

**REKAYASA**

**LAPORAN PENELITIAN**  
Nomor : 12/Sugas-Pen/LPPM-UKP/2012



**KLASIFIKASI OTOMATIS KELOMPOK BINTIK MATAHARI UNTUK  
MENGANALISA KONDISI CUACA ANTARIKSA**

Oleh:

1. Gregorius Satia Budhi, M.T.
2. Rudy Adipranata, M.Eng.
3. Bambang Setiahadhi, D.Sc.

Dibiayai oleh : LPPM Universitas Kristen Petra Tahun 2011/2012  
Sesuai Surat Pelaksanaan Hibah Penelitian Nomor : 12/Sugas-Pen/LPPM-UKP/2012  
dan Surat Keterangan Perpanjangan Penelitian Nomor: 01/Suket-Pen/LPPM-UKP/2012

**FAKULTAS TEKNOLOGI INDUSTRI**  
**UNIVERSITAS KRISTEN PETRA, SURABAYA**  
**NOPEMBER 2012**

## HALAMAN PENGESAHAN LAPORAN PENELITIAN

1. Judul Penelitian : Klasifikasi Otomatis Kelompok Bintang Matahari Untuk Menganalisa Kondisi Cuaca Antariksa
2. Ketua Peneliti
- a. Nama Lengkap : Gregorius Satia Budhi, M.T.
  - b. Jenis Kelamin : Pria
  - c. NIP/Golongan : 02-030
  - d. Strata/Jabatan Fungsional : S2 / Lektor
  - e. Jabatan Struktural : Kepala Bidang Studi Informatika
  - f. Bidang Keahlian : Kecerdasan Buatan, Jaringan Syaraf Tiruan, Data Mining
  - g. Fakultas/Jurusan : Fakultas Teknologi Industri / Teknik Informatika
  - h. Perguruan Tinggi : Universitas Kristen Petra
  - i. Tim Peneliti :

No.	Nama dan Gelar Akademik	Bid. Keahlian	Fak./Prodi.	Perguruan Tinggi / Lembaga
1.	Rudy Adipranata, M.Eng	Computer Vision, Pengolahan Citra Digital	Fakultas Teknologi Industri/ Prodi Teknik Informatika	Universitas Kristen Petra
2.	Bambang Setiahadhi, D.Sc.	Simulasi Magnetohidrodinamika (MHD) Matahari	-	Lembaga Penerbangan dan Antariksa Nasional

3. Lokasi Penelitian : Surabaya dan Watukosek Pasuruan
4. Jangka Waktu Penelitian : 11 (sebelas) bulan
5. Pembiayaan
- a. Biaya total yang diusulkan : Rp. 7,500,000
  - b. Biaya yang disetujui tahun 1 : Rp. 7,500,000

Mengetahui,  
Dekan Fakultas Teknologi Industri

Surabaya, 22 Nopember 2012  
Ketua Peneliti,

Ir. Djoni Haryadi Setiabudi, M.Eng  
NIP. 85-009

Gregorius Satia Budhi, M.T.  
NIP. 02-030

Menyetujui,  
Kepala LPPM

Prof. Ir. Lilianny Sigit Arifin, M.Sc.,Ph.D  
NIP. 84-011

## RINGKASAN DAN SUMMARY

Matahari merupakan sumber energi tak terbatas bagi kehidupan di bumi, namun dapat pula memberikan ‘gangguan’ yang mempengaruhi lingkungan antariksa sekitar bumi serta kehidupan di bumi, dalam bentuk badai matahari (*solar storm*). Gangguan tersebut perlu dideteksi sedini mungkin sehingga dapat dilakukan langkah-langkah antisipasi yang diperlukan. Gangguan yang terjadi pada umumnya berasal dari kelompok bintik matahari yang terbentuk sebagai akibat dari tabung medan magnet (*magnetic flux tube*) muncul dari dalam matahari ke fotosfer dan korona. Citra bintik matahari dapat diambil dengan menggunakan instrumen *Michelson Doppler Imager* (MDI) *Continuum/SOHO* (*Solar dan Heliospheric Observatory*).

Pada penelitian yang bekerja sama dengan LAPAN (Lembaga Penerbangan dan Antariksa Nasional) ini, akan dilakukan pengembangan sistem untuk melakukan klasifikasi otomatis terhadap kelompok bintik matahari yang muncul. Sistem ini nantinya dapat digunakan sebagai alat bantu untuk menganalisa kondisi cuaca antariksa dan memberikan informasi kepada masyarakat. Penelitian ini dilakukan selama tiga tahun dimana pada tahun kedua ada dua tujuan penelitian. Tujuan pertama menguji sub-sistem ekstraksi fitur serta database penyimpanan fitur. Tujuan berikutnya adalah membangun tiga sub-sistem klasifikasi dengan metode jaringan saraf tiruan yang akan dibandingkan akurasi. Analisa dan pemilihan salah satu metode untuk digunakan, dilakukan pada tahun ketiga.

Penelitian pada tahun kedua dapat dikelompokkan menjadi tiga tahap, yaitu: tahap pengujian sub-sistem ekstraksi fitur dan database penyimpanan fitur yang dibangun pada tahun pertama. Secara paralel, sambil melakukan verifikasi kebenaran hasil sub sistem ekstraksi fitur, peneliti LAPAN melakukan klasifikasi manual terhadap hasil ekstraksi fitur kelompok bintik matahari. Hasil klasifikasi manual ini digunakan sebagai data training sub-sistem klasifikasi otomatis. Tahap selanjutnya adalah pembuatan sub-sistem untuk tiga metode klasifikasi kelompok bintik matahari yang nantinya akan dianalisa dan dipilih satu yang terbaik pada awal tahun ketiga. Adapun ketiga metode klasifikasi yang dibangun dan dibandingkan itu adalah jaringan saraf tiruan *backpropagation* (BNN), *probabilistic* (PNN) dan *self-organizing map* (SOM). Untuk memudahkan proses perbandingan ketiga metode jaringan saraf tiruan, telah dibangun 6 buah prototipe sub-sistem. Dimana untuk setiap metode jaringan saraf tiruan, dibangun dua buah prototipe. Setiap prototipe selanjutnya ditraining dengan data sample yang telah diklasifikasi manual. Dari hasil pengujian yang dilakukan menggunakan data yang belum pernah ditrainingkan didapat hasil terbaik pada salah satu prototipe, yaitu yang menggunakan metode jaringan saraf

tiruan *probabilistic*, dengan tingkat akurasi maksimal sebesar 94%. Namun masih terlalu dini untuk menentukan apakah metode *probabilistic* yang dipilih, karena bisa jadi metode klasifikasi lain yang bukan termasuk metode JST akan lebih baik tingkat akurasinya. Pada tahun ketiga ini dibangun sebuah prototipe sub-sistem klasifikasi menggunakan metode klasifikasi ID3 / C4.5. Metode ini dipilih dengan mengikuti saran dari penelitian lain tentang klasifikasi bintik matahari (Nguyen, 2006). Dari analisa perbandingan terlihat bahwa sub-sistem klasifikasi bintik matahari dengan metode JST PNN masih memiliki akurasi lebih baik dibanding metode klasifikasi ID3 / C4.5. Selanjutnya pada tahun ketiga semua ini sub-sistem disatukan menjadi sebuah sistem lengkap untuk klasifikasi otomatis badai matahari (kelompok bintik matahari).

## **PRAKATA**

Kami sebagai penulis pertama - tama mengucapkan syukur kepada Tuhan Yang Maha Esa atas terselesaikannya penelitian hibah internal ini dan mendapatkan hasil sesuai dengan harapan.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada Pusat Penelitian Universitas Kristen Petra yang telah mendanai penelitian ini sehingga hasil penelitian yang diharapkan selesai secara utuh dan menyeluruh dapat tercapai.

Penulis, sebagai manusia yang jauh dari kesempurnaan, sadar bahwa hasil penelitian ini kemungkinan banyak terdapat kekurangan disana - sini. Oleh sebab itu penulis mengharapkan saran dan kritik yang membangun dari berbagai pihak demi kemajuan dan hasil terbaik bagi penelitian ini. Akhir kata, penulis berharap semoga penelitian ini dapat memberikan kontribusi bagi masyarakat serta nusa dan bangsa Indonesia.

Surabaya, 22 Nopember 2012

Penulis

# DAFTAR ISI

RINGKASAN DAN SUMMARY .....	iii
PRAKATA .....	v
DAFTAR ISI .....	vi
DAFTAR TABEL .....	viii
DAFTAR GAMBAR .....	ix
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan .....	2
1.3 Ruang Lingkup Pembahasan .....	3
1.4 Sistematika Penyusunan Laporan .....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Penelitian Sebelumnya yang Menunjang.....	5
2.2 Bintang Matahari.....	5
2.3 Jaringan Syaraf Tiruan.....	7
2.3.1 <i>Probabilistic Neural Network</i> .....	7
2.4 <i>Data Preprocessing</i> .....	11
2.4.1 <i>Min-Max Normalization</i> .....	11
2.4.2 <i>Image Processing</i> .....	11
2.5 <i>Decision Tree Induction</i> .....	12
2.6 <i>ID3</i> .....	12
2.6.1 <i>Information gain</i> .....	13
2.7 <i>C4.5</i> .....	14
2.7.1 <i>Gain ratio</i> .....	14
2.8 <i>Intuitive partitioning</i> .....	15
2.9 <i>Gini index</i> .....	15
BAB 3. TUJUAN DAN MANFAAT PENELITIAN .....	18
3.1 Tujuan Penelitian .....	18
3.2 Manfaat Penelitian .....	19
BAB 4. METODE PENELITIAN.....	20
BAB 5. HASIL DAN PEMBAHASAN.....	23
5.1 Perbandingan akurasi klasifikasi dari tiga macam sub-sistem Jaringan Syaraf Tiruan....	23

5.2	Desain sub-sistem klasifikasi kelompok bintang matahari dengan ID3 / C4.5.....	24
5.2.1	Metode Segmentasi .....	28
5.2.2	<i>Intuitive Partitioning</i> .....	28
5.3	<i>Gini Index</i> .....	30
5.4	Garis Besar Modul Training .....	35
5.5	Garis Besar Modul Klasifikasi.....	40
5.6	Antarmuka sub-sistem klasifikasi kelompok bintang matahari dengan ID3 / C4.5.....	41
5.7	Pengujian prototipe sub-sistem klasifikasi dengan metode ID3/C4.5.....	43
5.8	Software Sistem Klasifikasi Otomatis Tipe Badai Matahari .....	45
5.9	Antarmuka Sistem Klasifikasi Otomatis Tipe Badai Matahari .....	46
BAB 6.	KESIMPULAN DAN SARAN.....	53
6.1	Kesimpulan.....	53
6.2	Saran .....	53
DAFTAR	PUSTAKA.....	205
LAMPIRAN	.....	207

## DAFTAR TABEL

Tabel 5.1 Akurasi klasifikasi berdasarkan perubahan jumlah data training (dalam %)	43
Tabel 5.2 Akurasi klasifikasi berdasarkan perubahan jumlah region (dalam %)	44
Tabel 5.3 Perbandingan akurasi klasifikasi dengan dan tanpa proses segmentasi data (dalam %)	45



## DAFTAR GAMBAR

Gambar 2.1 Klasifikasi “ <i>Modified-Zurich Sunspot Classification</i> ” .....	6
Gambar 2.2 Arsitektur <i>Probabilistic Neural Network</i> .....	8
Gambar 2.3 Arsitektur <i>Probabilistic Neural Network</i> .....	8
Gambar 2.4 Arsitektur <i>PNN</i> yang Digunakan.....	10
Gambar 4.1 Bagan Alir Penelitian Tahun Ketiga.....	20
Gambar 5.1 Perbandingan akurasi klasifikasi untuk variasi jumlah data training .....	23
Gambar 5.2 Perbandingan akurasi klasifikasi untuk variasi pembagian region citra digital .....	24
Gambar 5.3 <i>Flowchart</i> garis besar aplikasi.....	25
Gambar 5.4 Konsep Penggunaan <i>Region</i> .....	25
Gambar 5.5 <i>Flowchart addregion</i> .....	26
Gambar 5.6 <i>Flowchart makesegmentation</i> .....	27
Gambar 5.7 <i>Flowchart Intuitive Partitioning</i> .....	28
Gambar 5.8 <i>Flowchart Intuitive Partitioning</i> (lanjutan) .....	29
Gambar 5.9 <i>Flowchart giniindex</i> .....	30
Gambar 5.10 <i>Flowchart giniindex</i> (lanjutan) .....	31
Gambar 5.11 <i>Flowchart giniindexprocess</i> .....	32
Gambar 5.12 <i>Flowchart setnolfrekuensi gini</i> .....	33
Gambar 5.13 <i>Flowchart setfrekuensi gini</i> .....	34
Gambar 5.14 <i>Flowchart getgini</i> .....	35
Gambar 5.15 <i>Flowchart</i> alur modul <i>training</i> .....	36
Gambar 5.16 <i>Flowchart settabelentropy</i> .....	37
Gambar 5.17 <i>Flowchart getentropy</i> .....	38
Gambar 5.18 <i>Flowchart setruletable</i> .....	39
Gambar 5.19 <i>Flowchart</i> alur modul <i>classification</i> .....	40
Gambar 5.20 Antarmuka Awal Prototipe.....	41
Gambar 5.21 Antarmuka untuk proses training data.....	41
Gambar 5.22 Antarmuka tampilan Decision Tree yang dihasilkan dari proses training .....	42
Gambar 5.23 Antarmuka untuk menampilkan hasil proses klasifikasi dengan ID3/C4.5. ....	42
Gambar 5.24 Perbandingan metode klasifikasi PNN dan ID3/C4.5 .....	45
Gambar 5.25 Desain Sistem Keseluruhan.....	46
Gambar 5.26 Antarmuka awal Sistem Kasifikasi Otomatis Tipe Badai Matahari.....	47

Gambar 5.27 Antarmuka awal saat menu “Input Gambar” pada tampilan awal dipilih .....	47
Gambar 5.28 Antarmuka hasil akhir saat menu “Automatic Process” pada tampilan “Input Gambar” dipilih.....	48
Gambar 5.29 Antarmuka “Klasifikasi Manual” .....	49
Gambar 5.30 Antarmuka proses training JST Probabilistik.....	50
Gambar 5.31 Antarmuka Klasifikasi Otomatis saat pertama kali dipanggil.....	51
Gambar 5.32 Antarmuka Klasifikasi Otomatis setelah sebuah proses klasifikasi berakhir .....	51
Gambar 5.33 Antarmuka untuk verifikasi dan update pada proses pindah data .....	52

# BAB 1. PENDAHULUAN

## 1.1 Latar Belakang

Sebagai sebuah sumber energi yang tak terbatas bagi kehidupan di bumi, matahari ternyata juga memberikan ‘gangguan’ ke lingkungan antariksa sekitar bumi serta kehidupan di bumi dalam bentuk badai matahari (*solar storm*). Gangguan ini dapat sangat mempengaruhi kehidupan manusia khususnya dengan ketergantungan manusia terhadap teknologi modern saat ini, sehingga kemunculan gangguan tersebut haruslah diantisipasi.

Terdapat dua tipe gangguan dari matahari yaitu berupa ledakan dahsyat (*flare*) dan pelontaran massa korona (*Coronal Mass Ejection/CME*) dimana keduanya pada umumnya berasal dari kelompok bintik matahari (*sunspot*). Fenomena bintik matahari terbentuk sebagai akibat dari tabung medan magnet (*magnetic flux tube*) muncul dari dalam matahari ke fotosfer dan korona (Setiahadi *et.al.*, 2006). Perpotongan tabung medan magnet dengan fotosfer membentuk suatu bintik matahari, yang tampak hitam dikarenakan medan magnet memiliki efek ‘pendinginan’ sehingga suhu (temperatur) bintik matahari lebih rendah dari fotosfer di sekitarnya (Bothmer and Daglis, 2007).

Bintik matahari berevolusi dari bintik kecil dengan aktivitas rendah berkembang menjadi konfigurasi yang sangat kompleks dengan kemungkinan mempunyai aktivitas tinggi, yaitu mengeluarkan ledakan-ledakan (*flares*) ataupun pelontaran massa korona (CME). Untuk mengetahui tingkat kompleksitas bintik matahari dan aktivitasnya digunakan klasifikasi bintik matahari *Modified-Zurich* guna mengklasifikasikan jenis bintik matahari tersebut. Informasi mengenai klasifikasi ini sangat penting untuk menganalisa cuaca antariksa (*space weather*).

LAPAN sebagai satu-satunya lembaga penelitian pemerintah non-departemen (LPND) mempunyai tugas dan fungsi memberikan informasi tentang kondisi cuaca antariksa kepada masyarakat, serta melakukan prakiraan (*forecasting*) tentang kemungkinan gangguan dari matahari yang mempunyai potensi memberikan gangguan (bencana) terhadap lingkungan antariksa di sekitar bumi dan kehidupan manusia di bumi. Penelitian ini diharapkan memberikan kemudahan bagi LAPAN untuk menyampaikan informasi tentang kondisi cuaca antariksa kepada masyarakat secara cepat.

Dari permasalahan tersebut, maka dibutuhkan sebuah aplikasi yang dapat mendeteksi dan mengelompokkan bintik-bintik matahari dari sebuah citra *digital*. Aplikasi ini dibangun selama tiga tahun. Pada tahun pertama telah dibangun sub-sistem untuk perbaikan kualitas citra matahari, segmentasi bintik matahari dengan metode *watershed*, serta ekstraksi fitur kelompok bintik

matahari yang memanfaatkan metode data mining DbScan. Di samping itu juga dilakukan pembangunan basis data untuk penyimpanan data tersebut.

Langkah kerja pada tahun kedua dapat dikelompokkan menjadi dua tahap, yaitu: tahap penyempurnaan ekstraksi fitur dan tahap pembuatan tiga metode klasifikasi kelompok bintang matahari yang nantinya akan dipilih satu yang terbaik pada awal tahun ketiga. Adapun ketiga metode klasifikasi yang dibangun dan dibandingkan itu adalah jaringan saraf tiruan *backpropagation* (BNN), *probabilistic* (PNN) dan *self-organizing map* (SOM).

Pada tahun ketiga ini, dilakukan analisa dan penentuan metode jaringan syaraf tiruan mana yang digunakan. Dimana akhirnya dipilih jaringan syaraf tiruan Probabilistik (PNN), karena dari hasil pengujian PNN memiliki tingkat akurasi tertinggi. Untuk memastikan apakah metode jaringan syaraf tiruan PNN ini lebih baik dari metode klasifikasi bintang matahari yang bukan jaringan syaraf tiruan, kami membangun sebuah prototype sub-sistem menggunakan metode klasifikasi yang disarankan dan digunakan oleh peneliti lain sebelumnya yaitu metode klasifikasi ID3 (Nguyen, 2006). Dari hasil pengujian didapat bahwa metode JST PNN masih lebih baik hasil akurasinya dibandingkan dengan metode ID3. Oleh sebab itu akhirnya diputuskan bahwa metode JST PNN akan dipakai sebagai inti dari Sistem Klasifikasi Badai Matahari Otomatis yang dibangun.

Penelitian pada tahun pertama dan kedua dibiayai dari dana PHB Dikti dan Kopertis Wilayah 7, sementara penelitian pada tahun ketiga ini dibiayai dari hibah penelitian internal Pusat Penelitian Universitas Kristen Petra.

## 1.2 Permasalahan

Permasalahan utama yang dihadapi dan diharapkan dapat diselesaikan melalui penelitian pada tahun kedua ini adalah bagaimana mencari dan mendapatkan metode klasifikasi dengan tingkat akurasi yang baik dan memiliki respon cepat. Karena itu, sambil melakukan proses ekstraksi fitur serta klasifikasi manual data training oleh peneliti dari LAPAN, pada tahun kedua ini dibangun prototype sub-sistem klasifikasi menggunakan tiga macam metode jaringan saraf tiruan. Ketiga metode tersebut adalah jaringan saraf tiruan *backpropagation*, *probabilistic*, dan *self-organizing map*. Setelah diketahui metode JST mana yang tepat untuk digunakan, sub-sistem dengan metode JST yang terpilih akan dibandingkan dengan sebuah sub-sistem yang dibangun dengan Sistem klasifikasi kelompok bintang matahari secara menyeluruh, yaitu dari input awal berupa citra matahari sampai dengan munculnya output berupa hasil klasifikasi otomatis kelompok bintang matahari akan dikerjakan pada tahun ketiga.

### 1.3 Ruang Lingkup Pembahasan

Dalam penelitian ini terdapat batasan, yaitu:

- Input berupa Microsoft Access database file yang merupakan hasil output dari sub-sistem ekstraksi fitur yang dikerjakan pada tahun pertama. Format dari input tersebut adalah sebagai berikut:
  - Tabel Picture  
Memiliki 4 field yaitu: *ID, FileName, History, dan DateTime.*
  - Tabel Sunspot  
Memiliki 7 field, yaitu: *ID, Area, Perimeter, EmX, EmY, Region, dan PictureID.*
  - Tabel Pixel  
Memiliki 6 field, yaitu : *ID, X, Y, Nei, NeiCount, dan SunSpotID.*
  - Tabel Cluster  
Memiliki 10 field, yaitu : *ID, Distance, PictureID, Area, Perimeter, Diameter, StartX, StartY, EndX, dan EndY.*
- Klasifikasi kelompok bintang matahari berdasarkan *Modified-Zurich Sunspot Classification.*
- Metode yang digunakan untuk membangun prototipe sub-sistem klasifikasi adalah jaringan saraf tiruan *Backpropagation, Probabilistic, Self-Organizing Map* (telah dikerjakan pada tahun kedua) serta metode data mining ID3 / C4.5.
- Data training untuk prototipe sistem dengan metode jaringan saraf tiruan dan ID3 / C4.5. menggunakan data hasil ekstraksi fitur yang telah diklasifikasi manual oleh peneliti dari LAPAN yang merupakan pakar tentang matahari.
- Output berupa tampilan hasil klasifikasi bintang matahari beserta database file dengan penambahan field untuk hasil klasifikasi berdasarkan metode klasifikasi bintang matahari *Modified-Zurich.*
- Aplikasi akhir yang didevelop adalah sebuah Sistem Klasifikasi Otomatis Tipe Badai Matahari yang menyeluruh. Dimulai dari memasukkan inputan data – data sample, klasifikasi manual pada data sample untuk membuat target klasifikasi dari data sample yang digunakan untuk training, proses training jaringan syaraf tiruan yang dipilih dan pada akhirnya melakukan proses klasifikasi otomatis tipe badai matahari terhadap sebuah citra matahari baru yang diambil dengan peralatan *Michelson Doppler Image (MDI) Continuum / SOHO (Solar dan Heliospheric Observatory).*
- Aplikasi akan dibuat dengan menggunakan bahasa pemrograman C#.

#### 1.4 Sistematika Penyusunan Laporan

Laporan penelitian ini secara keseluruhan terdiri dari enam bab dimana secara garis besar masing-masing bab membahas hal-hal sebagai berikut:

- BAB 1 **Pendahuluan:** berisi latar belakang, permasalahan, ruang lingkup permasalahan, dan sistematika penyusunan laporan.
- BAB 2 **Tinjauan Pustaka:** membahas tentang teori-teori dasar yang relevan dan metode yang digunakan untuk memecahkan persoalan yang dibahas pada penelitian ini.
- BAB 3 **Tujuan dan Manfaat Penelitian:** membahas mengenai tujuan yang ingin dicapai pada penelitian ini beserta dengan manfaat yang ingin dihasilkan.
- BAB 4 **Metode Penelitian:** membahas tentang metode penelitian yang dilakukan.
- BAB 5 **Hasil Penelitian dan Pembahasan:** berisi tentang hasil dari penelitian, berupa aplikasi yang telah dikembangkan beserta dengan pengujian aplikasi tersebut.
- BAB 6 **Kesimpulan dan Saran:** berisi kesimpulan yang mencakup beberapa hal penting pada hasil yang didapat dari penelitian dan saran-saran yang diajukan bagi penyempurnaannya.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Penelitian Sebelumnya yang Menunjang

Penelitian sebelumnya telah dilakukan pada tahun 2009 s/d 2010 yang meliputi penelitian mengenai segmentasi citra digital serta ekstraksi fitur. Ekstraksi fitur ini digunakan sebagai input pada proses klasifikasi. Proses klasifikasi dilakukan dengan menggunakan jaringan saraf tiruan. Penelitian ini telah dipublikasikan (Adipranata dkk, 2010; Budhi dkk, 2011a).

Pada tahun 2010 s/d 2011, telah dilakukan lanjutan penelitian mengenai klasifikasi atau pengenalan pola (*pattern recognition*) menggunakan jaringan saraf tiruan *Self Organized Map*, *Backpropagation* dan *Probabilistic Neural Network* yang diimplementasikan pada hasil ekstraksi fitur bintang matahari. Pada akhir penelitian ini telah didapatkan metode terbaik yang digunakan pada klasifikasi bintang matahari. Sebagian dari hasil penelitian ini telah dipublikasikan (Budhi dkk, 2011b).

Pada tahun 2012 dilakukan penelitian lanjutan yaitu membangun sebuah sub-sistem klasifikasi otomatis menggunakan metode *data mining* gabungan ID3/C4.5. Penelitian serupa yang pernah dilakukan oleh peneliti lain mengenai klasifikasi kelompok bintang matahari berdasarkan sistem klasifikasi "*Modified-Zurich Sunspot Classification*", dilakukan oleh Nguyen dan kawan-kawan. (Nguyen *et.al*, 2006). Mereka membandingkan empat metode klasifikasi, yaitu: *C4.5 Decision Tree* dan *k-Nearest Neighbour (kNN)* yang terdapat pada *tool WEKA* serta algoritma *LEM2* dan *RIONA* pada *tool RSES*. Dari empat metode yang digunakan, metode kNN memiliki akurasi terbaik sebesar 90%, sementara metode *C.4.5 Decision Tree* memiliki akurasi terbaik 85%.

Dalam penelitian ini digunakan pendekatan dengan metode yang berbeda dari penelitian sebelumnya yaitu melakukan klasifikasi dengan metode *data mining* ID3/C4.5. Pada penelitian ini pula akan diuji, dibandingkan dan dianalisa hasil klasifikasi berdasar metode jaringan saraf tiruan terbaik yang telah ditentukan dari hasil penelitian sebelumnya yaitu jaringan syaraf tiruan probabilistik dengan hasil dari sub-sistem klasifikasi dengan metode ID3/C4.5 yang dibangun pada penelitian ini.

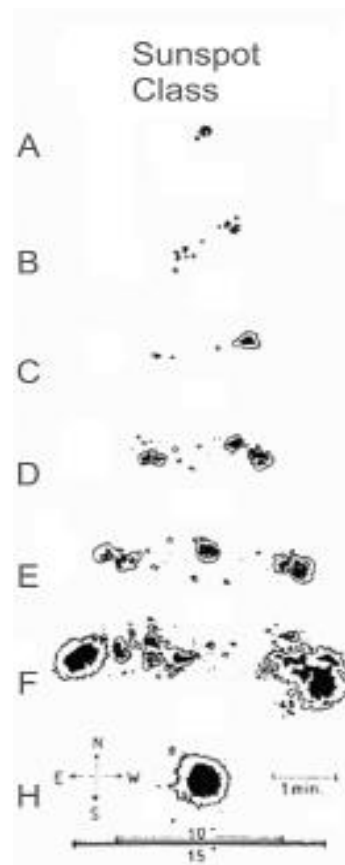
### 2.2 Bintang Matahari

Bintang matahari (*sunspot*) merupakan perpotongan tabung medan magnet matahari (*magnetic flux tube*) dengan fotosfer. Bintang matahari tampak hitam karena medan magnet mempunyai efek pendinginan sehingga suhu bintang matahari lebih rendah dari sekitarnya. Medan

magnet bintik matahari terbukti merupakan sumber energi gangguan dari matahari, misalnya ledakan dahsyat (*flare*) atau pelontaran massa korona (*coronal mass ejection/CME*) (Setiahadi, 2006). Fenomena ini dapat mengganggu lingkungan antariksa di sekitar Bumi maupun fasilitas teknologi dan kehidupan di permukaan Bumi (Lanzerotti, 2001; Boteler, 2001; Bothmer and Daglis, 2007).

Karena bintik matahari merupakan perpotongan tabung medan magnet dengan fotosfer, pemunculan bintik matahari diawali dengan sebuah bintik hitam berukuran kecil. Seiring dengan waktu, bila tabung medan magnet terus-menerus keluar dari dalam matahari akibat gaya apung (*buoyancy force*), maka akan tampak dua buah bintik dengan polaritas medan magnet berlawanan (*bipolar*). Bintik matahari berkembang menjadi konfigurasi lebih kompleks, yaitu jumlah bintik dan luas bertambah. Semakin kompleks suatu konfigurasi bintik matahari, semakin besar kemungkinan terjadi ketidakstabilan (*instability*) medan magnet sehingga memicu peristiwa *flare* atau CME (Anwar *et.al.*, 1993; Setiahadi, 2005).

Tahapan evolusi bintik matahari dinyatakan dalam “*Modified-Zurich Sunspot Classification*”, yaitu kelas A, B, C, D, E, F (tingkat kompleksitas bertambah) dan kemudian berangsur-angsur menurun hingga akhirnya menjadi kelas H. Gambar klasifikasi terdapat pada Gambar 2.1 (McIntosh, 1990).



Gambar 2.1 Klasifikasi “*Modified-Zurich Sunspot Classification*”



Perubahan kelas bintik matahari terkait dengan tingkat aktivitas bintik tersebut. Sebagai contoh, bintik matahari kelas E memiliki tingkat aktivitas lebih tinggi dibandingkan kelas D.

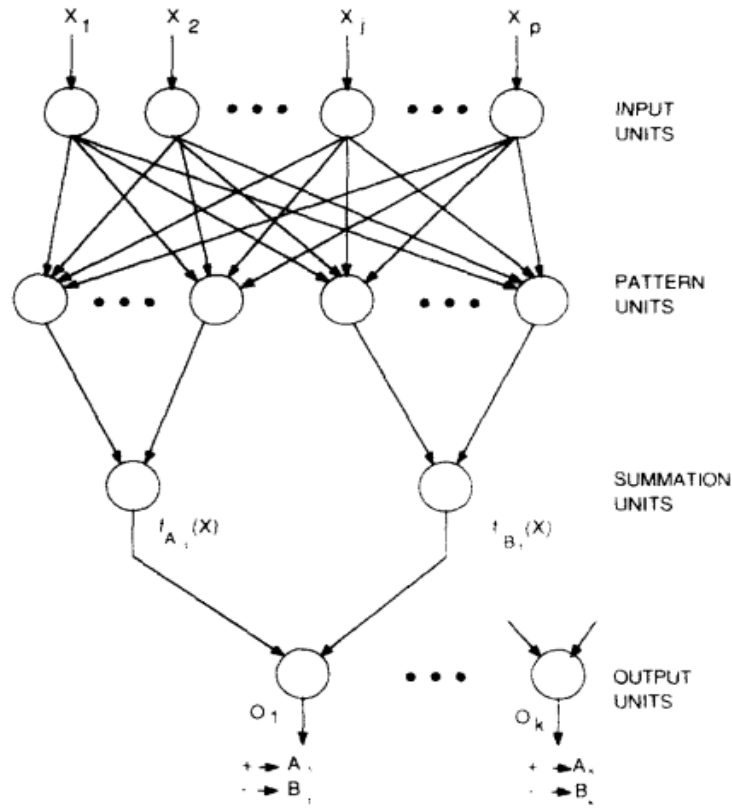
## 2.3 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah sebuah model cara berpikir yang didasarkan pada cara kerja otak manusia (Negnevitsky, 2005). Sama seperti otak manusia, jaringan syaraf tiruan terdiri dari sel-sel neuron yang saling berhubungan. Tiap sel neuron menerima sejumlah sinyal *input* yang diproses untuk menghasilkan satu sinyal *output*. Jaringan syaraf tiruan dapat belajar seperti otak manusia dengan cara memberi bobot (*weight*) untuk setiap *input* pada sel neuron. Dengan menggunakan bobot tersebut, jaringan syaraf tiruan dapat mempelajari *input* yang diberikan.

### 2.3.1 Probabilistic Neural Network

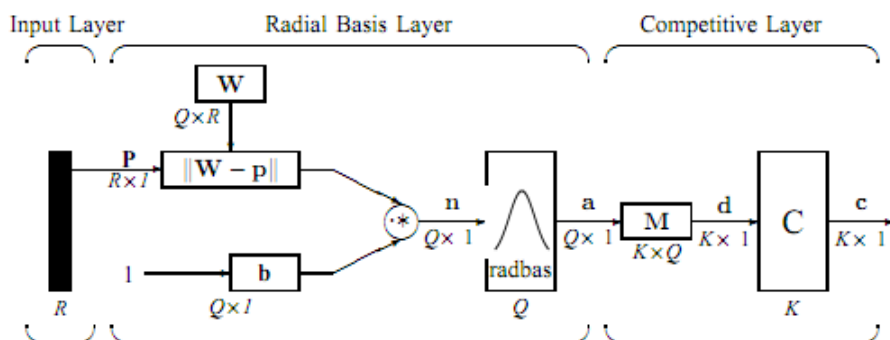
*Probabilistic Neural Network (PNN)* adalah sebuah metode jaringan syaraf tiruan yang menggunakan prinsip dari teori statistik yaitu *Bayesian Classification* untuk menggantikan prinsip heuristik yang digunakan oleh algoritma *Backpropagation* (Specht, 1990). Karena itulah *PNN* biasa digunakan untuk melakukan *pattern classification* (Fausett, 1994).

Arsitektur dari *PNN* terdiri dari 4 *layer*, yaitu *input layer*, *pattern layer*, *summation layer*, dan *decision layer / output layer* seperti dapat dilihat pada Gambar 2.4 (Specht, 1990).



Gambar 2.2 Arsitektur *Probabilistic Neural Network*

*Input layer* tidak melakukan perhitungan apapun, hanya mentransfer data *input* menuju tiap *neuron* pada *pattern layer*. Tiap *neuron* pada *pattern layer* akan melakukan perhitungan probabilitas (jarak) antara *input* dengan data yang tersimpan pada *neuron pattern layer*. Selanjutnya, *summation layer* akan menerima *input* dari tiap *neuron pattern layer* dan akan menjumlahkannya sehingga akan didapat berapa kemungkinan suatu *input x* masuk ke dalam suatu kelompok  $t$ . Dan terakhir, *output layer* akan menghasilkan hasil klasifikasi berdasarkan hasil dari *summation neuron* yang memiliki nilai yang paling besar. Dalam penelitian ini akan digunakan arsitektur seperti pada Gambar 2.5.



Gambar 2.3 Arsitektur *Probabilistic Neural Network*

Berikut ini adalah algoritma *training* dan pengklasifikasian dari jaringan saraf tiruan *probabilistic* (Wu et al, 2007):

Melakukan inisialisasi bobot awal (*radial basis layer weight*)  $W$  yang didapat dari hasil transpose dari matriks  $R \times Q$  di mana  $R$  adalah dimensi input dan  $Q$  adalah jumlah *data training*.

Melakukan inisialisasi konstanta *spread* ( $s$ ) dari *PNN*.

Melakukan inisialisasi bobot bias ( $b$ ).

$$b = \frac{\sqrt{-\ln 0.5}}{s} \quad (2.1)$$

Melakukan inisialisasi bobot akhir (*competitive layer weight*)  $M$  yang merupakan matriks berukuran  $K \times Q$  di mana  $K$  adalah jumlah kelompok hasil klasifikasi dan  $Q$  adalah jumlah data training. Baris ke- $i$  pada matriks  $M$  mewakili *data training* ke- $i$  dan pada kolom ke- $j$  nilai matriks akan bernilai 1 bila *data training* masuk ke dalam kelompok tersebut, bila tidak maka akan bernilai 0.

Menghitung jarak antara vektor data input  $P$  dengan vektor tiap baris pada bobot awal  $W$  (*Euclidean Distance* antara vektor  $P$  dengan vektor  $W_i$ ) sehingga menghasilkan matriks jarak  $\|W - P\|$  yang berdimensi  $Q \times 1$ .

Menghitung nilai aktivasi dari jarak antara bobot awal dengan data input (vektor  $a$ ), dengan menggunakan fungsi *radbas*(.).

$$\text{radbas}(n) = e^{-n^2} \quad (2.2)$$

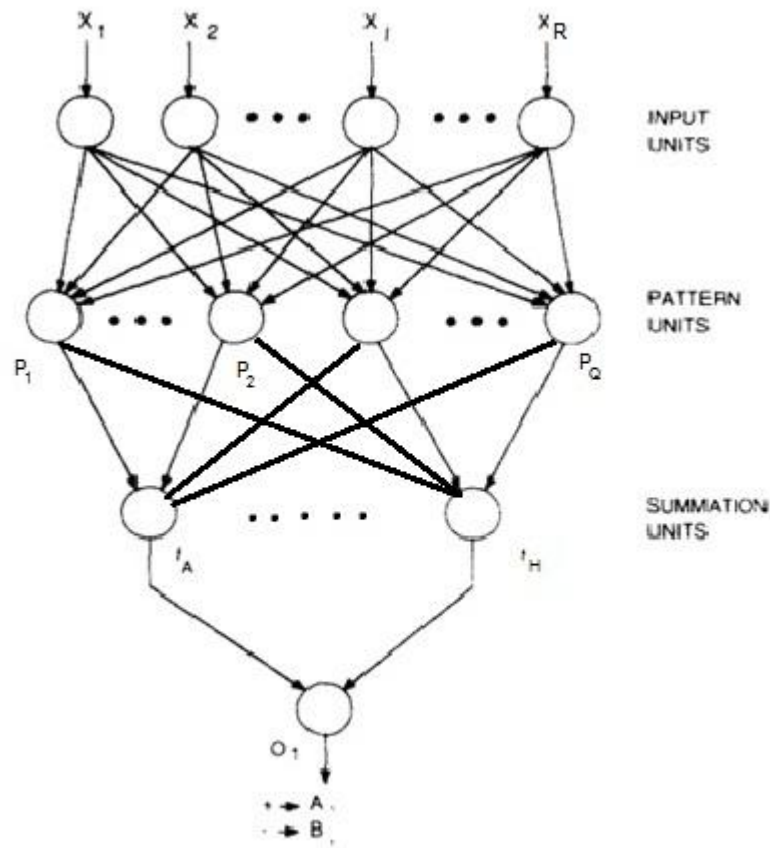
$$n = \|W - P\| * b \quad (2.3)$$

Mengalikan vektor  $a$  dengan matriks  $M$  sehingga menghasilkan vektor *output*  $d$ .

Mencari output dari *PNN* dengan *competitive function*  $C$  di mana *competitive function* ini akan mencari nilai terbesar dalam vektor  $d$ . Index dari nilai terbesar inilah yang menandakan hasil klasifikasi data input  $P$ .

Simpan bobot awal, *spread*, dan bobot akhir.

Untuk gambar arsitektur *PNN* yang digunakan secara lebih sederhana dapat dilihat pada Gambar 2.6 berikut.



Gambar 2.4 Arsitektur *PNN* yang Digunakan

Untuk jumlah *input unit* akan didapat dari besarnya dimensi *data input* ( $R$ ), yaitu *data* yang diolah dari *database input* yang berupa diameter, luas, keliling, jumlah bintik matahari tiap *cluster*, dan rata-rata *pixel* hitam untuk setiap *region* gambar *input*. Setelah itu, setiap *input unit* akan terhubung ke semua *pattern unit* yang berjumlah  $Q$  dengan sebuah *weight*, di mana  $Q$  adalah jumlah *data training*. Sehingga matriks *weight* awal akan berukuran  $Q \times R$ .

Untuk setiap *pattern unit* akan menghasilkan sebuah nilai jarak antara *weight* dengan *input*, hasil tersebut berupa matriks berdimensi  $Q \times 1$  yang kemudian akan digunakan untuk perhitungan selanjutnya dengan fungsi *radbas*.

Setiap *pattern unit* akan terhubung dengan semua *summation unit* melalui *weight* akhir yang merupakan matriks dengan ukuran  $K \times Q$ , di mana  $K$  adalah jumlah kelompok hasil klasifikasi, yaitu kelas A, B, C, D, E, F, dan H. Besarnya  $Q$  adalah jumlah data training. Baris ke- $i$  pada matriks  $M$  mewakili *data training* ke- $i$  dan pada kolom ke- $j$  nilai matriks akan bernilai 1 bila *data training* masuk ke dalam kelompok tersebut, bila tidak maka akan bernilai 0.

Lalu setiap *summation unit* akan dihubungkan dengan sebuah *output unit* yang berfungsi untuk mencari nilai *summation unit* yang terbesar dan mengambil indeks dari *summation unit* tersebut sebagai hasil klasifikasi.

## 2.4 Data Preprocessing

Agar data yang ada sesuai dengan yang dibutuhkan, perlu dilakukan beberapa proses sebelum data tersebut dapat diproses sebagai *input* oleh jaringan syaraf tiruan yang dibuat.

### 2.4.1 Min-Max Normalization

Pada algoritma jaringan syaraf tiruan, *input* yang diterima berupa bilangan dengan *range* antara 0 sampai dengan 1. Karena itu, diperlukan sebuah proses terlebih dahulu untuk menyesuaikan data yang sesungguhnya dengan *range* tersebut.

Proses yang digunakan adalah *min-max normalization*, yaitu metode normalisasi yang mentransformasi data secara linear terhadap sebuah *range* yang baru (Han, Kamber & Pei, 2012). Jika  $min_A$  dan  $max_A$  masing-masing adalah nilai minimum dan maksimum dari sebuah atribut  $A$ , maka *min-max normalization* memetakan sebuah nilai  $v$  dari atribut  $A$  menjadi nilai  $v'$  dalam *range*  $[new\_min_A, new\_max_A]$  melalui persamaan:

$$v' = \frac{v - \min_A}{\max_A - \min_A} (new\_max_A - new\_min_A) + new\_min_A \quad (2.4)$$

### 2.4.2 Image Processing

Salah satu *input* untuk aplikasi ini berupa *image file* yang berisi citra observasi dari bintang matahari. *Image file* tersebut dibagi menjadi beberapa *region* terlebih dahulu. Data yang pada akhirnya menjadi *input* adalah jumlah *pixel* berwarna gelap dari masing-masing *region*. Proses yang dilakukan adalah mengubah *image* menjadi *grayscale image*, kemudian menghitung berapa *pixel* pada masing-masing *region* yang memiliki nilai warna lebih tinggi daripada nilai *threshold* tertentu.

Untuk mengubah *image* menjadi *grayscale image*, perlu dihitung nilai *luminance* dari masing-masing *pixel*, yang kemudian digunakan sebagai nilai warna *red*, *green*, dan *blue* untuk *pixel* tersebut (Powell, 2000). Nilai *luminance* tersebut didapatkan melalui persamaan:

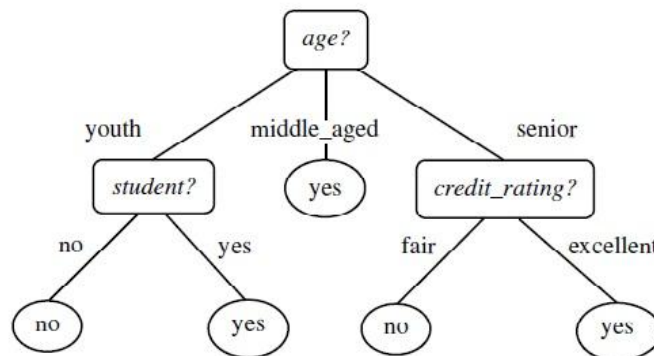
$$Y = 0.3RED + 0.59GREEN + 0.11BLUE \quad (2.5)$$

dimana *RED*, *GREEN*, dan *BLUE* masing-masing adalah nilai warna *red*, *green*, dan *blue* untuk satu *pixel* dan *Y* adalah nilai *luminance* untuk *pixel* tersebut.

## 2.5 Decision Tree Induction

*Decision Tree Induction* adalah sebuah metode pembelajaran *decision tree*. *Decision tree* berbentuk seperti flowchart di mana setiap internal node menggambarkan sebuah tes pada sebuah atribut, setiap branch menggambarkan hasil dari tes, dan setiap leaf node menunjukkan class label. Node paling atas adalah root node (Han, Kamber & Pei, 2012).

Contoh dari *decision tree* dapat dilihat pada Gambar 2.1. Gambar ini menunjukkan prediksi apakah seorang customer di AllElectronics akan membeli komputer atau tidak. Internal node digambarkan dengan kotak, dan leaf node digambarkan dengan oval.



Gambar 2.1 Contoh *Decision Tree* (Han, Kamber, Pei, 2012)

Selama membuat tree, perlu memilih atribut yang paling baik dalam membedakan kelas-kelas. Untuk itu digunakan algoritma. Beberapa algoritma *decision tree* antara lain ID3, C4.5, dan CART.

## 2.6 ID3

ID3 (*Iterative Dichotomiser*) merupakan sebuah algoritma pohon keputusan yang dibuat oleh J. Ross Quinlan (Han, Kamber & Pei, 2012). ID3 menggunakan algoritma *basic tree induction* yang memberi atribut ke *node* pada *tree* berdasar berapa banyak informasi bertambah dari *node* tersebut. Metode ID3 memperbolehkan sebuah atribut untuk memiliki dua atau lebih nilai pada sebuah *node* atau titik *split* (Berry & Browney, 2006).

Atribut yang dipilih untuk setiap *node* adalah atribut yang memaksimalkan *information gain*.

### 2.6.1 Information gain

ID3 menggunakan *Information gain* untuk pemilihan atributnya. Atribut yang memiliki *Information gain* tertinggi dipilih menjadi atribut *splitting* untuk sebuah *node* (Han, Kamber & Pei, 2012).

Informasi yang diperkirakan dibutuhkan untuk mengklasifikasi sebuah *tuple* dalam D diberikan

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (2.6)$$

di mana  $p_i$  adalah kemungkinan sebuah arbitrary *tuple* dalam D termasuk kelas  $C_i$  dan diestimasi oleh  $|C_{i,D}|/|D|$ . Fungsi log dengan basis 2 digunakan karena informasi dihitung dalam bit.  $Info(D)$  hanya jumlah rata-rata informasi yang digunakan untuk mengidentifikasi class label dari sebuah *tuple* dalam D.  $Info(D)$  dikenal sebagai *entropy* dari D.

Jumlah informasi yang masih dibutuhkan untuk mencapai partisi yang tepat diukur dengan persamaan (2.7) (Han, Kamber & Pei, 2012).

$$Info_A(D) = a_0 + \sum_{j=1}^v \frac{|D_j|}{|D|} Info(D_j) \quad (2.7)$$

Term  $\frac{|D_j|}{|D|}$  berfungsi sebagai bobot dari partisi ke j.  $Info_A(D)$  adalah perkiraan informasi yang dibutuhkan untuk mengklasifikasikan sebuah *tuple* dari D berdasarkan partisi oleh A. Semakin kecil informasi yang diperkirakan dibutuhkan, semakin besar tingkat kemurnian dari partisi.

*Information gain* didefinisikan sebagai perbedaan antara kebutuhan informasi asal dengan kebutuhan yang baru (Han, Kamber & Pei, 2012). Yaitu

$$Gain(A) = Info(D) - Info_A(D) \quad (2.8)$$

Dengan kata lain,  $Gain(A)$  memberi tahu seberapa banyak akan bertambah dengan *branching* di A. Atribut A dengan *information gain* terbesar, ( $Gain(A)$ ) akan dipilih sebagai atribut *splitting* di *node* yang bersangkutan.

## 2.7 C4.5

Algoritma C4.5 merupakan pengembangan Quinlan atas metode yang pernah dibuatnya terdahulu yaitu ID3 untuk membuat *decision tree*. Algoritma C4.5 secara rekursif mendatangi setiap *decision node*, memilih *split* yang optimal, sampai tidak ada lagi kemungkinan untuk *split* (Larose, 2005).

Algoritma C4.5 menggunakan konsep *Information gain* atau *Entropy Reduction* untuk memilih *split* yang optimal. Asumsikan memiliki variabel X dengan k kemungkinan nilai memiliki peluang  $p_1, p_2, \dots, p_k$ . *Entropy* X didefinisikan

$$H(X) = - \sum_j^k P_j \log_2(P_j) \quad (2.9)$$

C4.5 menggunakan konsep *entropy* sebagai berikut. Asumsikan memiliki kandidat *split* S, di mana mempartisi set *data training* T menjadi beberapa subset, T1, T2, ..., Tk. Informasi berarti yang dibutuhkan dapat dihitung dengan

$$H_S(T) = \sum_{i=1}^k P_i H_S(T_i) \quad (2.10)$$

Di mana  $P_i$  merepresentasikan proporsi record dalam subset i. *Information gain* dapat ditentukan menjadi  $gain(S) = H(T) - H_S(T)$ , di mana, peningkatan dalam informasi yang diproduksi dari mempartisi *data training* T menurut kandidat *split* S. Di setiap *decision node*, C4.5 memilih *split* yang optimal untuk menjadi *split* yang memiliki *Information gain* terbesar,  $gain(S)$  (Larose, 2005).

Algoritma ini merupakan salah satu algoritma yang disarankan dalam pembuatan aplikasi klasifikasi bintang matahari (Nguyen, 2006).

### 2.7.1 Gain ratio

Pengukuran *Information gain* bisa melalui berbagai tes dengan banyak hasil. Ada kecenderungan untuk memilih atribut dengan nilai yang banyak. Sebagai contoh *split* dalam *product* ID akan lebih dipilih karena memiliki partisi yang banyak. Setiap partisi memiliki 1 *tuple*. Karena setiap partisi murni, maka informasi yang dibutuhkan untuk mengklasifikasi set data D berdasar partisi ini akan bernilai 0. Oleh karena itu, penambahan informasi dari partisi atribut ini maksimal. Jelas, partisi menjadi tidak berguna untuk klasifikasi (Han, Kamber & Pei, 2012).

C4.5, suksesor dari ID3, menggunakan ekstensi untuk *Information gain* yang dikenal sebagai *Gain ratio*, yang mencoba untuk mengatasi bias ini. Semacam normalisasi diberlakukan pada *Information gain* menggunakan nilai “*split information*” dianalogikan dengan  $Info(D)$  seperti



$$SplitInfo_A(D) = - \sum_{j=i}^v \frac{|D_j|}{|D|} \times \log_2 \left( \frac{|D_j|}{|D|} \right) \quad (2.11)$$

Nilai ini merepresentasikan informasi potensial yang dihasilkan dari *splitting* set *data training*,  $D$ , ke  $v$  partisi, sesuai pada  $v$  hasil dari tes pada atribut  $A$  (Han, Kamber & Pei, 2012). *Gain ratio* dapat didefinisikan

$$GainRatio(A) = Gain(A) / SplitInfo(A) \quad (2.12)$$

Atribut dengan maksimum *gain ratio* dipilih sebagai atribut *split*.

### 2.8 Intuitive partitioning

*Intuitive partitioning* disebut juga sebagai 3-4-5 rule. Partisi ini digunakan untuk mensegmentasi data numerik secara *uniform* (Han, Kamber & Pei, 2012). Aturannya adalah sebagai berikut

- a. Jika interval mengandung 3, 6, 7, atau 9 nilai yang berbeda pada *most significant digit*, maka partisi *range* menjadi 3 interval (3 equal-width interval untuk 3, 6, dan 9; dan 3 interval dalam kelompok of 2-3-2 untuk 7).
- b. Jika mengandung 2, 4, atau 8 nilai yang berbeda pada *most significant digit*, maka partisi *range* menjadi 4 equal-width interval.
- c. Jika mengandung 1, 5, atau 10 nilai yang berbeda pada *the most significant digit*, maka partisi *range* menjadi 5 equal-width interval.

Aturan tersebut dapat diaplikasikan secara rekursif untuk setiap interval, membuat suatu konsep hirarki untuk atribut numerik yang diberikan. Data dalam dunia riil sering mengandung banyak sekali nilai *outlier* positif dan atau negatif, yang bisa mempengaruhi metode *top-down discretization* berdasar nilai data minimal dan maksimal. Oleh karena itu digunakan aturan 5 persentil dan 95 persentil untuk mengurangi dampak ketidakakuratan partisi akibat data *outlier* tersebut.

### 2.9 Gini index

Gini index digunakan dalam CART (Classification And Regression Tree), (Han, Kamber & Pei, 2012). Gini index mengukur impurity dari  $D$ , partisi data atau set dari training tuples dengan

$$\text{Gini}(D) = 1 - \sum_{i=1}^m p_i^2 \quad (2.13)$$

di mana  $p_i$  adalah probabilitas sebuah tuple dalam  $D$  termasuk kelas  $C_i$  dan diestimasi oleh  $|C_i, D|/|D|$ . Jumlah ini dihitung atas  $m$  kelas.

Gini index menganggap split setiap atribut adalah biner. Anggap atribut  $A$  memiliki  $v$  nilai yang berbeda,  $\{a_1, a_2, \dots, a_v\}$ , terjadi di  $D$ . Untuk menentukan split biner terbaik di  $A$ , dicek semua kemungkinan subset yang dapat terbentuk menggunakan nilai yang diketahui dari  $A$ . Setiap subset,  $SA$ , dapatkah dianggap sebagai sebuah tes biner untuk atribut  $A$  dari bentuk “ $A \in SA?$ ”. Diberikan sebuah tuple, tes ini akan memuaskan bila nilai  $A$  untuk tuple berada dalam daftar nilai  $SA$ . Jika  $A$  memiliki  $v$  nilai yang memungkinkan, maka terdapat  $2^v$  kemungkinan subset. Sebagai contoh, jika  $income$  memiliki tiga kemungkinan nilai, yaitu  $\{low, medium, high\}$ , maka kemungkinan subset adalah  $\{low, medium, high\}$ ,  $\{low, medium\}$ ,  $\{low, high\}$ ,  $\{medium, high\}$ ,  $\{low\}$ ,  $\{medium\}$ ,  $\{high\}$ , dan  $\{\}$ . Kesampingkan power set,  $\{low, medium, high\}$  dan set kosong dari asumsi, karena tidak merepresentasikan sebuah split. Oleh karena itu, ada  $2^v - 2$  kemungkinan jalan untuk membentuk dua partisi data,  $D$ , berdasar split biner di  $A$ .

Ketika mempertimbangkan split biner, dihitung weighted sum dari impurity pada setiap partisi hasil (Han, Kamber & Pei, 2012). Sebagai contoh, jika split biner pada  $A$  mempartisi  $D$  menjadi  $D_1$  dan  $D_2$ , maka gini index pada partisi tersebut dapat dihitung

$$\text{GiniA}(D) = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2) \quad (2.14)$$

Untuk setiap atribut, setiap kemungkinan split biner diperhitungkan. Untuk atribut yang bernilai diskrit, subset yang memberikan Gini index minimum untuk atribut tersebut dipilih sebagai splitting subset

Untuk atribut yang bernilai kontinyu, setiap kemungkinan titik split dicek. Strateginya serupa dengan yang dijelaskan pada information gain, di mana midpoint antara setiap pasangan nilai diambil sebagai kemungkinan titik split. Titik yang memberikan Gini index minimum pada atribut tersebut akan diambil sebagai titik split pada atribut yang tersebut.  $D_1$  merupakan set yang bernilai  $\leq$  titik split, sedangkan  $D_2$  merupakan set yang bernilai  $>$  titik split

Reduksi dalam impurity yang akan terjadi karena split biner pada atribut  $A$  yang bernilai yang diskrit atau kontinyu adalah

$$\Delta\text{Gini}(A) = \text{Gini}(D) - \text{GiniA}(D), \quad (2.15)$$

Atribut yang memaksimalkan reduksi dalam impurity (atau yang memiliki Gini index minimum) dipilih sebagai atribut split. Atribut ini beserta splitting subset atau titik split bersama membentuk kriteria split (Han, Kamber & Pei, 2012).

## BAB 3. TUJUAN DAN MANFAAT PENELITIAN

### 3.1 Tujuan Penelitian

Tujuan yang ingin dicapai pada penelitian ini secara keseluruhan adalah membangun sistem otomatis untuk melakukan klasifikasi kelompok bintik matahari dimana sistem ini dapat digunakan sebagai alat bantu untuk menganalisa kondisi cuaca antariksa. Untuk mencapai tujuan tersebut, penelitian ini dilakukan selama tiga tahun.

Penelitian pada tahun ketiga ini merupakan tahap lanjutan penelitian yang telah dicapai pada tahun pertama dan kedua. Pada tahun pertama telah dibangun sub-sistem segmentasi bintik matahari (Adipranata dkk, 2010). Selain itu pada tahun pertama juga dibangun sub-sistem untuk menggali atau mengekstraksi fitur - fitur kelompok bintik matahari dari citra digital matahari yang telah disegmentasi (Budhi dkk, 2011a) dan juga database untuk menyimpan hasil ekstraksi fitur.

Pada tahun kedua dilakukan proses evaluasi sub-sistem ekstraksi fitur dan juga database penyimpanannya. Pengujian dan verifikasi hasil ekstraksi fitur ini dilakukan oleh peneliti dari LAPAN. Sambil melakukan verifikasi hasil sub-sistem ekstraksi fitur, peneliti LAPAN secara paralel juga melakukan klasifikasi manual pada data-data hasil ekstraksi fitur tersebut. Selanjutnya pada tahun kedua dibangun sub-sistem klasifikasi otomatis dengan menggunakan tiga metode jaringan saraf tiruan yang berbeda. Adapun ketiga metode klasifikasi yang dibangun dan dibandingkan itu adalah jaringan saraf tiruan *backpropagation*, *probabilistic* (PNN) dan *self-organizing map* (SOM). Untuk memudahkan proses perbandingan ketiga metode jaringan saraf tiruan, telah dibangun 6 buah prototipe sub-sistem. Dimana untuk setiap metode jaringan saraf dibangun dua buah prototipe. Setiap prototipe selanjutnya ditraining dengan data training yang sebelumnya telah diklasifikasi manual oleh peneliti LAPAN. Kemudian dipilih 1 prototipe terbaik untuk tiap jenis jaringan saraf tiruan. Salah satu hasil prototipe terbaik untuk jaringan syaraf tiruan SOM telah dipublikasi (Budhi dkk, 2011b).

Langkah pertama pada tahun ketiga adalah membandingkan hasil tiga prototipe yang dipilih dan menganalisa hasil perbandingan tersebut. Dari hasil analisa diketahui bahwa prototipe yang dibangun dengan jaringan syaraf tiruan Probabilistik memiliki kecepatan proses training dan akurasi klasifikasi terbaik. Hasil ini akan dipublikasi pada jurnal nasional terakreditasi KURSUSOR. Langkah selanjutnya adalah membangun sub-sistem klasifikasi kelompok bintik matahari dengan metode klasifikasi ID3 / C4.5 seperti saran dari penelitian lain sebelumnya (Nguyen, 2006). Hasil prototipe klasifikasi dengan metode ID3 / C4.5 ini dibandingkan dengan prototype terbaik JST yang menggunakan metode PNN. Dari analisa hasil perbandingan diketahui bahwa metode

jaringan syaraf tiruan Probabilistik memiliki tingkat akurasi yang lebih tinggi disbanding dengan metode klasifikasi ID3 / C4.5. Tujuan terakhir dari tahun ketiga adalah menggabungkan semua prototipe terbaik yang dibangun pada tahun pertama, kedua dan ketiga menjadi sebuah aplikasi utuh untuk melakukan klasifikasi kelompok bintik matahari, yang didapat dari sebuah citra digital matahari, berdasarkan metode klasifikasi *Modified-Zurich*. Hasil aplikasi utuh ini akan dipublikasi pada *International Journal of Multimedia and Ubiquitous Engineering*.

### 3.2 Manfaat Penelitian

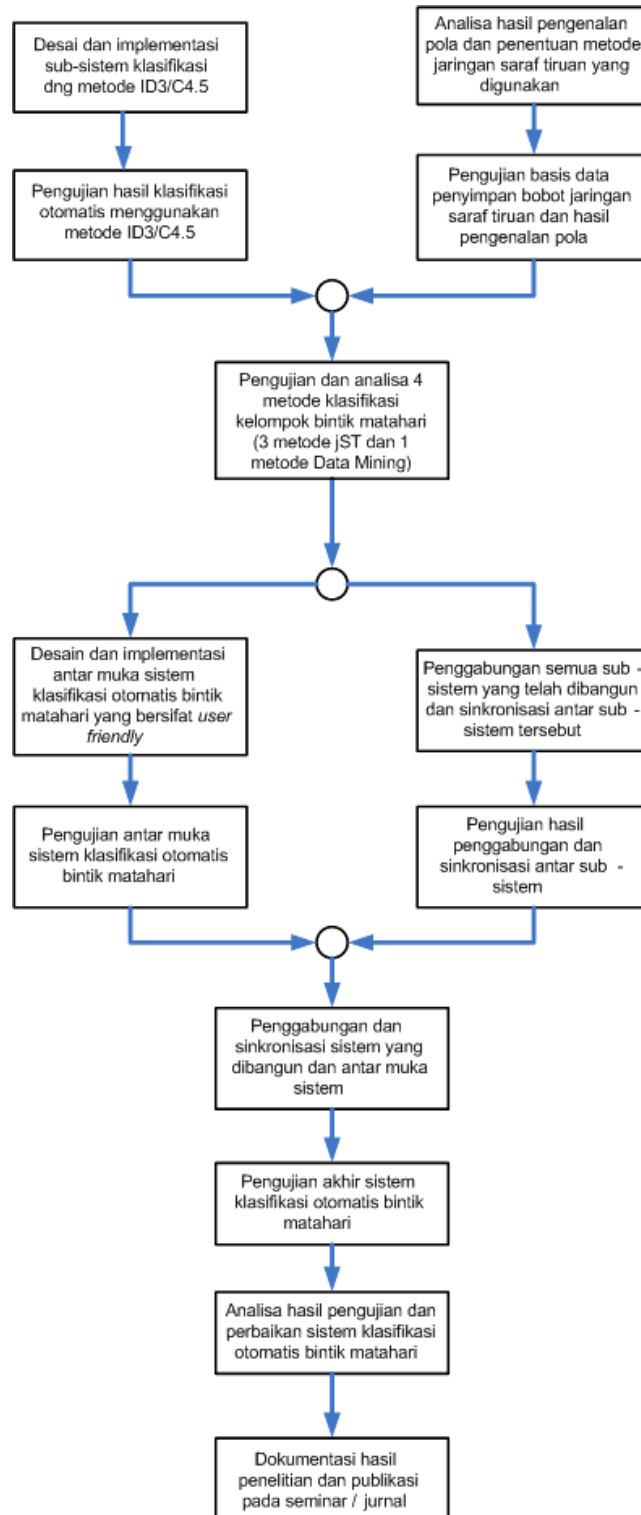
Penelitian yang dilakukan ini diharapkan mempunyai manfaat yaitu membantu LAPAN (Lembaga Penerbangan dan Antariksa Nasional) untuk menganalisa kondisi mengenai cuaca antariksa serta memberikan informasi kepada masyarakat dengan cepat akan gangguan, bahaya dan bahkan bencana yang mungkin terjadi di bumi karena munculnya badai matahari (*solar storm*), yang tampak sebagai bintik-bintik matahari (*sunspots*) bila diobservasi dari bumi.

Di samping itu, penelitian ini mempunyai manfaat yang terkait dengan pengembangan IPTEKS yaitu mendapatkan model segmentasi serta pembelajaran pola terbaik yang dapat diterapkan pada klasifikasi citra bintik matahari, sehingga dapat bermanfaat bagi penelitian lain yang membutuhkan dan di samping itu terdapat luaran berupa sebuah sistem klasifikasi otomatis untuk pengelompokan citra bintik matahari yang dapat dikembangkan lebih lanjut oleh peneliti lain.

Bagi pengembangan institusi, penelitian ini mempunyai manfaat yaitu memperkaya bahan pengajaran pada mata kuliah *computer vision*, jaringan saraf tiruan serta pengolahan citra digital. Dengan adanya penelitian ini maka bahan pengajaran tidak hanya berhenti pada penjelasan teori tetapi sudah merupakan penerapan teori tersebut ke kasus nyata.

## BAB 4. METODE PENELITIAN

Metode yang digunakan pada penelitian ini digambarkan berupa bagan alir yang terdapat pada Gambar 4.1.



Gambar 4.1 Bagan Alir Penelitian Tahun Ketiga

Penjelasan lebih rinci dari bagan alir pada Gambar 4.1. adalah sebagai berikut:

- Langkah pertama adalah membangun sebuah sub-sistem klasifikasi otomatis kelompok bintik matahari menggunakan metode *data mining* ID3/C4.5.
- Langkah kedua adalah menguji dan menganalisa hasil klasifikasi dari sub-sistem yang dibangun pada langkah pertama.
- Langkah ketiga adalah melakukan analisa hasil pengujian pengenalan pola tiap sub-sistem jaringan saraf tiruan yang telah dilakukan pada tahun kedua dan menentukan metode jaringan saraf tiruan mana yang paling baik untuk digunakan pada sistem klasifikasi citra bintik matahari. Langkah ini dilakukan secara bersamaan dengan langkah pertama.
- Langkah keempat adalah melakukan pengujian pada basis data yang akan digunakan untuk menyimpan bobot / weight hasil pelatihan sub sistem jaringan saraf tiruan serta hasil pengenalan pola dan klasifikasinya. Langkah ini dilakukan secara parallel dengan langkah kedua.
- Langkah kelima adalah melakukan pengujian dan analisa perbandingan antara 3 macam sub-sistem dengan metode jaringan saraf tiruan yang telah dibangun pada tahun kedua dan sebuah sub-sistem yang menggunakan metode data mining ID3/C4.5. Dari hasil perbandingan ini akan diambil satu sub-sistem terbaik.
- Langkah keenam adalah mendesain dan mengimplementasikan antar muka (*interface*) sistem klasifikasi otomatis bintik matahari yang enak digunakan (*user friendly*) dan memudahkan pengguna dalam melakukan proses klasifikasi otomatis bintik matahari serta kemudian menganalisa hasil klasifikasi tersebut.
- Langkah ketujuh adalah menguji antar muka yang dibuat.
- Langkah kedelapan adalah melakukan penggabungan semua sub-sistem yang telah dibuat serta melakukan sinkronisasi antar sub-sistem tersebut.
- Langkah kesembilan adalah menguji hasil penggabungan dan sinkronisasi sub-sistem pada langkah kedelapan. Untuk mempercepat proses implementasi, maka langkah keenam - ketujuh serta langkah kedelapan – kesembilan dilakukan secara paralel.
- Langkah kesepuluh adalah penggabungan serta sinkronisasi sistem klasifikasi otomatis bintik matahari dengan antar muka yang dirancang untuknya.
- Langkah kesebelas adalah pengujian hasil penggabungan dan sinkronisasi pada langkah ketujuh.

- Langkah duabelas adalah analisa hasil pengujian pada langkah kedelapan serta melakukan perbaikan - perbaikan yang diperlukan untuk sistem klasifikasi otomatis bintik matahari yang telah dibangun.
- Langkah terakhir pada tahun ketiga adalah melakukan dokumentasi hasil penelitian serta publikasi pada seminar / jurnal nasional.



## BAB 5. HASIL DAN PEMBAHASAN

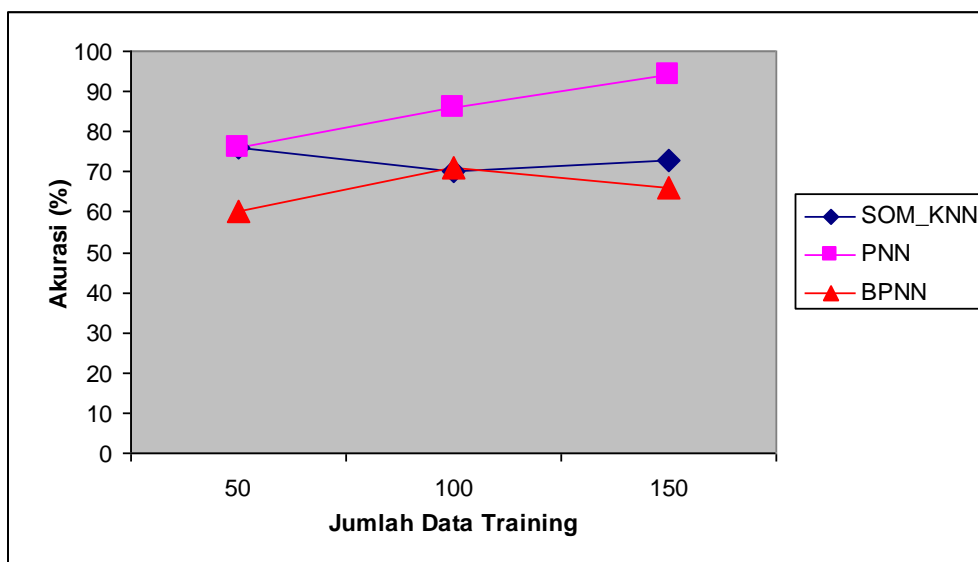
Langkah awal dari penelitian tahun ketiga adalah membandingkan tiga macam sub-sistem klasifikasi otomatis kelompok bintik matahari yang telah dibangun pada tahun kedua menggunakan metode jaringan syaraf tiruan Backpropagation, Probabilistik dan Self – Organising Map. Hasil perbandingan dan analisisnya dapat dilihat pada bab 5.1.

### 5.1 Perbandingan akurasi klasifikasi dari tiga macam sub-sistem Jaringan Syaraf Tiruan

Berikut dibandingkan tiga sub-sistem klasifikasi otomatis tipe badai matahari yang telah dibuat. Karena tiap - tiap metode memiliki ciri berbeda dan harus diuji dengan bentuk pengujian yang spesifik, maka perbandingan hanya dapat dilakukan pada pengujian - pengujian yang serupa yaitu:

1. Perbandingan akurasi hasil klasifikasi otomatis untuk variasi jumlah data training.

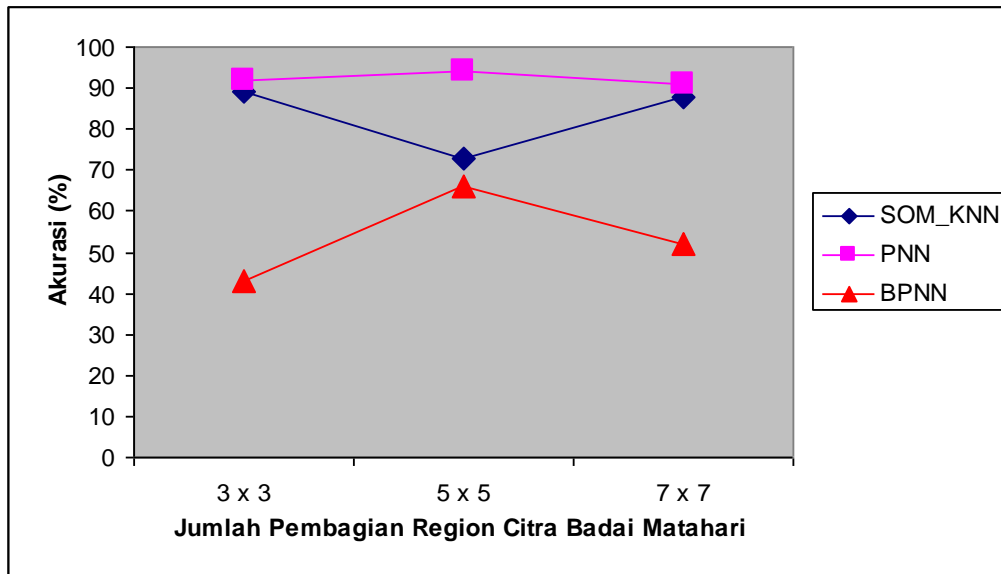
Hasil dapat dilihat pada Gambar 5.1. Dari hasil perbandingan tampak bahwa sub-sistem yang dibangun dengan memanfaatkan metode jaringan syaraf tiruan probabilistik memiliki performa lebih baik dibanding lainnya.



Gambar 5.1 Perbandingan akurasi klasifikasi untuk variasi jumlah data training

2. Perbandingan akurasi hasil klasifikasi otomatis untuk variasi pembagian region citra digital badai matahari.

Jumlah data training yang digunakan  $\cong$  150 data. Hasil dapat dilihat pada Gambar 5.2. Dari hasil perbandingan tampak bahwa sub-sistem yang dibangun dengan metode jaringan syaraf tiruan probabilistik memiliki performa terbaik. Sementara sub-sistem yang menggunakan metode jaringan syaraf tiruan backpropagation memiliki performa paling buruk.

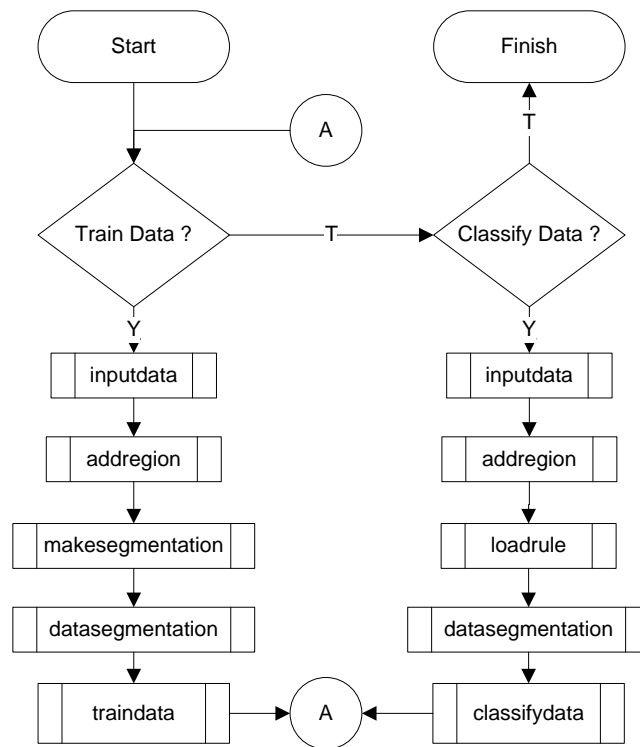


Gambar 5.2 Perbandingan akurasi klasifikasi untuk variasi pembagian region citra digital

Dari hasil perbandingan antara tiga sub-sistem klasifikasi otomatis tipe badai matahari yang dibangun, tampak bahwa performa sub - sistem dengan metode PNN secara keseluruhan lebih baik dari metode metode jaringan syaraf tiruan lainnya.

## 5.2 Desain sub-sistem klasifikasi kelompok bintik matahari dengan ID3 / C4.5.

Berikut ini adalah disain prototipe sub-sistem yang dibangun dengan metode ID3 / C4.5. Diagram alir dari aplikasi secara keseluruhan dapat dilihat pada Gambar 5.3 berikut:

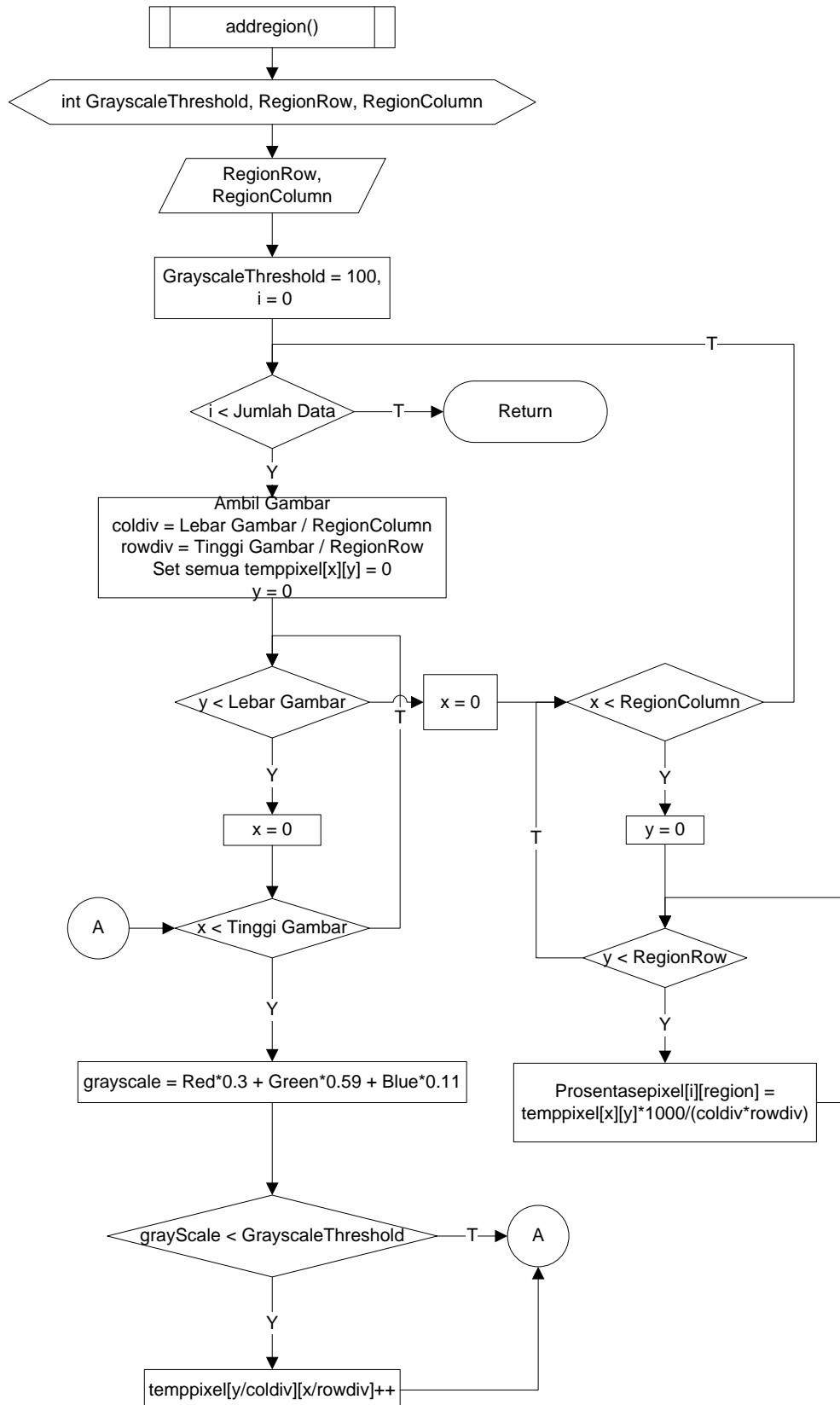


Gambar 5.3 Flowchart garis besar aplikasi

Untuk awalnya, *user* memilih akan melakukan *training* pada data atau melakukan klasifikasi data. Kedua aktivitas tersebut diawali dengan *procedure inputdata* dan *addregion*. *Procedure inputdata* adalah proses pengambilan data dari *database* yang digunakan. Sedangkan *addregion* adalah *procedure* untuk menambah *field* yang digunakan dengan prosentase *pixel* hitam pada setiap *region*. Pada prinsipnya, gambar dalam 1 *record* dipecah dalam beberapa *region*. Dari setiap *region* dihitung prosentase jumlah *pixel* hitamnya, yaitu *pixel* yang memiliki nilai warna yang lebih tinggi daripada nilai *threshold* tertentu setelah sebelumnya *image* tersebut dirubah menjadi *grayscale image*. Konsep ini dapat dilihat pada Gambar 5.4, sedangkan prosesnya dapat dilihat pada Gambar 5.5.



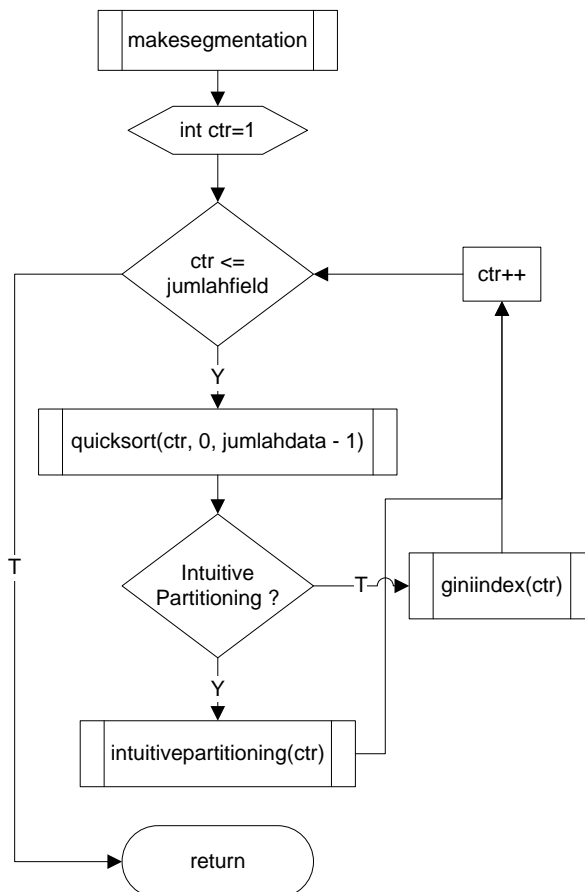
Gambar 5.4 Konsep Penggunaan *Region*



Gambar 5.5 Flowchart addregion

Tujuan dari fungsi *addregion* adalah menghitung dan memasukkan prosentase jumlah titik hitam dalam setiap *region* menjadi *field-field* yang digunakan dalam membentuk *decision tree* dan *rules*. Jumlah *region* ditentukan oleh *user*.

Setelah *field* yang digunakan sudah lengkap barulah segmentasi data dan *training* ataupun klasifikasi pada data dilakukan. Tentu saja segmentasi data yang dilakukan berdasar aturan segmentasi yang dibentuk. Segmentasi data dilakukan oleh fungsi *datasegmentation*. Untuk proses dalam *training* data, proses pembuatan aturan segmentasi dilakukan oleh fungsi *makesegmentation*, sedangkan untuk klasifikasi aturan segmentasi diambil dari *file rule* yang dipilih, menggunakan fungsi *loadrule*. Proses pembuatan aturan segmentasi dalam fungsi *makesegmentation* dapat dilihat pada Gambar 5.6.



Gambar 5.6 Flowchart *makesegmentation*

Proses pembuatan aturan segmentasi dalam fungsi *makesegmentation* dilakukan untuk semua *field* yang dipakai. Tentu saja, sebelum pembuatan aturan segmentasi dilakukan maka

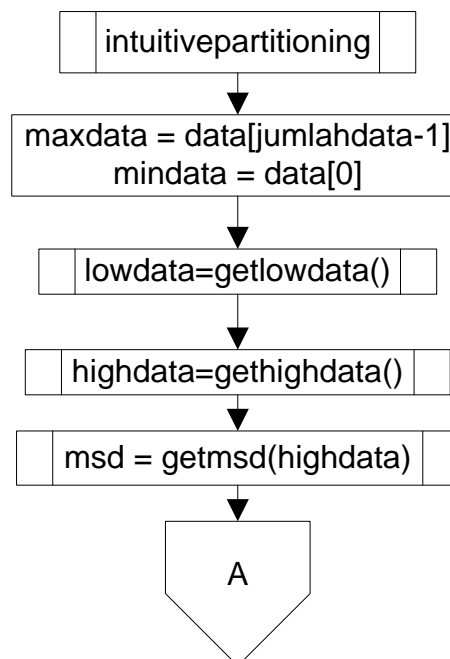
dilakukan *sorting* data menggunakan teknik *quicksort* yang dalam program diberi nama fungsi *quicksort*. Adapun penjelasan mulai dari metode segmentasi sampai *training* ataupun klasifikasi data akan diberikan selanjutnya.

### 5.2.1 Metode Segmentasi

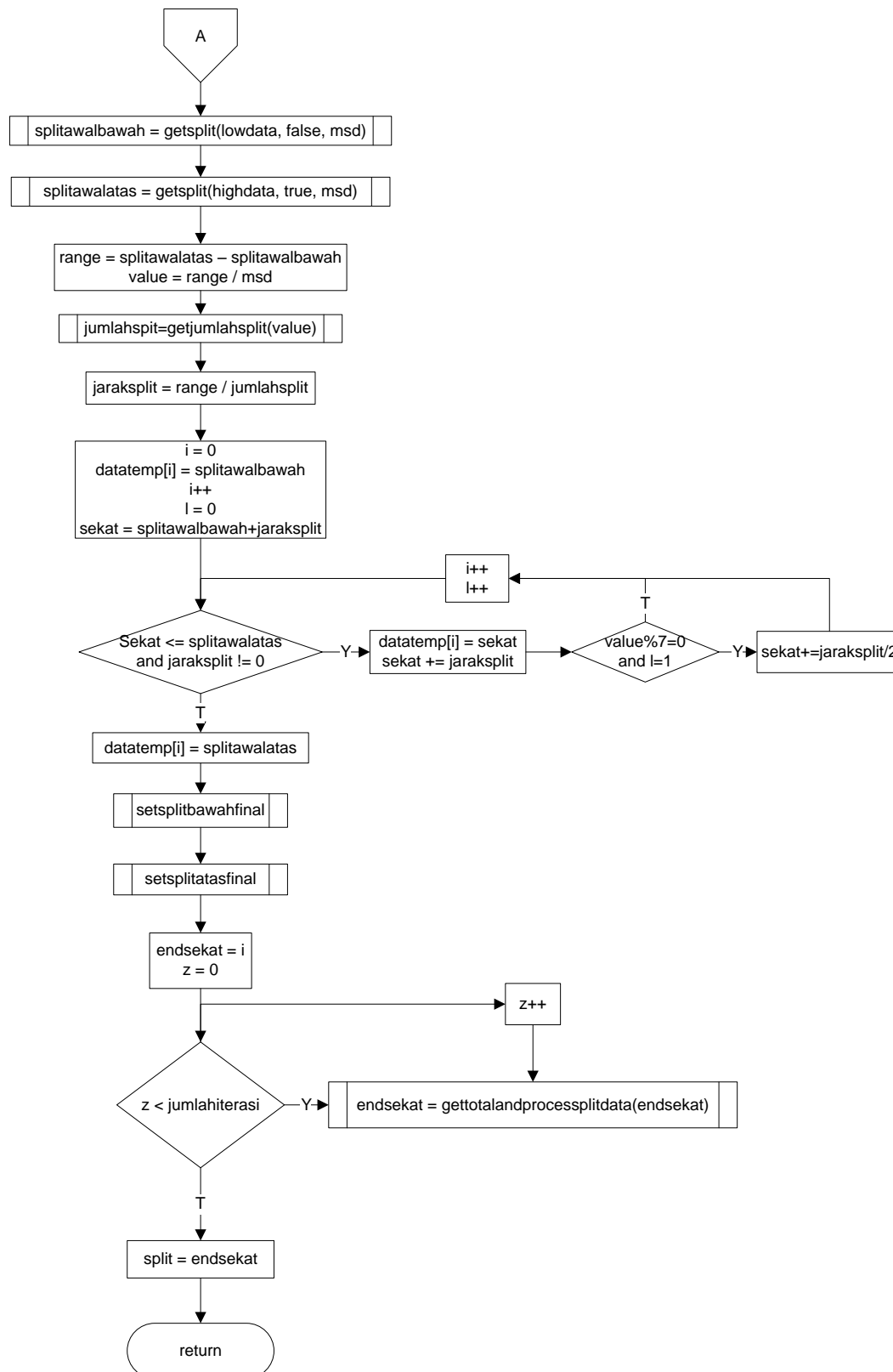
Data yang digunakan dalam aplikasi ini berupa angka yang sangat bervariasi dalam *range* tertentu. Oleh karena itu butuh adanya segmentasi data atau penggunaan interval agar jumlah variasi nilai yang ada dapat diperkecil sehingga dapat lebih efektif untuk diproses menjadi *decision tree* dan *rule* maupun dalam mengklasifikasi data berdasar *rule* yang telah ada. Jika tidak dibuat interval-interval, maka *decision tree* yang dihasilkan akan menjadi sangat rumit dan *rule* yang dihasilkan terlalu detail, dalam artian nilai beda sedikit maka sudah tidak termasuk dalam *rule*. Dalam aplikasi ini disediakan 2 metode segmentasi data yang memang akan diperbandingkan dalam penelitian ini, yaitu *Natural Partitioning* dan *Gini Index*.

### 5.2.2 Intuitive Partitioning

*Procedure* untuk proses segmentasi data menggunakan metode *Intuitive Partitioning* dapat dilihat pada Gambar 5.7. dan Gambar 5.8. Proses ini dilakukan untuk setiap *field* data yang akan digunakan.



Gambar 5.7 Flowchart Intuitive Partitioning



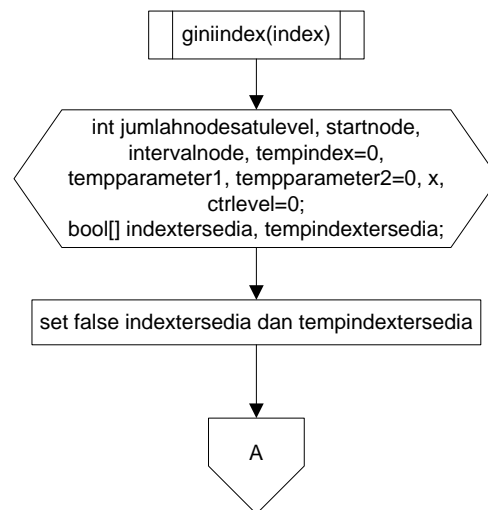
Gambar 5.8 Flowchart Intuitive Partitioning (lanjutan)

Untuk awal proses *intuitivepartitioning* dicari nilai minimal dan maksimal serta nilai *low* dan *high* maupun *msd* (*most significant digit*) awal dari data yang ada. Tentu saja data sudah

dalam kondisi urut. *Low* data adalah data ke 5% dari keseluruhan data, *high* data adalah data ke 95% dari keseluruhan data, sedangkan *msd* adalah kelipatan 10 yang terdekat dengan range data awal (hasil bagi *range* dengan *msd* bernilai 1 sampai 10). Setelah itu ditentukan batas dalam setiap interval yang ada berdasar aturan dalam algoritma *Intuitive Partitioning* seperti yang telah dijelaskan di bab 2.

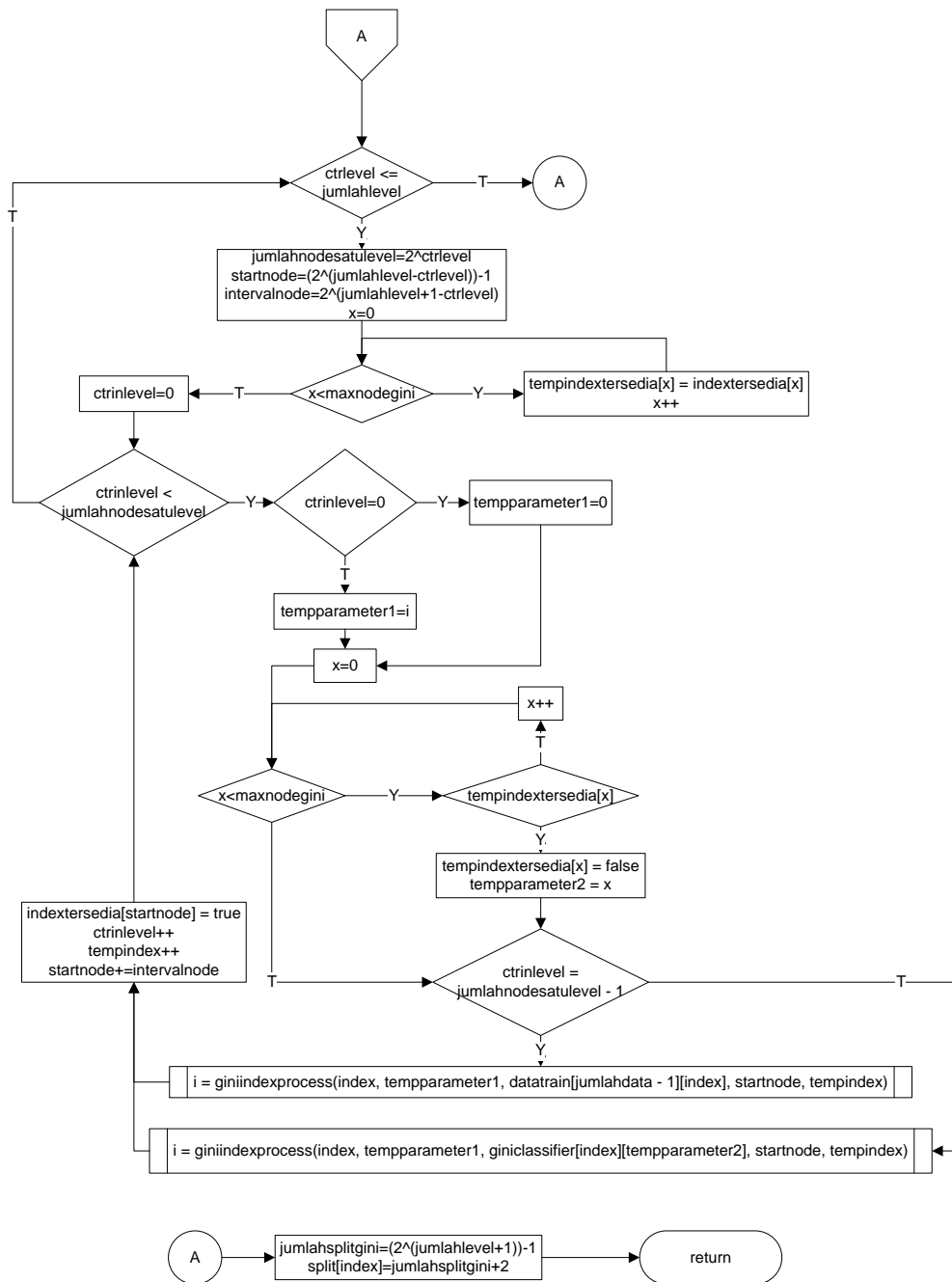
### 5.3 Gini Index

*Flowchart* untuk alur proses keseluruhan segmentasi data menggunakan *Gini Index* dapat dilihat pada Gambar 5.9. dan Gambar 5.10. Sedangkan proses untuk menentukan nilai sebagai batas interval dapat dilihat di Gambar 5.11.



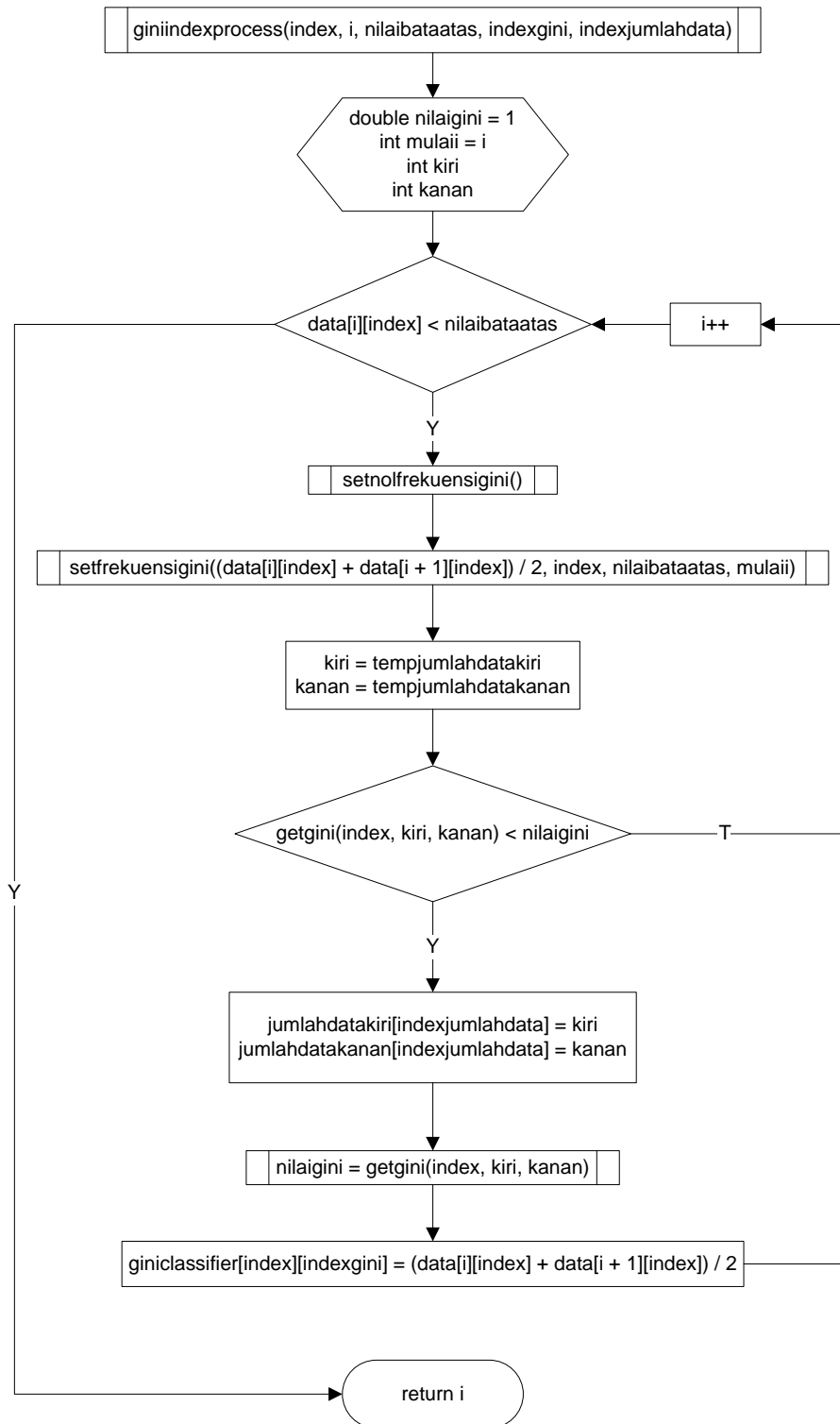
Gambar 5.9 *Flowchart giniindex*





Gambar 5.10 Flowchart giniindex (lanjutan)

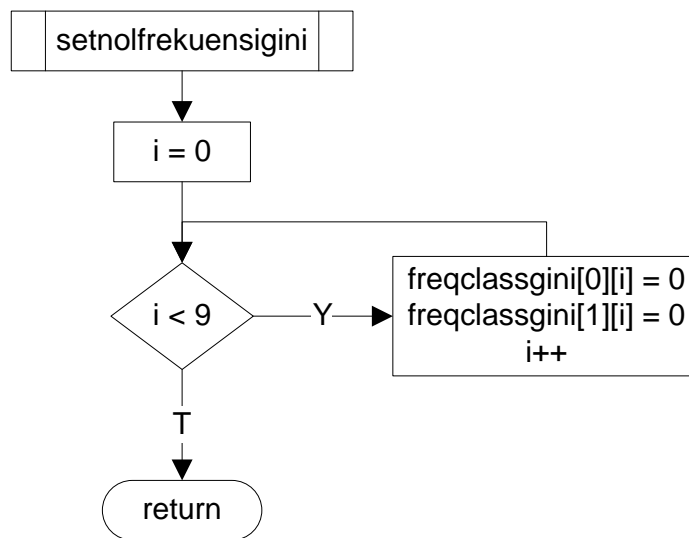
Proses yang lebih detail ada di fungsi *giniindexprocess*. Pola iterasi dilakukan bertahap setiap lapis, semisal jika iterasi dilakukan sejumlah 2 kali seperti Gambar 5.11, proses dilakukan dari lapisan awal (titik 3), tengah (titik 1 dan 5), sampai akhir (titik 0, titik 2, titik 4, dan titik 6).



Gambar 5.11 Flowchart *giniindexprocess*

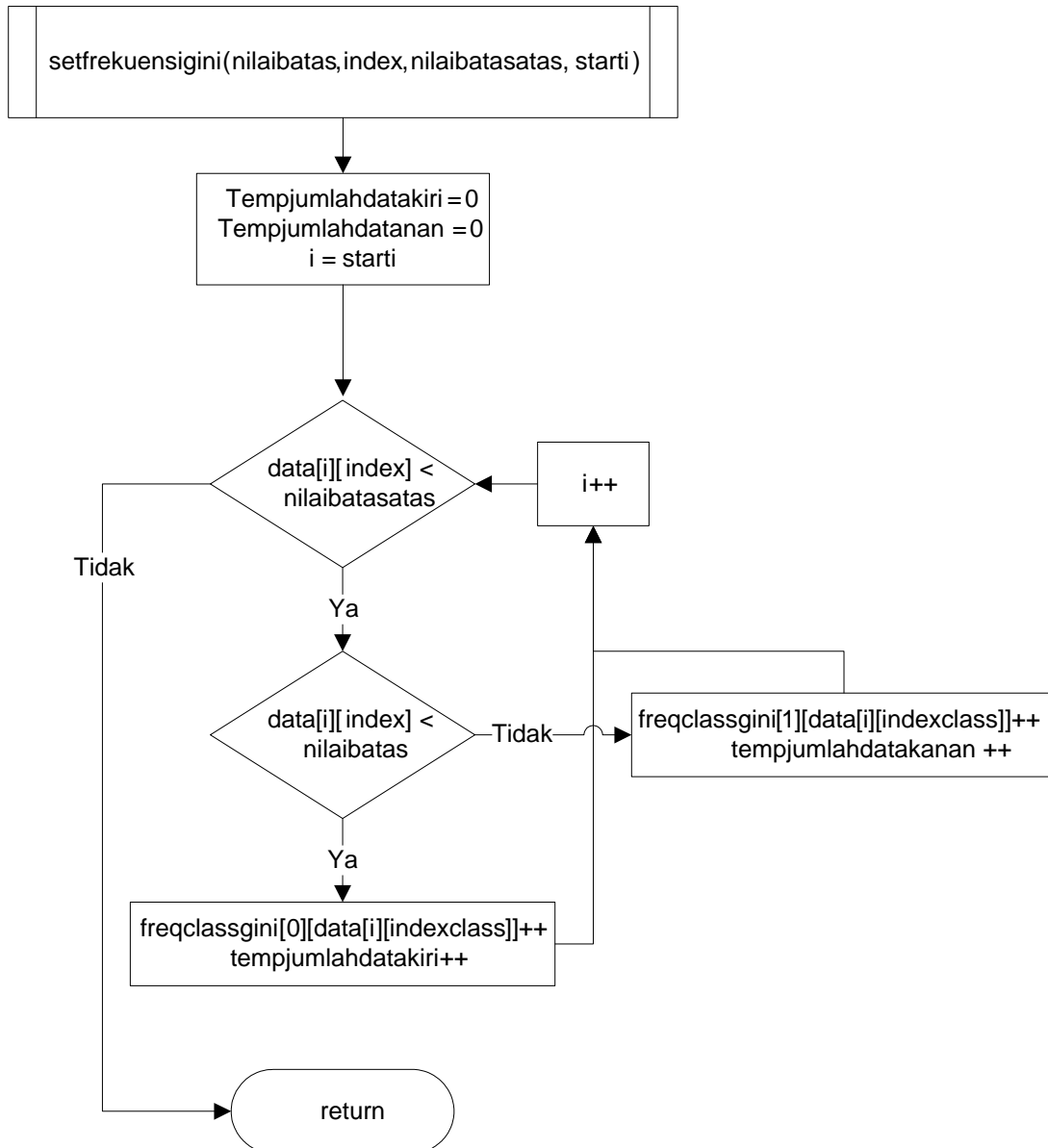
Inti dari proses *Gini Index* adalah membagi interval yang ada menjadi 2 interval baru. Agar dapat mengetahui titik pembagian yang baik untuk interval yang ada, maka setiap titik yang ada di dalam interval tersebut perlu diperbandingkan. Oleh karena itu dilakukan *looping* dari batas interval awal (nilai  $\text{data}[i][\text{index}]$  yang pertama kali dalam *looping* tersebut) sampai batas interval

akhir (*nilaibatatas*). Dimulai dengan menseset ulang variabel yang digunakan untuk menyimpan frekuensi data interval pertama dan kedua menggunakan fungsi *setnolfrekuensigini*, dapat dilihat pada Gambar 5.12. Setelah itu dilanjutkan dengan mensesetnya sesuai titik yang diuji saat ini menggunakan fungsi *setfrekuensigini*, dapat dilihat pada Gambar 5.13. Setelah itu nilai *gini* yang ada saling diperbandingkan untuk mendapat yang terbaik. Proses penghitungan nilai *gini* dapat dilihat pada Gambar 5.14. Titik batas interval dipilih yang memiliki nilai *gini* terkecil, karena semakin kecil nilai *gini* maka semakin efektif penempatan titik batas intervalnya.



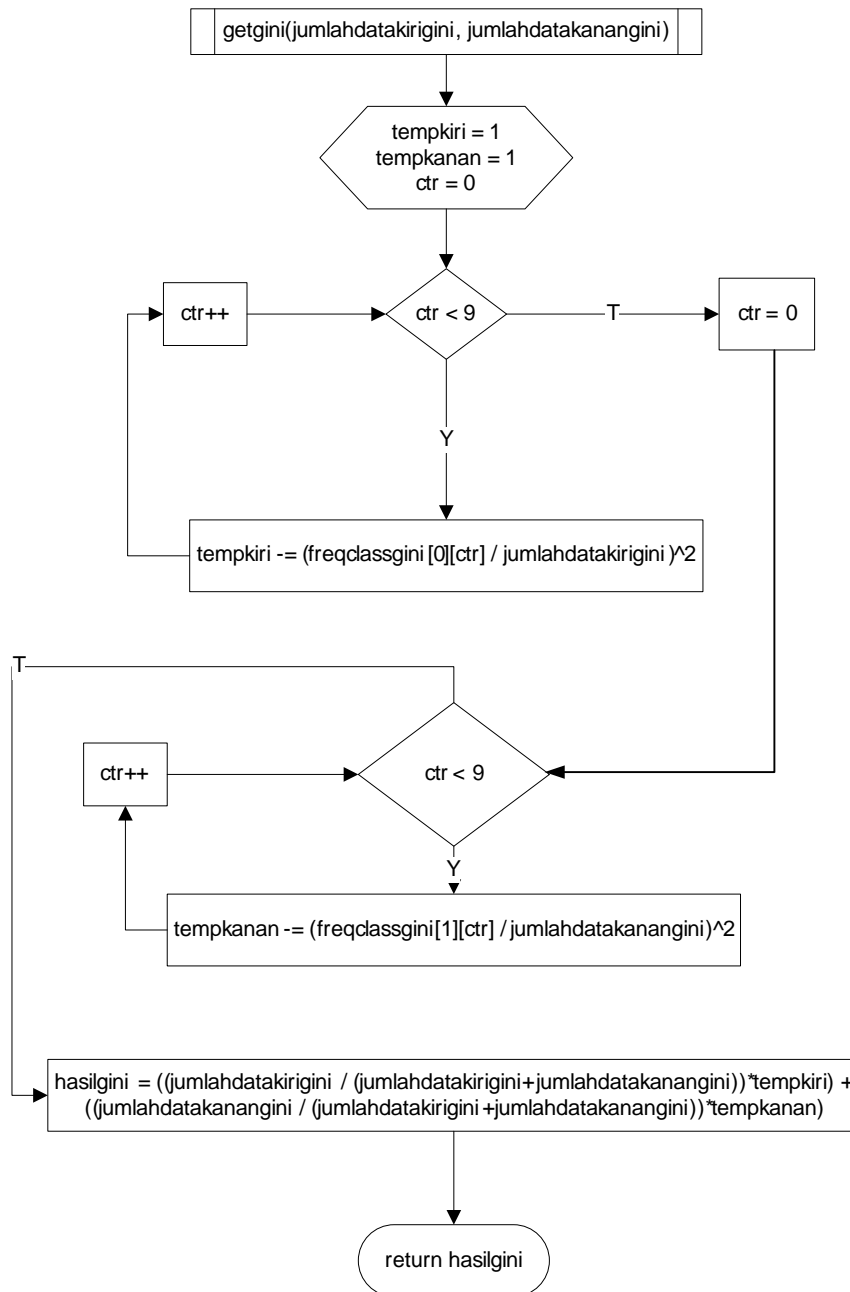
Gambar 5.12 Flowchart *setnolfrekuensigini*

Fungsi *setnolfrekuensigini* berguna untuk menseset ulang variabel *freqclassgini* yang dipakai untuk menampung frekuensi *class* di interval pertama dan kedua menjadi 0



Gambar 5.13 Flowchart setfrekuensi

Fungsi *setfrekuensi* berguna untuk menghitung jumlah frekuensi setiap *class* yang ada di interval pertama dan kedua. Proses ini dilakukan di dalam interval awal, sebelum dipecah menjadi interval pertama dan kedua.

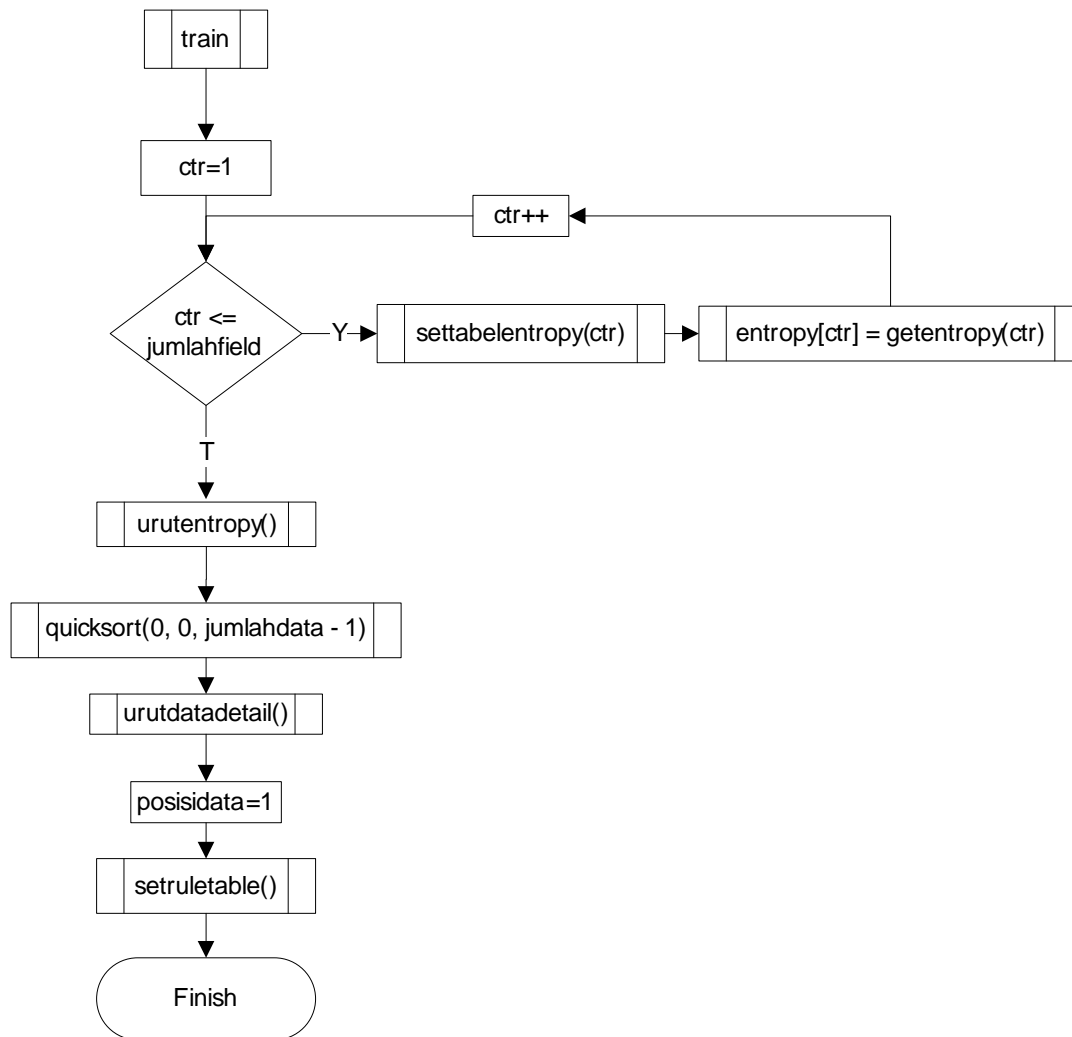


Gambar 5.14 *Flowchart getgini*

Fungsi *getgini* berguna untuk menghitung nilai *gini index*.

#### 5.4 Garis Besar Modul Training

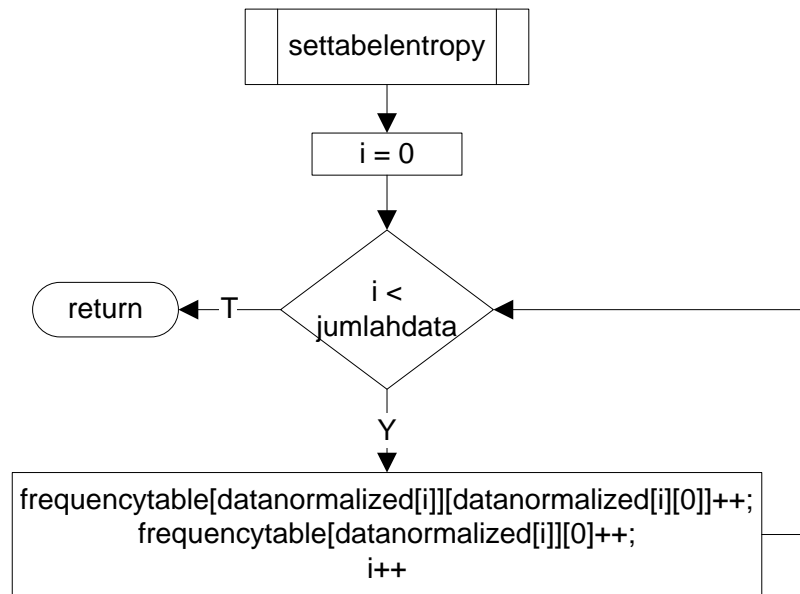
Alur proses dalam modul training dapat dilihat pada Gambar 5.15.



Gambar 5.15 Flowchart alur modul *training*

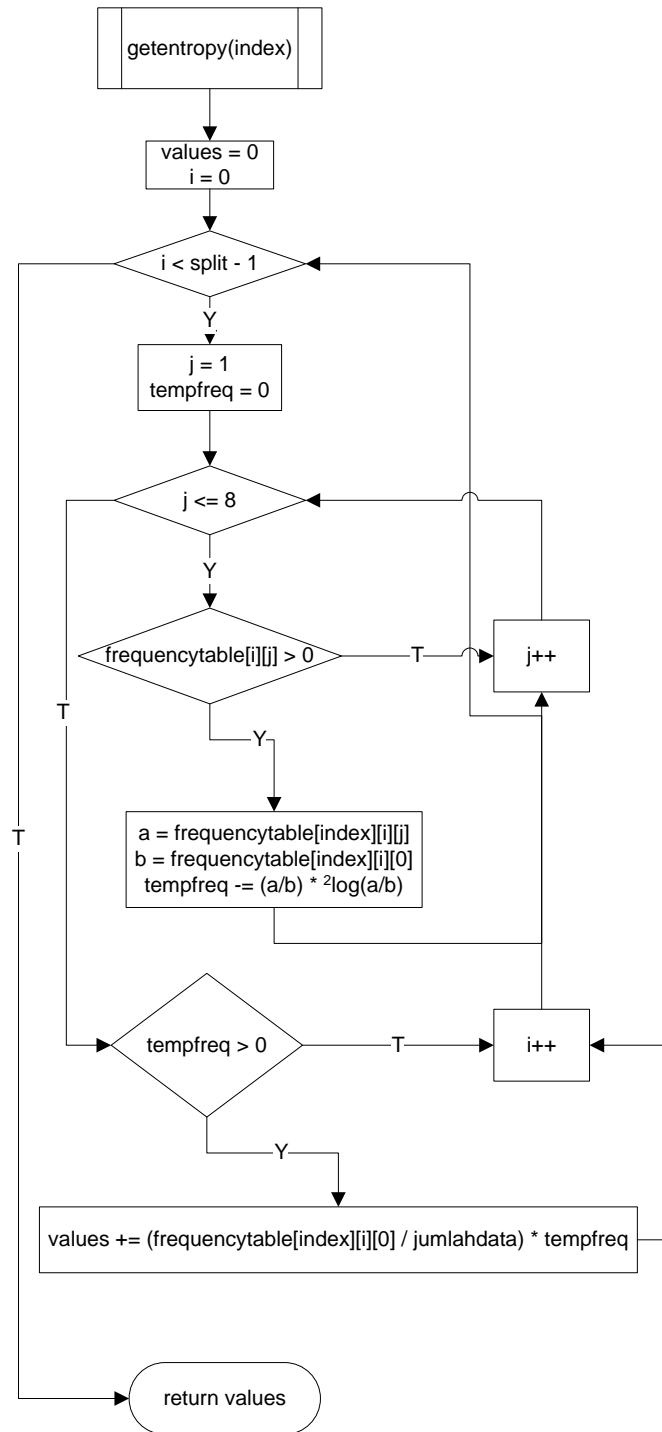
Proses training secara sederhana dan berurutan adalah sebagai berikut :

- a. Menghitung *entropy* setiap *field* dengan fungsi *getentropy* (dapat dilihat pada Gambar 5.17), di mana sebelumnya data-data yang dibutuhkan dalam proses perhitungan *entropy* seperti frekuensi dan lain sebagainya telah dihitung menggunakan fungsi *settabelentropy* (dapat dilihat pada Gambar 5.16) dan mengurutkannya dalam rangka menentukan urutan *field* yang dipakai dalam proses pembentukan *decission tree*.
- b. Mengurutkan data secara detail bertahap berdasar nilai dalam urutan *field* yang digunakan menggunakan fungsi *urutdatadetail*.
- c. Membuat *decission tree* dan *rule* (dapat dilihat pada Gambar 5.18)



Gambar 5.16 *Flowchart settabelentropy*

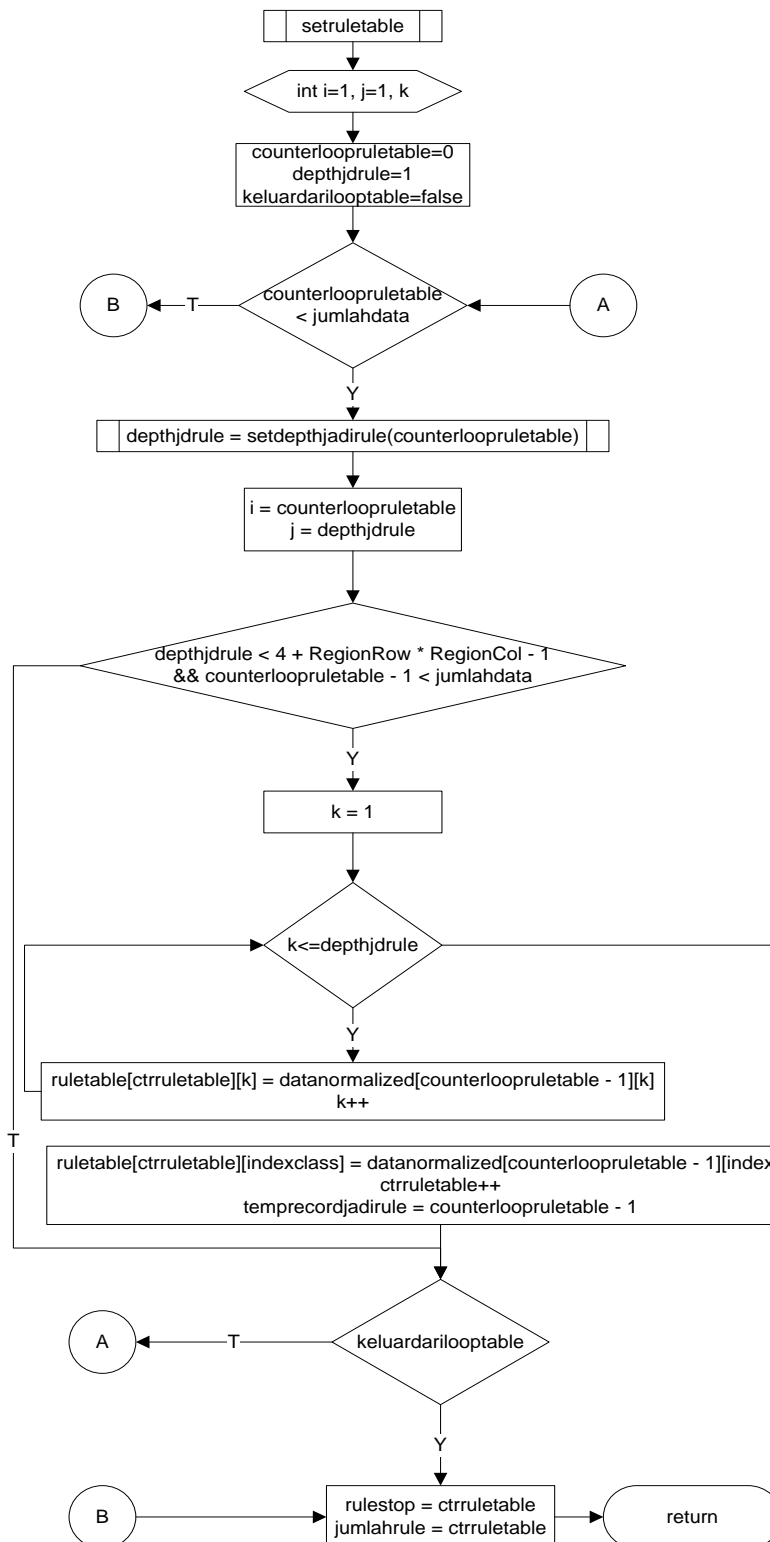
Fungsi *settabelentropy* digunakan untuk menset frekuensi jumlah setiap *class* dan nilai datanya (yang sudah disegmentasi) dalam setiap *field* yang digunakan. Hal ini dibutuhkan untuk menunjang perhitungan *entropy*.



Gambar 5.17 Flowchart *getentropy*

Fungsi *getentropy* digunakan untuk menghitung nilai *entropy* setiap *field* yang digunakan. Rumus yang digunakan sesuai dengan yang ada di algoritma ID3 / C4.5.





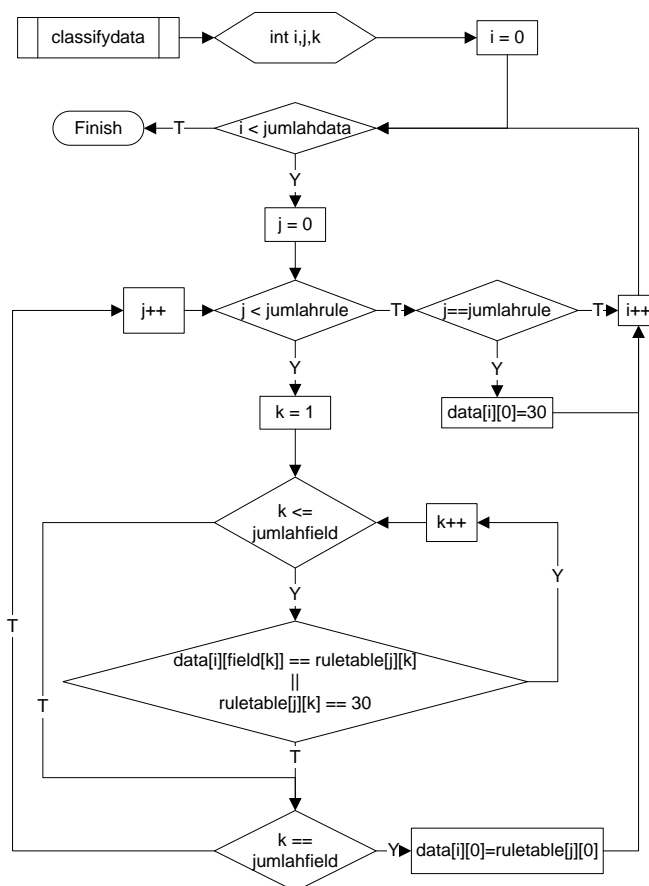
Gambar 5.18 Flowchart setruletable

Fungsi *setruletable* digunakan untuk membuat *rules* dari data yang sudah diurutkan secara detail sesuai *field* yang telah diurutkan dalam rangka metode *decision tree* yang digunakan. *Looping* dilakukan dari data pertama sampai terakhir. Setiap ditemukan *class* yang sama dengan

nilai beberapa *field* yang sama, maka nilai-nilai tersebut dijadikan *rules*. Proses untuk menemukan *rule* tersebut dikerjakan oleh fungsi *setdepthjadirule*. Caranya adalah melihat setiap *field* secara berurutan. Misal *field* awal adalah *field area*. Untuk nilai *area* yang sama dilakukan *looping* sampai nilai *area* berubah. Jika dalam nilai *area* yang sama *class* yang ada sama semua, maka nilai *area* tersebut dijadikan *rules*. Jika tidak, maka *field* yang dilihat ditambah 1, misalnya *perimeter*. Jadi sekarang yang dilihat adalah data yang memiliki nilai *area* dan nilai *perimeter* tertentu. Dengan demikian akan terbentuk *rules*.

### 5.5 Garis Besar Modul Klasifikasi

Alur proses dalam modul klasifikasi dapat dilihat pada Gambar 5.19.

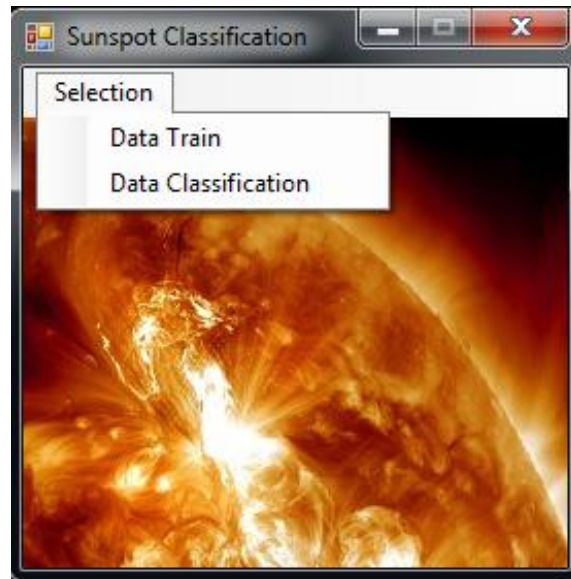


Gambar 5.19 Flowchart alur modul *classification*

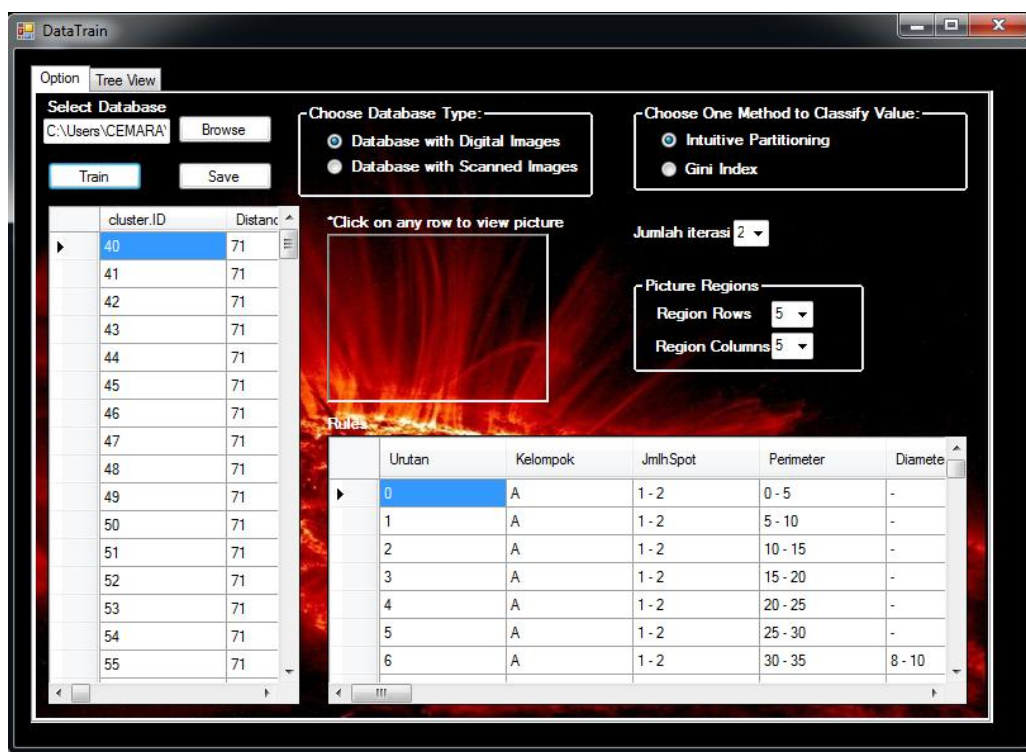
Cara kerja fungsi *classifydata* dalam modul *classification* adalah dengan mencocokkan nilai dari *record* yang diklasifikasi dengan *rule* yang ada.

## 5.6 Antarmuka sub-sistem klasifikasi kelompok bintang matahari dengan ID3 / C4.5.

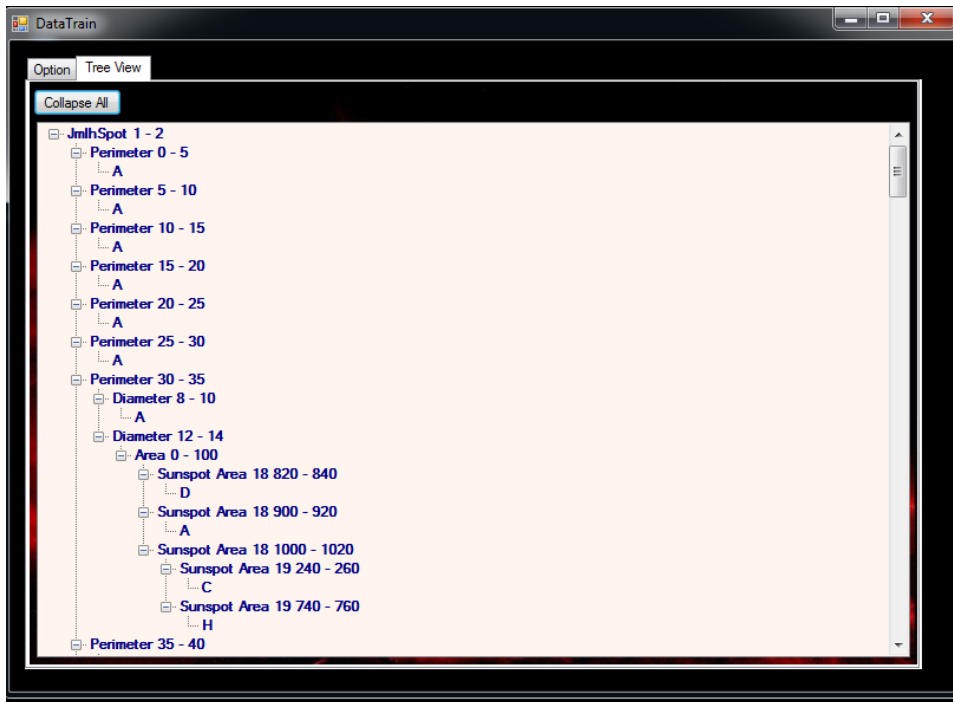
Berikut ini adalah antarmuka prototipe sub-sistem klasifikasi kelompok bintang matahari dengan ID3 / C4.5. Antarmuka dapat dilihat pada Gambar 5.20. sampai dengan Gambar 5.23.



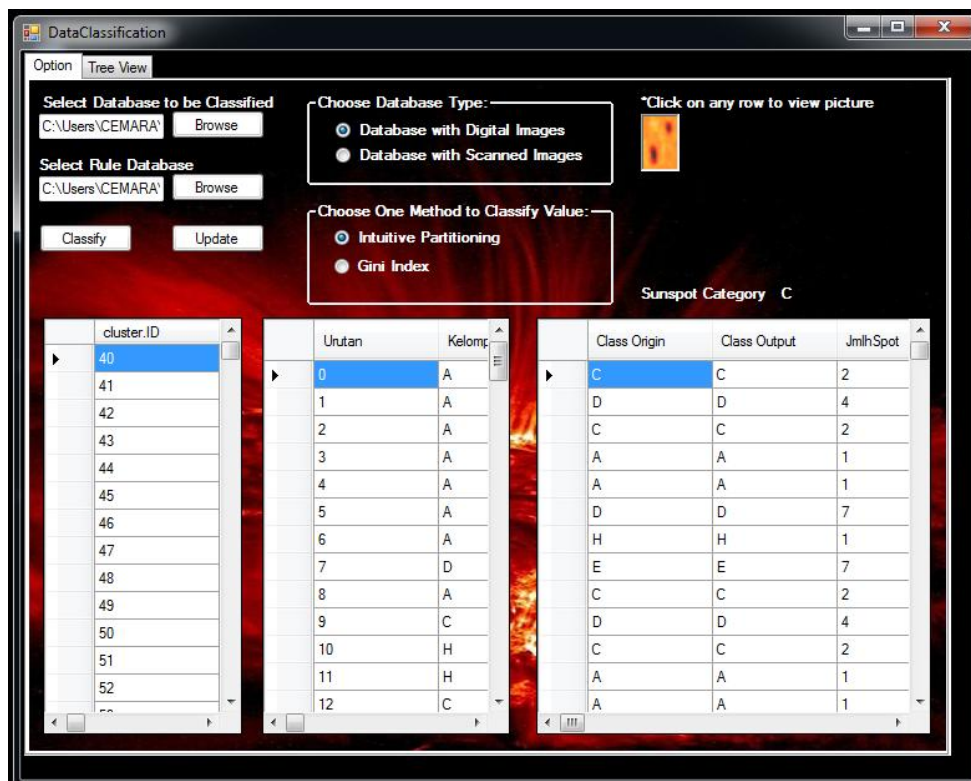
Gambar 5.20 Antarmuka Awal Prototipe



Gambar 5.21 Antarmuka untuk proses training data



Gambar 5.22 Antarmuka tampilan Decision Tree yang dihasilkan dari proses training



Gambar 5.23 Antarmuka untuk menampilkan hasil proses klasifikasi dengan ID3/C4.5.

### 5.7 Pengujian prototipe sub-sistem klasifikasi dengan metode ID3/C4.5.

Pengujian kecepatan proses dalam training metode ID3 / C4.5. saat membentuk Decision Tree dan juga rule – rule decision tidak dilakukan karena sangat cepat. Kecepatan proses training selalu dibawah 1 detik, dimana ini kurang lebih setara dengan kecepatan training metode JST Probabilistik yang dipilih. Jadi untuk prototipe ini pengujian dilakukan hanya pada akurasi hasil klasifikasinya. Pada setiap pengujian akurasi akan dibandingkan pula hasil akurasi segmentasi data menggunakan metode segmentasi *Intuitive Partitioning* dan *Gini Index*. Pengujian pengujian tersebut adalah sebagai berikut:

1. Pengujian akurasi klasifikasi berdasarkan perubahan jumlah data training yang digunakan. Hasil pengujian yang ditampilkan pada Tabel 5.1. diambil dari hasil akurasi pengujian dengan setting terbaik untuk setiap jumlah data training. Sementara jumlah region yang digunakan adalah 5 x 5.

Tabel 5.1 Akurasi klasifikasi berdasarkan perubahan jumlah data training (dalam %)

Jumlah Data Training	Data Training		Data Testing	
	Metode			
	Intuitive	Gini	Intuitive	Gini
50	100	100	58,53	68,29
100	99	100	71,5	72,8
150	99,33	100	84,37	82,81
214	99,6	100	-	-

2. Pengujian akurasi klasifikasi berdasarkan perubahan jumlah region. Hasil pengujian yang ditampilkan pada Tabel 5.2. Untuk pengujian ini digunakan data training sebanyak 150 data dan data testing sebanyak 64 data.

Tabel 5.2 Akurasi klasifikasi berdasarkan perubahan jumlah region (dalam %)

Jumlah Region	Intuitive		Gini	
	Akurasi Terhadap Data Training	Akurasi Terhadap Data Testing	Akurasi Terhadap Data Training	Akurasi Terhadap Data Testing
-	90	82,81	96,66	78,12
	93,33	82,81	96,66	79,68
	96,66	65,62	63,33	70,31
3 x 3	98,66	79,68	100	78,12
	96	78,12	100	81,25
	95,33	54,68	100	82,81
5 x 5	98,66	84,37	100	78,12
	97,33	82,81	100	81,25
	98	59,37	100	82,81
7 x 7	99,33	82,81	100	78,12
	98	81,25	100	81,25
	50,66	78,12	100	82,81

**Ket:** Tanda ‘-’ artinya pada pengujian tersebut tidak digunakan pembagian region.  
 Jumlah region = jumlah pixel dari citra yang diproses.

- Perbandingan pengujian akurasi klasifikasi antara data yang disegmentasi dalam bentuk range-range data dan data asli tanpa disegmentasi.

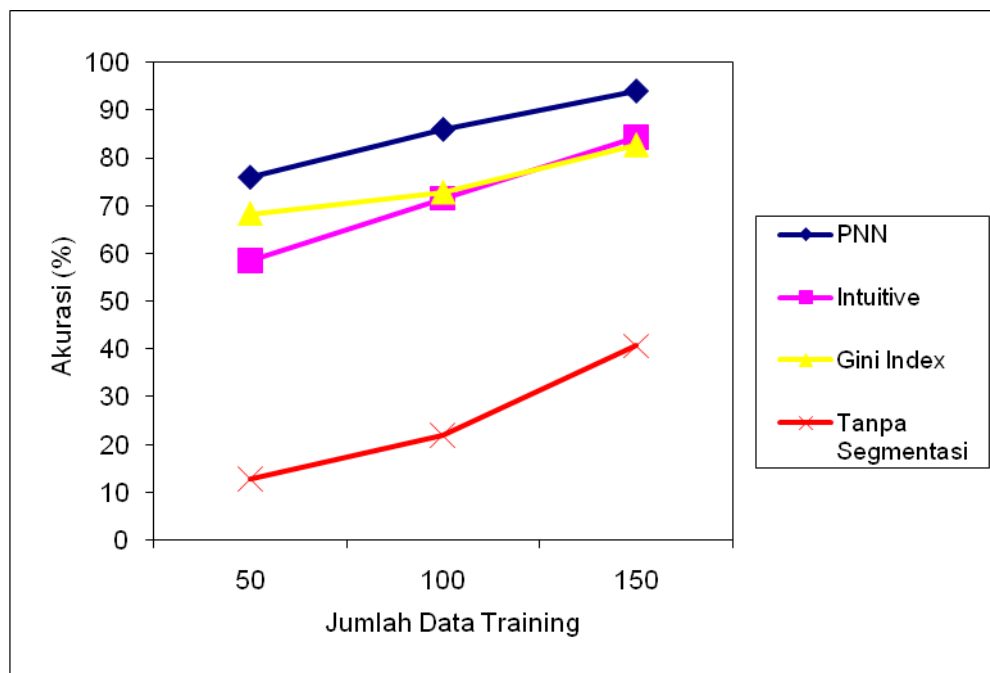
Sub-sistem dibuat dengan mengasumsikan bahwa nilai – nilai data terlalu beraneka ragam sehingga perlu disegmentasi / generalisasi menjadi range – range data menggunakan metode *Intuitive Partitioning* atau *Gini Index*. Belakangan muncul pemikiran jangan – jangan bila data mentah langsung diklasifikasi tanpa di-segmentasi (generalisasi), akan menghasilkan tingkat akurasi yang lebih baik. Untuk itu kami juga melakukan pengujian dengan meniadakan proses segmentasi. Hasil perbandingan pengujian akurasi klasifikasi dengan dan tanpa proses segmentasi dapat dilihat pada Tabel 5.3. Pada tabel tersebut tampak bahwa akurasi proses klasifikasi dengan ID3/C4.5 yang tidak didahului dengan proses segmentasi / generalisasi data akan sangat buruk untuk data – data yang tidak ditrainingkan (data testing).

Tabel 5.3 Perbandingan akurasi klasifikasi dengan dan tanpa proses segmentasi data (dalam %)

Jumlah Data Training	Jumlah Data Testing	Akurasi Klasifikasi terhadap Data Training			Akurasi Klasifikasi terhadap Data Testing		
		Metode Segmentasi					
		-	Intuitive	Gini	-	Intuitive	Gini
50	164	100	100	100	12,8	58,53	68,29
100	114	100	99	100	21,92	71,5	72,8
150	64	99,33	99,33	100	40,62	84,37	82,81
214	-	99,53	100	100	-	-	-

**Ket:** Tanda '-' artinya tanpa menggunakan proses segmentasi / generalisasi data.

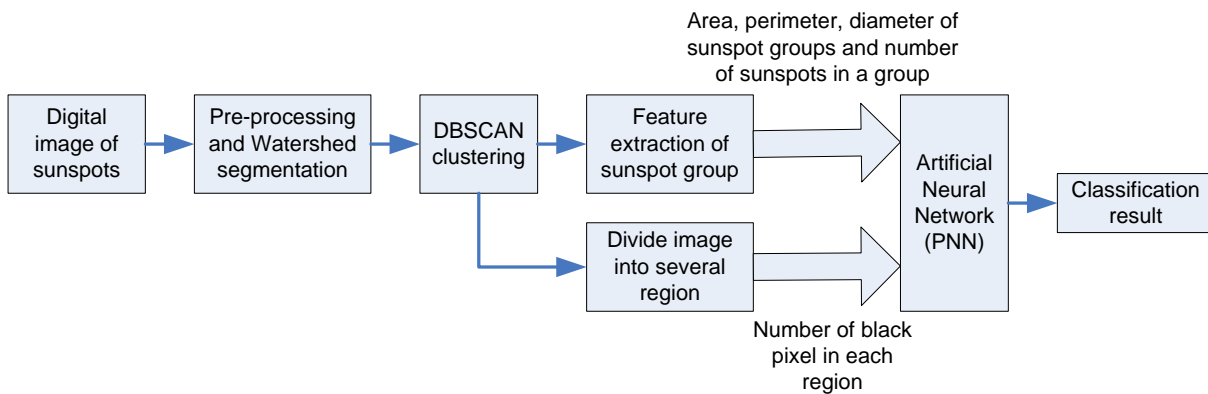
Terakhir, untuk melihat apakah hasil klasifikasi tipe badai matahari menggunakan metode ID3/C4.5. lebih baik atau tidak bila dibandingkan klasifikasi menggunakan metode JST terpilih, yaitu Jaringan Syaraf Tiruan Probabilistik (PNN), kami bandingkan hasil pada Tabel 5.3. dengan hasil pengujian PNN. Hasil perbandingan dapat dilihat pada Gambar 5.24.



Gambar 5.24 Perbandingan metode klasifikasi PNN dan ID3/C4.5

### 5.8 Software Sistem Klasifikasi Otomatis Tipe Badai Matahari

Pada Gambar 5.25. dapat dilihat blok diagram disain Sistem Klasifikasi Otomatis Tipe Badai Matahari / Kelompok Bintik Matahari berdasarkan metode klasifikasi *Modified – Zurich Sunspot Classification*. Sistem ini merupakan gabungan dari beberapa sub-sistem terpilih yang telah dibangun pada tahun pertama, kedua dan ketiga.



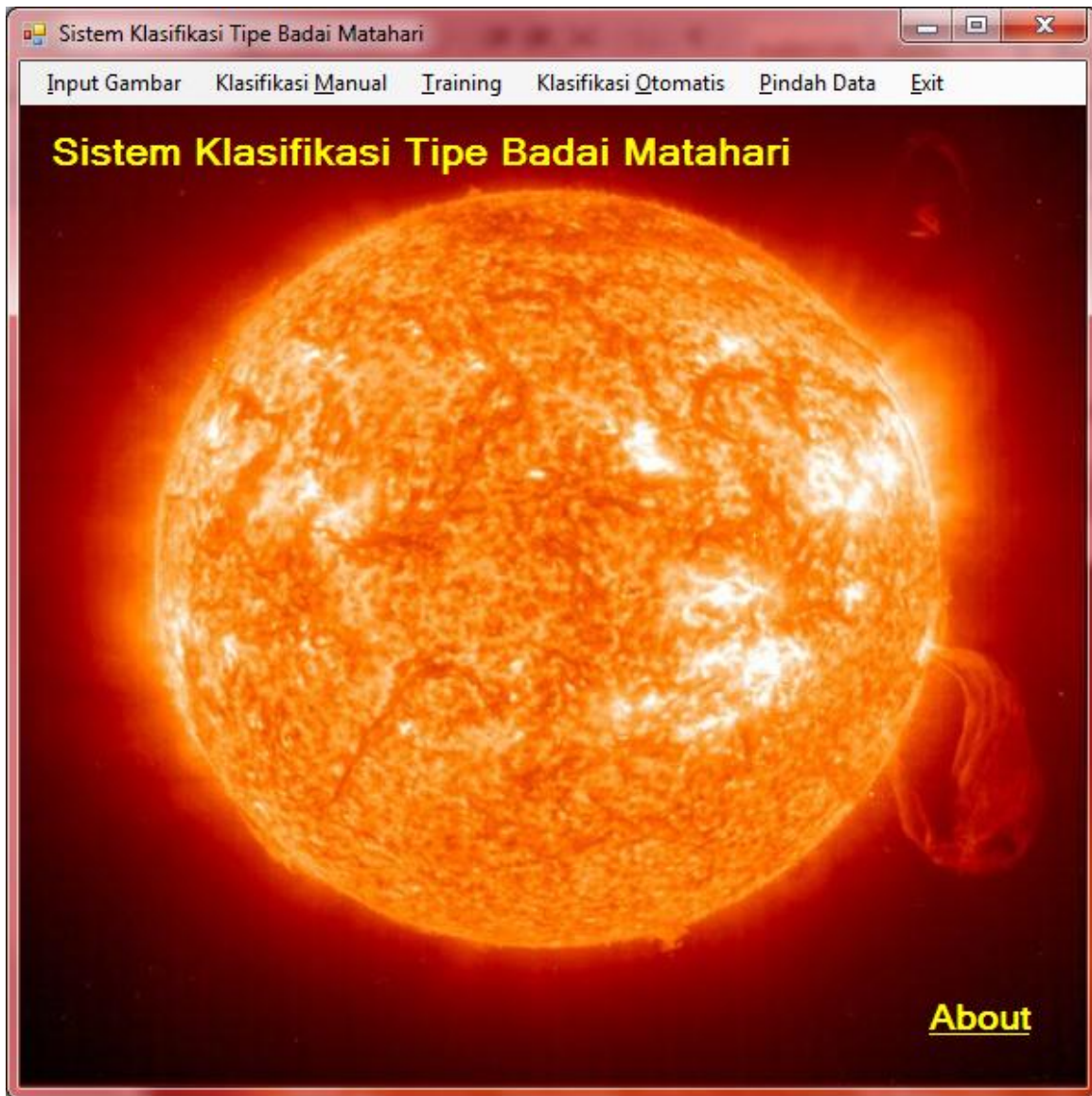
Gambar 5.25 Desain Sistem Keseluruhan

Proses secara detail dari masing – masing sub-sistem telah dibahas pada dua laporan penelitian hibah bersaing untuk penelitian ini yang telah dibuat pada tahun – tahun sebelumnya (Adipranata dkk, 2010; Budhi dkk, 2011).

### 5.9 Antarmuka Sistem Klasifikasi Otomatis Tipe Badai Matahari

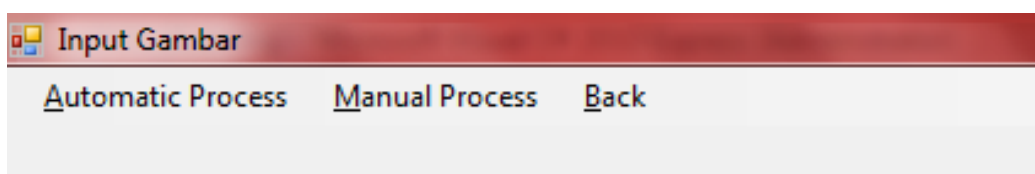
Berikut ini adalah sejumlah antarmuka Sistem Klasifikasi Otomatis Tipe Badai Matahari secara keseluruhan dari mulai awal data – data sample digunakan pada proses training jaringan syaraf tiruan sanmapi dengan proses klasifikasi sebuah citra matahari. Antarmuka awal saat system dijalankan dapat dilihat pada Gambar 5.26.





Gambar 5.26 Antarmuka awal Sistem Kasifikasi Otomatis Tipe Badai Matahari

Menu “Input Gambar” pada tampilan awal bila dipilih akan memunculkan tampilan seperti pada Gambar 5.27. Menu ini digunakan untuk memasukkan dan memproses data – data sample menjai data training yang siap digunakan.



Gambar 5.27 Antarmuka saat menu “Input Gambar” pada tampilan awal dipilih

Pada saat menu “Automatic Process” pada antarmuka “Input Gambar” dipilih, maka computer secara otomatis akan melakukan proses untuk membuka sebuah citra digital matahari, melakukan *pre-processing* dan *watershed segmentation*, serta mengelompokkan bintik – bintik matahari yang telah tersegmentasi menjadi kelompok – kelompok bintik matahari menggunakan metode data mining DBScan. Hasil dari proses akan ditampilkan pada antarmuka seperti pada Gambar 5.28.

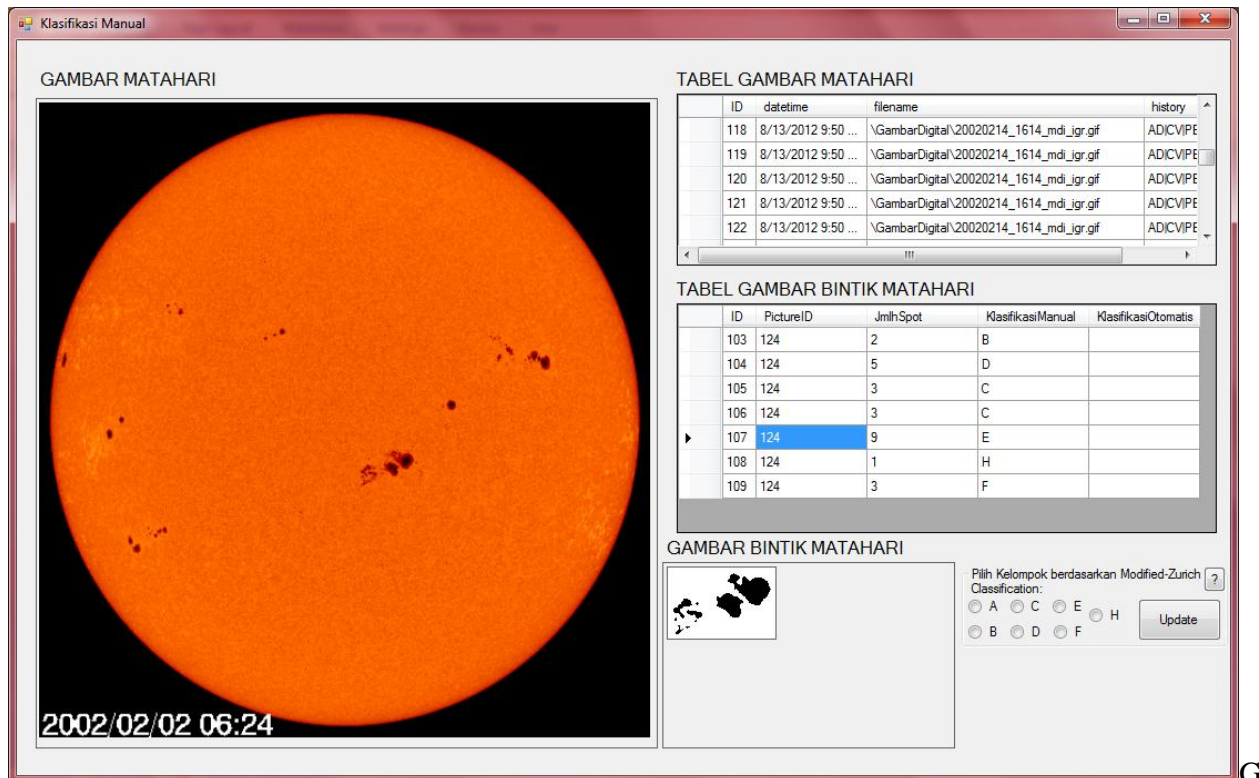
Sementara itu, bila diinginkan untuk melakukan proses input data, pre-processing, segmentasi dan pengelompokkan secara mandiri, dapat digunakan menu “Manual Process” pada antarmuka “Input Data”.

ID	PictureID	JmlhSpot
253	146	2
254	146	4
255	146	2
256	146	1
257	146	7
258	146	1
259	146	9

Gambar 5.28 Antarmuka hasil akhir saat menu “Automatic Process” pada tampilan “Input Gambar” dipilih.

Menu “Klasifikasi Manual” pada tampilan Awal system digunakan untuk melakukan klasifikasi manual pada data – data sample yang telah diinputkan dan diproses. Klasifikasi manual ini harus dilakukan oleh ahli badai matahari dari LAPAN. Guna dari klasifikasi manual ini digunakan untuk menyiapkan data – data training, berupa pasangan data sample dan target klasifikasi dari data tersebut. Antarmuka yang digunakan untuk klasifikasi manual dapat dilihat pada Gambar 5.29. Pada antarmuka ini akan disajikan kelompok – kelompok bintik matahari dan citra lengkap matahari pada pakar badai matahari yang akan mengklasifikasi. Selanjutnya

pakar tersebut dapat memilih tipe klasifikasi (A s/d H) yang sesuai. Hasil klasifikasi manual akan disimpan pada database klasifikasi manual.



ambar 5.29 Antarmuka “Klasifikasi Manual”

Menu “Training” pada antarmuka tampilan awal digunakan untuk melakukan proses training Jaringan Syaraf Tiruan Probabilistik menggunakan data – data sample dan target yang ada. Antarmuka proses training ini dapat dilihat pada Gambar 5.30. Pada proses ini pertama-tama dapat dipilih range database klasifikasi manual yang akan digunakan untuk training. Selanjutnya tombol “preprocessing” digunakan untuk merubah format data – data training menjadi bentuk yang dapat diterima oleh jaringan syaraf tiruan (skala 0 s/d 1). Setelah ditentukan spreadnya (default: 0.2 karena terbaik saat pengujian), maka dengan menekan tombol “Train” proses training akan berjalan secara otomatis. Bila telah selesai, user dapat menyimpan data “weight” yang dihasilkan pada proses training ke dalam file.

Form Training

Training Data

Tanggal Awal Wednesday, August 01, 2012

Tanggal Akhir Friday, October 12, 2012

Pilih Database Input

Input Konstanta PNN

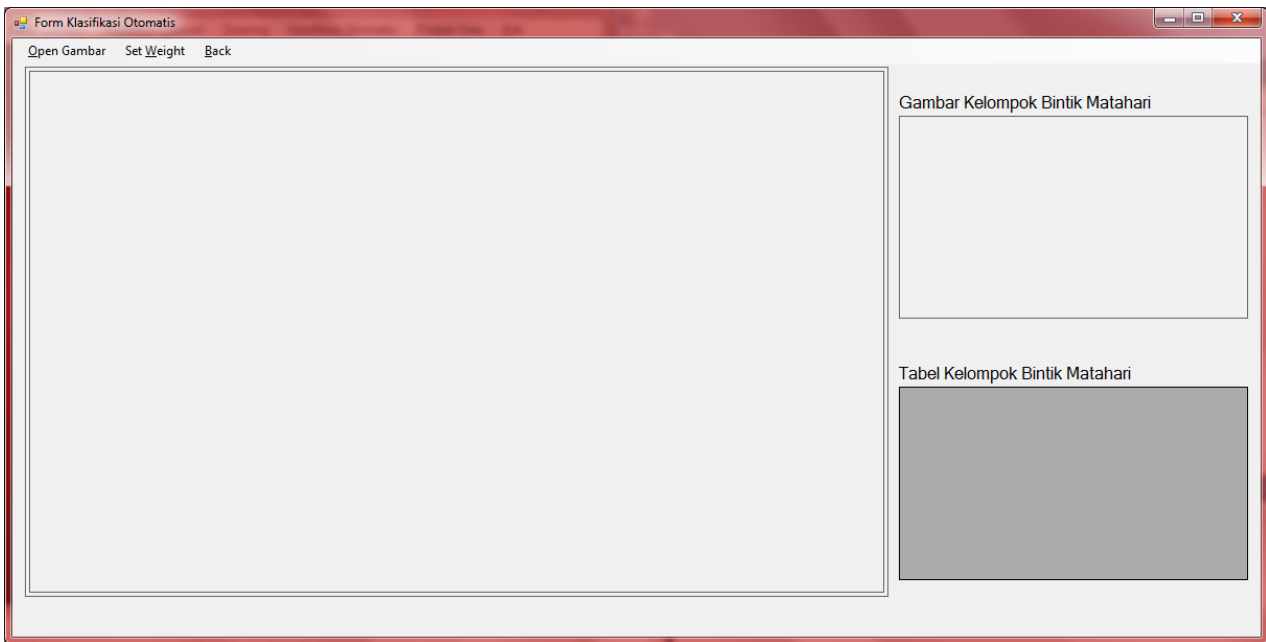
Spread 0.2

Tabel Gambar Yang Akan Di-Training

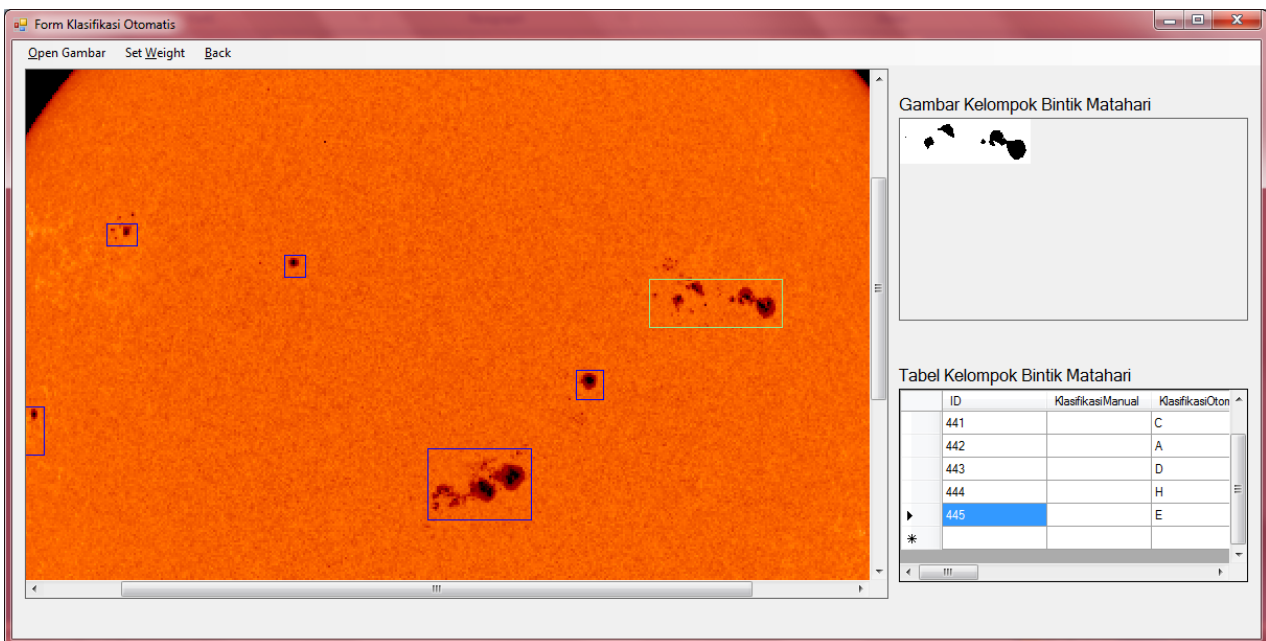
ID	FileName	DateTime	History
114	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
115	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
116	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
117	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
118	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
119	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
120	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
121	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
122	\GambarDigital\20...	8/13/2012 9:50 PM	AD CV F
138	\GambarDigital\20...	9/11/2012 8:34 PM	AD CV F
139	\GambarDigital\20...	9/11/2012 8:34 PM	AD CV F

Gambar 5.30 Antarmuka proses training JST Probabilistik

Bila menu “Klasifikasi Otomatis” pada tampilan awal dipilih maka akan memanggil antarmuka seperti pada Gambar 5.31. Bagian ini digunakan untuk melakukan klasifikasi otomatis terhadap data citra matahari baru yang didapat dari peralatan *Michelson Doppler Image (MDI) Continuum/SOHO (Solar dan Heliospheric Observatory)*. Pada langkah awal user dapat melakukan set ulang “weight” yang akan digunakan dan menjadikan set yang dipilih menjadi default bila lain kali aplikasi digunakan kembali. Selanjutnya bila user memilih menu “Open Gambar” sistem akan melakukan meminta inputan citra digital matahari, dan selanjutnya secara otomatis melakukan proses segmentasi bintang matahari, pengelompokan bintang matahari dan klasifikasi kelompok bintang matahari / tipe badai matahari. Hasil klasifikasi akan disimpan apada database hasil klasifikasi otomatis. Contoh hasil dapat dilihat pada Gambar 5.32.



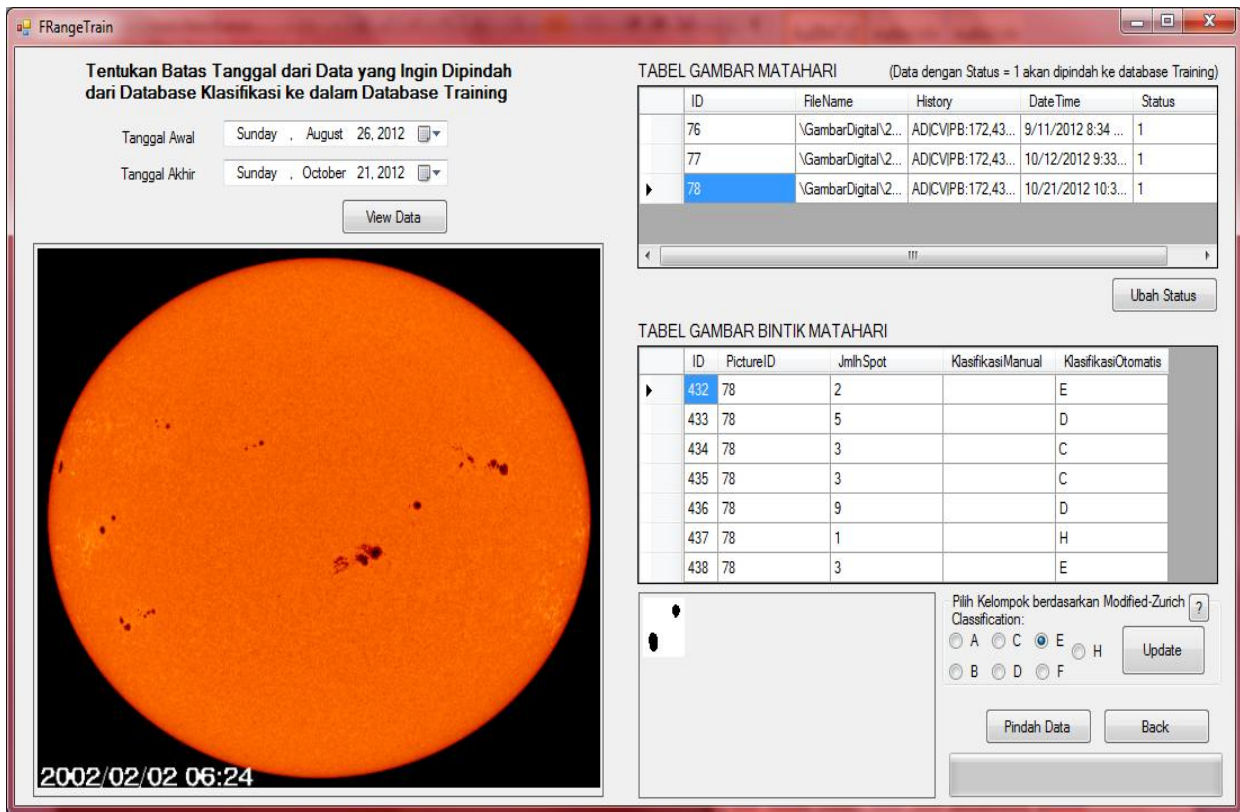
Gambar 5.31 Antarmuka Klasifikasi Otomatis saat pertama kali dipanggil



Gambar 5.32 Antarmuka Klasifikasi Otomatis setelah sebuah proses klasifikasi berakhir

Hasil klasifikasi otomatis ini dapat pula dijadikan data sample untuk proses training di masa depan dengan cara ditambahkan kedalam database klasifikasi manual. Caranya dengan memilih menu “Pindah Data” pada tampilan awal sistem. Untuk memindahkan data klasifikasi otomatis ke database klasifikasi manual diperlukan proses verifikasi serta update (bila dianggap

perlu) dari seorang ahli badai matahari. Antarmuka untuk proses verifikasi dan update ini dapat dilihat pada Gambar 5.33.



Gambar 5.33 Antarmuka untuk verifikasi dan update pada proses pindah data

## BAB 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Dari hasil pengujian dapat dilihat bahwa metode jaringan saraf tiruan Probabilistik (PNN) memiliki tingkat akurasi klasifikasi lebih baik bila dibandingkan dengan dua metode jaringan syaraf tiruan lain, yaitu jaringan syaraf tiruan *Backpropagation (BPNN)* dan kombinasi Self-Organising Map dan K-Nearest Neighbor (SOM-KNN). Dari pengujian selanjutnya juga telah dibuktikan bahwa klasifikasi menggunakan jaringan PNN memiliki akurasi yang lebih baik bila dibandingkan dengan klasifikasi menggunakan metode ID3 / C4.5. Untuk itu pada akhirnya metode PNN diputuskan untuk dipakai pada sistem klasifikasi otomatis tipe badai matahari yang dibangun pada penelitian ini.

### 6.2 Saran

Guna perbandingan, dapat pula dibangun sebuah sub sistem klasifikasi otomatis lain yang menggunakan metode klasifikasi selain jaringan saraf tiruan dan *ID3 / C4.5*.

Penelitian dapat dilanjutkan dengan memdisain dan memdevelope suatu sistem pengenalan badai matahari berdasarkan data spektrumnya dan juga deteksi arah gerak / motion dari badai matahari. Dari hasil kedua komponen tersebut, dikombinasi dengan hasil klasifikasi tipe badai matahari yang dibangun pada penelitian ini akan dapat dibuat sebuah sistem *early warning* gangguan badai matahari terhadap kondisi cuaca antariksa di bumi dan sekitarnya.

## DAFTAR PUSTAKA

- Adipranata, R., Budhi, G.S., Setiahadhi, B. dan Anwar, B. (2010). *Segmentasi Bintik Matahari Menggunakan Metode Watershed*. Konferensi Nasional Sistem & Informatika 2010, Bali.
- Adipranata, R., Budhi, G.S., Setiahadhi, B. dan Anwar, B. (2010). *Klasifikasi Otomatis Kelompok Bintik Matahari Untuk Menganalisa Kondisi Cuaca Antariksa*. Laporan Akhir Penelitian Hibah Bersaing tahun pertama, Dikti.
- Anwar, B., Acton, L.W., Makita, M., Hudson, H.S., McClymont, A.N. and Tsuneta, S. (1993). *Rapid Sunspot Motion During A Major Solar Flare*. Solar Physics, 147, pp. 287-303, Kluwer Academic Publisher.
- Boteler, D.H. (2001). *Space Weather Effects on Power Systems*. *Space Weather*. Song, P., Singer, H.J. and Siscoe, G.L. (Eds), Geophysical Monograph, 125, pp. 347.
- Bothmer, V. and Daglis, I.A. (2007). *Space Weather, Physics and Effects*. Springer-Praxis Publishing.
- Budhi, G.S., Adipranata, R., Sugiarto, M., Anwar, B., dan Setiahadhi, B. (2011a). *Pengelompokan Sunspot Pada Citra Digital Matahari Menggunakan Metode Clustering DbScan*. Prosiding Seminar Nasional Aplikasi Teknologi Informasi 2011, Yogyakarta.
- Budhi G.S., Adipranata, R., Setiahadhi, B., Anwar, B., Hartanto, A., Tjondrowiguno, A.N. (2011b) *Kombinasi Self-Organizing Map Neural Network Dan K-Nearest Neighbor Untuk Klasifikasi Otomatis Citra Kelompok Bintik Matahari*, Prosiding Konferensi Nasional Sistem & Informatika, Bali, 2011.
- Budhi, G.S., Adipranata, R. dan Setiahadhi, B. (2011c). *Klasifikasi Otomatis Kelompok Bintik Matahari Untuk Menganalisa Kondisi Cuaca Antariksa*. Laporan Akhir Penelitian Hibah Bersaing tahun kedua, kopertis wilayah 7.
- Fausett, L. (1994). *Fundamentals of neural networks: Architectures, algorithms, and applications*. Prentice-Hall International., Inc.
- Larose, D. T. (2005). *Discovering knowledge in data : An introduction to data mining*. New Jersey: John Wiley & Sons, Inc.
- Han, J., Kamber, M., and Pei, J. (2012). *Data mining: Concepts and techniques (3rd ed.)*. Waltham USA: Morgan Kaufmann Publishers.
- Lanzerotti, L.J. (2001). *Space Weather Effects on Technologies*. *Space Weather*, Song, P., Singer, H.J. and Siscoe, G.L. (Eds), Geophysical Monograph, 125, pp. 11.



- McIntosh, Patrick. S. (1990). *The Classification of Sunspot Groups*. Solar Physics, 125 vol. 125 no.2, pp. 251-267.
- Negnevitsky, M. (2005). *Artificial intelligence: A guide to intelligence systems (2nd ed.)*. New York: Addison Wesley.
- Nguyen, Trung Thanh., Claire P. Willis, Derek J. Paddon, Sinh Hoa Nguyen, and Hung Son Nguyen. (2006). *Learning Sunspot Classification*. Fundamenta Informaticae, Fundamenta Informaticae, vol. 72 (1-3) pp. 295-309. Amsterdam: IOS Press.
- Powell, R. (2000). *How to convert a colour image to grayscale*. Diunduh April 7, 2011, dari BobPowell.net: <http://www.bobpowell.net/grayscale.htm>.
- Setiahadi, B. (2006). *Automatic Determination of the Relative Sunspot Number from White-Light Full Disk Solar Digital Data Using Cluster Method and Turtle Algorithm*. Proceeding of International Conference on Mathematics and Natural Sciences (ICMNS), Institut Teknologi Bandung.
- Setiahadi, B., Sakurai, T., Miyazaki, H., and Hiei, E. (2006). *Research on Magnetohydrodynamic Transport Phenomena in Solar-Terrestrial Space at LAPAN Watukosek 2006*. Prosiding Seminar Antariksa Nasional III, Pusat Pemanfaatan Sains Antariksa, LAPAN, Bandung, pp. 17.
- Setiahadi, B. (2005). *Problems of Equilibria and Instabilities on Solar Coronal Magnetic Fields and Its Evolution Towards Energetic Energy Liberation: Effect to Interplanetary Space*. Prosiding Seminar Nasional Matematika, FMIPA UNDIP, E1, pp. 1.
- Specht, Donald F. (1990). *Probabilistic Neural Network*. Neural Networks, vol. 3, no. 1, pp. 109–118, 1990.
- Wu, Stephen Gang, Bao, Sheng Forrest, Xu, Eric You, Wang, Yu-Xuan, Chang, Yi-Fan, and Xiang, Qiao-Lang. (2007). *A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network*.

# LAMPIRAN

## PERSONALIA PENELITIAN

### 1. Ketua Peneliti:

- a. Nama Lengkap : Gregorius Satia Budhi
- b. Pangkat / Golongan / NIP : Pembina / IV B / 02030
- c. Jabatan Fungsional : Lektor
- d. Jabatan Struktural : Kepala Bidang Studi Informatika
- e. Fakultas/Program Studi : Teknologi Industri/Teknik Informatika
- f. Perguruan Tinggi : Universitas Kristen Petra
- g. Bidang Keahlian : Kecerdasan Buatan, Jaringan Saraf Tiruan, *Data Mining*
- h. Waktu untuk penelitian ini : 3 jam / minggu

### 2. Anggota Peneliti I:

- a. Nama Lengkap : Rudy Adipranata
- b. Pangkat / Golongan / NIP : Pembina / IV B / 99015
- c. Jabatan Fungsional : Lektor Kepala
- d. Jabatan Struktural : -
- e. Fakultas/Program Studi : Teknologi Industri/Teknik Informatika
- f. Perguruan Tinggi : Universitas Kristen Petra
- g. Bidang Keahlian : *Computer Vision*, Pengolahan Citra Digital
- h. Waktu untuk penelitian ini : 3 jam / minggu

### 3. Anggota Peneliti II:

- a. Nama Lengkap : Bambang Setiahari
- b. Pangkat / Golongan / NIP : Pembina / IV A / 19570214.198603.1.002
- c. Jabatan Fungsional : Peneliti Madya
- d. Jabatan Struktural : Kepala SPD LAPAN Watukosek
- e. Lembaga : Lembaga Penerbangan dan Antariksa Nasional
- g. Bidang Keahlian : Simulasi MHD Matahari
- h. Waktu untuk penelitian ini : 3 jam / minggu

**MAHASISWA YANG MEMBANTU PENELITIAN INTERNAL**

<b>No.</b>	<b>Nama/ NRP</b>	<b>Judul Tugas Kuliah / Tugas Akhir</b>	<b>Status Kemajuan</b>
1	Cemara Avriliwan Putra. / 26408001	Perancangan Dan Pembuatan Aplikasi Klasifikasi Citra Observasi Bintik Matahari Menggunakan Metode ID3 / C4.5	Mahasiswa sudah menyelesaikan pengerjaan tugas akhir dan telah lulus dari UKP.

**DRAF PUBLIKASI ILMIAH PADA  
JURNAL NASIONAL TERAKREDITASI  
KURSOR VOL 6 NO.4, JULI 2012  
(Telah diterima, belum terbit)**

## Solar Storm Type Classification Using Probabilistic Neural Network compared with the Self-Organizing Map

**Gregorius S. Budhi<sup>1</sup>, Rudy Adipranata<sup>2</sup>, Bambang Setiahadi<sup>3</sup>, Adrian H. N.<sup>4</sup>**

<sup>1,2,4</sup> Information Engineering Department, Petra Christian University,  
Siwalankerto 121 – 131, Surabaya

<sup>3</sup> Aerospace Observer Station (SPD) LAPAN,  
Watukosek, Gempol, POBox 04, Pasuruan

E-Mail: greg@petra.ac.id, rudya@petra.ac.id, bambangsetiahadi@rocketmail.com

### Abstrak

Salah satu tugas LAPAN adalah melakukan pengamatan dan peramalan gangguan badai matahari. Gangguan ini dapat mempengaruhi medan elektromagnet bumi sehingga mengacaukan peralatan elektronik dan navigasi yang ada di bumi. LAPAN menginginkan adanya aplikasi komputer yang secara otomatis dapat mengklasifikasi tipe badai matahari, yang menjadi bagian dari sistem early warning yang akan dibuat. Klasifikasi terhadap citra digital badai matahari / bintik matahari ini berbasis pada sistem klasifikasi "Modified - Zurich Sunspot Classification". Metode klasifikasi yang kami gunakan disini adalah Jaringan Saraf Tiruan Probabilistik. Hasil dari pengujian cukup menjanjikan karena memiliki akurasi sebesar 94% untuk data testing. Akurasi ini lebih baik dari akurasi aplikasi serupa yang dibangun dengan kombinasi metode *Self-Organizing Map* dan *K-Nearest Neighbor*.

*Kata kunci:* Klasifikasi Tipe Badai Matahari, Sistem Klasifikasi *Modified - Zurich Sunspot Classification*, Jaringan Syaraf Tiruan Probabilistik.

### Abstract

*One of the task of the LAPAN is making observation and forecasting of solar storms disturbance. This disturbances can affect the earth's electromagnetic field that disrupt the electronic and navigational equipment on earth. LAPAN wanted a computer application that can automatically classify the type of solar storms, which became part of early warning systems to be created. The classification of the digital images of solar storm / sunspot is based on "Modified - Zurich Sunspot Classification System". Classification method that we use here is the Probabilistic Neural Networks. The result of testing is promising because it has an accuracy of 94% for testing data. The accuracy is better than the accuracy of similar applications we've built with a combination of methods Self-Organizing Map and K-Nearest Neighbor.*

*Key words:* Solar Storm Type Classification, Modified - Zurich Sunspot Classification, Probabilistic Neural Network.

## INTRODUCTION

Disruption from the sun that recently known as solar storm is strongly concerned. The disruption can affect the Earth's electromagnetic field, in addition to others dangers that can be happening. Today, human life is highly dependent on modern technologies especially in the field of electrical, electronics, computer and navigate the sea and air, that are susceptible to interference from electromagnetic field changes. Therefore, the emergence of 'interference' from solar storms should be anticipated quickly and precisely.

Actually the solar storm term is a great explosion (Flare) and Coronal Mass Ejection / CME on the surface of the sun. Before the famous term solar storm, this two type of disruptions is more commonly called a group of sunspots, because that's the kind seen in the image of the sun taken by the Michelson Doppler Image instruments (MDI) Continuum / SOHO (Solar and Heliospheric Observatory). Sunspots evolved from a tiny spot with low activity evolved into a very complex configuration with the possibility of having high activity, which issued a mass explosion and hurling corona. To determine the level of complexity of sunspots and sunspot activity Modified - Zurich classification is used in order to classify the types of sunspots.

This study is part of a larger study entitled "Automated Sunspot Group Classification for Analyzing Space Weather Conditions" we've done in collaboration with researchers from the National Aeronautics and Space Agency (LAPAN). Classification of sunspots groups is based on Modified - Zurich Sunspot Classification system. In this paper we publish some of the results of our study that is the use of probabilistic neural network to the classification of sunspots groups / solar storms type, which looks at the digital image of the sun.

Before the classification process, it is necessary proceeding digital image enhancement of the sun images. It should also detect the location of sunspots and sunspot groupings. All of our process has been published previously [1, 2].

For the testing, we also compared the results of this sub-system testing with other classification sub-systems that we have made

previously by using a combination of artificial neural network method Self-Organizing Map and data mining methods K-Nearest Neighbor [3]. This comparison purposes so that we can obtain the most appropriate classification method to be used in the classification system of solar storms that will be created.

## Sunspot Classification

Sunspots are the intersection of the solar magnetic flux tube with the photosphere. Sunspots appear dark because the magnetic field has the effect of sunspot cooling so that the temperature is lower than the surrounding area. The magnetic field of sunspots proved to be a source of disturbance energy from the sun, such as explosives, or hurling coronal mass [4]. The latter phenomenon known as solar storms can disrupt space environment around the Earth and technology facilities and life on the Earth's surface [5, 6, 7].

Because of the sunspot's magnetic field is the intersection of the tube with the photosphere, the appearance of sunspots begins with a small black spot. Over time, if the tube magnetic field is constantly out of the sun due to the buoyant force, it would appear two spots with opposite magnetic polarity (bipolar). Sunspots evolved into more complex configurations, in example, the number of spots and area increases. The more complex a configuration of sunspots are, the greater the possibility of instability magnetic field that triggers a flare or CME events [8, 9].

Stages of the evolution of sunspots is expressed in the "Modified - Zurich Sunspot Classification", namely class A, B, C, D, E, F (the level of complexity increases) and then gradually decreased until it became a class H. Image of the sunspot evolution classification found in Figure 1 [10].



Figure 1. Modified - Zurich Sunspot Classification [10]

**Artificial Neural Networks**

Artificial neural network is a processing model that is based on the workings of the human brain [11]. Just like the human brain, neural networks composed of cells interconnected neurons. Each cell is a neuron receives input signals are processed to produce an output signal. Neural networks can learn to like the human brain by giving weight for each input on neuron cells. By using weights, the neural network can learn a given input.

**Probabilistic Neural Network**

Probabilistic Neural Network (PNN) is a method of artificial neural networks using the principles of statistical theory Bayesian Classification to replace heuristic principles used in the Back-propagation algorithm [12]. That is

why PNN is usually used to perform pattern classification [13].

Architecture of the PNN consists of four layers, namely input layer, pattern layer, summation layer and decision layer / output layer as can be seen in Figure 2 [13].

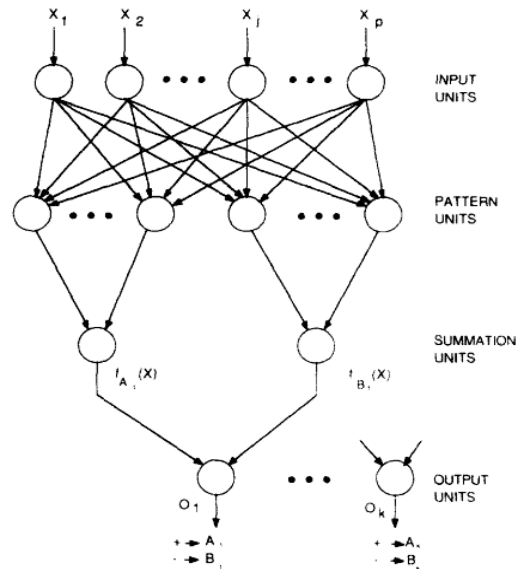


Figure 2. Detail Architecture of PNN [13]

The input layer does not perform any calculations, just transfer the data input to each neuron in the pattern layer. Each neuron in the pattern layer will perform probability calculations (distance) between the inputs to the data stored in the pattern layer neurons. Furthermore, the summation layer will receive input from neurons of pattern layer and sum it. And last, the output layer will produce classification results based on the results of the summation neuron that has the greatest value. In general, probabilistic neural network architecture is shown in Figure 3.

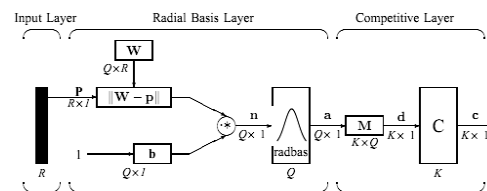


Figure 3. PNN Architecture [14]

Here is the training algorithm of probabilistic neural network [14]:

1. Initializing initial weight (radial basis layer weight)  $W$  obtained from the transpose of the matrix  $R \times Q$  where  $R$  is the dimension of input and  $Q$  is the number of training data.

2. Initializing constant spread ( $s$ ) of PNN.

3. Initializing the bias weight ( $b$ ).

$$b = \frac{\sqrt{-\ln 0.5}}{s} \quad (1)$$

4. Initializing final weight (competitive layer weight)  $M$  which is a matrix of size  $K \times Q$  where  $K$  is the number of the classification results and  $Q$  is the number of training data.  $i$ -th row of the matrix  $M$  represents the  $i$ -th training data and the  $j$ -th column matrix values will be worth 1 if the training data into the group, otherwise it will be 0.

5. Calculate the distance between the vectors of input data  $P$  with vector of each row in the initial weight  $W$  (Euclidean Distance between vector  $P$  with vector  $W_i$ ) resulting distance matrix  $\|W - P\|$  that have  $Q \times 1$  dimension.

6. Calculating the activation value of the distance between the initial weight of the input data (vector  $a$ ), using  $\text{radbas}()$  function.

$$\text{radbas}(n) = e^{-n^2} \quad (2)$$

$$n = \|W - P\| * b \quad (3)$$

7. Multiplying vector  $a$  by matrix  $M$  to produce output vector  $d$ .

8. Looking for an output of PNN with competitive function  $C$  where the competitive function will locate the largest value in the vector  $d$ . Index of greatest value is to indicate the results of the classification of data input  $P$ .

9. Save the initial weight, spreads, and the final weight.

**Min - Max Normalization**

Min-max normalization is a method of normalization that transforms the data in a linear fashion into a new range [15]. Formula of this method is shown in Equation 4.

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A \quad (4)$$

**DESAIN OF SOLAR STORM CLASSIFICATION SUB-SYSTEM**

**Probabilistic Neural Network Architecture used**

Probabilistic neural network architecture used is shown in Figure 4.

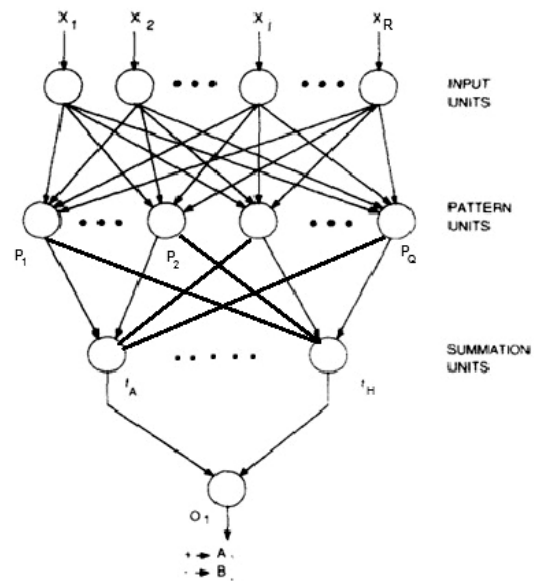


Figure 4. PNN architecture used

For the number of input units will be obtained from the large dimension of the data input ( $R$ ), the processed data from the database input diameter, area, perimeter, number of sunspots each group, and the average black pixel for each region of the input image. After that, each input unit is connected to all the pattern units amounting to a weight  $Q$ , where  $Q$  is the amount of training data. So the initial weight matrix size  $Q \times R$ .

Each pattern unit will generate a value of the distance between the weight and the input, the result is a matrix dimension  $Q \times 1$  which will then be used for further calculations using  $\text{radbas}()$  function.

Each pattern unit will be connected with all summation units through the final weight, which is a matrix of size  $K \times Q$ , where  $K$  is the number of the classification results, which is a class  $A, B,$

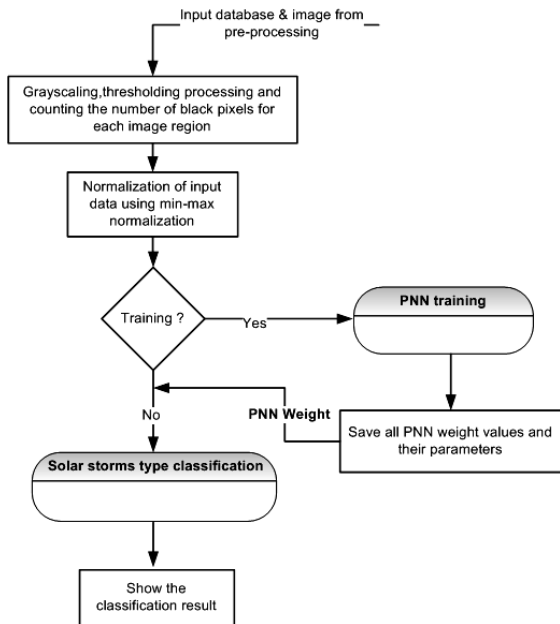


C, D, E, F, and H. The magnitude of Q is the amount of training data. I-th row of the matrix M represents the i-th training data and the j th column matrix values will be worth 1 if the training data into the group, otherwise it will be 0.

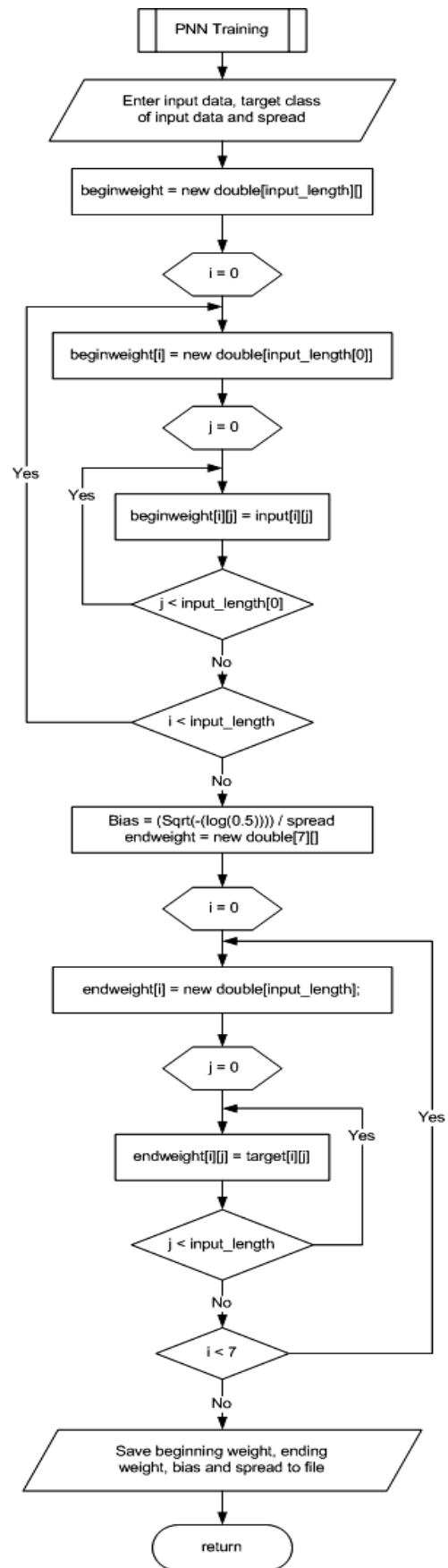
Then each summation unit will be connected to an output unit whose function is to seek the greatest value of summation unit and take the index of the summation unit of as a result of the classification.

**Sub-System Design**

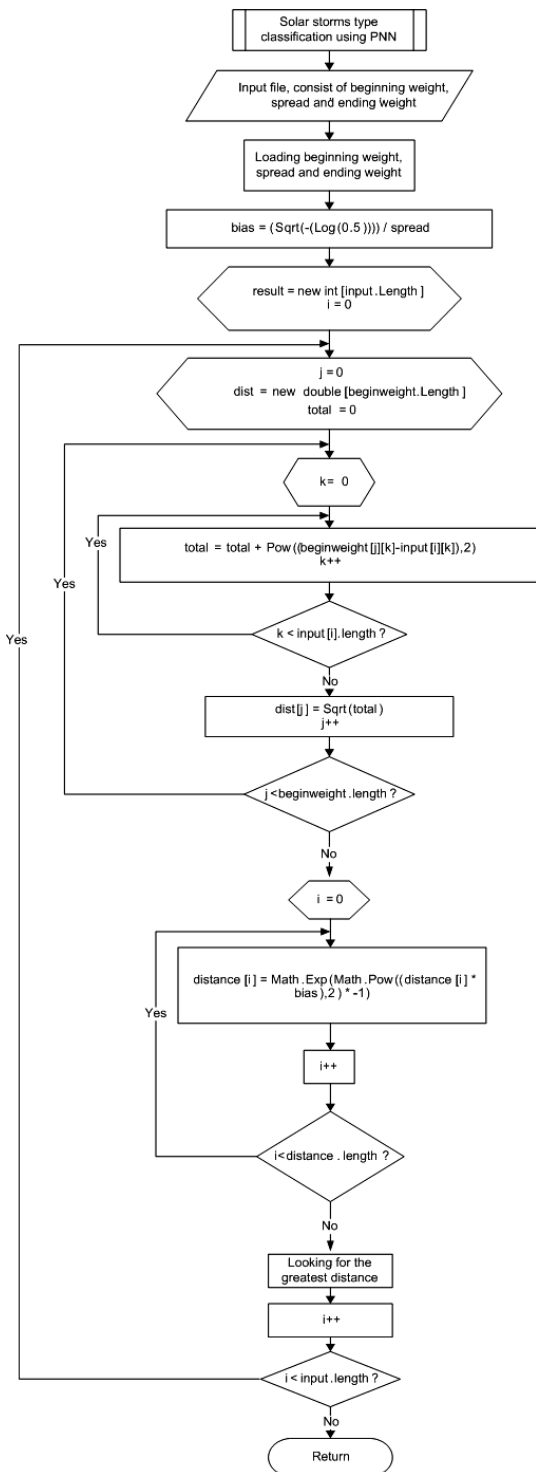
In Figure 5, 6 and 7 can be seen a block diagram and flowchart of the design sub-systems that are built for the solar storms type classification using probabilistic neural network method.



**Figure 5.** Block diagram of solar storm type classification sub-system using PNN



**Figure 6.** Flowchart of the PNN Training process



**Figure 7.** Flowchart of the solar storm type classification process using PNN

Input from the sub-system is derived from the sub-system that we created earlier and have been published [1, 2]. Any data that goes into the sub-type classification system solar storm consists of 4 kinds of values and a digital image. It consists of four different values: area, perimeter,

diameter, and the number of spots available to a group of sunspots. Meanwhile, the input digital image is a digital image of the solar storm as shown in Figure 8. Furthermore, the solar storm digital image is converted to data that can be received by the input neurons of PNN. To change, the image is processed into black and white image using the grayscale processing and thresholding method. After becoming a black-and-white image, it is proceeded to divide of regions in the image, where the number of regions specified by user. Examples of the process can be seen in Figure 8. Furthermore, the number of black pixels in each region is counted.



**Figure 8.** Examples of solar storm digital image and the process of region dividing

Next is the process of normalization of all the input data, using the Min - Max Normalization formula. These data are the area, perimeter, diameter, number of spots and values derived from counting the black pixels of each region. The values that have been normalized are used for input neurons PNN.

Once all the input data has been prepared, the user can select the process to be executed, namely: ANN training or classification. For training the neural network, as in general the process of training the PNN method, all sample data will be processed to update all the weight until convergent. After converging, the weight is stored and can then be used in the classification process. The process of training for a sample data is shown in Figure 6.

For the classification process, input required is a set of PNN weights that have been convergent, PNN architecture used during training, and image the solar storms that will be classified. With passing the solar storms image to be classified in the sub-system of PNN (with a weight that has converged), we will get the value of the classification based on Modified - Zurich Sunspot Classification, which is a class A, B, C, D, E, F or H. The process of classification for a data can be seen in Figure 7.

From the results of this classification can be determined quickly whether a solar storm (sunspot groups) is dangerous or not. Class types to be concerned are D, E and F.

## RESULT AND DISCUSSION

There is some testing that has been done on the sub-systems, namely:

a. Testing the speed of the training and qualification process. It would be tested with a maximum total data that we have (214 kinds of sample data). Test results state that each process has a speed less than 0.1 milliseconds. It can be concluded that in terms of processing speed, the sub-system is very good.

b. Tests with constant variation spread of PNN. The test is performed with the number of training data = 150 data and digital images of solar storms that is divided into 5 x 5 regions. The test results can be seen in Table 1. So, we could conclude that the higher spread value will be, the decrease of classification accuracy value would be produced.

**Table 1. The constant variation spread testing**

Spread	Classification accuracy of the training data	Classification accuracy of the testing data
0.2	97%	94%
0.5	85%	89%
0.9	72%	86%

**Note:** The testing data are samples data that did not participate in the training process.

c. Tests on the amount of training data variation. Setting the test is spread = 0.2 and a digital image of solar storms is divided into 5 x 5 regions. The test results can be seen in Table 2. From the results it can be concluded that if we tested with the data had been trained, the accuracy of sub-systems did not have a clear pattern. However, when tested with data testing, it can be seen that the more training data is used, the higher the accuracy of the sub-systems.

**Table 2. Testing variations in the amount of training data**

Training data number	Classification accuracy of the training data	Classification accuracy of the data testing
50	98%	76%
100	96%	86%
150	97%	94%

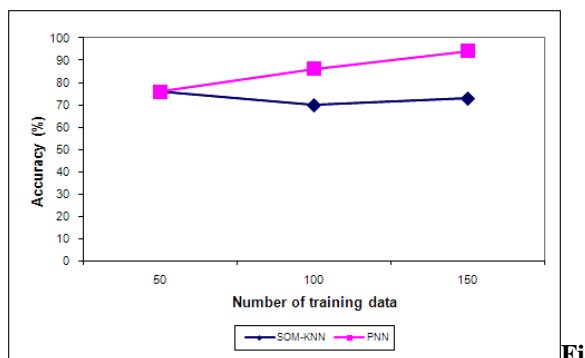
d. The next test is to vary the number of regions of the solar storm digital image that are processed. The test is performed with the number of training data = 150 data and the constant spread = 0.2. The test results are shown in Table 3. The test results stated that for the classification of the training data, the best regions number is 7x7. However, if the classification performed on the data testing, the best regions number is 5x5.

**Table 3. Regions number variations testing**

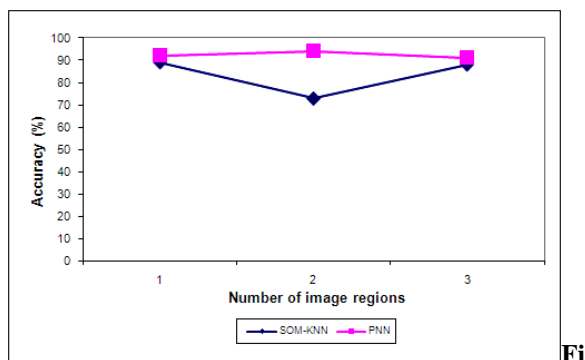
The number of region	Classification accuracy of the training data	Classification accuracy of the data testing
3 x 3	88%	92%
5 x 5	97%	94%
7 x 7	99%	91%

Further, we will compare this sub system with similar sub-system that we have built previously used a combination of methods Self-Organizing Neural Network and K-Nearest Neighbor [3]. Testing variations spread constants cannot be compared because the constant spread only on PNN and not used in the SOM. In comparison of the speed of the training process, the PNN method was much better. Speed the process of training the SOM-KNN is still above 1 second for as many as 50 pieces of training data, while the PNN has speed training under 0.1 milliseconds for maximum training data (214 training data). Furthermore, the results of comparison of the classification accuracy of PNN and SOM-KNN can be seen in Figure 9 and 10. These accuracy comparisons are performed only

on the data testing (sample data that not used during training).



**Figure 9.** Comparison of classification accuracy for variation in the amount of training data with the distribution of digital image region = 5 x 5



**Figure 10.** Comparison of classification accuracy for variations of digital image regions with the number of training data = 150 data

From the results of the comparison between the solar storms type classification sub-systems built with PNN and SOM-KNN, it appears that the performance of the PNN sub - system was better.

## CONCLUSION

Solar storm type classification sub-system using Probabilistic Neural Network has been created properly. It can be seen from the rapid process of training and classification, as well as the high level of classification accuracy. Besides that the sub-system has also proven to have a better performance compared to the similar sub-system that has been built previously using a combination of Self-Organizing Map and K-Nearest Neighbor methods.

However, we have not decided yet whether the sub-system will be used or not. The test results must be compared with the results of similar system test that we built using Feed-Forward Back-propagation method. From the

comparison result, analysis and further testing, we will determine which sub-system to use.

## ACKNOWLEDGMENT

The research was funded by Research Competitive Grant DIPA-PT Coordination of Private Higher Education region VII, East Java (0082/SP2H/PP/K7/KL/IV/2011) fiscal year 2011 and Research Center, Petra Christian University, Surabaya (12/Sugas-Pen/LPPM-UKP/2012), fiscal year 2012, entitled "Automated Sunspot Group Classification for Analyzing Space Weather Conditions".

## DEDICATION

We dedicate this paper to a co-researcher from the National Aeronautics and Space Agency (LAPAN), namely: Dr. Bachtiar Anwar M.Sc. who had died of illness at the end of 2011.

## REFERENCES

- [1] Adipranata R, Budhi GS, Setiahadhi B, dan Anwar B, Segmentation of Sunspot Group using Watershed Method. Proceeding of the National Conference of System and Informatics, Bali: 332-337, 2010.
- [2] Budhi, GS, Adipranata R, Sugiarto M, Anwar B, dan Setiahadhi B, Sunspot Grouping of Solar Digital Image using DBSCAN Clustering Method. Proceeding of the National Seminar of Information Technology Application, Yogyakarta: F81 – F85, 2011.
- [3] Budhi GS, Adipranata R, Setiahadhi B, Anwar B, Hartanto A, Tjondrowiguno AN, Combination of Self-Organizing Map Neural Network and K-Nearest Neighbor for Automatic Classification of Sunspot Group Image. Proceeding of the National Conference of System and Informatics, Bali: 369 - 374, 2011.
- [4] Setiahadhi B, Sakurai T, Miyazaki H and Hiei E, Research on Magnetohydrodynamic Transport Phenomena in Solar-Terrestrial Space at LAPAN Watukosek 2006, Proceeding of the National Space Seminar III Space Science Utilization Center, LAPAN Bandung: 17 – 34, 2006.
- [5] Boteler DH, Space Weather Effects on Power Systems. Space Weather. Geophysical Monograph, vol. 125: 347 - 355, 2001.
- [6] Bothmer V and Daglis IA, Space Weather, Physics and Effects, Chicester UK: Springer - Praxis Publishing, 2007.

- [7] Lanzerotti LJ, Space Weather Effects on Technologies. Space Weather, Geophysical Monograph, vol. 125: 11 - 22, 2001.
- [8] Anwar B, Acton LW, Makita M, Hudson HS, McClymont AN and Tsuneta S, Rapid Sunspot Motion During A Major Solar Flare. Solar Physics, vol. 147: 287 - 303, 1993.
- [9] Setiahadi B, Problems of Equilibria and Instabilities on Solar Coronal Magnetic Fields and Its Evolution Towards Energetic Energy Liberation: Effect to Interplanetary Space, Proceedings of the National Seminar on Mathematics, FMIPA, University of Diponegoro, E1: 1 - 7. 2005.
- [10] McIntosh PS, The Classification of Sunspot Groups, Solar Physics, vol. 125 no.2: 251 - 267, 1990.
- [11] Negnevitsky M, Artificial intelligence: A guide to intelligence systems 2<sup>nd</sup> Edition, New York: Addison Wesley, 2005.
- [12] Specht DF, Probabilistic Neural Network, Neural Networks, vol. 3 no. 1: 109 – 118, 1990.
- [13] Fausett L, Fundamentals of neural networks: Architectures, algorithms, and applications, New Jersey: Prentice - Hall International, 1994
- [14] Wu SG, Bao FS, Xu EY, Wang YX, Chang YF and Xiang QL, A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network, IEEE 7th International Symposium on Signal Processing and Information Technology, Cairo :12 – 17, 2007.
- [15] Han J, Kamber M and Pei J, Data Mining: Concept and Techniques 3<sup>rd</sup> Edition, Waltham USA: Morgan Kaufmann Publishers, 2012.

**DRAF PUBLIKASI ILMIAH PADA  
INTERNATIONAL JOURNAL OF  
MULTIMEDIA AND UBIQUITOUS  
ENGINEERING**

(Telah dikirim, akan di-review)

## Automatic Classification of Sunspot Groups for Space Weather Analysis

Rudy Adipranata <sup>1)</sup>, Gregorius Satia Budhi <sup>2)</sup>, Bambang Setiahadhi <sup>3)</sup>  
*Informatics Dept., Petra Christian University, Surabaya, Indonesia <sup>1,2)</sup>,  
 Indonesian National Institute of Aeronautics and Space (LAPAN), Watukosek, Indonesia <sup>3)</sup>  
 rudya@peter.petra.ac.id <sup>1)</sup>, greg@peter.petra.ac.id <sup>2)</sup>, bambangsetiahadhi@rocketmail.com <sup>3)</sup>*

### Abstract

The sun is unlimited energy source for life on the earth. But, besides as energy source, the sun also gives disruptions to the universe around earth and also to the life on earth. Source of the disruptions from the sun are flares and Coronal Mass Ejection/CME. Both of those disruptions in general come from group of sunspots. With growing of dependency of human life with modern technology, either facility on the surface of the earth or in universe around the earth, the disruptions from the sun should be anticipated. To know the complexity level of sunspot groups and their activity, Modified-Zurich sunspot classification is used. Image of sunspots can be taken using the Michelson Doppler Imager instrument (MDI) Continuum / SOHO (Solar and Helispheric Observatory).

This research conducted on the automatic classification of sunspot group that can be used to analyze the space weather conditions and provide information to the public. There are two stages to classify sunspot groups, feature extraction and pattern recognition. For feature extraction, we used digital image processing to get features of sunspot group, and for pattern recognition we used artificial neural network. We compared 3 methods of artificial neural networks to get the best result of classification, namely backpropagation, probabilistic and combination between self-organizing map and k-nearest neighbor. Among three of them, probabilistic neural network gave the best classification result.

**Keywords:** sunspot groups classification, artificial neural network, pattern recognition

## 1. Introduction

Life on earth is dependent on the sun as a source of unlimited energy. But, besides as energy source, the sun also gives disruptions to the universe around earth and also to the life on earth. There are two kinds of disruptions from the sun, flares and Coronal Mass Ejection/CME. Both of those disruptions in general come from group of sunspots. The sunspots phenomenon is formed as a result of the magnetic flux tube emerged from the sun to the photosphere and corona [1]. The intersection of the magnetic flux tube with the photosphere forms a sunspot, which appear black because the magnetic flux has cooling effect so that the sunspot temperature is lower than the surrounding photosphere [2].

Sunspots evolved from a tiny spot with low activity into a very complex configuration with the possibility of having high activity, which issued an explosion and hurling corona mass. To determine the complexity level of sunspot groups and their activity, Modified-Zurich sunspot classification is used [3]. Modified-Zurich sunspot classification has class A, B, C, D, E, F (growing level of complexity) and then gradually declined until became a class H, can be seen at Figure 1 [3]. This classification is very important to analyze the space weather.

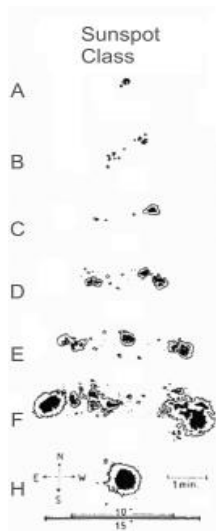


Figure 1. Modified-Zurich sunspot classification

## 2. Feature Extraction

Feature extraction is a stage to obtain features of sunspot groups in an image. Image of sunspots can be taken using the Michelson Doppler Imager instrument (MDI) Continuum / SOHO (Solar and Helispheric Observatory). These features are area of sunspot group, perimeter of sunspot group, diameter of sunspot group and the number of sunspots in group. To extract the features, the image of sunspots should be segmented. Watershed method will be used for segmentation process. To get the features of sunspot group, after being segmented, sunspots will be grouped using DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) clustering algorithm [4].

### 2.1. Watershed Segmentation

Watershed is one of the methods used to perform digital image segmentation. The concept of watershed is to assume an image has three-dimensional shape that is the position of  $x$  and  $y$  with its color pixel level. The  $x$  and  $y$  is a position of color pixel level, which in this case, the gray level is the height of the assumption that the value closer to white has a higher altitude. Assuming the form of topography, there are three different points: (a) a point which is the regional minimum, (b) a point that if there is a drop of water, then the water will fall down to a certain minimum position, and (c) a point that if there is a drop of water, the water has the possibility to fall into one minimum position (not always fall to a minimum point, but it may fall into a certain minimum point or another). To a certain minimum regional, a set of points that satisfy the condition (b) is called a catchment basin, while the set of points that satisfy the condition (c) is called the watershed line [5].

From the explanation above, this watershed segmentation method has the objective to perform a search watershed line. The basic idea for the workings of this segmentation is assumed there is a hole made on the regional minimum, and then the whole topography flooded by water from the hole with a constant velocity. When the water rises from the two catchment basins was about to join, then built a dam to prevent the merger. The flow of water will reach the desired level and stops flowing when only the upper part of the dam is visible. The edge of the dam is called the watershed line. And the watershed line is the result of segmentation, assuming that line the edge of the watershed is about to segmented objects. For more details on the drawing can be seen in Figure 2 [5].



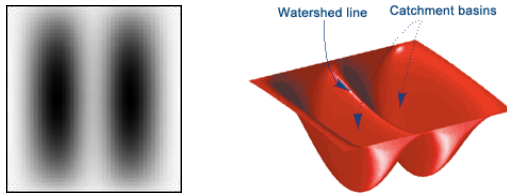


Figure 2. Watershed concept

Figure 2(a) displays a two-dimensional image of the watershed concept in which two dark colored part is the catchment basin and an area between two catchment basin is an area where the watershed line will be, whereas in Figure 2(b) displays three-dimensional images of the watershed concept.

### 3. Artificial Neural Network

Artificial neural network is a thinking model based on the workings of the human brain [6]. Just like the human brain, neural networks composed of interconnected neurons. Each neuron receives input signals, and processed to produce an output signal. Neural networks can learn to like the human brain by giving weight for each input on neuron. By using weights, the neural network can learn a given input.

#### 3.1. Backpropagation Neural Network

Backpropagation is a method of neural networks with multilayer feedforward neural network structure [6]. Neurons are divided into several layers, namely input layer, output layer and several hidden layers. Neurons in the hidden layer store the patterns of input in the form of weights of the connections between neurons. The structure of backpropagation neural network can be seen at Figure 3 [6].

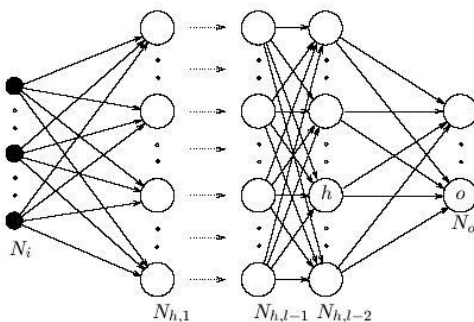


Figure 3. Backpropagation structure

The steps in the backpropagation algorithm are as follows [6]:

#### 1. Initialization

Determine all the weights and thresholds of the neural network with random numbers that are in the range (1).

$$\left( -\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right) \quad (1)$$

$F_i$  is the total input of neuron  $i$  in the neural network

#### 2. Activation

Calculate output of neurons in the hidden layer and output layer using equation (2).

$$y(p) = \text{sigmoid} \left[ \sum_{i=1}^n x_i(p) \times w_i(p) - \theta \right] \quad (2)$$

$n$  is the number of input neuron,  $x$  is the input signal,  $w$  is the weight,  $\theta$  is a threshold,  $p$  indicating  $p^{\text{th}}$  iteration, and *sigmoid* is the sigmoid activation function that follows the equation (3).

$$Y^{\text{sigmoid}} = \frac{1}{1 + e^{-X}} \quad (3)$$

### 3. Weight Training

Calculate the error signal to the neurons in the output layer using the equation (4).

$$e(p) = y_d(p) - y(p) \quad (4)$$

$y_d$  is the desired output results.

Calculate the error gradient for the neurons in the output layer using equation (5).

$$\delta(p) = y(p) \times [1 - y(p)] \times e(p) \quad (5)$$

Calculate the weight corrections for neurons in the output layer using equation (6).

$$\Delta w(p) = \alpha \times y(p) \times \delta(p) \quad (6)$$

$\alpha$  is the learning rate

Change weight to the neurons in the output layer using equation (7).

$$w(p+1) = w(p) + \Delta w(p) \quad (7)$$

Calculate the error gradient for the neurons in the hidden layer using equation (8).

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^l \delta_k(p) \times w_{jk}(p) \quad (8)$$

$l$  is the number of neurons in the previous layer.

Calculate the weight correction for the neurons in the hidden layer using equation (9).

$$\Delta w(p) = \alpha \times x(p) \times \delta(p) \quad (9)$$

$\alpha$  is the learning rate

Change weight to neurons in the hidden layer using the following equation (10).

$$w(p+1) = w(p) + \Delta w(p) \quad (10)$$

### 4. Iteration

Repeat the above process starting from step 2 to obtain the value of sum of squared errors below 0.001.

## 3.2. Probabilistic Neural Network

Probabilistic Neural Network (PNN) is a method of artificial neural networks using the principles of Bayesian statistical theory is to replace classification heuristic principle used by backpropagation algorithm [7]. That's why PNN is used to perform pattern classification [8].

Architecture of the PNN consists of four layers, namely input layer, pattern layer, summation layer and decision layer / output layer as shown in Figure 4 [7]. The input layer does not perform any calculations, just transfer the input data to each neuron in the pattern layer. Each neuron in the pattern layer will calculate probability (distance) between input data to the data stored in the neurons pattern layer. Furthermore, the summation layer will receive input from each neurons pattern layer and calculate the summation of them, so it will get the probability an input  $x$  is the member of a group  $t$ . And the last, output layer will produce classification results based on the results of the summation neuron that has the greatest value.

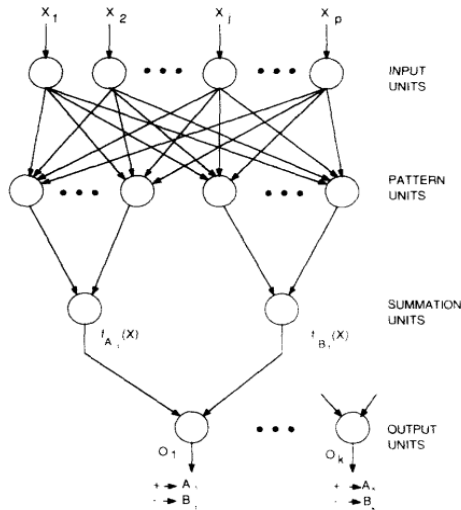


Figure 4. Probabilistic neural network architecture

### 3.3. Combination of Self-Organizing Map and K-Nearest Neighbor Neural Network

Self-organizing map is a method of artificial neural network that implement competitive learning [8]. In general, self-organizing map is used to perform the classification or grouping (clustering). Self-organizing map consists of  $m$  cluster units and  $n$  input units. Cluster units and input units were connected to the weight vector.

In general, the principle of self-organizing map is by selecting one of the  $m$  cluster unit with weight vector most similar to the input pattern using minimum squared euclidean distance. Weight vector of the cluster unit and several other adjacent clusters unit are updated. This method can be used to group multiple input vectors into  $m$  groups. The structure of self-organizing maps can be seen in Figure 5 [8].

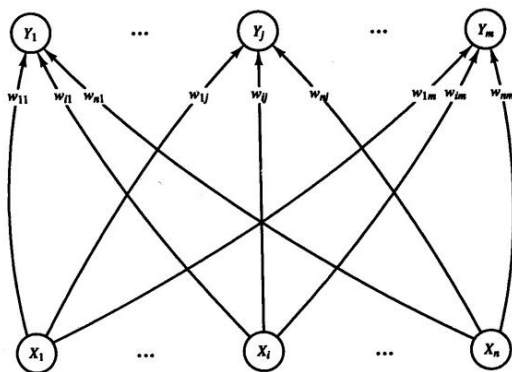


Figure 5. Self-organizing map structure

K-Nearest Neighbor (KNN) is a method to classify the object based on the data that were located closest to the object. This method is the most common method used for the estimation and prediction [9]. The steps in the KNN algorithm are as follows [10]:

**Input:** a set of pre-classified training instances, a query instances  $q$ , and a parameter  $k$ , defining the number of nearest neighbors to use

**Output:** a label indicating the class of the query instance  $q$

**Step 1:** Find the  $k$  closest training instances to  $q$  according to a distance measure

**Step 2:** Select the class of  $q$  to be the class held by the majority of the  $k$  nearest training instances

To calculate the distance measure in Step 1, euclidean distance equation (11) is used. To count majority vote in Step 2, weighted voting equation (12) is used [10].

$$d_{Euclidean}(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \quad (11)$$

$$vote = \frac{1}{d(x, y)^2} \quad (12)$$

#### 4. System Design and Experimental Result

Overall system design for automatic classification of sunspot groups can be seen in Figure 6, and example of digital image of sunspots can be seen in Figure 7.

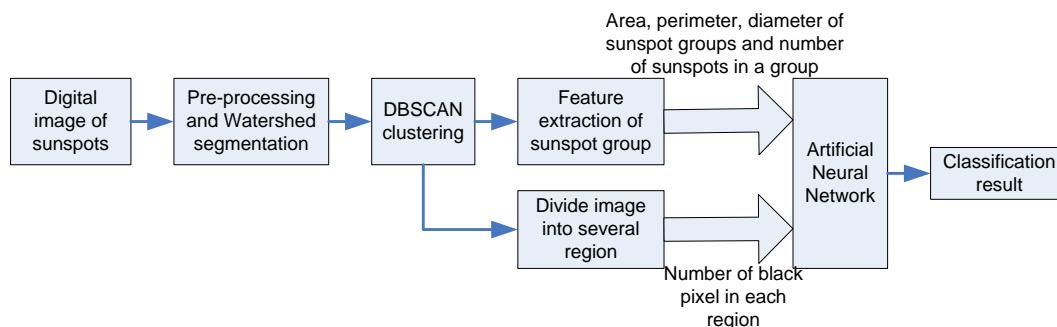


Figure 6. System design



Figure 7. Digital image of sunspots

The watershed segmentation processing will produce over segmentation, so that the desired object cannot be segmented properly, can be seen in Figure 8 [11]. To overcome the over segmentation, we should apply pre-processing before watershed segmentation.

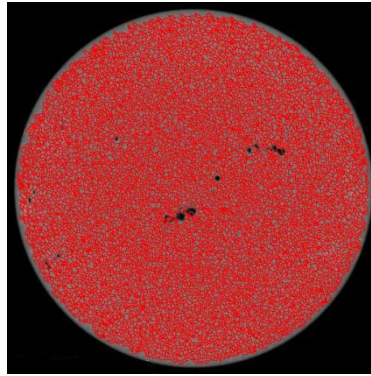


Figure 8. Result of watershed segmentation

The pre-processing implemented are opening, closing, erosion, dilation, canny and sobel [5]. We conducted research to find the best combination of those pre-processing to obtain the best segmentation results. From experiment, the combination of opening, erosion, sobel, canny, dilation and following by closing is the best combination to obtained the best segmentation result, can be seen in Figure 9 [11].

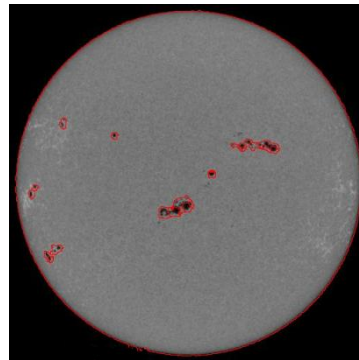


Figure 9. Result of watershed segmentation using pre-processing

After being segmented, the sunspot should be grouped. When viewed manually, sunspots form groups with irregular curve. This irregular curve is difficult to be detected automatically using common clustering methods such as k-means or k-medoids. Because using those methods, the number of clusters, cluster central point should be determined, and will have oval or elliptical cluster result, while the curve of sunspot groups has irregular shapes. Therefore, to classify sunspots, we used DBSCAN (Density Based Spatial Clustering of Applications with Noise) clustering method [4].

Using DBSCAN, number of clusters generated is not limited and depends on the distance of each sunspot against other spots. DBSCAN clustering method is moving from one point to another and produce irregular clusters curved, according to their positions. this corresponds to the shape of the sunspot group. The results of clustering/grouping of sunspots with DBSCAN can be seen in Figure 10 [12]. After being clustered using DBSCAN method, next step is to search and mark the edges of each sunspot group, so that later, the image can be cropped easily for each group to extract the features. The features being extract for each group are area of sunspot group, perimeter of sunspot group, diameter of sunspot group and the number of sunspots in group. Edge marking results of each group of sunspots can be seen in Figure 11 [12].

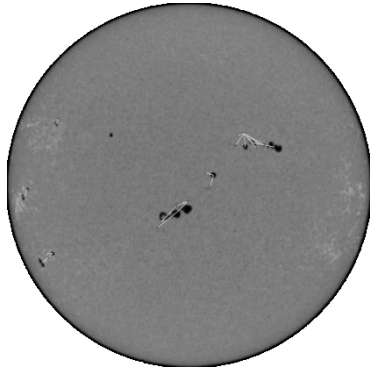


Figure 10. DBSCAN clustering result

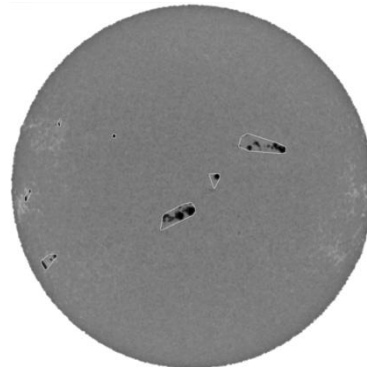


Figure 11. Edge marking of sunspot groups

Furthermore, the digital image of sunspot groups was converted into data that can be received by the input of the artificial neural network. To convert, the original color image is processed into black and white image by using grayscale and thresholding method. After being a black-and-white image, we divide the image into several regions. Examples of the process can be seen in Figure 12 [13]. After dividing into several regions, the number of black pixels in each region is counted.



Figure 12. Dividing sunspot groups image into several regions

Next is the process of normalization of the four group features and number of black pixels of each region, using the min-max normalization [14]. Min-max normalization is a method of normalization that transforms the data in linear fashion into a new range. Formula for min-max normalization can be seen in equation (13). The normalized values are used as neuron input of artificial neural network.

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A \quad (13)$$

We used and compared three artificial neural network methods to get the best result of classification, namely backpropagation, probabilistic and combination between self-organizing map and k-nearest neighbor.

In backpropagation, all input data is processed and update all the weight value to achieve a convergent condition. The mapping of sunspot class to backpropagation output layer can be seen in Table 1.

Table 1. Mapping of sunspot class to backpropagation output layer

Output Layer			Sunspot Class
Neuron 1	Neuron 2	Neuron 3	
0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E
1	0	1	F
1	1	0	Unknown
1	1	1	H

In this backpropagation, there are three types of experiments have been done. For the first one, we tested on variation in the number of hidden layer. We have 214 data, where 164 data was used for training and 50 data was used for testing. We used learning rate parameter at 0.2, the number of region = 5x5. Testing results can be seen in Table 2. From the results it can be seen that 5 hidden layers result on the highest accuracy.

Table 2. Backpropagation testing results based on variations of the number of hidden layers

Hidden Layer	Accuracy	% accuracy
3 layers	32/50	64%
5 layers	33/50	66%
7 layers	15/50	30%

The second is testing on variation of learning rate. We used same parameters as first one, but use 5 hidden layers. Testing result can be seen in Table 3. From the results it can be seen that learning rate = 0.2 produce the highest accuracy.

Table 3. Backpropagation testing results based on variation of learning rate

Learning Rate	Accuracy	% accuracy
0,2	33/50	66%
0,5	10/50	20%
0,8	15/50	30%

The last test is testing on variation of the number of image region. We used same parameter as testing before and 5 hidden layers. Testing result can be seen in Table 4. From the results it can be seen that 5x5 regions results on the highest accuracy.

Table 4. Backpropagation testing results based on variation of the number of image region

The number of regions	Accuracy	% accuracy
3 x 3	22/50	44%
5 x 5	33/50	66%
7 x 7	26/50	52%

In probabilistic neural network, the number of input units will be obtained from dimension size of the input data ( $R$ ), consist of diameter, area, perimeter of sunspot group, number of sunspots for each group, and the number of black pixel for each image region. After that, each input unit is connected to  $Q$  pattern units, where  $Q$  is the amount of training data. So the size of initial weight matrix is  $Q \times R$ . Each pattern unit will generate a distance value between the input and weight. Also each pattern unit will be connected with

all summation units through the last weight, which is  $K \times Q$  matrix, where  $K$  is the number of the classification results, class A, B, C, D, E, F, or H. Then each summation unit will be connected to an output unit whose function is to seek the largest value of summation unit and take the index of the summation unit as a result of classification.

We have done two types of experiments using 214 data which is 164 data for training and 50 data for testing. The first one is done by varying the spread of probabilistic neural network, and the input image divided by 5x5 regions. Testing result can be seen in Table 5, the higher the spread, accuracy of classification will decrease.

Table 5. Probabilistic neural network testing results based on spread variation

Spread	Accuracy	% accuracy
0.2	47/50	94%
0.5	45/50	90%
0.9	43/50	86%

The second experiment was done on variation of the number of image region. We used 0.2 for spread value. Testing result can be seen in Table 6.

Table 6. Probabilistic neural network testing results based on variation of the number of image region

Number of regions	Accuracy	% accuracy
3 x 3	46/50	92%
5 x 5	47/50	94%
7 x 7	45/50	90%

In self-organizing maps, all sample data will be trained until all the weight values that connect all input neurons to all self-organizing maps become convergent. After converging, the weight is stored and can later be used in the classification process. For the classification process, input required is a set of weights that have been converging, self-organizing maps architecture, sample of data used for training process, and sunspot group data that will be classified. The first step is to perform clustering of all sample data that will form clusters on available maps. Then we will look for the winning map that best fits the data to be classified. Once obtained, then all sample data that belong to a cluster with winning map will be processed using the k-NN method to get the classification result.

For testing, same as two methods before, we used 164 data for training and 50 data for testing. We have done testing on variation of the learning rate and used 5x5 regions size. The result can be seen in Table 7.

Table 7. Self-organizing maps testing results based on variation of learning rate

Learning Rate	Accuracy	% accuracy
---------------	----------	------------



0.2	36/50	72%
0.5	37/50	74%
0.9	36/50	72%

Another testing based on variation of region size have been done also. The result can be seen in Table 8.

Table 8. Self-organizing maps testing results based on variation of number of region

Region size	Accuracy	% accuracy
3 x 3	39/50	78%
5 x 5	38/50	76%
7 x 7	37/50	74%

After completion of the experiments on each of the three artificial neural network methods, we compare the result of those methods. The comparison has been done between the three methods are comparison of the accuracy with a varying number of training data and comparison of the accuracy with a varying number of image regions. For comparison of the accuracy with a varying amount of training data, probabilistic neural network has performed better than the others. The result can be seen in Figure 13.

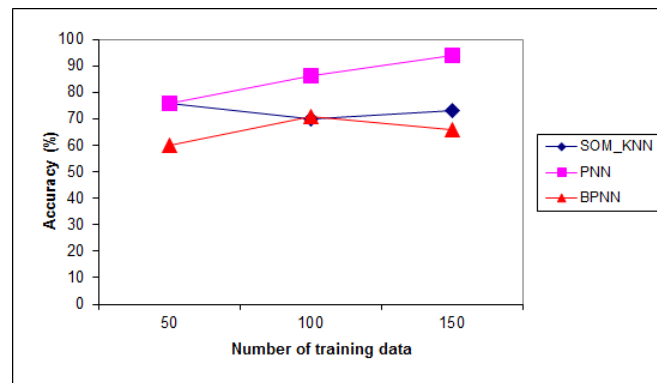


Figure 13. Comparison result based on varying number of training data

For comparison of the accuracy with a varying number of image regions, probabilistic neural network has performed better than the others again. The result can be seen in Figure 14.

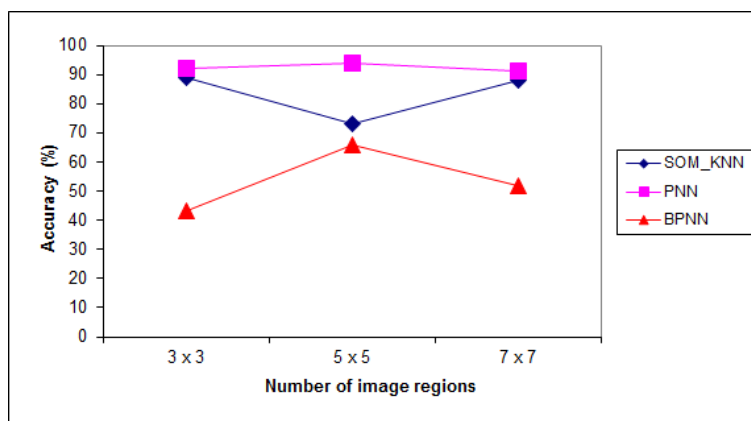


Figure 14. Comparison result based on varying number of image regions

From the comparison result, it can be concluded that probabilistic neural network perform better for classification of sunspot groups than the other two methods. So in our system, we used probabilistic neural network. The screen view of our system can be seen in Figure 15. Using our system, the image of sunspot can be classified automatically and the classification result can be used for space weather analysis.

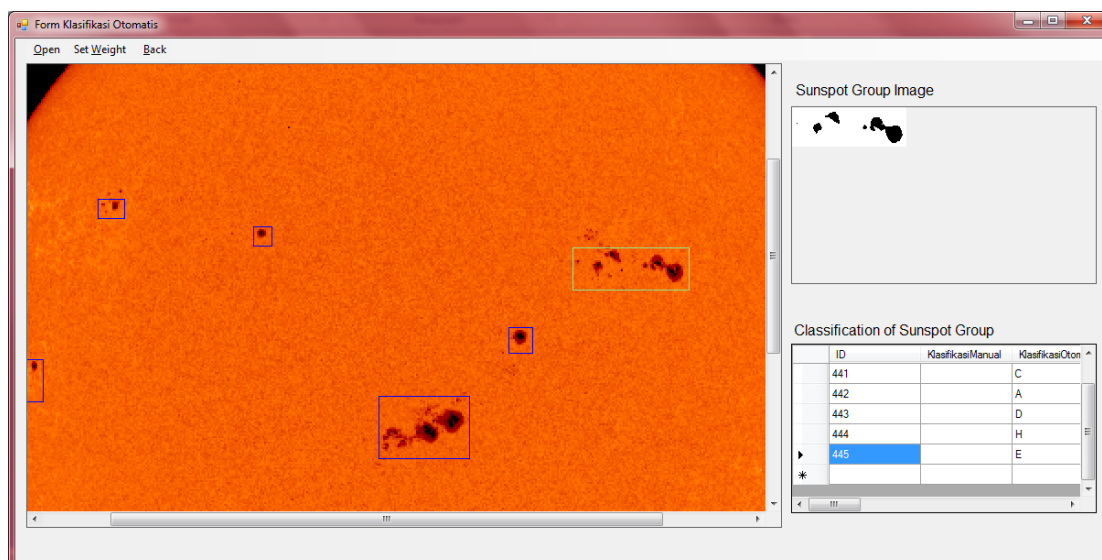


Figure 15. Screenshot of Automatic Classification System

## 5. Conclusion

In this paper has been discussed system development for automatic classification of sunspot groups from digital solar image. The classification is done using artificial neural network. From experimental result, it can be concluded that backpropagation method less appropriate for use on the system, because maximal accuracy is only 71%. Probabilistic neural network method is more appropriate for our automatic classification system because from the comparison result, it appears that the performance of a probabilistic neural network is always better than the other two methods.




## 6. Acknowledgments

This research was funded by Research Competitive Grant DP2M Directorate General of Higher Education (101/SP2H/PP/DP2M/3/2010), National Education Ministry, Indonesia,

fiscal year 2010, Research Competitive Grant DIPA-PT Coordination of Private Higher Education Region VII, East Java, Indonesia (0082/SP2H/PP/K7/KL/IV/2011), fiscal year 2011 and Research Center, Petra Christian University, Surabaya, Indonesia (12/Sugas-Pen/LPPM-UKP/2012), fiscal year 2012, entitled “Automated Sunspot Group Classification or Analyzing Space Weather Conditions”. Also thanks to Adrian Hartanto N. for his work on system coding.

## 7. References

- [1] Setiahadhi, B. Automatic Determination of the Relative Sunspot Number from White-Light Full Disk Solar Digital Data Using Cluster Method and Turtle Algorithm. Proceeding of International Conference on Mathematics and Natural Sciences (ICMNS). (2006): Bandung, Indonesia.
- [2] Bothmer, V. and Daglis, I.A.. Space Weather, Physics and Effects. Springer-Praxis Publishing, (2007).
- [3] McIntosh, Patrick. S. The Classification of Sunspot Groups. Solar Physics, vol. 125 no.2, pp. 251-267 (1990)
- [4] Ester, M., Kriegel H.P., Sander J., and Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining. (1996).
- [5] Gonzalez, Rafael C., and Woods, Richard E. Digital Image Processing 3rd Edition. Upper Saddle River, New Jersey (2008).
- [6] Negnevitsky, M. Artificial intelligence: A Guide to Intelligence Systems (2nd ed.). Addison Wesley, New York (2005).
- [7] Specht, Donald F. Probabilistic Neural Network. Neural Networks, vol. 3, no. 1, pp. 109–118 (1990).
- [8] Fausett, L. Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice-Hall International., Inc (1994).
- [9] Larose, D.T. Discovering Knowledge in Data : An Introduction to Data Mining. John Wiley & Sons, Inc (2005).
- [10] Schenker, A., Bunke, H., Last, M. and Kandel, A. Graph-Theoretic Techniques For Web Content Mining. World Scientific (2005).
- [11] Adipranata, R., Budhi, Gregorius S., Setiahadhi, B., and Anwar, B. Segmentation of Sunspot Group using Watershed Method. Proceeding of National Conference of System and Informatics. (2010): Bali, Indonesia.
- [12] Budhi, Gregorius S., Adipranata, R., Sugiarto, M., Anwar, B., and Setiahadhi B., Sunspot Grouping of Solar Digital Image using DBSCAN Clustering Method. Proceeding of National Seminar of Information Technolgy Application. (2011): Yogyakarta, Indonesia.
- [13] Budhi, Gregorius, S., Adipranata, R., Setiahadhi B, Anwar B, Hartanto A, Tjondrowiguno A.N. Combination of Self-Organizing Map Neural Network and K-Nearest Neighbor for Automatic Classification of Sunspot Group Image. Proceeding of National Conference of System and Informatics. (2011):Bali, Indonesia.
- [14] Han, J., Kamber, M., Pei, J. Data Mining: Concept and Techniques 3rd Edition. Morgan Kaufmann Publishers, Waltham USA (2012).

	<p><b>Rudy Adipranata</b> is currently a senior lecturer in Informatics Department, Petra Christian University, Surabaya, Indonesia. He received his bachelor degree in Electrical Engineering from Petra Christian University, Surabaya, Indonesia, and master degree in Software Engineering from Graduate School of Software, Dongseo University, Busan, South. His research interests are image processing and computer vision.</p>
	<p><b>Gregorius Setia Budhi</b> is currently a senior lecturer in Informatics Department, Petra Christian University, Surabaya. He received his bachelor degree in Electrical Engineering, minority on computer science from Adhi Tama Institute of Technology Surabaya, Indonesia and master degree in Informatics from Sepuluh November Institute of Technology, Surabaya, Indonesia. His research interests are artificial intelligence and data mining.</p>
	<p><b>Bambang Setiahari.</b> Senior researcher at LAPAN (Indonesian National Institute of Aeronautics and Space). He Received degree in ITB-Astronomy for BSc and Drs, and his MSc and DSc at NAOJ (National Astronomical Observatory of Japan). Permanent office at Watukosek Solar-Terrestrial Observatory, in East Java, Indonesia. Specialization in Solar Magnetohydrodynamics and Solar Activity Early Warnings.</p>