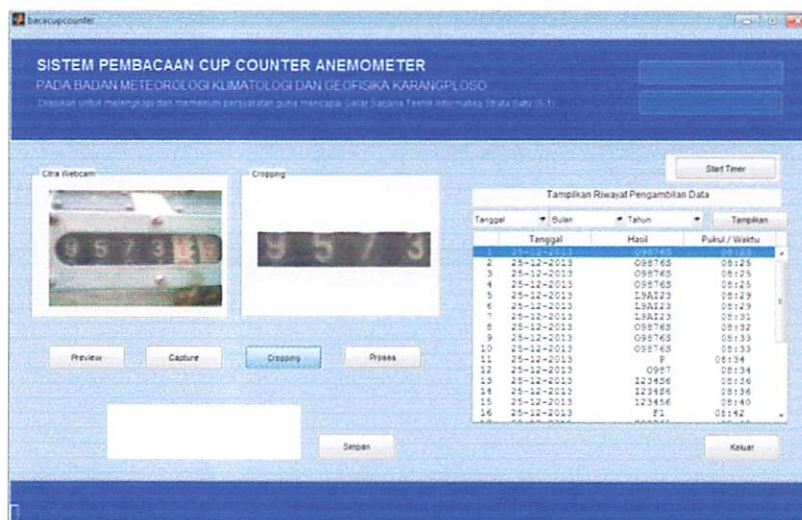
Source code yang diterapkan untuk melakukan penyimpanan gambar dari citra streaming webcam adalah sebagai berikut :



Gambar 4.5 Tampilan Capture

4.1.6 Tombol Cropping

Untuk memudahkan program dalam pengenalan karakter maka dilakukan cropping, berfungsi untuk menghilangkan gambar yang tidak diperlukan dan hanya diambil gambar yang berupa karakter.



Gambar 4.6 Tampilan Cropping

4.1.7 Tombol Proses

Pada tombol ini berisikan source code atau perintah optical character recognition untuk menganali karakter dari data gambar yang telah di crop.



Gambar 4.7 Tampilan Proses

4.1.8 Tombol Simpan

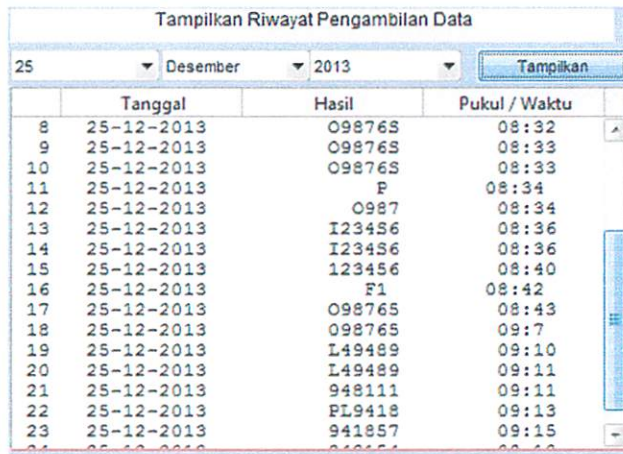
Menyimpan dari hasil pengenalan karakter dan disimpan kedalam database MySql untuk ditampilkan kedalam tabel Riwayat pengambilan data.



Gambar 4.8 Tampilan Simpan

4.1.9 Tombol Tampilkan

Untuk menampilkan riwayat hasil dari pengambilan data yang pernah dilakukan berdasarkan tanggal, bulan, dan tahun. Pada menu ini menampilkan informasi date / tanggal pengambilan, waktu pengambilan data dan hasil pengenalan data.

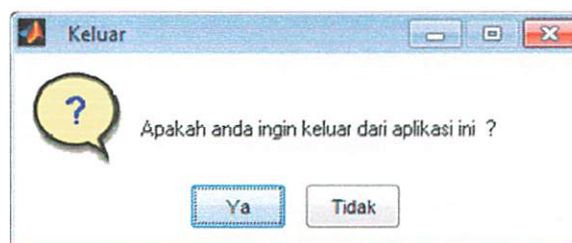


	Tanggal	Hasil	Pukul / Waktu
8	25-12-2013	09876S	08:32
9	25-12-2013	09876S	08:33
10	25-12-2013	09876S	08:33
11	25-12-2013	F	08:34
12	25-12-2013	0987	08:34
13	25-12-2013	I23456	08:36
14	25-12-2013	I23456	08:36
15	25-12-2013	123456	08:40
16	25-12-2013	F1	08:42
17	25-12-2013	09876S	08:43
18	25-12-2013	09876S	09:7
19	25-12-2013	L49489	09:10
20	25-12-2013	L49489	09:11
21	25-12-2013	948111	09:11
22	25-12-2013	FL9418	09:13
23	25-12-2013	941857	09:15

Gambar 4.9 Tampilan Riwayat

4.1.10 Tombol Keluar

Tombol keluar digunakan untuk keluar dari program aplikasi.



Gambar 4.10 Tampilan Dialog Keluar

4.2 Pengujian Aplikasi

Proses pengujian merupakan bagian yang penting karena dalam proses ini akan diketahui apakah program yang dibuat sudah sesuai dengan apa yang diharapkan. Dalam pengujian ini dilakukan beberapa langkah yang berbeda untuk memastikan bahwa program berjalan sesuai keinginan.

1. Pengujian terhadap Paper Number

Pengujian ini dilakukan dengan cara menempatkan sebuah kertas yang bergambar angka dengan tujuan menguji sistem apakah sistem dapat membaca angka yang ditunjukkan pada sebuah kertas tersebut.

2. Pengujian terhadap Cup Counter Anemometer.

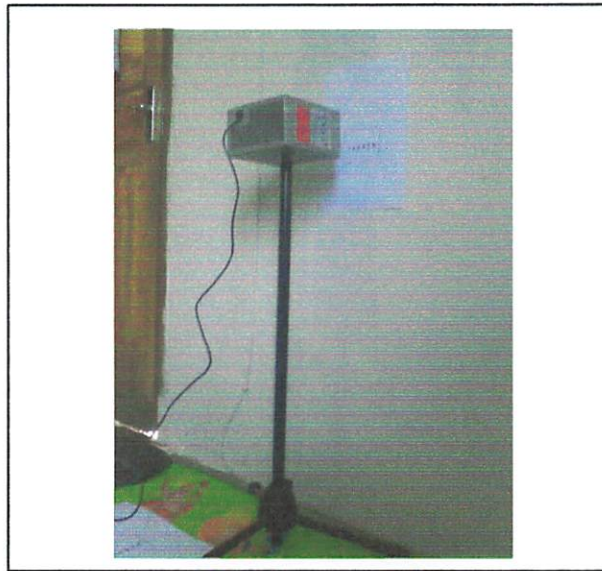
Dengan jarak pengambilan citra yang berbeda.

4.2.1 Prototipe Pengujian

Prototipe pengujian merupakan gambaran umum penempatan webcam untuk pengujian system pembacaan alat pengukur kecepatan sistem cup counter anemometer.

1. Prototipe Pengujian Paper Number.

Gambaran umum penempatan webcam untuk pengujian sistem pembacaan alat pengukur kecepatan angin cup counter anemometer terhadap Paper Number dapat dilihat pada gambar 4.11



Gambar 4.11 Prototipe pengujian terhadap Paper Number.

2. Prototipe Pengujian Cup Counter Anemometer

Gambaran umum penempatan webcam untuk pengujian sistem pembacaan alat pengukur kecepatan angin cup counter dapat dilihat pada gambar 4.12



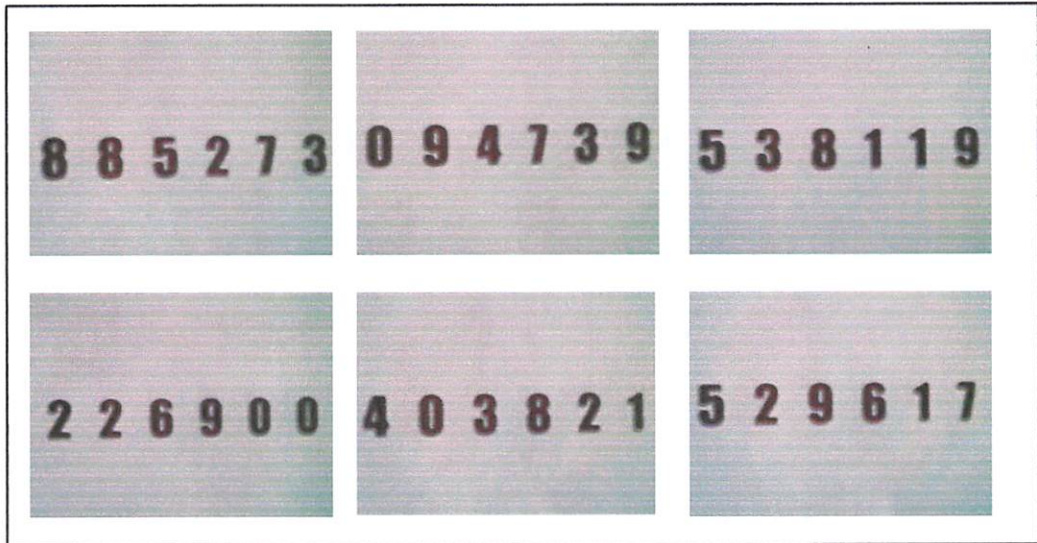
Gambar 4.12 Prototipe Anemometer jarak 10 cm.

4.2.2 Pengujian Berdasarkan Jarak Pengambilan Citra

Proses pengujian berdasarkan jarak ini adalah proses pengujian yang ditujukan pada number yang menggunakan sampel pengujian jarak pengambilan gambar yang tidak sama. Pengujian ini menggunakan proses pengambilan gambar dengan webcam, seperti pada contoh sampel berikut ini :

1. Citra Paper Number yang diambil dari jarak 5 Cm.

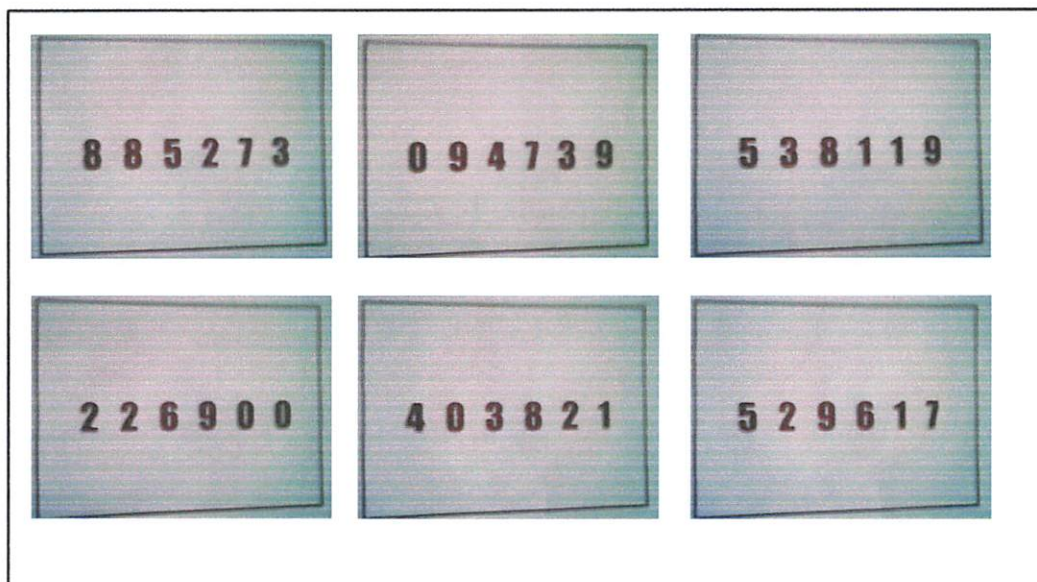
sampel citra ini menggunakan jarak pengambilan yaitu 5 cm pada Paper Number. Dengan menggunakan webcam resolusi 640X480 px. Hasil pengambilan gambar ditunjukkan pada gambar 4.13



Gambar 4.13 Sampel Citra Data Paper Number 5 cm

2. Citra Paper Number yang diambil dari jarak 10 Cm.

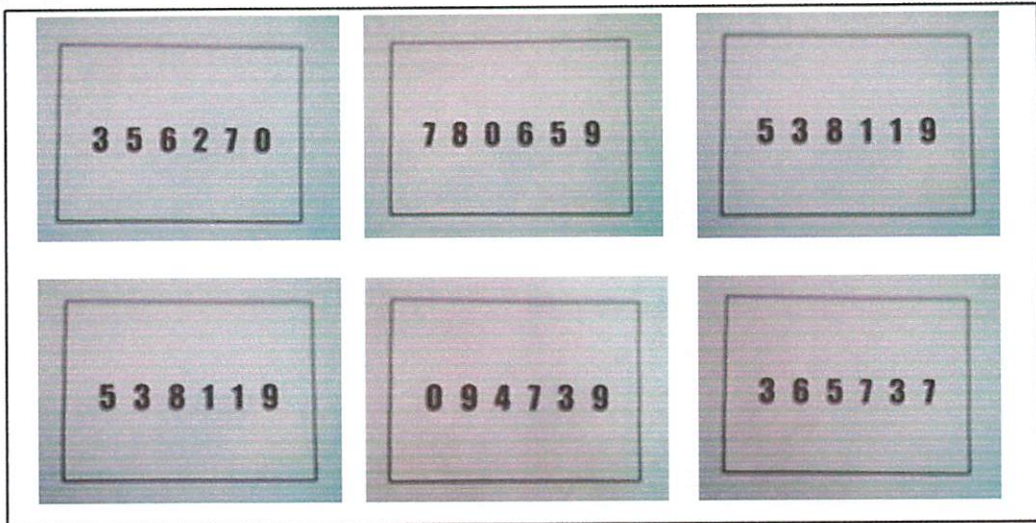
sampel citra ini menggunakan jarak pengambilan yaitu 10 cm pada Paper Number. Dengan menggunakan webcam resolusi 640X480 px. Hasil pengambilan gambar ditunjukkan pada gambar 4.14



Gambar 4.14 Sampel Citra Data Paper Number 10 cm

2. Pengambilan citra dari jarak 15 cm terhadap Paper Number.

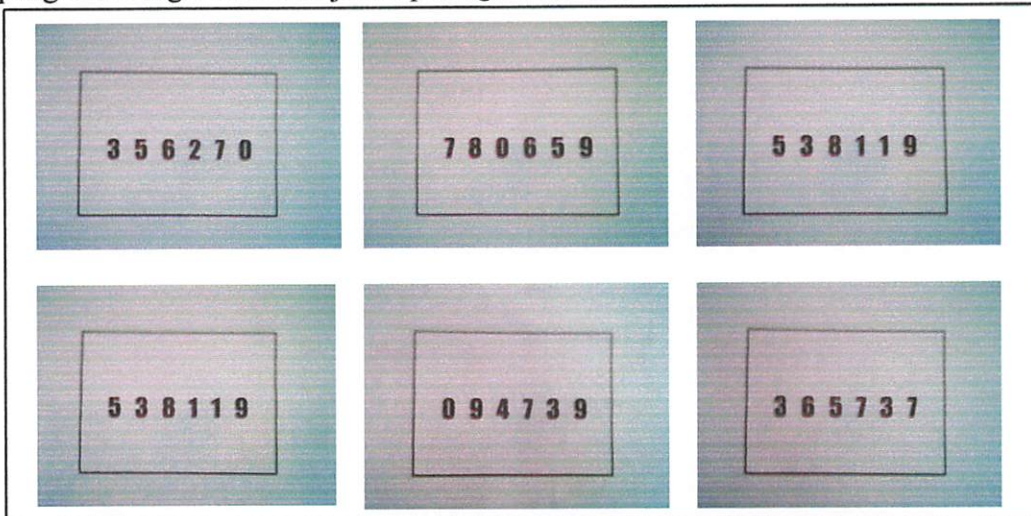
Sampel citra ini menggunakan jarak pengambilan yaitu 15 cm pada Paper Number. Dengan menggunakan webcam resolusi 640X480 px. Hasil pengambilan gambar ditunjukkan pada gambar 4.15



Gambar 4.15 Sampel Citra Data Paper Number 15cm.

3. Pengambilan citra dari jarak 20 cm terhadap Paper Number.

Sampel citra ini menggunakan jarak pengambilan yaitu 20 cm pada Paper Number. Dengan menggunakan webcam resolusi 640X480 px. Hasil pengambilan gambar ditunjukkan pada gambar 4.16



Gambar 4.16 Sampel Citra Data Paper Number 20cm.

3. Citra Cup Counter Anemometer diambil dari jarak 10 cm sampel citra dengan jarak pengambilan yaitu 10 cm pada cup counter anemometer. Dengan menggunakan webcam resolusi 640X480 px. Hasil pengambilan gambar ditunjukkan pada gambar 4.17



Gambar 4.17 Sampel Citra Data Anemometer jarak 10 cm.

4.2.3 Pengujian Berdasarkan Jenis Font

Proses pengujian berdasarkan Jenis Font adalah proses pengujian yang ditujukan pada number yang menggunakan sampel pengujian jenis font yang tidak sama. Pengujian ini menggunakan proses pengambilan gambar dengan webcam, jenis font yang digunakan dalam pengujian ini adalah berikut :

Tabel 4.2 Pengujian berdasarkan jenis Font

No	Jenis Font	Sample Font
1	Tahoma	5657689
2	Times New Roman	5657689
3	Arial	5657689
4	Impact	5657689
5	MV Boli	5657689

4.2.4 Hasil Pengujian

Hasil pengujian berdasarkan penempatan kamera dan dari jarak yang telah tidak sama adalah sebagai berikut :

- 1. Hasil pengujian terhadap Paper Number dengan jarak berbeda ditunjukkan pada tabel 4.3

Tabel 4.3 Hasil Pengujian

No	Jenis Font	Data Number	Hasil Pembacaan	Keterangan
1	5 Cm.	780650	780650	Berhasil
		356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
		689114	689114	Berhasil
		582092	582092	Berhasil
		753084	753084	Berhasil
		930199	930199	Berhasil
		018942	018942	Berhasil
		780650	780650	Berhasil
2	10 Cm.	356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil

3	15 Cm	094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
		689114	689114	Berhasil
		582092	582092	Berhasil
		753084	753084	Berhasil
		930199	930199	Berhasil
		018942	018942	Berhasil
	15 Cm	356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
		689114	689114	Berhasil
		582092	582092	Berhasil
		753084	753084	Berhasil
		930199	930199	Berhasil
		018942	018942	Berhasil
	20 Cm.	356270	7356270	Gagal
		365737	7365737	Gagal
		226900	7226900	Gagal
		529617	7529617	Gagal
		403821	7403821	Gagal
		538119	7538119	Gagal
		094739	7094739	Gagal
		885273	7885273	Gagal

	057803	7057803	Gagal
	689114	7689114	Gagal
	582092	7582092	Gagal
	753084	7753084	Gagal
	930199	7930199	Gagal
	018942	7018942	Gagal

2. Hasil pengujian terhadap Paper Number dengan jarak berbeda ditunjukkan pada table 4.4

Tabel 4.4 Hasil Pengujian Berdasarkan Jenis Font

No	Jarak Pengambilan Citra	Data Number	Hasil Pembacaan	Keterangan
1	Tahoma	780650	780650	Berhasil
		356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
2	Times New Roman	780650	780650	Berhasil
		356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil

		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
3	Arial	780650	780650	Berhasil
		356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
4	Impact	780650	780650	Berhasil
		356270	356270	Berhasil
		365737	365737	Berhasil
		226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil
		<i>780650</i>	780650	Berhasil
		<i>356270</i>	356270	Berhasil
		<i>365737</i>	365737	Berhasil

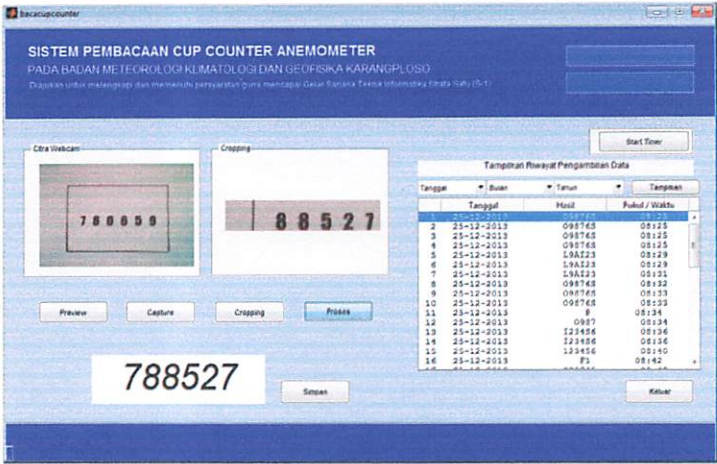
5	MV Boli	226900	226900	Berhasil
		529617	529617	Berhasil
		403821	403821	Berhasil
		538119	538119	Berhasil
		094739	094739	Berhasil
		885273	885273	Berhasil
		057803	057803	Berhasil

Tampilan pengujian dengan jarak 10 Cm dapat dilihat pada gambar 4.18



Gambar 4.18 Pengujian Paper Number 10 cm.

Tampilan pengujian dengan jarak 15 Cm dapat dilihat pada gambar 4.19



Gambar 4.19 Pengujian Paper Number 15 cm.

1. Hasil pengujian terhadap Cup Counter Anemometer dengan jarak 10 Cm ditunjukkan pada tabel 4.5

Tabel 4.5 Hasil Pengujian cup counter jarak 10 cm.

No	Citra yang diambil	Prosses Cropping	Prosses Threshold	Pembacaan	Ket.
1				9 5 7 3	Berhasil
2				9 5 7 7	Berhasil
3				9 5 7 9	Berhasil
4				9 5 8 3	Berhasil

Tampilan pengujian dengan jarak 10 Cm dapat dilihat pada gambar 4.20



Gambar 4.20 Hasil Pengujian cup 10 cm

Dari pengujian yang sudah dilakukan didapat bahwa pengujian sistem pembacaan alat pengukur kecepatan angin cup counter anemometer yang diterapkan pada paper number dan alat cupcounter anemometer dapat menunjukkan hasil yang diharapkan. Pengujian sistem yang diterapkan pada paper number berhasil 100% sesuai dengan yang diharapkan.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah membuat aplikasi Sistem Pembacaan Cup Counter Anemometer Menggunakan Webcam dengan Optical Character Recognition Pada BMKG Karangploso dan membuat laporan, maka dapat disimpulkan :

1. Proses pengujian dinilai 100% berhasil, dengan hasil pembacaan 4 digit angka depan yang ditunjukkan oleh display pada Cup Counter Anemometer terbaca sesuai yang diharapkan.
2. Pengambilan jarak citra dengan webcam mempengaruhi hasil pengenalan karakter oleh sistem.

5.2. Saran

Setelah melakukan pembuatan aplikasi Sistem Pembacaan Cup Counter Anemometer Menggunakan Webcam dengan Optical Character Recognition Pada BMKG Karangploso, maka pengembangan yang dapat dilakukan :

1. Aplikasi Sistem Pembacaan Cup Counter Anemometer ini dapat dikembangkan dengan menggunakan IP Camera. Sehingga monitoring data display Cup Counter Anemometer dapat dilakukan dimana saja dan kapan saja.
2. Sistem ini dapat dikembangkan dengan menambah lampu LED pada area box kamera sehingga penangkapan citra maksimal pada saat terjadi hujan dan pencahayaan cuaca tidak stabil.

DAFTAR PUSTAKA

Erick Paulus, Yessica Nataliani. 2007. *"GUI Matlab"*. Yogyakarta : Penerbit Andi.

Gunaidi, Abdia Away. 2006. *"Matrix Laboratory Matlab Programing"*. 1st edition, Bandung: Informatika.

Jek Siang M.Sc. 2005. *"Jaringan Syaraf Tiruan dan Pemrograman Menggunakan Matlab"*. 1st edition. Yogyakarta : Andi.

Prasetyo, Eko. 2011. *"Pengolahan Citra Digital dan Aplikasinya menggunakan Matlab"*. Yogyakarta: Penerbit Andi.

Sofian Bahri, Raden., Irfan Maliki. 2012. *"Perbandingan Algoritma Template Matching Dan Feature Extraction Pada Optical Character Recognition"*. Program Studi Teknik Informatika. Universitas Komputer Indonesia. Bandung.

Sugiharto, Aris. 2006. *"Pemrograman GUI dengan MATLAB"*. Yogyakarta : Penerbit Andi.

Kusumadewi, Sri. 2003. *"Artifical Intelegence Teknik dan Aplikasinya"*. Yogyakarta: Graha Ilmu.

Naba, Agus. 2009. *"Belajar Cepat Fuzzy Logic Menggunakan Matlab"*. Yogyakarta : Penerbit Andi.

LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

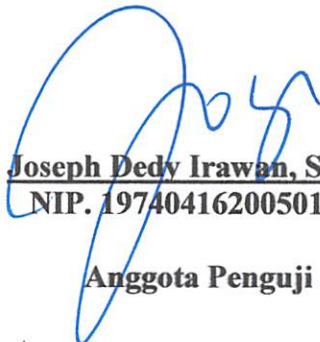
Nama : Muhamad Ma'shum Arif
NIM : 0918157
Jurusan : Teknik Informatika S-1
Judul : Sistem Pembacaan Cup Counter Anemometer Menggunakan Webcam Dengan Optical Character Recognition Pada BMKG Karangploso

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1)
pada :

Hari : Rabu
Tanggal : 19 Februari 2014
Nilai : 87,5 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji

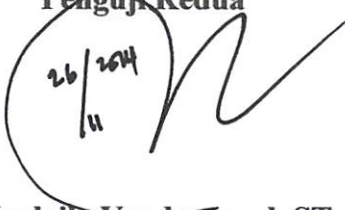

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Penguji Pertama


Febriana Santi W, Skom, Mkom
NIP.P. 1031000425

Penguji Kedua


Nurlaily Vendyansyah, ST



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Muhamad Ma'shum Arif
NIM : 0918157
Jurusan : Teknik Informatika S-1
Judul : Sistem Pembacaan Cup Counter Anemometer Menggunakan Webcam
Dengan Optical Character Recognition Pada BMKG Karangploso

Tanggal	Penguji	Uraian	Paraf
19 Februari 2014	I	1. Tambahkan Teori Citra, Operasi yang digunakan	
19 Februari 2014	II	1. Tambahkan Teori Citra, Operasi yang digunakan 2. Kesimpulan 3. Tabel Pengujian 4. Flowchart Pre-Processing	

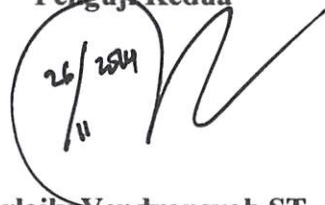
Anggota Penguji :

Penguji Pertama



Febriana Santi W, Skom, Mkom
NIP.P. 1031000425

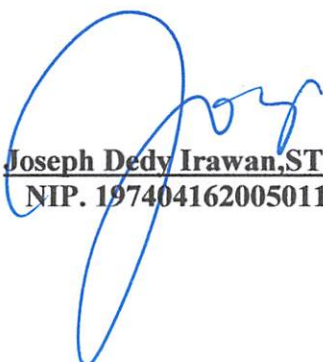
Penguji Kedua



Nurlaily Vandyansyah,ST

Mengetahui

Dosen Pembimbing I



Joseph Dedy Irawan,ST, MT
NIP. 197404162005011002

Dosen Pembimbing II



Sonny Prasetyo, ST.MT
NIP.P. 1031000433



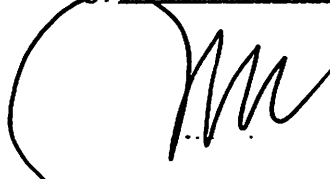
FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : MUHAMMAD MA'SHUM ARIF
NIM : 09.18.157
Perbaikan Meliputi :

1. Tambahkan Teori Citra , Operasi yang digunakan apasaja ?
2. Kesimpulan
3. Tabel pengujian
4. Flowchart preprocessing .

Malang, 19 FEBRUARI 2014


(NURLAILI)



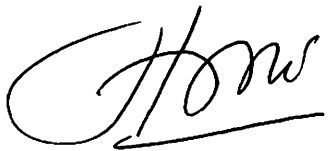
INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1

FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : MUHAMMAD MA'SHUM ARIF
NIM :
Perbaikan Meliputi :

Malang, 15-2-2014


(FEBRIANA S.W)



FORMULIR PERBAIKAN SEMINAR JUDUL

Dalam pelaksanaan Seminar Judul Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : Muhammad Ma'shum Arif
NIM : 0918157
Perbaikan Meliputi : _____

1. Tambahkan referensi

2. Tuliskan tentang cara penempatan Webcam

Malang, 27 - 4 - 2013

(SENYUT PRASETYU, S.H.)



FORMULIR PERBAIKAN SEMINAR JUDUL

Dalam pelaksanaan Seminar Judul Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : Mohammad Ma'shum Arij
NIM : 091 257
Perbaikan Meliputi : _____

1) Perbaiki daftar pustaka.
Gunakan buku, e-book

Malang, 27 April 2013

(AL) MAH UD, I



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

(PERSERO) MALANG
KAMPUS NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 20 Januari 2014

Nomor : ITN-1-227/IX.T.INF/2014
Aspek : -
Materi : PENELITIAN SKRIPSI

Kepada : Yth. Bapak Amat Subekti, SSI
Kepala Badan Meteorologi Klimatologi dan Geofisika (BMKG)
Jl. Zentana No.33 Karangploso
Malang

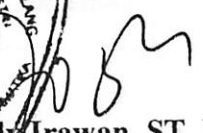
Dengan hormat,

Bersama dengan surat ini kami mohon kebijaksanaan Bapak/Ibu agar Mahasiswa kami dari **Fakultas Teknologi Industri, Program Studi Teknik Informatika S-1** dapat di ijinkan untuk melaksanakan Survey, untuk keperluan penelitian skripsi.

Survey akan dilakukan pada : 23 Januari 2014 s/d 24 Januari 2014
Adapun mahasiswa tersebut adalah :

M. Ma'shum Arif NIM : 09.18.157

Setelah melaksanakan survey, hasil dari survey akan digunakan untuk penulisan laporan penelitian/skripsi.
Demikian agar maklum dan atas perhatian serta bantuannya kami ucapkan banyak terima kasih.

Program Studi
Teknik Informatika S-1
Ketua

Joseph Dedy Irawan, ST, MT.
NIP. 19740416 200501 1 002

Tembusan Kepada :
1. Arsip



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

NI (PERSERO) MALANG
ANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-161/T.INF/TA/2013
Lampiran : -
Perihal : Bimbingan Skripsi

10 Oktober 2013

Kepada : Yth. Sdr. Joseph Dedy Irawan, ST, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

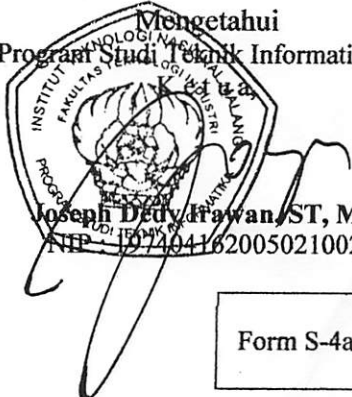
Nama : MUHAMMAD MA'SUM ARIF
Nim : 0918157
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

12 November 2013 – 12 Mei 2014

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1

Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002

Form S-4a