

# Desain dan Implementasi Komputasi Paralel dengan Algoritma Genetik untuk Prapemrosesan *Probabilistic Neural Network*

Kudang Boro Seminar, Agus Buono, Teguh Pratama Januzir Sukin

## Abstrak

*Probabilistic Neural Network* (disingkat PNN) merupakan salah satu jaringan saraf tiruan yang banyak dikembangkan bagi kepentingan manusia. PNN memiliki tingkat akurasi klasifikasi yang cukup tinggi dan waktu pelatihan yang cukup singkat. Salah satu faktor yang mempengaruhi akurasi klasifikasi PNN adalah parameter penghalus ( $\sigma$ ).

Proses pencarian nilai  $\sigma$  yang optimum merupakan salah satu prapemrosesan yang harus dilakukan sebelum masuk ke PNN. Nilai parameter  $\sigma$  tidak dapat ditentukan secara langsung. Oleh sebab itu algoritma genetik dapat digunakan untuk mencari nilai  $\sigma$  yang optimum. Pada umumnya solusi dari algoritma genetik semakin baik apabila jumlah populasi yang dibangkitkan cukup besar dan proses alamiah evolusi sering terjadi. Hal ini tentunya membutuhkan banyak iterasi dan sumberdaya komputasi yang cukup besar. Permasalahan ini dapat diatasi dengan menggunakan komputasi paralel yang menerapkan metode pelemparan pesan. Dengan mengkombinasikan metode algoritma genetik dan pemrosesan paralel untuk menduga parameter penghalus ( $\sigma$ ), maka diharapkan akan diperoleh model PNN yang optimum.

Penelitian ini bertujuan untuk menguji dan membandingkan kinerja dari algoritma genetik sekuensial, algoritma genetik paralel mode vektor baris, dan algoritma genetik mode vektor kolom. Penelitian menggunakan basis data bunga iris yang dikembangkan oleh R.A. Fisher. Berbagai kriteria yang digunakan untuk uji analisis adalah akurasi klasifikasi PNN, waktu eksekusi total, rasio kebergantungan mesin, peningkatan kecepatan, dan efisiensi.

Kompleksitas waktu terbaik dari algoritma genetik sekuensial adalah  $\Omega(N_M N_P)$  dan algoritma genetik paralel adalah  $\Omega(2N_{Pro}^2 + N_M N_P)$ . Sedangkan Kompleksitas waktu terburuk dari algoritma genetik sekuensial adalah  $O(N_{Itr} N_{Pop} (N_{Pop} + N_M N_P))$  dan algoritma genetik paralel adalah  $O\left(N_{Itr} \left(C_{P2} + \frac{N_{Pop} (N_M N_P)}{N_{Pro}} + N_{Pop}^2\right)\right)$ .

Dari percobaan didapatkan kesimpulan bahwa secara umum kinerja algoritma genetik paralel jauh lebih baik apabila dibandingkan dengan algoritma genetik sekuensial. Sedangkan kinerja antara algoritma genetik paralel mode vektor baris dan kinerja algoritma genetik paralel mode vektor kolom tidak terlampau jauh berbeda.

## PENDAHULUAN

### Latar Belakang

Dalam perkembangan komputasi yang pesat pada saat ini, telah banyak bidang kegiatan manusia yang diaplikasikan dengan menggunakan komputer. Beberapa fenomena komputasi yang cukup berkembang dengan cepat pada saat ini

adalah mengenai komputasi paralel, algoritma genetik (disingkat AG) dan jaringan saraf tiruan (disingkat JST).

Dengan adanya JST, suatu komputer dapat diprogram sedemikian rupa sehingga dapat melakukan pembelajaran sebagaimana layaknya manusia. Hal ini dimungkinkan disebabkan informasi yang ditangkap dan diolah oleh komputer disimulasikan dengan menggunakan

sekumpulan neuron-neuron yang terkait satu sama lain mirip dengan pengolahan informasi pada jaringan saraf biologis makhluk hidup.

Salah satu JST yang banyak diteliti dan dikembangkan akhir-akhir ini oleh berbagai kalangan untuk berbagai kepentingan manusia adalah *Probabilistic Neural Network* (disingkat PNN). Keunggulan yang dimiliki PNN adalah tingkat keakuratan yang cukup tinggi dan waktu pelatihannya yang cukup singkat.

Salah satu kendala utama dalam PNN adalah sulitnya mencari nilai parameter penghalus ( $\sigma$ ). Proses pencarian nilai  $\sigma$  ini merupakan salah satu prapemrosesan sebelum masuk ke PNN. Bila kita mendapatkan nilai  $\sigma$  yang tepat maka PNN memiliki keakuratan sampai mendekati 100%. Tetapi tingkat keakuratan dari PNN bisa berkurang sampai dengan 50% apabila terjadi kesalahan dalam menentukan nilai parameter penghalus yang tepat.

Untuk mendapatkan parameter penghalus yang optimum, maka dapat digunakan AG yang merupakan suatu algoritma pencarian solusi kombinatorik yang bersifat acak sistematis. AG merupakan suatu teknik proses komputasi yang pada dasarnya meniru teori evolusi dari bidang ilmu Biologi.

Pada umumnya solusi yang didapatkan oleh AG semakin baik apabila jumlah populasi yang dibangkitkan cukup besar dan proses alamiah evolusi seperti mutasi dan rekombinasi sering terjadi. Keadaan ini tentunya membutuhkan banyak iterasi dan sumberdaya komputasi. Untuk mengatasinya maka digunakanlah komputasi paralel yang menggunakan metode pelemparan pesan.

Untuk mewujudkan komputasi paralel ini maka digunakanlah *Library Message Passing Interface* (disingkat MPI) yang dapat mendukung metode *Single Program Multiple Data* (disingkat SPMD) pada beberapa mesin secara *realtime*. Dengan mengkombinasikan metode AG dan pemrosesan paralel untuk menduga parameter penghalus ( $\sigma$ ), maka diharapkan akan diperoleh model PNN yang optimum.

### Tujuan

Penelitian ini bertujuan untuk:

1. Mendapatkan metode pencarian nilai  $\sigma$  yang tepat, efektif, efisien, dan cepat untuk diterapkan pada fungsi kernel PNN.
2. Mengetahui pengaruh nilai parameter  $\sigma$  persamaan kernel terhadap keakuratan klasifikasi PNN.
3. Mengetahui keunggulan dan cara kerja metode PNN, AG, dan komputasi paralel.
4. Membandingkan akurasi klasifikasi PNN, waktu total (kecepatan eksekusi program), rasio kebergantungan mesin (*Machine Dependent Ratio*), peningkatan kecepatan (*Speed Up*), dan efisiensi (*Efficiency*) antara pemrosesan AG sekuensial, AG paralel mode vektor baris, dan AG paralel mode vektor kolom.

### Ruang Lingkup Penelitian

Ruang lingkup penelitian ini dibatasi oleh AG yang diterapkan pada komputasi sekuensial dan komputasi paralel untuk menentukan nilai  $\sigma$  kernel pada PNN.

### Hasil dan Manfaat

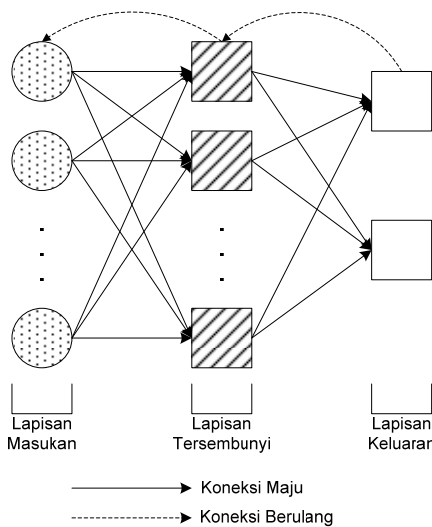
Hasil dari percobaan ini adalah kesimpulan komprehensif mengenai perbandingan proses komputasi yang dilakukan secara sekuensial terhadap proses komputasi yang dilakukan secara paralel. Dengan adanya perbandingan terhadap nilai-nilai akurasi klasifikasi PNN, waktu total, rasio kebergantungan mesin, peningkatan kecepatan, dan efisiensi pemrograman yang diuji oleh MANOVA, akan didapatkan kesimpulan mengenai kompleksitas dan efisiensi untuk setiap mode komputasi. Dengan dilakukannya penelitian ini diharapkan bisa menjadi bahan acuan dalam pengembangan program komputasi yang cukup kompleks di masa yang akan datang.

## TINJAUAN PUSTAKA

### Jaringan Saraf Tiruan

Menurut Fu (1994), Jaringan Saraf Tiruan (JST) merupakan suatu sistem

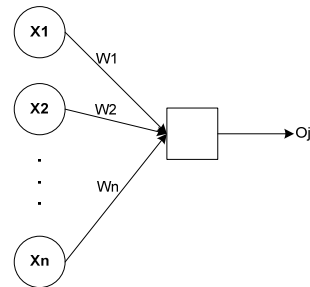
pemrosesan informasi digital yang memiliki karakteristik-karakteristik seperti jaringan saraf pada makhluk hidup.



Gambar 1. Bagan Model Sederhana Jaringan Saraf Tiruan (JST).

Pada dasarnya, pemrosesan informasi pada JST mengacu pada pemrosesan informasi yang terjadi pada sel-sel saraf biologis, yaitu dengan pemancaran sinyal elektro kimia melalui serabut-serabut saraf (neuron). Pada jaringan saraf biologis, apabila suatu neuron mendapatkan suatu sinyal elektro kimia yang melebihi suatu ambang tertentu, maka neuron tersebut akan meneruskan sinyal tersebut kepada neuron yang lain melalui sinapsis. Dan begitu pula sebaliknya, apabila sinyal yang diterima neuron tidak mencapai ambang tersebut maka sinyal tidak akan diteruskan kepada neuron-neuron yang lain.

Pada JST, setiap neuron menerima input ( $x$ ) dari setiap neuron lain yang dikalikan dengan suatu nilai pembobotan ( $W$ ) yang sesuai. Total penjumlahan akumulatif dari himpunan input terboboti dinamakan dengan level aktivasi. Level aktivasi inilah yang akan menentukan kemungkinan apakah suatu neuron dapat meneruskan sinyal ataukah tidak kepada neuron-neuron yang lain.



Gambar 2. Bagan Model Aktivasi Sinyal Jaringan Saraf Tiruan (JST).

$$LevelAktivasi = \sum_{i=1}^n X_i W_i \quad (1)$$

Dengan

$X_i$  : Input ke- $i$

$W_i$  : Bobot untuk input ke- $i$

Sebelum dapat digunakan, JST harus diberikan pelatihan terlebih dahulu. Pelatihan ini diperlukan untuk menemukan nilai pembobotan yang tepat bagi JST agar keluarannya menjadi benar.

Pola pelatihan yang diterapkan bagi JST ada dua macam, yaitu pelatihan *Supervised* dan pelatihan *Unsupervised*. JST dikatakan belajar dengan *Supervised* apabila output yang diinginkan telah diketahui. Sedangkan JST dikatakan belajar dengan *Unsupervised* bila output yang diinginkan tidak diketahui.

### Pengklasifikasian Bayes

Menurut Rish (2004), pengklasifikasian Bayes merupakan suatu metode klasifikasi yang menggunakan suatu fungsi diskriminan pada peluang posterior kelas. Fungsi klasifikasi yang digunakan oleh kelas  $i$  dapat dinotasikan dengan :

$$P(\omega_i)P(x | \omega_i) \quad (2)$$

Dengan

$P(\omega_i)$  : Peluang kelas  $i$

$P(x|\omega_i)$  : Peluang bersyarat  $x$  jika masuk ke dalam kelas  $i$

$X$  : Vektor Input

$\omega_i$  : Kelas  $i$

Pengklasifikasian Bayes banyak digunakan pada PNN. Dengan menggunakan metode Bayes, klasifikasi

dapat dilakukan seoptimal mungkin dengan cara meminimalisir nilai kerugian yang terjadi bila terjadi kesalahan klasifikasi. Untuk mengklasifikasikan input  $x$  agar masuk ke dalam kelas A, maka harus dipenuhi syarat :

$$h_A c_A f_A(x) > h_B c_B f_B(x) \quad (3)$$

Dengan

- $h_A$  : Kemungkinan contoh terambil dari kelas A
- $h_B$  : Kemungkinan contoh terambil dari kelas B
- $c_A$  : Biaya yang dikorbankan bila terjadi kesalahan klasifikasi input A
- $c_B$  : Biaya yang dikorbankan bila terjadi kesalahan klasifikasi input B
- $f_A$  : Fungsi Kepekatan A
- $f_B$  : Fungsi Kepekatan B

Apabila syarat persamaan diatas tidak terpenuhi, maka input  $x$  dimasukkan kedalam kelas B.

#### Penduga Kepekatan Parzen

Menurut Fu (1994), Fungsi kepekatan yang digunakan untuk PNN yang berkaitan dengan data multivariat adalah fungsi kepekatan Parzen. Fungsi Parzen merupakan suatu prosedur non parametrik yang mensintesis Penduga Probability Density Function (PDF) Gauss. Fungsi Parzen akan memberikan keputusan klasifikasi setelah menghitung PDF untuk setiap kelas melalui pola pelatihan yang ada.

Pada fungsi Parzen terdapat fungsi pembobot yang disebut dengan fungsi Kernel ( $K(x)$ ). Fungsi Parzen untuk data multivariat dapat dinotasikan dengan:

$$g(x) = \frac{1}{n\sigma^2} \sum_{i=1}^n K\left(\frac{x-x_i}{\sigma}\right) \quad (4)$$

Sedangkan fungsi Kernel yang digunakan adalah fungsi Gauss dinotasikan dengan :

$$K(x) = \frac{1}{\sigma(2\pi)^{\frac{d}{2}}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (5)$$

Maka kita mendapatkan fungsi kepekatan untuk kelas A sebagai berikut :

$$f_A(x) = \frac{1}{N_A \sigma^d (2\pi)^{\frac{d}{2}}} \sum_{i=1}^{N_A} \exp\left(\frac{-(x-x_{Ai})^T(x-x_{Ai})}{2\sigma^2}\right) \quad (6)$$

Jadi dapat dinotasikan :

$$P(\omega_A) = \frac{N_A}{N} \quad (7)$$

$$P(x|\omega_A) = \frac{1}{N_A \sigma^d (2\pi)^{\frac{d}{2}}} \sum_{i=1}^{N_A} \exp\left(\frac{-(x-x_{Ai})^T(x-x_{Ai})}{2\sigma^2}\right) \quad (8)$$

Sehingga

$$P(\omega_A)P(x|\omega_A) = \frac{1}{N\sigma^d(2\pi)^{\frac{d}{2}}} \sum_{i=1}^{N_A} \exp\left(\frac{-(x-x_{Ai})^T(x-x_{Ai})}{2\sigma^2}\right) \quad (9)$$

Dengan

- $P(\omega_A)$  : Peluang kelas A
- $P(x|\omega_A)$  : Peluang bersyarat  $x$  jika masuk ke dalam kelas A
- $x_{Ai}$  : Pola pelatihan ke- $i$  kelas A
- $d$  : Dimensi vektor input
- $N_A$  : Jumlah pola pelatihan kelas A
- $N$  : Jumlah pola pelatihan seluruh kelas
- $\sigma$  : Faktor penghalus

#### Faktor Penghalus Sigma ( $\sigma$ )

Menurut Lee, D. X, *et. al.* (2004),  $\sigma$  merupakan suatu nilai parameter yang berguna untuk menghaluskan fungsi kernel. Secara tidak langsung nilai  $\sigma$  berperan pula dalam menentukan ketepatan klasifikasi PNN. Nilai  $\sigma$  tidak dapat ditentukan secara langsung, akan tetapi bisa didapatkan melalui metode statistik maupun dari hasil coba-coba. Pada penelitian ini, nilai  $\sigma$  didapatkan melalui AG.

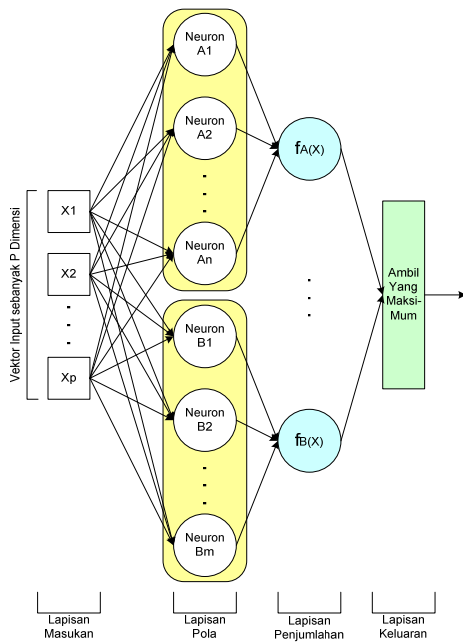
#### Probabilistic Neural Network (PNN)

Menurut Rahmani (2004), PNN merupakan JST yang menggunakan

teorema probabilitas klasik seperti pengklasifikasian Bayes dan penduga kepekatan Parzen. Proses yang dilakukan oleh PNN dapat berlangsung lebih cepat bila dibandingkan dengan JST Propagasi Balik. Hal ini terjadi disebabkan PNN hanya membutuhkan satu kali iterasi pelatihan bila dibandingkan dengan JST Propagasi Balik yang membutuhkan beberapa kali iterasi dalam proses pelatihannya.

Walaupun demikian, keakuratan dari klasifikasi PNN sangat ditentukan oleh nilai  $\sigma$  dan pola pelatihan yang diberikan. Bila nilai  $\sigma$  yang diterapkan pada PNN tepat, maka akurasi klasifikasi akan mendekati atau mencapai 100%. Bila nilai  $\sigma$  yang diterapkan tidak tepat maka akurasi klasifikasi PNN akan berkurang.

Demikian pula dengan pola pelatihan PNN. Apabila pola pelatihan dan data masukan pada satu kelas yang sama sangat berbeda jauh nilainya, maka PNN akan mengekstrapolasi data masukan tersebut. Hal inilah yang nantinya akan mengakibatkan akurasi klasifikasi PNN turun cukup drastis.



Gambar 3. Bagan Model Jaringan Saraf Tiruan *Probabilistic Neural Network*.

## Algoritma Genetik

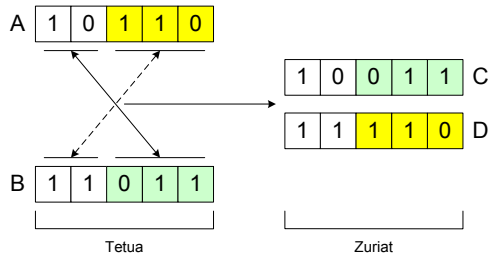
Menurut Michalewicz (1995), Algoritma genetik adalah suatu teknik proses komputasi yang pada dasarnya meniru teori evolusi alamiah. Pada algoritma ini, terdapat proses-proses utama yang menjadi prinsip utama dalam evolusi, yaitu kawin silang (*recombination*) dan mutasi (*mutation*). Algoritma ini menggunakan sistem seleksi alamiah (*natural selection*) terhadap individu baru yang muncul dari evolusi individu sebelumnya. Diharapkan individu solusi yang terseleksi merupakan individu solusi yang lebih baik bila dibandingkan individu solusi yang lainnya. Istilah-istilah alam yang berkaitan dengan AG adalah :

Tabel 1. Daftar Istilah Algoritma Genetik.

No.	Alam	Komputer
1.	Individu	Solusi dari permasalahan
2.	Populasi	Gugus Solusi
3.	Kecocokan / <i>Fitness</i>	Kualitas dari solusi
4.	Kromosom	Untaian gen
5.	Gen	Bagian kromosom berupa variabel permasalahan atau berupa digit biner
6.	Lokus	Posisi gen pada kromosom
7.	Alel	Gen yang memiliki lokus yang sepadan
8.	Rekombinasi & Mutasi	Operator pencarian (Operator Genetik)

## Rekombinasi

Menurut Tandriarto (2002), rekombinasi adalah proses materi genetik dengan menggabungkan dua individu leluhur untuk mendapatkan satu atau lebih keturunan. Penggambaran rekombinasi pada lokus ke-3 adalah sebagai berikut :

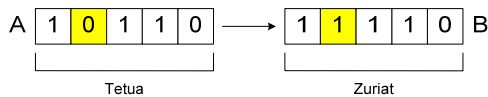


Gambar 4. Bagan Model Sederhana Proses Rekombinasi Satu Titik.

Pada individu A dan Individu B terjadi rekombinasi dari lokus ke-3 sampai lokus terakhir. Masing-masing fragmen kromosom kemudian saling bertukar posisi sehingga terbentuk dua individu baru C dan D.

### Mutasi

Menurut Tandriarto (2002), mutasi adalah proses materi genetik dengan perubahan alel, membalik urutan lokus, atau mengacak urutan lokus individu leluhur sehingga terbentuk individu baru. Operator mutasi yang sering digunakan dalam AG adalah mutasi membalikkan. Penggambaran mutasi membalikkan pada lokus ke-2 adalah sebagai berikut :



Gambar 5. Bagan Model Sederhana Proses Mutasi Satu Titik.

Pada individu A terjadi mutasi pada lokus ke-2 saja. Gen pada lokus yang bermutasi ini mengalami membalikkan gen dari 0 menjadi 1 atau dari 1 menjadi 0 sedemikian rupa sehingga terbentuk Individu B.

### Seleksi

Menurut Tandriarto (2002), seleksi merupakan proses penyaringan individu-individu dari populasi yang ada berdasarkan nilai ketahanan hidupnya (*fitness*) secara alamiah.

Semakin tinggi nilai ketahanan hidupnya maka peluang individu tersebut untuk bertahan hidup dan membentuk

regenerasi individu baru semakin besar. Sedangkan individu-individu yang memiliki nilai ketahanan hidup kecil secara alamiah akan tersisih dan punah sehingga tidak mungkin membentuk generasi baru.

### Pemrosesan Paralel

Menurut Pacifico, *et. al.* (1998), Pemrosesan Paralel adalah penggunaan banyak prosesor yang saling bekerjasama satu sama lain untuk mencari suatu solusi tunggal dari suatu permasalahan. Pemrosesan paralel dapat digunakan untuk beberapa keperluan, diantaranya adalah untuk mempercepat waktu eksekusi dan mendistribusikan pencarian solusi dari permasalahan yang sangat kompleks.

### Hukum Amdahl

Menurut Amdahl (1967), peningkatan dari pemrosesan paralel tidak hanya bergantung pada banyaknya prosesor yang digunakan, akan tetapi lebih dipengaruhi oleh fraksi rasio antara intruksi sekuensial dengan keseluruhan instruksi pada suatu program.

$$SpeedUp = \frac{1}{\left(\frac{1-F}{N}\right) + F}$$

$$F = \frac{I_{Sek}}{I_{Sek} + I_{Par}}$$

(14)

Dengan

- SpeedUp* : Peningkatan Kecepatan
- F* : Fraksi rasio intruksi sekuensial terhadap keseluruhan instruksi
- 1-F* : Fraksi rasio intruksi paralel terhadap keseluruhan instruksi
- I<sub>Sek</sub>* : Instruksi Sekuensial
- I<sub>Par</sub>* : Instruksi Paralel
- N* : Banyaknya prosesor

Hukum Amdahl menunjukkan bahwa peningkatan kinerja dari kecepatan (*Speed Up*) proses paralel dengan  $F = 0,01$  hanya naik 10 kali lipat walaupun jumlah prosesor dinaikkan dari 10 buah menjadi

1000 buah (sebanyak 100 kali lipat). Jadi nilai F pada program harus terus diperkecil untuk meningkatkan kecepatan pemrosesan paralel.

Bila hanya diketahui waktu pemrosesan intruksi paralel dan waktu pemrosesan intruksi sekuensial saja, maka perumusan *Speed Up* menjadi :

$$SpeedUp = \frac{T_{Sek} + T_{Par}}{T_{Sek} + \frac{T_{Par}}{N}} \quad (15)$$

Dengan

*SpeedUp* : Peningkatan Kecepatan

$T_{Par}$  : Waktu yang dibutuhkan sebuah prosesor untuk mengeksekusi perintah paralel

$T_{Sek}$  : Waktu yang dibutuhkan sebuah prosesor untuk mengeksekusi perintah sekuensial

$N$  : Banyaknya prosesor

#### Efisiensi (Efficiency)

Efisiensi adalah rasio antara *Speed Up* dengan banyaknya prosesor ( $N$ ).

$$Efficiency = \frac{SpeedUp}{N} \quad (16)$$

Dengan

*SpeedUp* : Peningkatan Kecepatan

$N$  : Banyaknya Prosesor

Bila nilai efisiensi semakin tinggi, maka kinerja program semakin baik. Dan bila sebaliknya maka kinerja dari program semakin buruk.

#### Machine Dependent Ratio (MDR)

MDR adalah suatu rasio antara waktu komunikasi dengan waktu kalkulasi. Semakin besar MDR maka waktu pemrosesan paralel semakin lambat. Demikian pula bila sebaliknya.

$$MDR = \frac{T_{Comm}}{T_{Calc}} \quad (17)$$

Dengan

$T_{Comm}$  : Waktu yang dibutuhkan untuk mentransfer sebuah pesan antara dua buah simpul (node).

$T_{Calc}$  : Waktu yang dibutuhkan untuk melakukan suatu operasi kalkulasi floating point.

Salah satu strategi untuk memperkecil MDR ialah dengan memperbesar *overlapping* antara waktu komunikasi dan waktu kalkulasi pada setiap *node* komputer.

#### Message Passing Interface (MPI)

Menurut Toonen (1998), MPI merupakan kumpulan fungsi-fungsi *library* yang digunakan untuk saling menghubungkan dan mensubstitusi pesan-pesan aplikasi antar komputer.

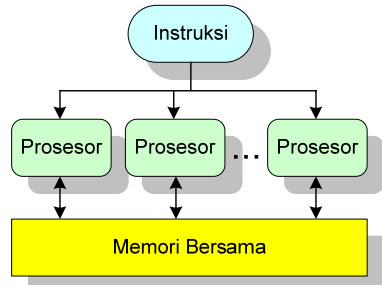
MPI digunakan pada sistem yang mendukung arsitektur pemrograman SPMD (*Single Program Multiple Data*) dengan menggunakan pemrograman soket (*Socket Programming*). Kelebihan dari MPI adalah adanya suatu mekanisme untuk sinkronisasi proses pada komputasi paralel. Fungsi sinkronisasi yang digunakan dalam penelitian ini adalah mekanisme *blocking* dalam pengiriman data (MPI *Isend*), *blocking* dalam penerimaan data (MPI *Irecv*), dan *blocking* dalam eksekusi (MPI *Barrier*).

#### Pemrograman Soket

Menurut Quinton (1997), Pemrograman Soket adalah pemrograman jaringan untuk mengembangkan sistem *Client Server* dengan menggunakan protokol TCP/IP atau UDP/IP. Pemrograman ini meliputi sistem pengalamatan jaringan yang terdiri atas *services*, *sockets*, dan *port*.

#### Single Instruction Multiple Data (SIMD)

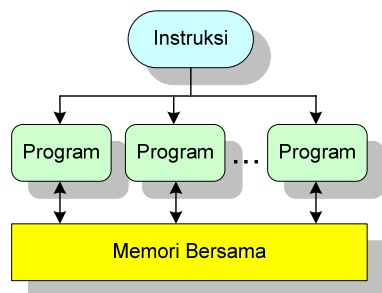
Menurut Flynn (1966), SIMD adalah suatu klasifikasi arsitektur perangkat keras komputer yang menerapkan satu buah pengontrol (*controller*) pada beberapa unit elemen pemroses data untuk memroses beberapa aliran data secara bersamaan.



Gambar 6. Bagan Model SIMD.

### Single Processing Multiple Data (SPMD)

Menurut Flynn (1966), SPMD merupakan suatu paradigma proses paralel pada program, dimana terminologinya hampir mirip dengan SIMD pada perangkat keras.

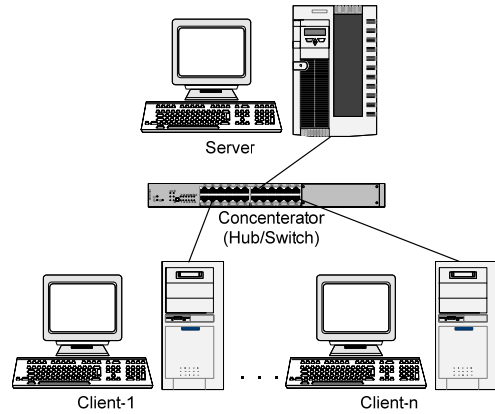


Gambar 7. Bagan Model SPMD.

SPMD merupakan bentuk arsitektur pemrograman tidak sinkron (*asynchronous*) yang diterapkan pada beberapa buah sistem komputer berarsitektur MIMD (*Multiple Instruction Multiple Data*).

### Topologi Jaringan Berbentuk Bintang

Topologi model ini dirancang agar setiap *node* terkoneksi ke jaringan melewati sebuah *concentrator*. Keuntungan Topologi jaringan model bintang diantaranya adalah : mudah di pasang dan pengabelan tidak rumit; tidak mengakibatkan gangguan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya; dan mudah untuk mendeteksi kesalahan yang terjadi pada kabel maupun *node*.



Gambar 8. Bagan Model Jaringan Topologi Bintang.

### Analisis keragaman (ANOVA)

Menurut Moore (1994), ANOVA (*Analysis of Variance*) Merupakan sebuah metode penarikan kesimpulan dengan cara membandingkan keragaman dalam contoh dengan keragaman antar individu yang serupa. Khususnya, ANOVA menguji apakah beberapa populasi mempunyai rata-rata sama dengan membandingkan lebar rentang rata-rata contoh dan banyaknya variasi yang ada di dalam contoh.

MANOVA (*Multivariate Analysis of Variance*) adalah metode pengujian ANOVA yang diterapkan pada variabel - variabel kategori tidak bebas dan variabel - variabel bebas. Bila ANOVA menguji beda rata-rata dari sebuah variabel bebas terhadap variabel-variabel tidak bebas, maka MANOVA menguji beda vektor dari rata-rata variabel-variabel bebas terhadap variabel-variabel tidak bebas.

## METODE

Tahapan - tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut :

### Definisi Masalah

Definisi masalah dilakukan dengan cara melakukan studi pustaka untuk mempelajari permasalahan - permasalahan yang berkaitan dengan perancangan sistem. Hasil tahapan ini telah diuraikan pada bab Tinjauan Pustaka.

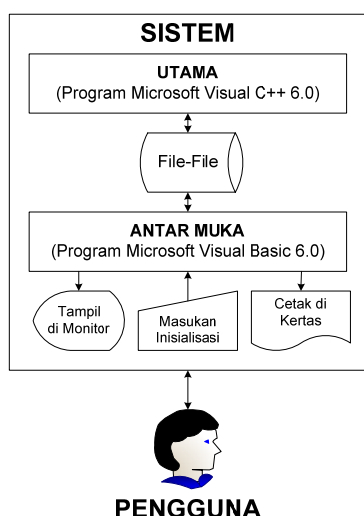


## Perancangan

Perancangan sistem dilakukan dengan cara menyusun strategi umum proses penyusunan dan perbandingan algoritma. Hal ini dapat dimaksudkan sebagai dasar acuan pada tahapan implementasi dan pengujian. Perancangan sistem secara garis besar terdiri atas :

### 1. Perancangan Model Interaksi Antara Sistem dengan Pengguna

Untuk model perancangan interaksi antara sistem yang akan dibangun dengan pengguna dapat dilihat pada gambar dibawah ini :



Gambar 9. Bagan Model Interaksi Sistem dan Pengguna

### 2. Perancangan Representasi Program

Program yang digunakan dalam penelitian ini ada dua macam, yaitu :

#### 1. Program Utama

Program Utama merupakan program yang menjadi inti dari penelitian ini. Program inilah yang nantinya akan memroses seluruh data - data sesuai dengan inisialisasi yang diberikan. Program Utama ini membutuhkan *library* MPI. Oleh sebab itu, *daemon* MPICH (*mpichd.exe*) harus dijalankan terlebih dahulu sebelum Program Utama dieksekusi. Program Utama dapat dieksekusi melalui konsol (*command prompt*) maupun melalui

lingkungan GUI MPIRun. Dalam penelitian ini, seluruh eksekusi Program Utama dilakukan didalam lingkungan GUI MPIRun.

#### 2. Program Antar Muka

Program Antar Muka merupakan program yang digunakan untuk memudahkan pengguna dalam menginisialisasi dan menjalankan Program Utama. Pada program ini dapat dilakukan berbagai macam aksi seperti :

- Memulai eksekusi Program Utama terhadap data-data yang ada.
- Memanipulasi data inisialisasi
- Memanipulasi file-file data masukan
- Melaporkan data-data hasil eksekusi Program Utama kepada pengguna.

Ketika akan mengeksekusi sistem, kedua program yang ada harus dijalankan secara bersamaan. Program Utama dan Program Antar Muka akan saling mengecek kondisi status pada file-file berekstensi STA (\*.STA). Bila kondisi status dari kedua program sudah siap, maka eksekusi sistem dapat segera dilakukan.

Dalam prosesnya, Program Utama akan mencatat dan memberikan laporan terhadap proses yang terjadi kedalam file-file keluaran. Data-data file-file keluaran diantaranya adalah :

#### 1. Waktu Total

Waktu total adalah waktu yang dibutuhkan sistem untuk menyelesaikan satu kali eksekusi secara keseluruhan.

#### 2. Waktu Komunikasi

Waktu komunikasi adalah waktu yang dibutuhkan komputer *Server* untuk saling berkomunikasi dengan komputer-komputer *Client*.

#### 3. Waktu Kalkulasi

Waktu kalkulasi adalah waktu yang dibutuhkan setiap komputer untuk menyelesaikan seluruh proses eksekusi tanpa melakukan proses komunikasi.

#### 4. Waktu Instruksi Sekuensial

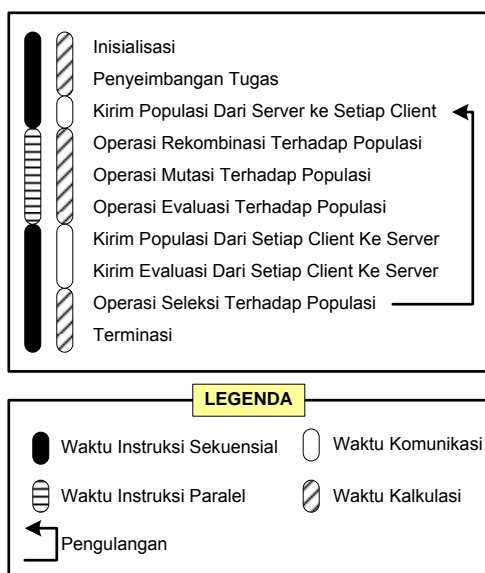
Waktu instruksi sekuensial adalah waktu yang dibutuhkan komputer *Server* untuk melakukan proses-proses sekuensial.

#### 5. Waktu Instruksi Paralel.

Waktu intruksi paralel adalah waktu yang dibutuhkan oleh setiap komputer untuk melakukan instruksi yang dikerjakan secara paralel.

6. Nilai Sigma ( $\sigma$ ) terbaik
7. Tingkat Klasifikasi PNN terbaik
8. Rasio Kebergantungan Mesin
9. Peningkatan Kecepatan
10. Efisiensi

Fungsi waktu yang mencatat perubahan waktu pada Program Utama hanya memiliki tingkat ketelitian satu detik saja. Oleh sebab itu apabila suatu proses tertentu berlangsung kurang dari satu detik, maka sistem akan mencatat perubahan waktu sama dengan nol.



**Gambar 10. Bagan Model Waktu Instruksi**

Pada gambar diatas dapat dilihat bagaimana cara Program Utama mencatat waktu dari setiap proses yang terjadi. Waktu total bisa didapatkan dari penjumlahan waktu komunikasi ditambah waktu kalkulasi atau dari waktu instruksi sekuensial ditambah waktu instruksi paralel.

Bila mode komputasi dilakukan secara sekuensial, maka waktu instruksi paralel dan waktu komunikasi sama dengan nol. Sehingga waktu total, waktu instruksi sekuensial, dan waktu kalkulasi akan selalu bernilai sama.

### 3. Perancangan Representasi Struktur Data

Dalam melakukan percobaan, setiap inisialisasi dan data-data yang masuk maupun yang keluar direpresentasikan dalam bentuk file-file teks data. Secara umum, file-file data yang digunakan dalam percobaan ini dibagi ke dalam 4 kategori, yaitu :

1. File-file Inisialisasi (\*.INI)
2. File-file Masukan (\*.TXT)
3. File-file Keluaran (\*.TXT, \*.STA)
4. File-file Sementara (\*.TD7, \*.TMP, \*.DBL, \*.OLD, \*.MTR)

### 4. Perancangan Nilai Batas Parameter dan Metode Eksekusi Program

Batasan-batasan nilai parameter dan mode eksekusi program yang digunakan adalah sebagai berikut :

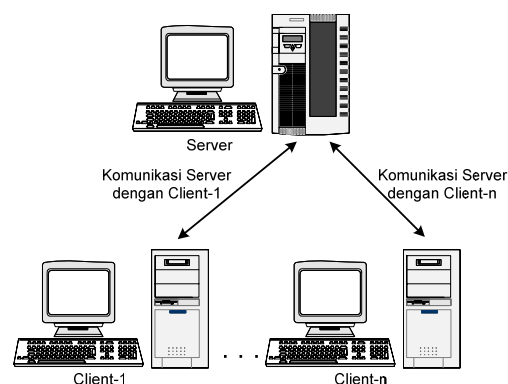
1. Operator mutasi yang digunakan dalam AG adalah operator mutasi satu titik dan operator rekombinasi yang digunakan dalam AG adalah operator rekombinasi satu titik.
2. Nilai peluang mutasi AG adalah 0,6 dan Nilai peluang rekombinasi AG adalah 0,35. Sedangkan banyaknya generasi yang dibangkitkan setiap eksekusi AG adalah 10 dan ketelitian pencarian nilai  $\sigma$  kernel dibagi dalam 9.007.199.254.740.991 interval. Nilai-nilai parameter ini selalu konstan untuk setiap percobaan dan perulangan.
3. PNN digunakan sebagai fungsi operator evaluasi bagi tiap individu AG. Akurasi pengklasifikasian PNN akan menentukan nilai ketahanan hidup bagi masing-masing individu yang dibangkitkan oleh AG.
4. Operator seleksi AG yang digunakan adalah Roda Roulette (*Roulette Whell*).
5. Setiap individu dalam AG hanya mewakili satu buah variabel tunggal, yaitu nilai parameter  $\sigma$  kernel PNN.
6. Parameter  $\sigma$  kernel PNN yang digunakan pada penelitian ini bersifat homogen, karena data

- pelatihan dan data masukan hanya terdiri atas satu tipe data kontinyu yaitu satuan cm.
7. Data yang digunakan sebagai acuan pelatihan dan contoh masukan PNN berasal dari basis data bunga Iris yang dibuat oleh R. A. Fisher. Pengelompokan PNN dibagi atas 3 kelas, yaitu :
    - a. Bunga *Iris Setosa* sebanyak 50 vektor data.
    - b. Bunga *Iris Versicolor* sebanyak 50 vektor data.
    - c. Bunga *Iris Virginica* sebanyak 50 vektor data.
  8. Jumlah faktor perlakuan yang digunakan dalam analisis MANOVA terdiri atas 3 faktor, yaitu :
    - a. Mode komputasi, terdiri dari :
      - 1)Komputasi Sekuensial.
      - 2)Komputasi Paralel vektor baris.
      - 3)Komputasi Paralel vektor kolom.
    - b. Rasio jumlah vektor pelatihan dengan jumlah vektor contoh masukan PNN, terdiri dari :
      - 1)135 vektor (90%) pelatihan dengan 15 vektor (10%) contoh masukan.
      - 2)120 vektor (80%) pelatihan dengan 30 vektor (20%) contoh masukan.
      - 3)105 vektor (70%) pelatihan dengan 45 vektor (30%) contoh masukan
      - 4)90 vektor (60%) pelatihan dengan 60 vektor (40%) contoh masukan
      - 5)75 vektor (50%) pelatihan dengan 75 vektor (50%) contoh masukan
      - 6)60 vektor (40%) pelatihan dengan 90 vektor (60%) contoh masukan
      - 7)45 vektor (30%) pelatihan dengan 105 vektor (70%) contoh masukan
      - 8)30 vektor (20%) pelatihan dengan 120 vektor (80%) contoh masukan
      - 9)15 vektor (10%) pelatihan dengan 135 vektor (90%) contoh masukan
    - d. Populasi AG, terdiri dari :

- 1)10 individu
- 2)100 individu
- 3)1000 individu
- 4)5000 individu

Nilai respon yang akan diuji oleh MANOVA adalah nilai akurasi klasifikasi PNN, waktu total, rasio kebergantungan mesin, peningkatan kecepatan, dan efisiensi.

### 5. Perancangan Mekanisme Komunikasi dan Kalkulasi

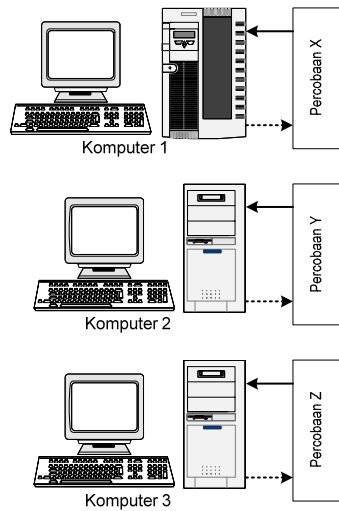


Gambar 11. Bagan Model Komunikasi Komputasi Paralel.

Dalam mode komputasi paralel, setiap komputer *Client* hanya berkomunikasi dengan komputer *Server* dan antar sesama komputer *Client* tidak berkomunikasi satu sama lain. Komunikasi hanya terjadi pada saat proses inialisasi, pengimbangan tugas, pengiriman populasi dari *Server* ke *Client*, penerimaan populasi dari *Client* ke *Server*, dan penerimaan evaluasi dari *Client* ke *Server*.

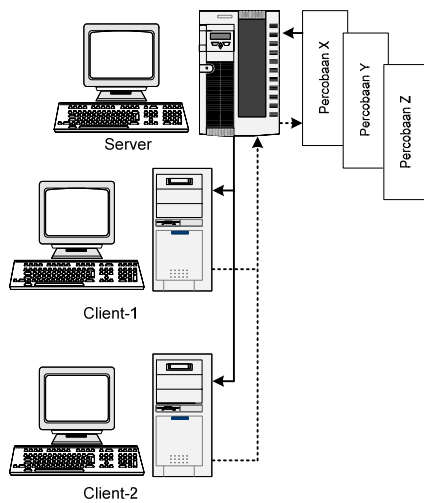
Proses-proses kalkulasi dilakukan oleh komputer *Server* dan setiap komputer *Client*. Dengan kata lain komputer *Server* tidak hanya melakukan pengontrolan proses pada setiap komputer *Client* saja, akan tetapi ikut pula memproses data yang sudah terbagi-bagi bersama-sama dengan komputer-komputer *Client*.

## 6. Perancangan *Setting Up* Jaringan dan Komputer



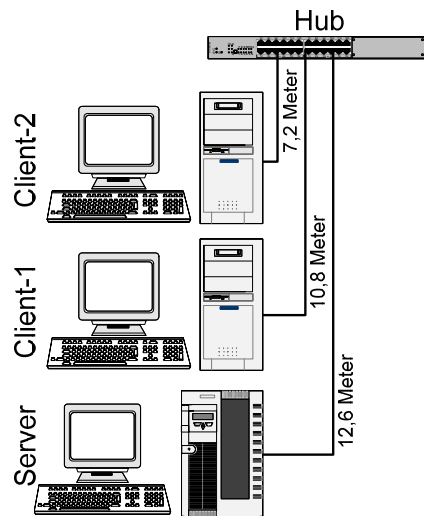
Gambar 12. Bagan Percobaan Komputasi Sekuensial.

Pada proses sekuensial, setiap komputer bekerja memproses pekerjaannya sendiri-sendiri tanpa adanya komunikasi satu sama lain. Disebabkan oleh lambatnya waktu yang diperlukan komputasi sekuensial untuk melakukan suatu operasi AG, maka setiap komputer yang identik mengerjakan percobaan yang berbeda-beda. Diharapkan dengan dilakukannya mekanisme ini dapat memberikan hasil keluaran beberapa buah percobaan yang berbeda-beda dalam waktu yang jauh lebih singkat.



Gambar 13. Bagan Percobaan Komputasi Paralel.

Pada proses paralel, ketiga komputer ini bahu membahu melakukan komunikasi dan pembagian data untuk diproses secara paralel. Satu buah komputer berfungsi sebagai *Server* dan dua buah komputer yang lain berfungsi sebagai *Client*. Setiap komputer akan mendapatkan bagian data yang berimbang untuk di proses. Pengimbangan ini dilakukan agar pemrosesan paralel pada setiap komputer dapat diselesaikan secara bersamaan.



Gambar 14. Bagan *Setting Up* Jaringan Komputer.

Dalam *Setting Up* jaringan digunakan model topologi bintang. Kabel yang digunakan adalah model UTP (*Unshielded Twisted Pair*) dengan konektor RJ-45. Kecepatan transfer data tiap komputer adalah 10 Mbps. Untuk *Concenterator* digunakan Hub *Half Duplex*. Sedangkan panjang pengkabelan antara Hub dengan komputer *Server* adalah 12,6 Meter; Hub dengan komputer *Client-1* adalah 10,8 Meter; dan Hub dengan komputer *Client-2* adalah 7,2 Meter.

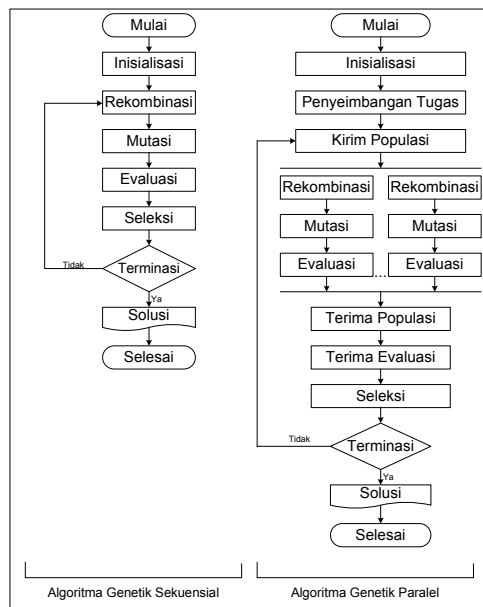
Dalam *setting up* terdapat beberapa kelemahan, diantaranya adalah :

1. Kabel UTP dipasang berdekatan dengan kabel sumber daya PLN sehingga memungkinkan terjadinya interferensi (*Cross Talk*) pada kabel UTP.
2. Hub diletakkan pada posisi yang berdekatan dengan lampu Neon.

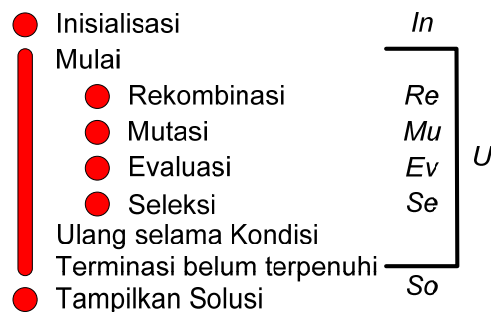
3. Hub diletakkan pada posisi berdekatan dengan jendela ruangan, sehingga mudah terganggu oleh interferensi kilat disaat hujan.
4. Panjang kabel antara komputer dengan Hub tidak seragam.

**Prosedur Percobaan**

AG paralel mengikuti proses AG sekuensial seperti terlihat pada bagan dibawah ini :



Gambar 15. Bagan Diagram Alir Algoritma Genetik Sekuensial dan Algoritma Paralel.



Gambar 16. Bagan Kompleksitas Waktu Algoritma Genetik Sekuensial.

$$T_{Total} = T_{In} + U(T_{Re} + T_{Mu} + T_{Ev} + T_{Se}) + T_{So} \tag{18}$$

Karena

- $T_{In} + T_{So}$  = Biaya Rutin ( $C_R$ )
- $T_{Re}$   $\approx$  Populasi AG ( $P$ )
- $T_{Ev}$   $\approx$  Populasi AG ( $P$ ) \* Jumlah Data Masukan PNN ( $N_M$ ) \* Jumlah Data Pelatihan PNN ( $N_P$ )
- $T_{Mu}$   $\approx$  Populasi AG ( $P$ ) \* Jumlah Bit yang diperlukan untuk merepresentasikan satu buah individu ( $R$ )
- $T_{Se}$   $\approx$  Populasi AG ( $P$ ) \* (Populasi AG ( $P$ )+3)

Maka

$$T_{Total} = C_R + U[P + (PR) + (PN_M N_P) + (P(P + 3))] \tag{19}$$

$$T_{Total} = C_R + UP[P + R + N_M N_P + 4]$$

Analisis AG sekuensial :

1. Kasus terbaik  
Kasus terbaik terjadi apabila banyak Bit yang digunakan dalam representasi kromosom AG adalah satu buah, jumlah populasi adalah dua individu, dan kondisi terminasi dapat terpenuhi dalam satu kali iterasi.

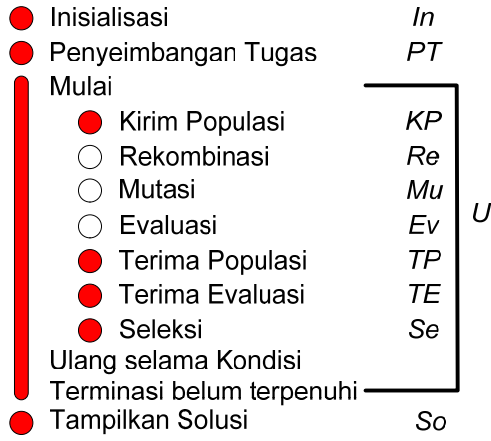
$$T_{Total} = C_R + 2N_M N_P + 14$$

$$\text{Kompleksitas} = \Omega(N_M N_P)$$

2. Kasus Terburuk  
Kasus terburuk terjadi apabila banyak Bit yang digunakan dalam representasi kromosom AG adalah 53 buah, jumlah populasi adalah  $N_{Pop}$  individu, dan kondisi terminasi dapat terpenuhi dalam  $N_{Itr}$  kali iterasi.

$$T_{Total} = C_R + (N_{Itr} N_{Pop} (N_{Pop} + N_M N_P + 57))$$

$$\text{Kompleksitas} = O(N_{Itr} N_{Pop} (N_{Pop} + N_M N_P))$$



Gambar 17. Bagan Kompleksitas Waktu Algoritma Genetik Paralel.

$$T_{Total} = T_{In} + T_{PT} + U \left( T_{KP} + \left( \frac{T_{Re} + T_{Mu} + T_{Ev}}{N_{Pro}} \right) + T_{TP} + T_{TE} + T_{Se} \right) + T_{So} \quad (20)$$

Karena

- $T_{In} + T_{So}$  = Biaya Rutin ( $C_R$ )
- $T_{PT}$  = Biaya Komunikasi Paralelisme 1 ( $C_{P1}$ )
- $T_{KP} + T_{TP} + T_{TE}$  = Biaya Komunikasi Paralelisme 2 ( $C_{P2}$ )
- $T_{Ev}$   $\approx$  Populasi AG ( $P$ ) \* Jumlah Data Masukan PNN ( $N_M$ ) \* Jumlah Data Pelatihan PNN ( $N_P$ )
- $T_{KP}, T_{Re}, T_{TP}, T_{TE}$   $\approx$  Populasi AG ( $P$ )
- $T_{Mu}$   $\approx$  Populasi AG ( $P$ ) \* Jumlah Bit yang diperlukan untuk merepresentasikan satu buah individu ( $R$ )
- $T_{Se}$   $\approx$  Populasi AG ( $P$ ) \* (Populasi AG ( $P$ )+3)
- $N_{Pro}$  = Banyaknya Prosesor

Maka

$$T_{Total} = U \left( C_{P2} + \left( \frac{C_R + C_{P1} + P + (PR) + PN_M N_P}{N_{Pro}} \right) + (P(P+3)) \right)$$

$$T_{Total} = UP \left( \left( \frac{C_R + C_{P1} + UC_{P2} + R + N_M N_P + 1}{N_{Pro}} \right) + P + 3 \right) \quad (21)$$

Analisis AG paralel :

1. Kasus terbaik  
Kasus terbaik terjadi apabila banyak Bit yang digunakan dalam representasi kromosom AG adalah satu buah, jumlah populasi adalah  $2N_{Pro}$  individu, kondisi terminasi dapat terpenuhi dalam satu kali iterasi, dan biaya paralelisme ( $C_{P1}$  dan  $C_{P2}$ )  $\approx 0$ .

$$T_{Total} = C_R + 4N_{Pro}^2 + 6N_{Pro} + 2N_M N_P + 4$$

$$\text{Kompleksitas} = \Omega(2N_{Pro}^2 + N_M N_P)$$

2. Kasus Terburuk  
Kasus terburuk terjadi apabila banyak Bit yang digunakan dalam representasi kromosom AG adalah 53 buah, jumlah populasi adalah  $N_{Pop}$  individu, kondisi terminasi dapat terpenuhi dalam  $N_{Itr}$  kali iterasi, dan biaya paralelisme ( $C_{P1}$  dan  $C_{P2}$ ) sangat tinggi.

$$T_{Total} = N_{Itr} \left( C_{P2} + \frac{C_R + C_{P1} + N_{Pop}(N_M N_P + 54)}{N_{Pro}} + N_{Pop}^2 + 3N_{Pop} \right)$$

Kompleksitas=

$$O \left( N_{Itr} \left( C_{P2} + \frac{N_{Pop}(N_M N_P)}{N_{Pro}} + N_{Pop}^2 \right) \right)$$

Kode semu dibawah ini menunjukkan metode AG sekuensial yang digunakan dalam penelitian ini.

#### Algoritma Genetika Sekuensial

1. Buat Populasi Inisial
2. Selama Kriteria Terminasi belum tercapai, lakukan :
  - a. Lakukan proses Rekombinasi pada populasi
  - b. Lakukan proses Mutasi pada populasi
  - c. Lakukan proses Evaluasi pada populasi
  - d. Lakukan proses Seleksi pada populasi
3. Tampilkan Solusi

Gambar 18. Kode Semu Algoritma Genetik Sekuensial.

Sedangkan kode semu dibawah ini menunjukkan metode AG paralel yang digunakan dalam penelitian ini.

**Algoritma Genetika Paralel**

**Pada Komputer Server**

1. Buat Populasi Inisial
2. Seimbangkan tugas untuk setiap prosesor dengan asumsi semua prosesor memiliki spesifikasi yang sama.
3. Selama Kriteria Terminasi belum tercapai, lakukan :
  - a. Kirim sub-sub populasi kepada tiap komputer yang ada

**Pada Komputer Server dan Client**

- b. Lakukan proses Rekombinasi pada tiap sub populasi
- c. Lakukan proses Mutasi pada tiap sub populasi
- d. Lakukan proses Evaluasi pada tiap sub populasi
- e. Kumpulkan sub-sub populasi menjadi satu populasi beserta hasil evaluasi pada Komputer Server

**Pada Komputer Server**

- f. Lakukan proses Seleksi pada populasi

4. Tampilkan Solusi

Gambar 19. Kode Semu Algoritma Genetik Paralel.

Kode semu dibawah ini menunjukkan metode rekombinasi yang digunakan dalam AG paralel.

**Rekombinasi Algoritma Genetika Paralel**

**Pada Komputer Server dan Client**

1. Tentukan Peluang Rekombinasi [0..1] ( $P_c$ )
2. Bangkitkan bilangan acak [0..1] ( $r_i$ ) untuk setiap individu ( $V_i$ ) sub populasi
3.  $i = 0$
4. Selama  $i <>$  Jumlah Sub Populasi, lakukan
  - a. Jika  $r_i < P_c \rightarrow$  Pilih  $V_i$
  - b. Jika sudah ada dua buah individu yang terpilih maka lakukan :
    - i. Cari bilangan acak [1..Panjang Kromosom Individu] ( $s$ )
    - ii. Lakukan pindah silang fragmen kromosom mulai dari lokus ke- $s$  sampai lokus terakhir pada kedua individu tersebut
  - c.  $i = i + 1$

Gambar 20. Kode Semu Rekombinasi Algoritma Genetik Paralel.

Kode semu dibawah ini menunjukkan metode mutasi yang digunakan dalam AG paralel.

**Mutasi Algoritma Genetika Paralel**

**Pada Komputer Server dan Client**

1. Tentukan Peluang Mutasi [0..1] ( $P_m$ )
2. Bangkitkan bilangan acak [0..1] ( $r_i$ ) untuk setiap gen individu ( $B_i$ )
3.  $i = 0$
4. Selama  $i <>$  (Jumlah Sub Populasi \* Panjang Kromosom Tiap Individu), lakukan
  - a. Jika  $r_i < P_m \rightarrow$  Pilih  $B_i$   
Lakukan :  
Jika  $B_i = 0 \rightarrow B_i = 1$   
Selainnya  $B_i = 0$
  - b.  $i = i + 1$

Gambar 21. Kode Semu Mutasi Algoritma Genetik Paralel.

Kode semu dibawah ini menunjukkan metode evaluasi yang digunakan dalam AG paralel.

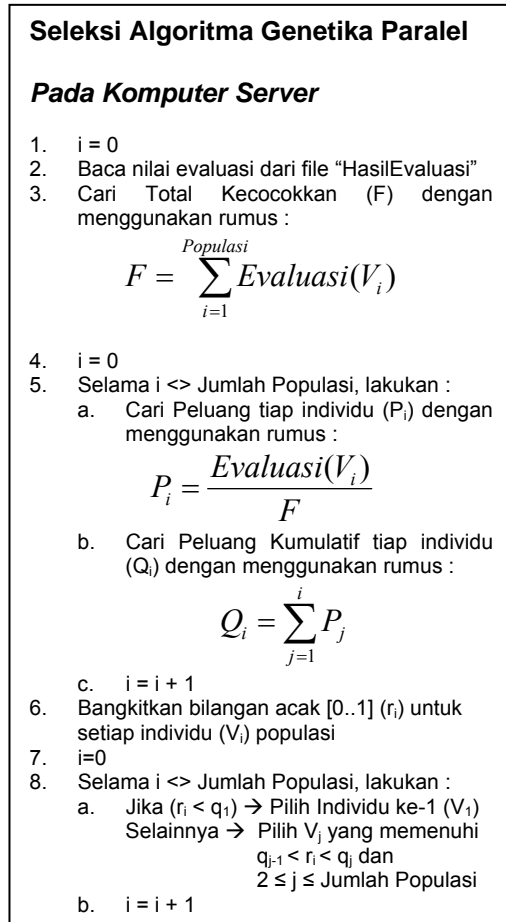
**Evaluasi Algoritma Genetika Paralel**

**Pada Komputer Server dan Client**

1.  $i = 0$
2. Selama  $i <>$  Jumlah Sub Populasi, lakukan
  - a. Evaluasi setiap individu  $V_i$  kedalam fungsi evaluasi PNN
  - b. Masukkan Nilai Evaluasi kedalam file "HasilEvaluasi"
  - c.  $i = i + 1$

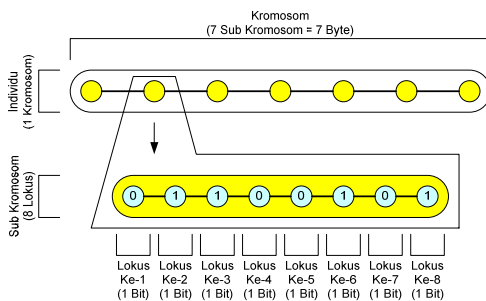
Gambar 22. Kode Semu Evaluasi Algoritma Genetik Paralel.

Kode semu dibawah ini menunjukkan metode seleksi yang digunakan dalam AG paralel.



Gambar 23. Kode Semu Seleksi Algoritma Genetik Paralel.

Secara umum, representasi tipe data untuk satu individu yang terdiri atas satu buah kromosom pada AG dapat dilihat pada gambar dibawah ini.



Gambar 24. Bagan Representasi data Kromosom, Sub Kromosom, dan Lokus.

Pendefinisian tipe data kromosom hanya menggunakan 7 Byte (56 Bit) saja. Batasan ini ada disebabkan oleh

keterbatasan efektivitas proses pengonversian tipe data *double* ke dalam tipe data *integer*. Pengonversian ini hanya efektif sebanyak 53 Bit saja. Disebabkan tipe data terkecil yang terdapat dalam bahasa C++ adalah char (1 Byte), maka banyaknya Bit yang dimungkinkan untuk merepresentasikan kromosom adalah kelipatan 8 Bit. Oleh sebab itu kromosom yang digunakan untuk menampung 53 buah gen adalah 7 Byte (56 Bit). Dengan kata lain, kromosom hanya menggunakan 53 Bit secara efektif. Sedangkan 3 Bit sisanya tidak digunakan sama sekali.

Untuk merepresentasikan individu yang terdiri atas satu buah kromosom, maka dilakukan pendefinisian suatu tipe data baru di dalam bahasa pemrograman C++. Tipe data yang dibuat ini kemudian dapat disimpan kedalam file biner guna pemrosesan lebih lanjut.

**struct ModelSubKromosom**

```
{
  unsigned char Lokus8 : 1;
  unsigned char Lokus7 : 1;
  unsigned char Lokus6 : 1;
  unsigned char Lokus5 : 1;
  unsigned char Lokus4 : 1;
  unsigned char Lokus3 : 1;
  unsigned char Lokus2 : 1;
  unsigned char Lokus1 : 1;
};
```

Gambar 25. Pendefinisian Tipe Data Untuk Representasi Sub Kromosom dalam bahasa C++.

Dari model tipe data diatas, dapat diperhatikan bahwa sesungguhnya **Model Sub Kromosom** berukuran char (1 Byte). Oleh sebab itu akan lebih baik bila dibuat suatu bentuk union untuk memudahkan proses pembuatan populasi inisial.

**union ModelGrupLokusSubKromosom**

```
{
  unsigned char PtrGrupLokus;
  ModelSubKromosom SubKromosom;
};
```

Gambar 26. Pendefinisian Tipe Data Union Untuk Pengolahan Tiap Sub Kromosom dalam bahasa C++.

Untuk mewujudkan kromosom yang terdiri atas beberapa buah sub kromosom, maka dapat dilakukan pembuatan tipe data



baru berbentuk union yang diturunkan dari tipe data larik **ModelSubKromosom**.

```

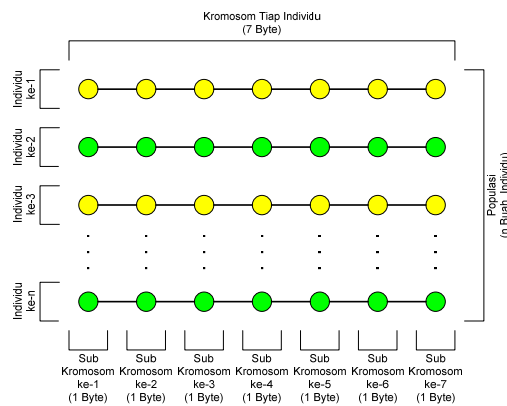
union ModelKromosom
{
  unsigned char PtrSubKromosom[7];
  ModelSubKromosom SubKromosom[7];
};

ModelKromosom *Individu;
    
```

**Gambar 27. Pendefinisian Tipe Data Union Untuk Representasi Kromosom dalam bahasa C++.**

Bila representasi kromosom untuk tiap individu telah didefinisikan dengan baik, maka kita dapat melakukan pembentukan dan penyimpanan populasi yang terdiri atas n buah individu dengan menggunakan memori ataupun dengan menggunakan file.

Penelitian ini menggunakan file untuk pembentukan dan penyimpanan populasi. Hal ini dilakukan karena penggunaan file lebih murah dan fleksibel terhadap perbedaan sumberdaya komputer (walaupun lebih lambat kinerjanya) bila dibandingkan menggunakan memori.

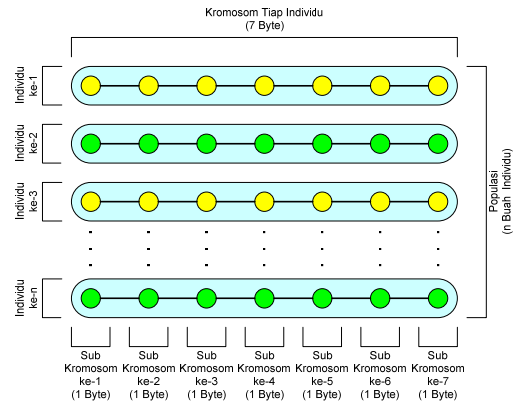


**Gambar 28. Bagan Representasi data Populasi dan Individu dalam bentuk matriks.**

Tipe data vektor MPI dapat digunakan untuk melempar paket-paket data yang berisi data-data sub populasi yang digunakan pada AG paralel. Bila diperhatikan lebih lanjut, matriks populasi dapat dipecah menjadi dua metode pelemparan vektor yaitu :

### 1. Pelemparan data vektor per baris matriks

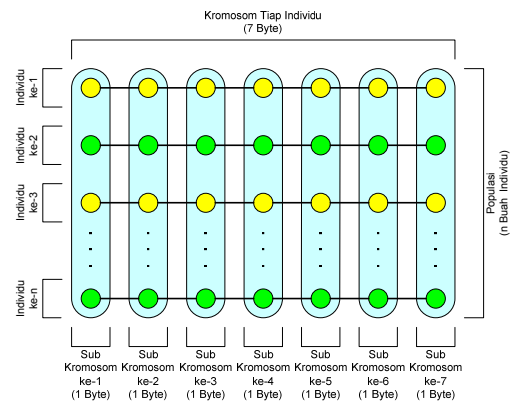
Bila menggunakan metode ini, banyaknya paket data yang dikirimkan relatif besar (n kali pelemparan pesan). Akan tetapi besar data didalam satu buah paket data cukup kecil (7 Byte).



**Gambar 29. Bagan Representasi Model Pelemparan perbaris matriks.**

### 2. Pelemparan data vektor per kolom matriks

Bila menggunakan metode ini, banyaknya paket data yang dikirimkan relatif kecil (7 kali pelemparan pesan). Akan tetapi besar data didalam satu buah paket data cukup besar (n Byte).



**Gambar 30. Bagan Representasi Model Pelemparan per kolom matriks.**

### Implementasi

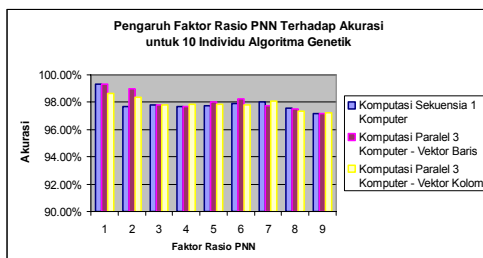
Setelah semua komponen sistem telah terpenuhi dengan baik dan benar, maka

langkah selanjutnya adalah mengimplementasikan sistem pada komputer. Implementasi sistem dilakukan dengan menggunakan perangkat lunak Microsoft Visual C++ 6.0, Microsoft Visual Basic 6.0, dan *Library* MPICH untuk Microsoft Windows NT pada komputer dengan spesifikasi prosesor Intel Pentium 4 1,7 GHz, RAM 128 MB, kapasitas Hardisk Maxtor 20 GB, dan Kartu Jaringan Realtek RTL8139 Family PCI Fast Ethernet 10 Mbps. Sedangkan sistem operasi yang digunakan adalah Microsoft Windows XP Professional. Untuk *Concenterator* jaringan digunakan Hub FlexHub/18 16 Port 10Base-T with BNC-AUI. Untuk pengujian komputasi sekuensial digunakan 3 buah komputer yang bekerja secara mandiri tanpa komunikasi satu sama lain. Sedangkan untuk pengujian komputasi paralel digunakan 3 buah komputer yang saling bahu membahu bekerja sama menyelesaikan tugas.

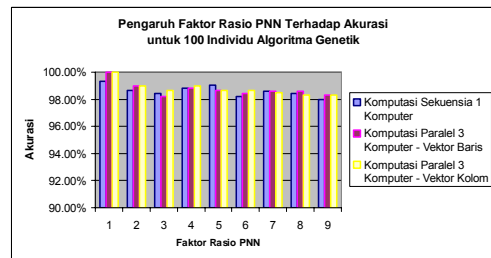
## HASIL DAN PEMBAHASAN

### Pengamatan Grafik Pengaruh Faktor Rasio PNN

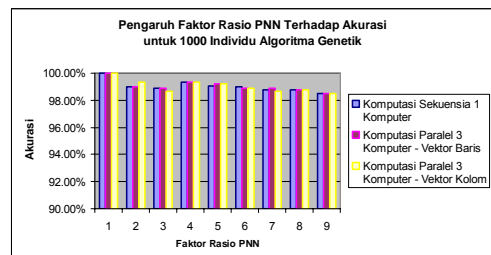
#### 1. Percobaan Pengaruh Faktor Rasio PNN terhadap Akurasi Klasifikasi PNN



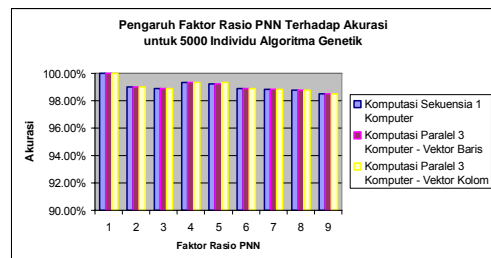
Gambar 31. Bagan Pengaruh Faktor Rasio PNN Terhadap Akurasi Klasifikasi PNN untuk 10 Individu Algoritma Genetik.



Gambar 32. Bagan Pengaruh Faktor Rasio PNN Terhadap Akurasi Klasifikasi PNN untuk 100 Individu Algoritma Genetik.



Gambar 33. Bagan Pengaruh Faktor Rasio PNN Terhadap Akurasi Klasifikasi PNN untuk 1000 Individu Algoritma Genetik.



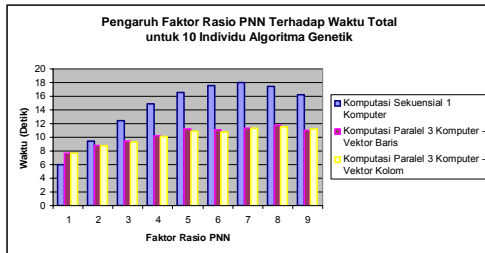
Gambar 34. Bagan Pengaruh Faktor Rasio PNN Terhadap Akurasi Klasifikasi PNN untuk 5000 Individu Algoritma Genetik.

Dari pengamatan grafik diatas terlihat bahwa pola keteraturan dari akurasi klasifikasi PNN semakin meningkat seiring dengan kenaikan jumlah populasi. Walaupun demikian setiap grafik menunjukkan pola keteraturan yang tetap, yaitu berbentuk gelombang yang turun, naik, kemudian turun kembali.

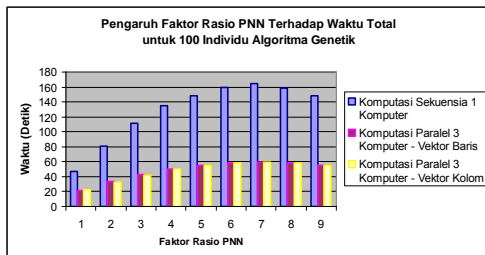
Akurasi klasifikasi PNN tertinggi selalu terjadi pada saat faktor rasio data masukan PNN 10% - data pelatihan PNN 90%. Sedangkan akurasi klasifikasi PNN terendah selalu terjadi pada saat faktor rasio data masukan PNN 90% - data pelatihan 10%.

Dari pola keteraturan grafik terlihat bahwa tingkat akurasi tidak bisa ditentukan oleh mode komputasi yang digunakan.

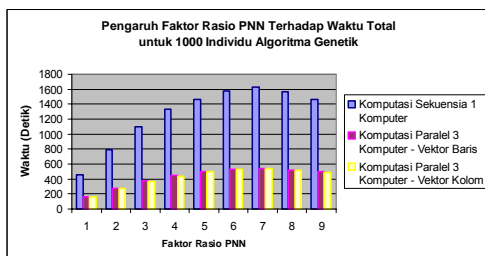
## 2. Percobaan Pengaruh Faktor Rasio PNN terhadap Waktu Total



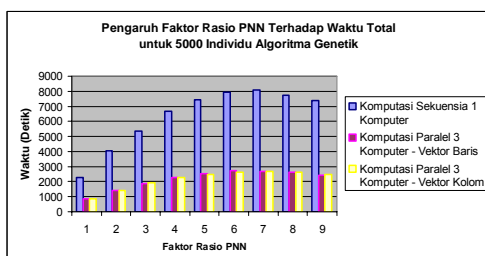
Gambar 35. Bagan Pengaruh Faktor Rasio PNN Terhadap Waktu Total untuk 10 Individu Algoritma Genetik.



Gambar 36. Bagan Pengaruh Faktor Rasio PNN Terhadap Waktu Total untuk 100 Individu Algoritma Genetik.



Gambar 37. Bagan Pengaruh Faktor Rasio PNN Terhadap Waktu Total untuk 1000 Individu Algoritma Genetik.



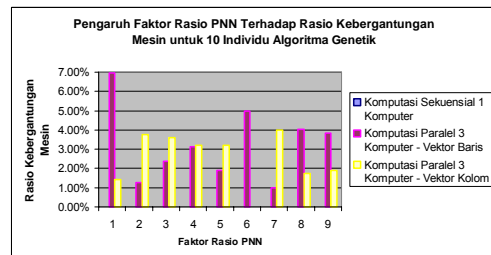
Gambar 38. Bagan Pengaruh Faktor Rasio PNN Terhadap Waktu Total untuk 5000 Individu Algoritma Genetik.

Dari pengamatan grafik diatas terlihat bahwa pola keteraturan waktu total semakin meningkat seiring dengan kenaikan jumlah populasi. Walaupun demikian setiap grafik menunjukkan pola keteraturan yang tetap, yaitu berbentuk gelombang yang naik kemudian turun kembali.

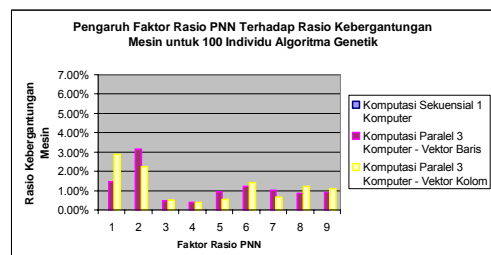
Pada grafik terlihat bahwa waktu total yang diperlukan oleh komputasi sekuensial selalu lebih besar bila dibandingkan dengan komputasi paralel, kecuali pada Gambar 35 saat faktor rasio data masukan PNN 10 % - data pelatihan PNN 90%. Anomali ini bisa terjadi diakibatkan terlalu kecilnya populasi AG yang diproses dan adanya biaya paralelisme pada proses komputasi paralel.

Dari pola grafik terlihat bahwa waktu kalkulasi antara komputasi paralel mode vektor baris dan komputasi mode vektor kolom tidak terlalu berbeda nyata.

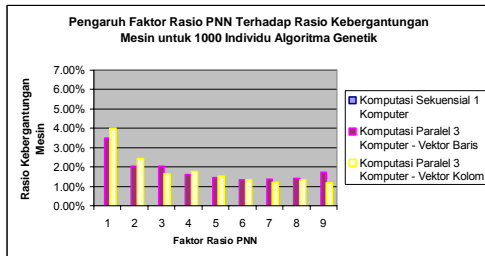
## 3. Percobaan Pengaruh Faktor Rasio PNN terhadap Rasio Kebergantungan Mesin



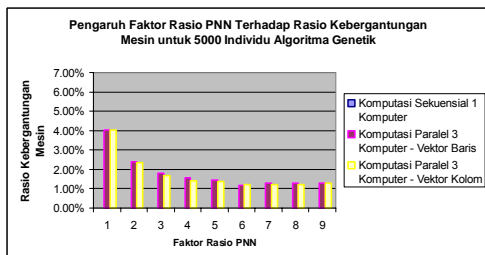
Gambar 39. Bagan Pengaruh Faktor Rasio PNN Terhadap Rasio Kebergantungan Mesin untuk 10 Individu Algoritma Genetik.



Gambar 40. Bagan Pengaruh Faktor Rasio PNN Terhadap Rasio Kebergantungan Mesin untuk 100 Individu Algoritma Genetik.



Gambar 41. Bagan Pengaruh Faktor Rasio PNN Terhadap Rasio Kebergantungan Mesin untuk 1000 Individu Algoritma Genetik.

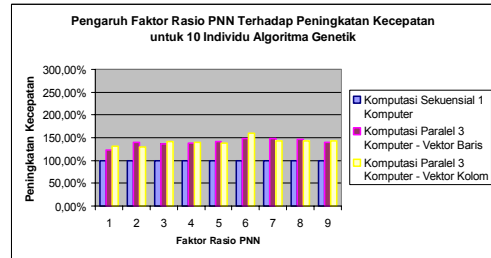


Gambar 42. Bagan Pengaruh Faktor Rasio PNN Terhadap Rasio Kebergantungan Mesin untuk 5000 Individu Algoritma Genetik.

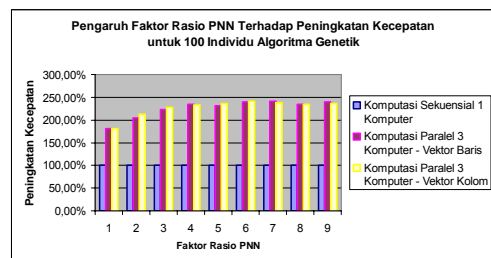
Dari pengamatan grafik diatas terlihat bahwa keteraturan rasio kebergantungan mesin belum tampak saat populasi AG 10 dan 100 individu. Keteraturan pola pada grafik mulai terlihat pada saat populasi AG 1000 dan 5000 individu. Dengan semakin bertambahnya populasi AG, bertambahnya data masukkan PNN, dan semakin kecilnya data pelatihan PNN maka keteraturan grafik cenderung berbentuk kurva logaritmik. Keteraturan ini dipengaruhi pula oleh kecepatan peningkatan waktu yang tidak imbang antara waktu kalkulasi dan waktu komunikasi. Waktu kalkulasi cenderung lebih cepat membesar bila dibandingkan dengan waktu komunikasi.

Pada pola grafik terlihat rasio kebergantungan mesin mode komputasi sekuensial sama dengan nol disebabkan oleh tidak adanya waktu komunikasi. Sedangkan rasio kebergantungan mesin antara komputasi paralel mode vektor baris dan komputasi paralel mode vektor kolom tidak terlalu berbeda nyata.

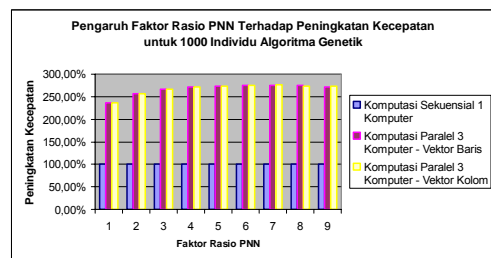
#### 4. Percobaan Pengaruh Faktor Rasio PNN terhadap Peningkatan Kecepatan



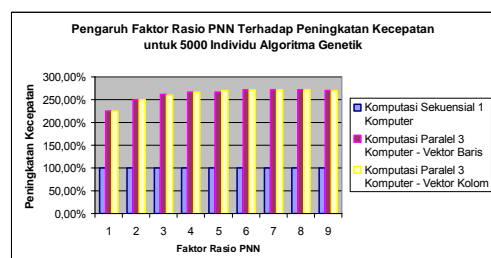
Gambar 43. Bagan Pengaruh Faktor Rasio PNN Terhadap Peningkatan Kecepatan untuk 10 Individu Algoritma Genetik.



Gambar 44. Bagan Pengaruh Faktor Rasio PNN Terhadap Peningkatan Kecepatan untuk 100 Individu Algoritma Genetik.



Gambar 45. Bagan Pengaruh Faktor Rasio PNN Terhadap Peningkatan Kecepatan untuk 1000 Individu Algoritma Genetik.



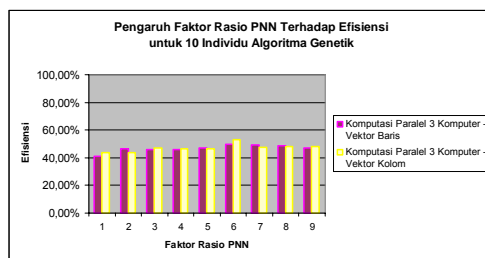
Gambar 46. Bagan Pengaruh Faktor Rasio PNN Terhadap Peningkatan Kecepatan untuk 5000 Individu Algoritma Genetik.

Dari pengamatan grafik diatas terlihat bahwa keteraturan peningkatan kecepatan semakin meningkat seiring dengan semakin bertambahnya data masukkan PNN dan semakin kecilnya data pelatihan PNN. Pola keteraturan semakin halus seiring dengan meningkatnya populasi AG.

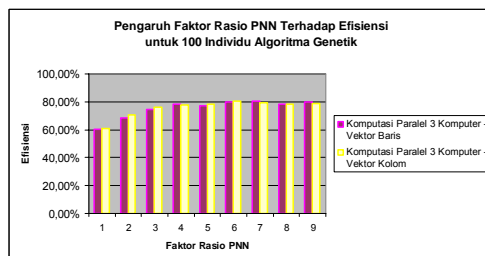
Pada grafik terlihat bahwa peningkatan kecepatan komputasi sekuensial selalu sama dengan 100% disebabkan oleh tidak adanya waktu instruksi paralel. Sedangkan peningkatan kecepatan antara komputasi paralel mode vektor baris dan komputasi paralel mode vektor kolom tidak terlalu berbeda nyata.

Peningkatan kecepatan untuk komputasi paralel tidak akan pernah mencapai angka 300%, disebabkan oleh selalu adanya fraksi rasio instruksi sekuensial terhadap keseluruhan instruksi.

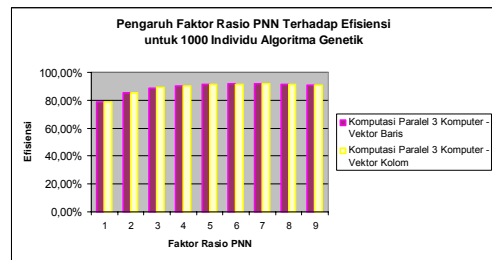
### 5. Percobaan Pengaruh Faktor Rasio PNN terhadap Efisiensi



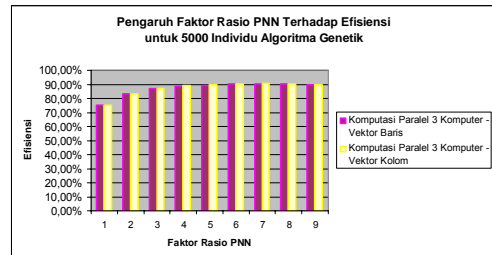
Gambar 47. Bagan Pengaruh Faktor Rasio PNN Terhadap Efisiensi untuk 10 Individu Algoritma Genetik.



Gambar 48. Bagan Pengaruh Faktor Rasio PNN Terhadap Efisiensi untuk 100 Individu Algoritma Genetik.



Gambar 49. Bagan Pengaruh Faktor Rasio PNN Terhadap Efisiensi untuk 1000 Individu Algoritma Genetik.



Gambar 50. Bagan Pengaruh Faktor Rasio PNN Terhadap Efisiensi untuk 5000 Individu Algoritma Genetik.

Dari pengamatan grafik diatas terlihat bahwa keteraturan peningkatan efisiensi paralelisme semakin meningkat seiring dengan semakin bertambahnya data masukkan PNN dan semakin kecilnya data pelatihan PNN. Keteraturan semakin halus seiring dengan meningkatnya populasi AG.

Peningkatan efisiensi paralelisme untuk komputasi paralel tidak pernah mencapai angka 100% disebabkan efisiensi tidak pernah mencapai 300%. Dari grafik terlihat bahwa peningkatan efisiensi antara komputasi paralel mode vektor baris dan komputasi paralel mode vektor kolom tidak terlalu berbeda nyata.

### Analisis eksplorasi data

Analisis eksplorasi data digunakan dalam penelitian ini untuk lebih menjelaskan statistika deskriptif yang disajikan dalam bentuk tabel. Selang kepercayaan yang digunakan oleh eksplorasi data pada penelitian ini adalah 95%.

**Tabel 2. Daftar Rataan Akurasi Klasifikasi PNN, Waktu Total, Rasio Kebergantungan Mesin, Peningkatan Kecepatan, dan Efisiensi Berdasarkan Eksplorasi Data.**

		Mode Komputasi		
		Sekuen-sial	Paralel Vektor Baris	Paralel Vektor Kolom
<b>Rataan</b>	Akurasi Klasifikasi PNN	98,60%	98,72%	98,69%
	Waktu Total	1717,39	662,77	658,59
	Rasio Kebergantungan Mesin	0,00%	2,02%	1,84%
	Peningkatan Kecepatan	100%	223,58 %	224,22 %
	Efisiensi	-	74,53%	74,74%

**Tabel 3. Daftar Ragam Akurasi Klasifikasi PNN, Waktu Total, Rasio Kebergantungan Mesin, Peningkatan Kecepatan, dan Efisiensi Berdasarkan Eksplorasi Data.**

		Mode Komputasi		
		Sekuen-sial	Paralel Vektor Baris	Paralel Vektor Kolom
<b>Ragam</b>	Akurasi Klasifikasi PNN	2,397 E-04	2,043 E-04	2,123 E-04
	Waktu Total	6734646	8803406	8618801
	Rasio Kebergantungan Mesin	0,00 E+00	9,982 E-04	8,125 E-04
	Peningkatan Kecepatan	0,000	0,282	0,279
	Efisiensi	-	3,182 E-02	3,095 E-02

**Pengujian dengan MANOVA pada Komputasi Sekuensial, Komputasi Paralel Mode Vektor Baris, dan Komputasi Paralel Mode Vektor Kolom**

Tujuan dari pengujian ini adalah untuk membantu dalam penarikan kesimpulan akhir dari data-data hasil percobaan dengan menggunakan pendekatan ragam. Tabel MANOVA untuk percobaan ini dapat dilihat pada tabel dibawah ini.

Tiga faktor perlakuan yang mengontrol percobaan adalah Faktor Mode Komputasi (3 perlakuan), Faktor rasio PNN (9 perlakuan), dan Faktor Populasi AG (4 perlakuan). Setiap faktor harus dikombinasikan satu sama lain didalam setiap percobaan. Dan didalam setiap percobaan dilakukan 10 kali perulangan. Ini ditujukan agar percobaan mendapatkan nilai hasil rata-rata yang bisa mewakili setiap keadaan yang mungkin terjadi. Jadi secara keseluruhan penarikan contoh dalam pengujian ini dilakukan sebanyak 1080 kali.

Selang kepercayaan yang digunakan dalam pengujian MANOVA adalah 95%. Jadi apabila suatu nilai uji ( $\alpha$ ) berada dibawah atau sama dengan 5% maka kesimpulan yang diambil adalah berbeda nyata. Sedangkan apabila nilai  $\alpha$  berada diatas 5%, maka kesimpulan yang diambil adalah tidak berbeda nyata

**Tabel 4. Pengaruh Faktor Perlakuan dan Interaksinya Terhadap Variabel Respon Berdasarkan Uji MANOVA pada Komputasi Sekuensial, Komputasi Paralel Vektor Baris, dan Komputasi Paralel Vektor Kolom.**

		Respon			
		Akurasi Klasifikasi PNN	Waktu Total	Rasio Kebergantungan Mesin	Peningkatan Kecepatan
<b>Faktor Perlakuan dan Interaksi</b>	Mode Komputasi	-	√	√	√
	Rasio PNN	√	√	√	√
	Populasi AG	√	√	√	√
	Interaksi Mode Komputasi & Rasio PNN	-	√	√	√
	Interaksi Mode Komputasi & Populasi AG	-	√	√	√
	Interaksi Rasio PNN & Populasi AG	-	√	-	√
	Interaksi Mode Komputasi, Rasio PNN, & Populasi AG	-	√	√	√

**Pengujian dengan MANOVA pada Komputasi Paralel Mode Vektor Baris dan Komputasi Paralel Mode Vektor Kolom**

Tabel MANOVA untuk percobaan ini dapat dilihat pada tabel dibawah ini. Tiga faktor perlakuan yang mengontrol percobaan adalah Faktor Mode Komputasi (2 perlakuan), Faktor rasio PNN (9 perlakuan), dan Faktor Populasi AG (4 perlakuan). Setiap faktor harus dikombinasikan satu sama lain didalam setiap percobaan. Dan didalam setiap percobaan dilakukan 10 kali perulangan. Ini ditujukan agar percobaan mendapatkan nilai hasil rata-rata yang bisa mewakili setiap keadaan yang mungkin terjadi. Jadi secara keseluruhan penarikan contoh dalam pengujian ini dilakukan sebanyak 720 kali.

Selang kepercayaan yang digunakan dalam pengujian MANOVA adalah 95%. Jadi apabila suatu nilai uji ( $\alpha$ ) berada dibawah atau sama dengan 5% maka kesimpulan yang diambil adalah berbeda nyata. Sedangkan apabila nilai  $\alpha$  berada diatas 5%, maka kesimpulan yang diambil adalah tidak berbeda nyata.

**Tabel 5. Pengaruh Faktor Perlakuan dan Interaksinya Terhadap Variabel Respon Berdasarkan Uji MANOVA pada Komputasi Paralel Vektor Baris dan Komputasi Paralel Vektor Kolom.**

		Respon				
		Akurasi Klasifikasi PNN	Waktu Total	Rasio Kebergantungan Mesin	Peningkatan Kecepatan	Efisiensi
Faktor Perlakuan dan Interaksi	Mode Komputasi	-	√	-	-	-
	Rasio PNN	√	√	√	√	√
	Populasi AG	√	√	√	√	√
	Interaksi Mode Komputasi & Rasio PNN	-	√	-	-	-
	Interaksi Mode Komputasi & Populasi AG	-	√	-	-	-
	Interaksi Rasio PNN & Populasi AG	-	√	-	√	√
	Interaksi Mode Komputasi, Rasio PNN, & Populasi AG	-	√	√	-	-

**KESIMPULAN**

- Komputasi paralel bisa dijadikan sebagai salah satu alternatif solusi terbaik bagi pemecahan masalah komputasional kompleks dengan waktu pemrosesan yang lebih cepat. Komputasi paralel telah membuktikan kinerja yang lebih unggul dalam hal kecepatan pemrosesan bila dibandingkan dengan komputasi sekuensial.
- Secara rata-rata kecepatan komputasi paralel mode vektor baris pada 3 buah komputer lebih cepat 2,59 kali lipat bila dibandingkan dengan komputasi sekuensial pada 1 buah komputer. Sedangkan rata-rata kecepatan komputasi paralel mode vektor kolom pada 3 buah komputer lebih cepat 2,61 kali lipat bila dibandingkan dengan

komputasi sekuensial pada 1 buah komputer.

- Akurasi klasifikasi PNN terbukti menjadi lebih baik apabila nilai parameter  $\sigma$  yang diberikan dalam PNN tepat. Dari percobaan didapatkan fakta bahwa nilai parameter  $\sigma$  yang tepat akan memberikan ketepatan klasifikasi PNN sebesar 91,11% sampai dengan 100%.
  - Dengan populasi AG yang semakin tinggi maka keakuratan klasifikasi PNN menjadi lebih tinggi walaupun waktu eksekusi menjadi lebih lambat
  - Rasio data pelatihan dengan data masukan PNN terbukti mempengaruhi keakuratan klasifikasi PNN. Secara rata-rata keakuratan klasifikasi tertinggi (99,33% s.d. 100%) terjadi saat rasio PNN 90%-10%. Sedangkan secara rata-rata keakuratan klasifikasi terendah (97,19% s.d. 98,52%) terjadi saat rasio PNN 10%-90%.
  - Pereduksian waktu pemrosesan pada komputasi paralel menjadi lebih baik apabila waktu komunikasi dan biaya paralelisme dapat ditekan sekecil mungkin. Dengan semakin kecilnya waktu komunikasi dan biaya paralelisme maka akan semakin memperbesar peningkatan kecepatan dari kinerja program.
  - Dalam uji MANOVA yang menggunakan faktor Komputasi Sekuensial, Komputasi Paralel mode vektor baris, dan Komputasi Paralel mode vektor kolom terlihat bahwa:
    - 1) Mode komputasi berpengaruh nyata pada :
      - a. Waktu Total
      - b. Rasio Kebergantungan Mesin
      - c. Peningkatan Kecepatan
      - d. Efisiensi
    - 2) Rasio PNN berpengaruh nyata pada :
      - a. Akurasi Klasifikasi PNN
      - b. Waktu Total
      - c. Rasio Kebergantungan Mesin
      - d. Peningkatan Kecepatan
      - e. Efisiensi
- 3) Populasi AG berpengaruh nyata pada :
    - a. Akurasi Klasifikasi PNN
    - b. Waktu Total
    - c. Rasio Kebergantungan Mesin
    - d. Peningkatan Kecepatan
    - e. Efisiensi
  - 4) Interaksi Mode komputasi dan Rasio PNN berpengaruh nyata pada :
    - a. Waktu Total
    - b. Rasio Kebergantungan Mesin
    - c. Peningkatan Kecepatan
    - d. Efisiensi
  - 5) Interaksi Mode komputasi dan Populasi AG berpengaruh nyata pada :
    - a. Waktu Total
    - b. Rasio Kebergantungan Mesin
    - c. Peningkatan Kecepatan
    - d. Efisiensi
  - 6) Interaksi Rasio PNN dan Populasi AG berpengaruh nyata pada :
    - a. Waktu Total
    - b. Peningkatan Kecepatan
    - c. Efisiensi
  - 7) Interaksi Mode Komputasi, Rasio PNN dan Populasi AG berpengaruh nyata pada :
    - a. Waktu Total
    - b. Rasio Kebergantungan Mesin
    - c. Peningkatan Kecepatan
    - d. Efisiensi
- Dalam uji MANOVA yang menggunakan faktor Komputasi Paralel mode vektor baris dan Komputasi Paralel mode vektor kolom terlihat bahwa :
    - 1) Mode komputasi berpengaruh nyata pada Waktu Total saja
    - 2) Rasio PNN berpengaruh nyata pada :
      - a. Akurasi Klasifikasi PNN
      - b. Waktu Total
      - c. Rasio Kebergantungan Mesin
      - d. Peningkatan Kecepatan
      - e. Efisiensi
    - 3) Populasi AG berpengaruh nyata pada :
      - a. Akurasi Klasifikasi PNN
      - b. Waktu Total
      - c. Rasio Kebergantungan Mesin
      - d. Peningkatan Kecepatan
      - e. Efisiensi



- 4) Interaksi Mode komputasi dan Rasio PNN berpengaruh nyata pada Waktu Total saja.
  - 5) Interaksi Mode komputasi dan Populasi AG berpengaruh nyata pada Waktu Total saja.
  - 6) Interaksi Rasio PNN dan Populasi AG berpengaruh nyata pada :
    - a. Waktu Total
    - b. Peningkatan Kecepatan
    - c. Efisiensi
  - 7) Interaksi Mode Komputasi, Rasio PNN dan Populasi AG berpengaruh nyata pada :
    - a. Waktu Total
    - b. Rasio Kebergantungan Mesin
- Perbedaan kinerja komputasi paralel mode vektor baris dan kinerja komputasi mode vektor kolom tidak terlalu jauh berbeda. Oleh sebab itu, dalam pengembangan sistem komputasi paralel dapat menggunakan metode pelemparan data mode baris maupun mode kolom secara bebas.
  - Untuk meningkatkan kinerja pada komputasi paralel, sebaiknya *setting up* jaringan ditata dengan lebih baik lagi. Sebaiknya dilakukan berbagai upaya untuk meminimalisir efek interferensi bila menggunakan pengkabelan UTP dan HUB.

## SARAN

- Untuk meningkatkan akurasi dari pencatatan waktu, sebaiknya digunakan fungsi pencatatan waktu yang bisa mencatat perubahan waktu sepersekian detik. Dengan semakin tingginya akurasi pencatatan waktu maka akan memberikan tingkat ketelitian penghitungan yang lebih baik, tepat, dan benar pada Rasio Kebergantungan Mesin, Peningkatan Kecepatan, dan Efisiensi .
  - Pada penelitian ini hanya digunakan data bunga IRIS dan parameter penghalus ( $\sigma$ ) yang homogen pada PNN. Untuk penelitian selanjutnya dapat digunakan pemodelan PNN yang lebih kompleks dengan melibatkan data dan parameter penghalus ( $\sigma$ ) yang heterogen.
- Pada penelitian ini hanya digunakan nilai peluang mutasi AG, peluang rekombinasi AG, interval ketelitian AG, dan jumlah generasi AG yang selalu tetap pada setiap percobaan. Untuk menghasilkan data percobaan yang lebih kompleks dan menyeluruh, maka akan lebih baik bila parameter AG pada percobaan diubah-ubah.

## DAFTAR PUSTAKA

- Flynn.** 1966. Tanpa Judul. <http://csep1.phy.ornl.gov/ca/node11.html> [11-12-2002]
- Fu, Limin.** 1994. *Neural Networks in Computer Intelligence*. ISBN 0-07-113319-4. Mc Graw-Hill. Singapura.
- Lee, D. X, G. Thoma, & H. Weschler.** 2004. "*Classification of Binary Document Images into Texture or Non-textual Data Blocks Using Neural Network Models*". [http://archive.nlm.nih.gov/pubs/doc\\_class/mv.php](http://archive.nlm.nih.gov/pubs/doc_class/mv.php). [09-08-2004]
- Michalewicz, Zbigniew.** 1995. *Genetic Algorithms + Data Structures = Evolution Programs*. ISBN 3-540-60676-9. Cetakan ketiga. Springer. Jerman.
- Moore, David S.** 1994. *The Basic Practice of Statistics*. ISBN 0-7167-2628-9. W. H. Freeman and Company. New York Amerika Serikat
- Pacifico, Mark & Mike Merrill.** 1998. Tanpa Judul. <http://www.cs.umd.edu/class/fall2001/cm411/projects/parallel2/proposal.html>. [11-12-2002]
- Quinton, Reggers.** 1997. Tanpa Judul. <http://www.uwo.ca/its/doc/courses/notes/socket/#INTRO>. [11-8-2003]
- Rahmani.** 2004. *Probabilistic Neural Networks*. <http://courses.iust.ac.ir/~rahmani/NN/pnn.htm>. [09-08-2004]

**Rish, I.** 2004. "An empirical study of the naive Bayes classifier".  
<http://www.intellektik.informatik.tu-darmstadt.de/~tom/IJCAI01/Rish.pdf>.  
[09-08-2004]

**Tandriarto, Naviansony.** 2002.  
"Penerapan Algoritme Genetik pada Travelling Salesman Problem".  
Departemen Ilmu Komputer Fakultas Matematika & Ilmu Pengetahuan Alam IPB. Indonesia.

**Toonen, Karen D.** 1998. Tanpa Judul.  
<http://www.fp.mcs.anl.gov/~lusk/papers/knoxville/node4.html>. [10-8-2003]

**Yuan, Shi.** 2003. Tanpa Judul. cis.  
<http://www.temple.edu/~shi/docs/amdahl/amdahl.html>. [11-12-2002]