

SKRIPSI

**PENERAPAN ALGORITMA GENETIKA DAN JARINGAN SYARAF
TIRUAN DALAM PENJADWALAN MATA KULIAH DI FAKULTAS
MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA**

***IMPLEMENTATION OF GENETIC ALGORITHM AND ARTIFICIAL
NEURAL NETWORK IN LECTURE SCHEDULING IN FACULTY OF
MATHEMATICS AND NATURAL SCIENCE
UNIVERSITAS GADJAH MADA***



**EKO BUDI PRASETYO
12/336144/PA/15079**

**PROGRAM STUDI ILMU KOMPUTER
JURUSAN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2014

SKRIPSI

**PENERAPAN ALGORITMA GENETIKA DAN JARINGAN SYARAF
TIRUAN DALAM PENJADWALAN MATA KULIAH DI FAKULTAS
MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA**

***IMPLEMENTATION OF GENETIC ALGORITHM AND ARTIFICIAL
NEURAL NETWORK IN LECTURE SCHEDULING IN FACULTY OF
MATHEMATICS AND NATURAL SCIENCE
UNIVERSITAS GADJAH MADA***

Diajukan untuk memenuhi salah satu syarat memperoleh derajat
Sarjana Komputer



**EKO BUDI PRASETYO
12/336144/PA/15079**

**PROGRAM STUDI ILMU KOMPUTER
JURUSAN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2014

HALAMAN PENGESAHAN

SKRIPSI

PENERAPAN ALGORITMA GENETIKA DAN JARINGAN SYARAF TIRUAN DALAM PENJADWALAN MATA KULIAH DI FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM UNIVERSITAS GADJAH MADA

Telah dipersiapkan dan disusun oleh

EKO BUDI PRASETYO

12/336144/PA/15079

Telah dipertahankan di depan Tim Penguji
pada tanggal 15 Desember 2014

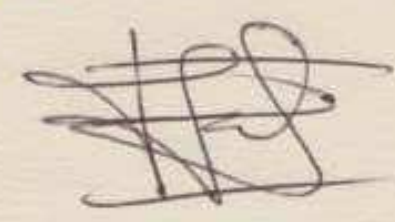
Susunan Tim Penguji



Faizah, S.Kom., M.Kom.
Pembimbing



Anifuddin Aziz, S.Si., M.Kom.
Penguji I



Retantyo Wardoyo, Drs., M.Sc., Ph.D.
Penguji II

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 21 November 2014



EKO BUDI PRASETYO

PRAKATA

Segala puji dan syukur kehadiran Allah SWT atas berkah limpahan rahmat dan karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik.

Banyak arahan, bantuan, serta dukungan dari berbagai pihak yang diberikan kepada penulis dalam rangka penyelesaian skripsi ini. Maka dari itu pada kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Ayah, Ibu, dan Adik penulis yang selalu mendoakan dan mendukung perjuangan penulis selama ini.
2. Ibu Sukati selaku nenek penulis yang telah banyak memberikan inspirasi perjuangan bagi penulis.
3. Ibu Faizah, S.Kom., M.Kom., selaku dosen pembimbing yang telah banyak membantu dan memberi bimbingan kepada penulis dalam menyelesaikan skripsi ini.
4. Bapak Anifuddin Aziz, S.Si., M.Kom., selaku dosen penguji I sidang skripsi penulis.
5. Bapak Retantyo Wardoyo, Drs., M.Sc., Ph.D., selaku dosen penguji II sidang skripsi penulis.
6. Bapak Janoe Hendarto, Drs., M.Kom., selaku dosen wali akademik atas bimbingan dan arahan selama penulis menempuh pendidikan S1 Ilmu Komputer.
7. Seluruh dosen pengajar serta civitas akademika di lingkungan fakultas MIPA UGM, khususnya program studi ilmu komputer.
8. Rekan-rekan seperjuangan Ilmu Komputer alih jenis angkatan 2012 (Bayu, Ani, Uzun, Joko) serta teman-teman alih jenis angkatan 2011 dan 2013.
9. Teman-teman kos Nusantara dan teman-teman Gamatechno divisi enterprise dan divisi egov.

10. Pihak-pihak lain yang telah membantu dalam penyelesaian skripsi ini dan tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa dalam skripsi ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik yang bersifat membangun sangat penulis harapkan. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pembaca.

Yogyakarta, 22 Desember 2014



Penulis

ABSTRACT

Implementation of Genetic Algorithm and Artificial Neural Network in Lecture Scheduling in Faculty of Mathematics and Natural Science Universitas Gadjah Mada

Eko Budi Prasetyo
12/336144/PA/15079

Lecture scheduling in faculty of Matematika dan Ilmu Pengetahuan Alam (MIPA) Universitas Gadjah Mada (UGM) is usually performed by faculty through delegation of each study program. So far its process takes a long time because after the schedule was established the faculty still need to synchronize for all schedules that have been given. Moreover, the limitations of time and room in scheduling can allow collision of lecture and incompatibility between the room capacity and the number of student who take the lecture.

One of the way to get over the scheduling problem is by using genetic algorithm and artificial neural network. Artificial neural network through *backpropagation* method can be used in the prediction of the number of lecture participants that is used for the determination of room. Genetic algorithms will be used to determine the optimal solution of some generated solutions schedule.

This research develop an *web* based application which capable to predict the number of lecture participants that have a role in determining lecture room and capable to scheduling the lecture in “one door”. The implementation of genetic algorithm and artificial neural network with appropriate parameters can produce optimal lecture schedule. Furthermore time required in scheduling process become faster.

Keyword: Lecture Scheduling, *Backpropagation*, Genetic Algorithm, Artificial Neural Network.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Algoritma adalah serangkaian langkah-langkah untuk menyelesaikan suatu permasalahan. Algoritma genetika merupakan algoritma yang digunakan dalam teknik pencarian *evolutionary computation* yang menemukan perkiraan solusi untuk optimasi dan masalah pencarian. Algoritma genetika termasuk algoritma heuristik (solusi tidak harus tepat namun mendekati) dan stokastik (menerapkan metode pengacakan pada proses pembangkitan individu dan seleksinya). Algoritma genetika sendiri telah banyak digunakan untuk permasalahan yang sulit (misalkan pada permasalahan *NP-complete*), untuk pembelajaran mesin dan juga untuk mengembangkan program sederhana (Deepa dkk., 2007). Salah satu penerapan untuk permasalahan yang sulit yaitu dalam permasalahan penjadwalan mata kuliah pada fakultas di perguruan tinggi.

Perguruan tinggi terdiri dari banyak fakultas yang di dalamnya terdapat berbagai macam program studi dan disiplin ilmu. Dalam pelaksanaan kegiatan perkuliahan, dibutuhkan banyak sumber daya pengajaran terutama dosen, waktu, dan ruang. Namun dalam kenyataannya jumlah sumber daya yang ada tersebut jumlahnya sangat terbatas. Oleh karena itu, untuk memanfaatkan sumber daya tersebut dengan optimal dibutuhkan penjadwalan yang baik sesuai dengan aturan atau batasan yang telah ditetapkan agar tidak terjadi permasalahan yang menyebabkan jadwal tidak dapat dilaksanakan. Hal ini menyebabkan pembuatan jadwal mata kuliah merupakan masalah klasik yang sulit untuk diselesaikan (Weare dkk., 1995).

Penjadwalan di fakultas Matematika dan Ilmu Pengetahuan Alam (MIPA) Universitas Gadjah Mada (UGM) telah dilakukan sebelum periode pengisian

Kartu Rencana Studi (KRS) di masing-masing program studi setiap semester. Proses pertama yang dilakukan adalah dengan membuat suatu rancangan jadwal yang mengalokasikan ruang dari blok program studi yang disediakan oleh fakultas dan juga alokasi dosen untuk mengampu mata kuliah yang terselenggara sesuai dengan kurikulum yang berlaku. Kemudian ditentukan ruang untuk masing-masing mata kuliah dengan cara melakukan perkiraan jumlah peserta berdasarkan jumlah mahasiswa yang mengambil mata kuliah yang sama pada tahun lalu. Hasil jadwal dari masing-masing program studi lalu dikumpulkan dan diverifikasi melalui rapat pada jurusan yang kemudian diserahkan kepada sekretariat fakultas MIPA UGM.

Hasil jadwal dari masing-masing jurusan yang diserahkan kepada sekretariat MIPA UGM biasanya belum sepenuhnya baik, karena ada kemungkinan kapasitas ruang yang besar dialokasikan untuk mahasiswa yang jumlahnya sedikit. Maka dari itu, setelah jadwal selesai dibuat biasanya akan diproses ulang oleh sekretariat MIPA UGM dengan melakukan penukaran ruang pada kelas mata kuliah agar sisa tempat duduk yang tidak terpakai menjadi seminimal mungkin. Permasalahan lain pada penjadwalan muncul pada saat pengisian KRS, yaitu ketika mahasiswa akan mengulang mata kuliah pilihan di semester tahun lalu tetapi tidak dapat dilakukan karena terjadi tumbukan waktu dengan mata kuliah wajib pada paket semester dan program studi yang sama.

Penelitian ini bermaksud untuk mengembangkan aplikasi guna menghasilkan jadwal mata kuliah sesuai dengan aturan atau batasan dalam penjadwalan yang ada pada fakultas MIPA UGM yang dapat digunakan pada masa pengisian KRS dengan penentuan ruang berdasarkan hasil prediksi jumlah peserta mata kuliah. Selain itu, penelitian ini diharapkan mampu menjadikan sistem “satu pintu” dalam pembuatan jadwal di fakultas MIPA UGM serta dapat memanfaatkan semua sumber daya ruang dan waktu yang ada secara optimal. Dengan adanya aplikasi ini diharapkan juga dapat mempercepat proses pembuatan

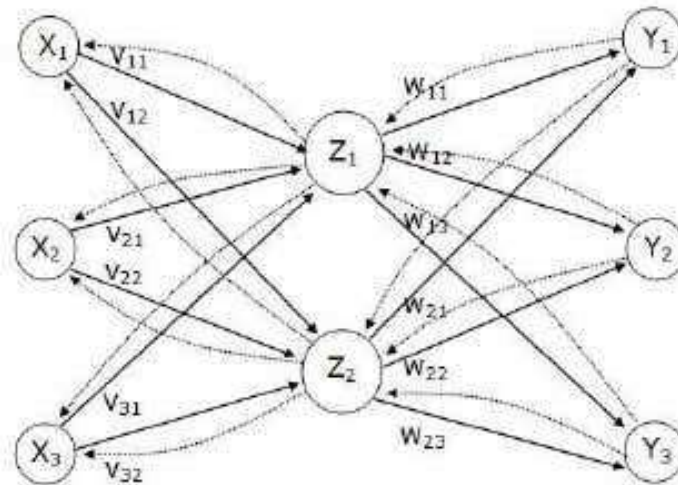
mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah *competitive*, *hebbian*, *kohonen*, *LVQ (Learning Vector Quantization)*, dan *neocognitron*.

c. *Hybrid learning* (pembelajaran hibrida).

Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*, sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah algoritma RBF.

3.2.6 Backpropagation

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*feed-forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi *sigmoid* (Puspitaningrum, 2006). Arsitektur jaringan *backpropagation* ditunjukkan pada gambar 3.15.



Gambar 3.15 Arsitektur jaringan *backpropagation*

Algoritma *backpropagation* (Puspitaningrum, 2006):

- a) Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
- b) Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai *false*:
 1. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Propagasi maju:

- a. Tiap-tiap unit *input* (X_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Tiap-tiap unit tersembunyi (Z_j , $j=1,2,3,\dots,p$) menjumlahkan sinyal-sinyal *input* terbobot, serta menambahkan bias v_{0j} jika diketahui. Proses tersebut ditunjukkan seperti pada persamaan 3.7.

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (3.7)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya seperti yang ditunjukkan pada persamaan 3.8.

$$z_j = f(z_in_j) \quad (3.8)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

- c. Tiap-tiap unit *output* (Y_k , $k=1,2,3,\dots,m$) menjumlahkan sinyal-sinyal *input* terbobot. Proses tersebut ditunjukkan seperti pada persamaan 3.9.

$$y_{in_k} = w_{0k} + \sum_{i=1}^p z_i w_{jk} \quad (3.9)$$

gunakan fungsi aktivasi untuk menghitung sinyal *output*-nya seperti yang ditunjukkan oleh persamaan 3.10.

$$y_k = f(y_{in_k}) \quad (3.10)$$

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Propagasi mundur:

- d. Tiap-tiap unit *output* (Y_k , $k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola *input* pembelajaran, hitung informasi *error*-nya seperti yang ditunjukkan oleh persamaan 3.11.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad (3.11)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}) seperti yang ditunjukkan oleh persamaan 3.12.

$$\Delta w_{jk} = \alpha \delta_k z_j \quad (3.12)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k}) seperti yang ditunjukkan oleh persamaan 3.13.

$$\Delta w_{0k} = \alpha \delta_k \quad (3.13)$$

kirimkan δ_k ini ke unit-unit yang ada di lapisan bawahnya.

- e. Tiap-tiap unit tersembunyi (Z_j , $j = 1,2,3,\dots,p$) menjumlahkan delta *input*-nya (dari unit-unit yang berada pada lapisan di atasnya) seperti yang ditunjukkan oleh persamaan 3.14.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (3.14)$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error* seperti yang ditunjukkan oleh persamaan 3.15.

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (3.15)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}) seperti yang ditunjukkan oleh persamaan 3.16.

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (3.16)$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}) seperti yang ditunjukkan oleh persamaan 3.17.

$$\Delta v_{0j} = \alpha \delta_j \quad (3.17)$$

- f. Tiap-tiap unit *output* (Y_k , $k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,\dots,p$) seperti yang ditunjukkan oleh persamaan 3.18.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (3.18)$$

Tiap-tiap unit tersembunyi (Z_j , $j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i = 0,1,2,\dots,n$) seperti yang ditunjukkan oleh persamaan 3.19.

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (3.19)$$

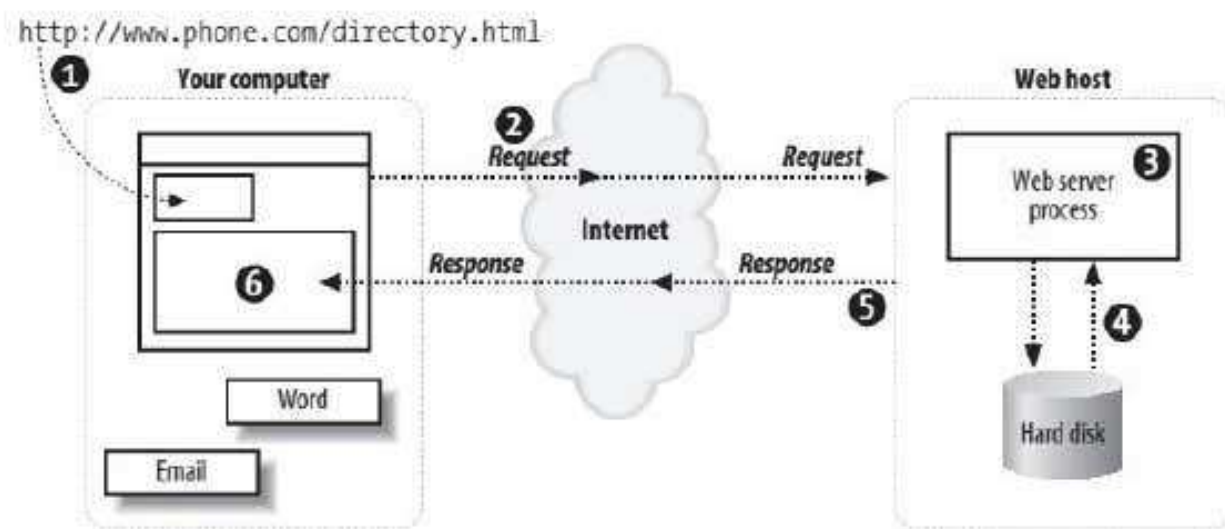
2. Tes kondisi berhenti apabila *error* ditemukan.

Untuk memeriksa *error* digunakan *Mean Search Error* (MSE) seperti yang ditunjukkan pada persamaan 3.20.

$$mse = \frac{\sum_{i=1}^n (target - output)^2}{2} \quad (3.20)$$

3.3 PHP

PHP adalah bahasa pengembangan *web* yang ditulis oleh dan untuk pengembang web. PHP singkatan dari *Hypertext Preprocessor*. PHP adalah bahasa *scripting server-side*, yang dapat tertanam dalam HTML atau digunakan sebagai biner mandiri (Converse dkk., 2004). Pemrosesan PHP pada server disebut dengan *server-side processing*. Proses tersebut dapat diilustrasikan seperti yang ditunjukkan pada gambar 3.16.



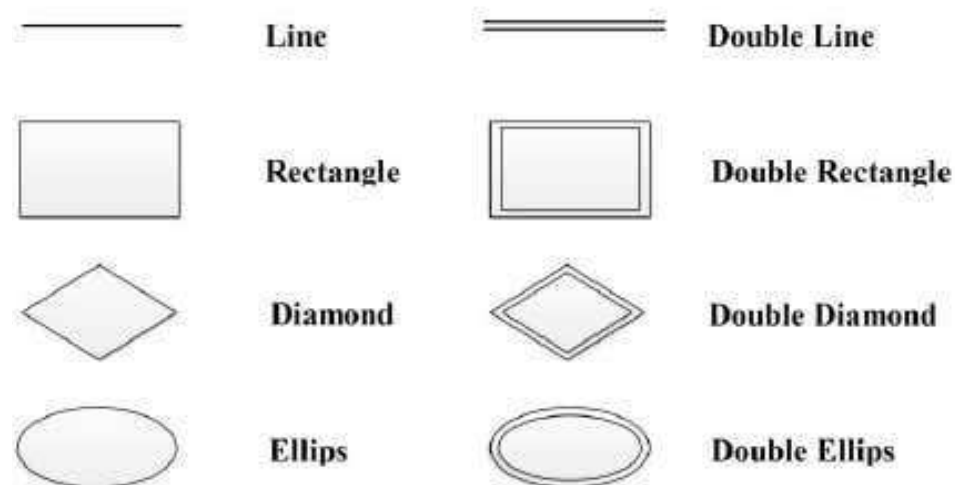
Gambar 3.16 Server-side processing (Davis dan Philips, 2007)

Pemrosesan PHP pada *server* atau *server-side processing* dimulai dengan adanya pengelompokan dari alamat web yang dimasukkan pada *browser* untuk mendapatkan nama berkas yang diminta dan alamat *server* yang dituju. Kemudian klien mengirimkan *request* atau permintaan kepada *web server*. *Request* yang dikirim tersebut kemudian diproses oleh *web server process* dan dilakukan pencarian terhadap berkas yang diminta. Kemudian *web server* membaca berkas yang terdapat pada *hard disk* dari *web server*. Hasil pembacaan berkas tersebut kemudian dikembalikan dalam bentuk *response* ke *browser* pada klien. *Web*

browser menggunakan markup HTML yang dikembalikan dari *web server* untuk menampilkan halaman *web* di layar komputer.

3.4 Diagram E-R

Diagram E-R merupakan diagram pemodelan data yang didasarkan pada persepsi pada dunia nyata yang terdiri dari kumpulan dasar objek yang disebut entitas dan hubungan antara objek-objek tersebut (Pallaw, 2010). Diagram terdiri dari *entity*, *relationship*, dan *attribute*. *Entity* merupakan penyimpan data yang namanya dapat merepresentasikan suatu kelas atau tipe. *Relationship* merupakan hubungan antara entitas yang biasanya dilambangkan dengan frase kata kerja. *Attribute* merupakan properti atau karakteristik dari suatu *entity* (Bagui dan Earp, 2003). Diagram E-R dapat tersusun dari komponen-komponen seperti yang tampak pada gambar 3.17.



Gambar 3.17 Komponen diagram E-R (Pallaw, 2010)

Berikut penjelasan dari gambar 3.17.

a. Line

Merupakan penghubung yang digunakan untuk menghubungkan *attribute* ke *entity set* dan penghubung antar *entity*.

b. Rectangle

Merepresentasikan *entity set*.

c. Diamond

Merepresentasikan hubungan antara *entity set*.

d. *Ellips*

Merepresentasikan *attribute*.

e. *Double Line*

Merepresentasikan partisipasi total dari sebuah *entity* pada suatu hubungan.

f. *Double Rectangle*

Merepresentasikan *entity set* lemah, yakni *entity* dimana keberadaan dari *entity* tersebut tergantung dari keberadaan *entity* lain.

g. *Double Diamond*

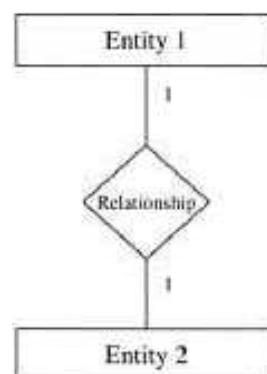
Merepresentasikan bahwa pada suatu hubungan terdapat *entity* lemah yang terlibat.

h. *Double Ellips*

Merepresentasikan *multi valued attribute*, yakni nilai dari suatu *attribute* yang mempunyai lebih dari satu nilai (*multi value*) dari *attribute* yang bersangkutan.

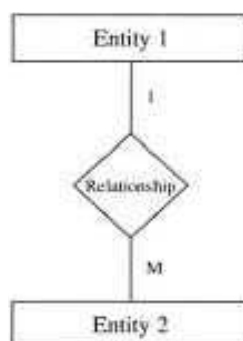
Menurut Bagui dan Earp (2003), *relationship* terbagi menjadi 4 tipe yaitu:

- 1) *One-to-one*: Pada *relationship* jenis ini, tiap satu anggota himpunan pada *entity* terhubung dengan satu anggota himpunan pada *entity* lain dan sebaliknya. Representasi dari *relationship one-to-one* ditunjukkan pada gambar 3.18.



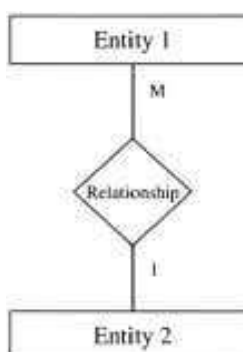
Gambar 3.18 One-to-one (Earp dkk., 2003)

- 2) *One-to-many*: Pada *relationship* jenis ini, satu anggota himpunan pada *entity* terhubung dengan banyak anggota himpunan pada *entity*. Representasi dari *relationship one-to-many* ditunjukkan pada gambar 3.19.



Gambar 3.19 *One-to-many* (Earp dkk., 2003)

- 3) *Many-to-one*: Pada *relationship* jenis ini, banyak anggota himpunan *entity* terhubung dengan satu anggota himpunan pada *entity* lain. Representasi dari *relationship many-to-one* ditunjukkan pada gambar 3.20.



Gambar 3.20 *Many-to-one* (Earp dkk., 2003)

- 4) *Many-to-many*: Pada *relationship* jenis ini, banyak anggota himpunan pada *entity* dapat terhubung dengan banyak anggota himpunan pada *entity* lain dan sebaliknya. Representasi dari *relationship many-to-many* ditunjukkan pada gambar 3.21.

menjadi 2 sks dan 1 sks. Maka pertemuan kelas tersebut diselenggarakan pada hari yang berbeda.

7. Kelas mata kuliah wajib yang berdekatan jenis semesternya (ganjil/genap) pada paket semester yang sama dan program studi yang sama, tidak diadakan pada waktu yang sama. Mata kuliah wajib memiliki prioritas dalam penyelenggaraannya daripada mata kuliah pilihan, karena mata kuliah wajib adalah mata kuliah yang harus diselesaikan dalam masa studi. Oleh karena itu mata kuliah wajib dijadwalkan pada waktu yang tidak bertabrakan satu sama lain.
8. Kelas yang memiliki mata kuliah yang sama (kelas paralel) diadakan pada waktu yang sama. Mata kuliah yang dibagi menjadi beberapa kelas, maka diselenggarakan dalam waktu yang sama. Hal ini digunakan untuk mempermudah pembuatan jadwal ujian pada kelas mata kuliah yang sama diadakan juga pada waktu yang sama.
9. Kelas kemipaan dari mata kuliah bersama diadakan pada sesi pertama dan dijadwalkan secara merata harinya. Kelas kemipaan terbentuk dari mata kuliah bersama yang biasanya terdiri dari beberapa kelas sekaligus. Sehingga penjadwalan waktunya diambil di sesi pertama dan disebar secara merata pada jadwal mingguannya. Misalnya kelas kemipaan terdiri dari 8 kelas, maka penjadwalannya tiap 2 kelas ditempatkan pada hari yang sama demikian seterusnya dari senin hingga kamis, maka selama 4 hari tersebut akan ada 2 kelas kemipaan pada jam yang sama di ruang yang berbeda.

4.1.4 Aturan khusus penjadwalan mata kuliah

Penjadwalan memiliki aturan yang tidak wajib dipenuhi namun jika dipenuhi maka semakin baik kualitas jadwal yang dihasilkan. Aturan-aturan tersebut dinamakan aturan khusus. Berdasarkan wawancara yang dilakukan

kepada Ibu Erri Ismiyati, S.I.P selaku narasumber dari fakultas MIPA, terdapat beberapa aturan khusus yang dipakai yakni sebagai berikut:

1. Kelas mata kuliah pilihan dengan kelas mata kuliah wajib pada program studi yang sama dan paket semester yang berdekatan jenis semesternya (ganjil/genap), tidak diadakan pada waktu yang sama. Hal ini digunakan mengingat banyaknya mahasiswa yang sering mengulang mata kuliah pilihan pada semester tahun sebelumnya namun kadang tidak dapat diambil karna bertabrakan waktunya dengan mata kuliah wajib pada semester saat mahasiswa sedang mengisi Kartu Rencana Studi (KRS).
2. Kelas mata kuliah diadakan pada ruang yang sesuai dengan program studi. Terdapat kesepakatan di fakultas, untuk membagi ruang yang tersedia kepada program studi untuk digunakan sebagai kelas perkuliahan. Namun kesepakatan ini tidak harus terpenuhi, karena kemungkinan dapat terjadi seperti kasus ruang dengan kapasitas 100 ditempati mahasiswa yang hanya berjumlah 10 orang. Sehingga perlu dialokasikan ke ruang yang lainnya yang kapasitasnya mendekati jumlah mahasiswa. Dan ruang yang diambil bisa dari ruang milik program studi lainnya.
3. Kelas mata kuliah yang memiliki paket yang sama maksimal berjumlah 8 sks dalam sehari. Hal ini dilakukan agar mahasiswa tidak merasa jenuh dalam mengikuti kegiatan perkuliahan dalam sehari.
4. Ruang kelas ditempati minimal sejumlah prosentase dari kapasitasnya. Biasanya fakultas memiliki kriteria perkiraan kapasitas ruang kelas minimal yang harus terpenuhi agar ruang terpakai secara optimal.
5. Waktu kelas mata kuliah diadakan sesuai dengan rekomendasi waktu dosen pengajar yang bersangkutan. Dosen pengajar kadangkala memiliki aktifitas yang padat di luar tugasnya dalam menjalankan perkuliahan. Sehingga dosen dapat memberi rekomendasi waktu jadwal sesuai yang diinginkan.

4.1.5 Kriteria mata kuliah

Pada dasarnya penjadwalan mata kuliah diimplementasikan dengan pemilihan ruang dan waktu yang terdapat pada fakultas sesuai dengan aturan yang telah ditetapkan. Pemilihan ruang hanya dapat dilakukan jika ruang tersebut ada dan disediakan oleh fakultas. Maka pada proses penjadwalan, mata kuliah yang dijadwalkan adalah mata kuliah yang dicalonkan untuk menempati ruang-ruang yang telah disediakan fakultas sesuai kriteria yang ditentukan. Terdapat beberapa kriteria mata kuliah yang tidak termasuk dalam penjadwalan jika diamati dari masing-masing karakter mata kuliah tersebut antara lain:

1. Mata kuliah praktikum. Mata kuliah praktikum diselenggarakan pada ruang praktikum yang telah disediakan khusus oleh fakultas dan tidak termasuk ke dalam referensi ruang yang digunakan dalam penjadwalan.
2. Mata kuliah seminar. Mata kuliah seminar seperti seminar tugas akhir biasanya telah disediakan ruang khusus seperti ruang siding.
3. Mata kuliah tugas akhir / tugas khusus / skripsi / penelitian. Mata kuliah tugas akhir tidak memerlukan ruang dalam pelaksanaannya.
4. Mata kuliah praktek / kerja lapangan. Mata kuliah praktek / kerja lapangan tidak menggunakan ruang dalam pelaksanaannya.
5. Mata kuliah KKN. Mata kuliah KKN tidak menggunakan ruang dalam pelaksanaannya.
6. Mata kuliah untuk kelas internasional. Mata kuliah kelas internasional biasanya telah disediakan ruang khusus oleh jurusan.
7. Mata kuliah yang diadakan di fakultas lain (agama budha, agama hindu, dan lain-lain). Mata kuliah ini biasanya diadakan di fakultas lain seperti fakultas filsafat sehingga tidak menggunakan ruang yang telah direferensikan dari fakultas MIPA UGM.

4.2 Analisis Kebutuhan Aplikasi

Pada analisis kebutuhan aplikasi yang dilakukan adalah analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

4.2.1 Kebutuhan fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisikan proses-proses yang mampu diberikan oleh aplikasi yang akan dibangun. Kebutuhan fungsional pada aplikasi yang dibuat meliputi:

1. Aplikasi mampu mengelola data yang digunakan dalam penjadwalan.
2. Aplikasi dapat melakukan prediksi jumlah peserta mata kuliah serta menampilkan hasil prediksinya.
3. Aplikasi mampu melakukan pembangkitan kelas dengan jumlah peserta dari hasil prediksi.
4. Aplikasi dapat melakukan penjadwalan mata kuliah serta menampilkan hasil jadwal yang didapatkan.

4.2.2 Kebutuhan non fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang memberikan batasan terhadap kebutuhan fungsional. Kebutuhan non-fungsional pada aplikasi meliputi:

1. Aplikasi menggunakan *browser* untuk menjalankannya.
2. Aplikasi membutuhkan *web server*.
3. Aplikasi membutuhkan database MySQL.
4. Aplikasi membutuhkan *software* pengolah *sheet* seperti Microsoft Excel.

4.3 Perancangan Basis Data

Perancangan basis data diperlukan untuk memodelkan hubungan antar data serta menyediakan struktur informasi bagi sistem. Pada perancangan basis data

dalam penelitian ini pertama terdapat rancangan bisnis yang terjadi kemudian dimodelkan dengan diagram E-R (*Entity Relationship*).

4.3.1 Perancangan bisnis proses

Perancangan bisnis proses terbagi menjadi 3 kebutuhan yakni kebutuhan entitas, atribut, dan relasi.

a. Entitas dan atribut

Dalam penjadwalan mata kuliah terdapat beberapa entitas yang digunakan serta atribut – atribut penyusunnya yaitu :

- 1) mata_kuliah_kur_rekap : digunakan untuk menyimpan jumlah peserta mata kuliah pada tahun tertentu.

Dengan atribut sebagai berikut:

- a) mkkurkp_id: merupakan atribut *primary key*.
- b) mkkurkp_tahun: merupakan tahun jumlah peserta mata kuliah.
- c) mkkurkp_jml_peminat: jumlah peserta untuk mata kuliah pada tahun tertentu.

- 2) mata_kuliah_kurikulum : digunakan untuk menyimpan data mata kuliah yang digunakan dalam penjadwalan.

Dengan atribut sebagai berikut:

- a) mkkur_id: merupakan atribut *primary key*.
- b) mkkur_nama: nama mata kuliah.
- c) mkkur_kode: kode mata kuliah.
- d) mkkur_sks: jumlah sks mata kuliah.
- e) mkkur_paket: paket semester mata kuliah.
- f) mkkur_smt: semester mata kuliah (ganjil atau genap).
- g) mkkur_sifat: sifat mata kuliah (wajib atau pilihan)

- 3) dosen : digunakan untuk menyimpan data dosen.

Dengan atribut sebagai berikut:

- a) dsn_id: merupakan atribut *primary key*.

nilainya sudah melebihi *threshold* yang ditentukan. Gambar 5.7 menunjukan perhitungan nilai MSE.

```
public function mse($t) {
    $mse = 0;
    $n = $this->jmlLayer;

    for($i=0; $i<$this->layersSize[$n-1]; $i++){
        $mse+=($t-$this->output[$n-1][$i]) * ($t-$this->output[$n-1][$i]);
    }
    $mse = $mse/2;

    return $mse;
}
```

Gambar 5.7 Kode sumber fungsi *mse()*

6. *unscaleOutput()*

Fungsi *unscaleOutput()* merupakan fungsi yang digunakan untuk mengembalikan hasil transformasi data ke bentuk asalnya. Gambar 5.8 merupakan bagaimana proses pengembalian transformasi data ke bentuk asalnya.

```
public function unscaleOutput($outputVector, $min, $max){
    $temp=0.0;

    $temp = ($max-$min) * (($outputVector - LO) / (HI-LO)) + $min ;
    $unscaledVector = $temp;

    return $unscaledVector;
}
```

Gambar 5.8 Kode sumber *unscaleOutput()*

5.3.2 Implementasi proses pembangkitan kelas

Proses pembangkitan kelas merupakan proses yang dilakukan untuk membuat kelas-kelas mata kuliah berdasarkan jumlah prediksi yang didapatkan. Kelas dibuat dengan beberapa aturan sesuai dengan jumlah peserta hasil prediksi dan batas jumlah kelas. Aturan tersebut terdiri dari 3 kondisi utama yakni apabila jumlah peserta prediksi kurang dari batas minimal kelas (i), jika lebih dari atau sama dengan batas minimal kelas dan kurang dari atau sama dengan batas

maksimal kelas (ii), dan jika lebih dari batas maksimal kelas (iii). Gambar 5.9 merupakan algoritma pembangkitan kelas.

```
//data prodi berdasarkan mata kuliah
mkProdi = get_base_mkprodid_by_mkid(idmk);
kelas = array();
if pred_jml_peminat < bts_kls_min do
    if sifat_makul = 'W' do
        //set nama kelas dengan kode mata kuliah
        nama_kelas = kd_mk;
        //set jumlah peserta kelas dengan hasil prediksi
        jumlah_per_kelas = pred_jml_peminat;
        //set data kelas
        kelas[] = daftarkanKelas(nama_kelas,jumlah_per_kelas);
    else
        kelas[] = null;
elseif pred_jml_peminat >= bts_kls_min
AND pred_jml_peminat <= bts_kls_max do
    if makul_is_universal() OR count(mkProdi)==0 do
        nama_kelas = kd_mk;
        jumlah_per_kelas = pred_jml_peminat;
        kelas[] = daftarkanKelas(nama_kelas,jumlah_per_kelas);
    else
        prodi_mk = mkProdi.prodi_kode;
        for i=1 to count(mkProdi) do
            prodi_mk .= mkProdi.prodi_kode;
            nama_kelas = prodi_mk.'-'.kd_mk;
            jumlah_per_kelas = pred_jml_peminat;
            kelas[] = daftarkanKelas(nama_kelas,jumlah_per_kelas);
else
    //set batas maks kelas
    a = bts_kls_max;
    //set kode mata kuliah
    b = kd_mk;
    if makul_is_universal() do
        prodi_mk = '';
        is_universal = true;
        kelas = klasifikasi(a, b, jml_porsi,prodi_mk,is_unvrsl);
    else
        if (count(mkProdi)>0) do
            for i=1 to count(mkProdi) do
                prodi_mk = mkProdi.prodi_kode;
                if i == 0 do
                    jml_porsi += mkProdi.sisa;
                    is_universal = false;
                    kelas_temp = klasifikasi(a,b,jml_porsi,prodi_mk,is_unvrsl);
                    for j=1 to count(kelas_temp) do
                        kelas[] = kelas_temp[j];
                else
                    prodi_mk = '';
                    is_universal = false;
                    kelas = klasifikasi(a,b,jml_porsi,prodi_mk,is_unvrsl);
```

Gambar 5.9 Algoritma pembangkitan kelas

Pada kondisi ketika jumlah peserta prediksi lebih dari batas maksimal kelas maka akan dibuat kelas paralel. Kelas paralel memiliki urutan abjad untuk membedakan masing-masing kelas. Pengimplementasian kelas ini diterapkan dengan memakai fungsi *klasifikasi()* seperti yang ditunjukkan pada gambar 5.10.

```
function klasifikasi(bts_kls_max, kd_mk, jml_porsi, prodi_mk, is_unvrsl){
    //hitung jumlah kelas dengan pembulatan ke atas
    kelas_bagi = ceil(jml_porsi/bts_kls_max) ;
    //hitung sisa
    mod = jml_porsi % kelas_bagi;
    for i=1 to kelas_bagi do
        //hitung daya tampung kelas
        jumlah_per_kelas = floor(jml_porsi / kelas_bagi);
        if mod > 0 AND i==1 do
            jumlah_per_kelas = floor(jml_porsi / kelas_bagi)) + mod;
        }
        kelas_nama = '';
        if is_unvrsl do
            if kelas_bagi>1 do
                kelas_nama = i+1;
                //set nama kelas paralel dengan urutan angka
                nama_kelas = kd_mk.'-'.kelas_nama;
            else
                if kelas_bagi>1 do
                    //set urutan abjad untuk nama kelas
                    kelas_nama = chr(i+65);

                if prodi_mk == '' do
                    //set nama kelas paralel
                    nama_kelas = kd_mk.'-'.(kelas_nama!=''?kelas_nama: '');
                else
                    kelas_nama!=''?kelas_nama.'-'.kd_mk:kd_mk;
                    //set nama kelas paralel
                    nama_kelas = prodi_mk.'-'.kelas_nama;
                //set data kelas
                kelas[] = daftarkanKelas(nama_kelas,jumlah_per_kelas);

        return kelas;
    }
```

Gambar 5.10 Kode sumber fungsi *klasifikasi()*

5.3.3 Implementasi proses pembuatan jadwal

Pembuatan jadwal mata kuliah hanya dapat dilakukan ketika telah terdapat kelas-kelas mata kuliah yang telah dilengkapi dengan dosen pengajarnya. Pembuatan jadwal mata kuliah ini diimplementasikan dengan menggunakan algoritma genetika. Untuk menghasilkan solusi jadwal mata kuliah, terdapat

beberapa fungsi yang harus dilalui diantaranya *generate_population()*, *update_population()*, *count_fitness()*, *roulette_wheel_selection()*, *crossover()*, *mutation()*, dan *update_selection()*. Gambar 5.11 merupakan proses-proses di dalam algoritma genetika untuk mendapatkan jadwal mata kuliah.

```
for i=1 to jumlah_generasi do
    if i == 0 do
        //pembangkitan populasi pada generasi pertama
        generate_population();
    else
        //pembaruan populasi
        update_population();

        //perhitungan nilai fitness
        count_fitness();
        //proses seleksi
        roulette_wheel_selection();
        crossover();
        mutation();
        update_selection();
    }
//pengambilan solusi dari hasil proses algoritma genetika
solusi = get_solution();
```

Gambar 5.11 Algoritma proses pembuatan jadwal

1. *generate_population()*

Fungsi *generate_population()* merupakan fungsi yang digunakan untuk membuat populasi awal yang tersusun dari individu-individu pembentuk solusi yang layak atau telah sesuai dengan aturan umum yang telah ditetapkan. Gambar 5.12 menunjukkan proses pembangkitan populasi. Fungsi ini akan mengeksekusi fungsi *create_ind()* untuk membentuk individu di dalam populasi.

```
public function generate_population() {
    //data kelas
    $this->kromosom = $this->create_information_class();
    $this->populasi = array();
    for ($i=0; $i < $this->post['jml_ind']; $i++) {
        //pembuatan ind
        $this->populasi[] = $this->create_ind();
    }
}
```

Gambar 5.12 Kode sumber fungsi *generate_population()*

Fungsi *create_ind()* membuat individu sebagai pilihan solusi jadwal mata kuliah yang telah layak. Gambar 5.13 menunjukkan proses pembuatan individu yang memenuhi aturan.

```
public function create_ind(){
    // untuk menampung sejumlah ind yang mewakili jadwal
    $ind = array();
    // untuk mengelompokan kelas berdasar matakuliahnya.
    $mkGrup = array();
    // data grup waktu_id kelas berdasar makulnya.
    $wktGrup = array();
    // matriks data ruang, hari, dan waktu
    $timeSpace = $this->timeSpace;
    foreach ($this->kromosom as $key => $val) {
        $arr_data = compact('timeSpace','ind','val','mkGrup','wktGrup');
        //tiap gen diberi value ruang dan waktu
        $ret_data = $this->get_feasible_ind($arr_data);
        extract($ret_data);
    }

    return $ind;
}
```

Gambar 5.13 Kode sumber fungsi *create_ind()*

2. *update_population()*

Fungsi *update_population()* merupakan fungsi untuk memperbarui populasi yang tersusun dari individu layak atau tidak melanggar aturan umum yang ditetapkan. Gambar 5.14 menunjukkan proses pembaruan populasi.

```
public function update_population(){
    $this->populasi = array();
    //individu hasil proses genetika diperbaiki sesuai aturan umum
    foreach ($this->populasi_baru as $key => $value) {
        $this->populasi[] = $this->repairing_individu($value);
    }
    $this->populasi_baru = null;
}
```

Gambar 5.14 Kode sumber fungsi *update_population()*

3. *count_fitness()*

Fungsi *count_fitness()* digunakan untuk menghitung nilai *fitness* pada suatu individu. Pada proses perhitungan nilai *fitness* dibagi dalam beberapa tahap

| No | Kode mata kuliah | Mata kuliah | Poin semester | Satuan | Program Studi | SKS | Prediksi Jumlah Peminat |
|----|------------------|---------------------------------------|---------------|--------|-------------------------------|-----|-------------------------|
| 1 | UNU100 | AGAMA ISLAM | 1 | ganj | | 2 | 523 |
| 2 | MIK2391 | ALGORITMA DAN STRUKTUR DATA II | 3 | genj | Ilmu Komputer | 3 | 125 |
| 3 | MIK2397 | ALGORITMA GENETIKA | 5 | ganj | Ilmu Komputer | 3 | 24 |
| 4 | NFF2125 | ALJABAR ABSTRAK DALAM FISIKA TEORETIK | 3 | ganj | Fisika | 3 | 4 |
| 5 | MMI1202 | ALJABAR LINEAR ELEMENTER | 1 | ganj | Matematika | 3 | 106 |
| 6 | MMI2204 | ALJABAR LINEAR NUMERIK | 5 | ganj | Matematika | 2 | 87 |
| 7 | MMI1207 | ALJABAR VEKTOR DAN MATRIKS | 1 | ganj | Matematika | 2 | 105 |
| 8 | MIK2393 | ANALISIS DAN DESAIN ALGORITMA I | 1 | ganj | Ilmu Komputer | 3 | 108 |
| 9 | NKK3381 | ANALISIS INSTRUMENTAL II | 5 | ganj | Kimia | 2 | 148 |
| 10 | NKK3381 | ANALISIS KIMIA LINGKUNGAN | 1 | ganj | Kimia | 2 | 11 |
| 11 | NIE2800 | ANALISIS PENGUKURAN FISIS | 3 | ganj | Elektronika dan Instrumentasi | 2 | 114 |
| 12 | MMS3403 | ANALISIS REGRESI TERAPAN | 5 | ganj | Statistika | 2 | 165 |
| 13 | MFG3548 | ANALISIS SPEKTRUM SINYAL DIGITAL | 5 | ganj | Statistika | 2 | 98 |
| 14 | MMS3417 | ANALISIS VARIANSI TERAPAN | 5 | ganj | Statistika | 2 | 10 |
| 15 | MMI2105 | ANALISIS VEKTOR | 3 | ganj | Matematika | 2 | 77 |

Gambar 5.24 Halaman mata kuliah beserta prediksi jumlah peserta

5.4.6 Implementasi halaman *input* pembangkitan kelas

Halaman *input* pembangkitan kelas digunakan untuk memberikan masukan parameter yang digunakan dalam membangkitkan kelas. Pembangkitan kelas ini memakai data jumlah peminat mata kuliah dari hasil prediksi yang dilakukan. Masukan parameter batas maksimal kelas yang digunakan pada halaman ini merupakan parameter standar yang akan dipakai dalam membentuk kelas apabila belum terdapat penentuan kapasitas maksimal kelas pada mata kuliah. Masing-masing mata kuliah sendiri dapat ditentukan jumlah kapasitas maksimal kelasnya dengan memanfaatkan fitur pengelolaan mata kuliah. Gambar 5.25 menunjukkan masukan parameter yang digunakan dalam pembangkitan kelas.

Gambar 5.25 Halaman *input* pembangkitan kelas

5.4.7 Implementasi halaman daftar kelas

Halaman daftar kelas merupakan halaman yang menampilkan data kelas yang telah dibangkitkan dengan menggunakan jumlah peminat mata kuliah dari hasil prediksi. Masing-masing kelas memiliki aksi untuk menentukan dosen yang akan mengampu kelas mata kuliah. Kelas tersebut harus memiliki dosen pengampu sebelum digunakan dalam proses penjadwalan mata kuliah. Daftar kelas mata kuliah ditunjukkan pada gambar 5.26.

| No | Nama Kelas | Mata kuliah | Paket Semester | Status | Jumlah Peserta | Dosen Kelas | Aksi |
|----|------------|-------------|----------------|--------|----------------|--|------------------------|
| 1 | LNUT1100-1 | AGAMA ISLAM | 1 | W | 50 | RIOTQ, Drs., M.Eng., Ph.D. | Daftar |
| 2 | LNUT1100-2 | AGAMA ISLAM | 1 | W | 50 | SUNARTA, Drs., M.S. | Daftar |
| 3 | LNUT1100-3 | AGAMA ISLAM | 1 | W | 50 | MCH, AU, JOKO WABONO, GEB., Dr. | Daftar |
| 4 | LNUT1100-4 | AGAMA ISLAM | 1 | W | 60 | SUTARNO, M.Si, Dr. | Daftar |
| 5 | LNUT1100-5 | AGAMA ISLAM | 1 | W | 50 | SUNARTA, Drs., M.S. | Daftar |
| 6 | LNUT1100-6 | AGAMA ISLAM | 1 | W | 50 | INRIANA KARTINI, S.Si., M.Si., Ph.D. | Daftar |
| 7 | LNUT1100-7 | AGAMA ISLAM | 1 | W | 60 | EDI SUHARYADI, S.Si., M.Eng., Dr.Eng. | Daftar |
| 8 | LNUT1100-8 | AGAMA ISLAM | 1 | W | 50 | ADHITYA RORNE EFFENDIE, M.S., M.Sc., Dr. | Daftar |
| 9 | LNUT1100-9 | AGAMA ISLAM | 1 | W | 60 | GUNTUR WARUTO, Drs., S.U., Dr. | Daftar |

Gambar 5.26 Halaman daftar kelas

5.4.8 Implementasi halaman *input* proses penjadwalan

Halaman *input* proses penjadwalan digunakan untuk memberi masukan parameter yang akan digunakan dalam proses penjadwalan mata kuliah. Proses penjadwalan mata kuliah ini menggunakan data kelas yang telah ditentukan dosen pengampunya. Proses penjadwalan mata kuliah menggunakan teknologi algoritma genetika dengan masukan parameter seperti jumlah individu dalam populasi, nilai P_c (probabilitas *crossover*), nilai P_m (probabilitas mutasi), dan jumlah generasi. Halaman *input* proses penjadwalan ditunjukkan pada gambar 5.27.

[illegible]

Gambar 5.27 Halaman *input* proses penjadwalan

5.4.9 Implementasi halaman jadwal mata kuliah

Halaman jadwal mata kuliah merupakan halaman untuk menampilkan jadwal mata kuliah dari masing-masing kelas sesuai dengan aturan umum dan aturan khusus yang ditentukan. Jadwal disusun menggunakan kombinasi antara ruang dan waktu dengan masing-masing sesinya seperti yang ditunjukkan pada gambar 5.28.

Home / Penjadwalan / Jadwal Kuliah

Jadwal Kuliah

Daftar Baru

RUANGWAKTU

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------|---|--|---|--|---|--|--|
| S2 01 | | STAT-B-4855388 PENG. STATISTIKA MATEMATIKA II S2 01, senin 08:30-11:00 | | | KIM-B-6802701 MATEMATIKA UNTUK KIMIA S2 01, senin 11:30-13:00 | | KIM-W902351 PENGANTAR KIMIA MAT S2 01, senin 13:30-15:00 |
| S2 02 | | KIM-W903381 NANO DAN BIOMATERIAL S2 02, senin 08:30-10:00 | | | | IKOM-MH4803 ENTERPRISE SYSTEM S2 02, senin 12:30-14:00 | |
| M2 13 | STAT-A-8852333 PENGENDALIAN KUALITAS STATISTIKA M2 13, Senin 07:30-09:10 | | | IKOM-A-680301 FILOSOFI DAN KOMPUTER M2 13, senin 10:30-12:10 | | | |
| S2 03 | | | | | KEMIPAAN-E MATEMATIKA KONTEKSTUAL S2 03, senin 11:30-13:00 | | IKOM-INTER S2 03 |
| K1 | | KIM-A-6803381 ELEKTROKIMIA K1, senin 08:30-10:00 | | MAT-B-4802231 PENG. STRUKTUR ALASAS II K1, senin 11:30-13:00 | | | ELING-PAHA K1, senin |
| S2 05 | | KIM-B-4803381 | | | | GEOFIS-WF02990 | GEOF |

© 2019

Gambar 5.28 Halaman jadwal mata kuliah

BAB VI

HASIL PENELITIAN DAN PENGUJIAN

6.1 Statistik Data

Statistik data dibutuhkan untuk menggambarkan banyaknya data pada proses pengujian yang akan digunakan mulai dari proses prediksi jumlah peserta mata kuliah hingga penjadwalan mata kuliah sehingga setelah pemrosesan tidak ada data hilang maupun data terduplikasi. Statistik data yang digunakan dalam penjadwalan semester ganjil 2014 ditunjukkan pada tabel 6.1.

Tabel 6.1 Statistik data

| Data | Statistik |
|---------------------|-----------------------|
| Total mata kuliah | 190 |
| Total program studi | 7 |
| Total ruang | 28 |
| Total waktu | 48 |
| Slot ruang waktu | $28 \times 48 = 1344$ |

6.2 Hasil Penelitian

Pada hasil penelitian ini bertujuan untuk menunjukkan hasil pemrosesan data mata kuliah agar dapat menghasilkan jadwal mata kuliah sesuai dengan algoritma yang dikembangkan mulai dari prediksi peserta mata kuliah hingga penjadwalan mata kuliah pada semester yang ditentukan. Kemudian dilakukan percobaan untuk mendapatkan jadwal kelas mata kuliah fakultas MIPA UGM pada semester ganjil 2014 yang dilakukan dengan menggunakan data mata kuliah mulai dari tahun 2008 pada jenjang S1 di 7 program studi dan berdasarkan statistik data yang terdapat pada tabel 6.1. Pengaruh parameter masukan pada hasil penjadwalan oleh sistem dalam percobaan belum dipertimbangkan karna lebih mengutamakan pada kesesuaian antara hasil dengan algoritma yang telah dibuat. Untuk mendapatkan hasil jadwal mata kuliah pada percobaan ini akan

melalui dalam 3 tahap, yakni prediksi peserta mata kuliah, pembagian kelas, dan penjadwalan mata kuliah.

6.2.1 Prediksi jumlah peserta mata kuliah

Prediksi jumlah peserta mata kuliah bertujuan untuk mengetahui banyaknya mahasiswa yang akan mengikuti mata kuliah pada semester ganjil 2014 yang diimplementasikan dengan menggunakan metode *backpropagation* dari jaringan syaraf tiruan berdasarkan data mata kuliah yang diambil mulai dari tahun 2008. Prediksi dilakukan pada mata kuliah dari semua program studi di fakultas MIPA dengan total sebanyak 190 mata kuliah. Representasi dari data rekapitulasi mata kuliah semester ganjil dari tahun 2008 hingga tahun 2013 ditunjukkan pada tabel 6.2.

Tabel 6.2 Gambaran data rekapitulasi mata kuliah pada 14 data pertama

| KODE | MAKUL | SIFAT | SKS | PAKET | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 |
|---------|---------------------------------------|-------|-----|-------|------|------|------|------|------|------|
| UNU1000 | AGAMA ISLAM | W | 2 | 1 | 273 | 362 | 240 | 716 | 678 | 528 |
| MIK2201 | ALGORITMA DAN STRUKTUR DATA II | W | 3 | 3 | 0 | 0 | 0 | 98 | 210 | 196 |
| MIK4207 | ALGORITMA GENETIKA | P | 3 | 5 | 20 | 0 | 0 | 20 | 47 | 21 |
| MFF2025 | ALJABAR ABSTRAK DALAM FISIKA TEORETIK | P | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 12 |
| MMM1202 | ALJABAR LINEAR ELEMENTER | W | 3 | 1 | 138 | 164 | 171 | 110 | 106 | 79 |
| MMM3204 | ALJABAR LINEAR NUMERIK | P | 2 | 5 | 51 | 50 | 42 | 51 | 65 | 57 |
| MMM1207 | ALJABAR VEKTOR DAN MATRIKS | W | 2 | 1 | 158 | 211 | 192 | 154 | 165 | 117 |
| MIK2203 | ANALISIS DAN DESAIN ALGORITMA I | W | 3 | 3 | 0 | 0 | 0 | 91 | 129 | 150 |
| MKK3501 | ANALISIS INSTRUMENTAL II | W | 2 | 5 | 0 | 0 | 0 | 156 | 139 | 222 |
| MKK3841 | ANALISIS KIMIA LINGKUNGAN | P | 2 | 1 | 0 | 0 | 0 | 57 | 101 | 122 |
| MIE2806 | ANALISIS PENGUKURAN FISIS | W | 2 | 3 | 0 | 0 | 0 | 99 | 121 | 133 |
| MMS3402 | ANALISIS REGRESI TERAPAN | W | 2 | 5 | 100 | 85 | 87 | 73 | 95 | 102 |
| MFG3948 | ANALISIS SPEKTRUM SINYAL DIGITAL | P | 2 | 5 | 0 | 0 | 0 | 19 | 70 | 99 |
| MMS3417 | ANALISIS VARIANSI TERAPAN | P | 2 | 5 | 74 | 69 | 63 | 80 | 54 | 80 |

Dari data rekapitulasi mata kuliah yang ditunjukkan tabel 6.2 kemudian digunakan untuk pelatihan pada jaringan syaraf dengan menggunakan metode *backpropagation* untuk mendapatkan bobot-bobotnya. Proses pelatihan menggunakan 1 keluaran yaitu jumlah peserta pada tahun target dan 3 masukan yaitu jumlah peserta pada 3 tahun sebelum tahun target. Setelah bobot-bobot telah terlatih maka jaringan syaraf digunakan untuk proses prediksi. Hasil prediksi dengan menggunakan data masukan yang sama pada proses pelatihan ditunjukkan pada tabel 6.3.

Tabel 6.6 Konfigurasi global pembangkitan kelas oleh fakultas MIPA

| Konfigurasi | Nilai |
|-----------------------|-------|
| Jumlah minimal kelas | 5 |
| Jumlah maksimal kelas | 90 |

Proses pembangkitan kelas diperoleh dengan cara membagi jumlah peserta dari hasil prediksi dengan jumlah maksimal kelas apabila jumlah peserta tersebut lebih besar atau sama dengan jumlah minimal kelas. Dari percobaan pembangkitan kelas menggunakan jumlah peserta hasil prediksi pada 190 mata kuliah di semester ganjil tahun 2014 menghasilkan kelas sebanyak 324. Representasi hasil kelas yang dibangkitkan dari jumlah peserta hasil prediksi tiap mata kuliah ditunjukkan pada tabel 6.7.

Tabel 6.7 Gambaran hasil kelas yang dibangkitkan pada 29 data pertama

| Kode | Mata Kuliah | Nama Kelas | Jumlah Peserta |
|---------|---------------------------------|------------------|----------------|
| UNU1000 | AGAMA ISLAM | UNU1000-1 | 59 |
| UNU1000 | AGAMA ISLAM | UNU1000-2 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-3 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-4 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-5 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-6 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-7 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-8 | 58 |
| UNU1000 | AGAMA ISLAM | UNU1000-9 | 58 |
| MIK2201 | ALGORITMA DAN STRUKTUR DATA II | ILKOM-A-MIK2201 | 61 |
| MIK2201 | ALGORITMA DAN STRUKTUR DATA II | ILKOM-B-MIK2201 | 60 |
| MIK4207 | ALGORITMA GENETIKA | ILKOM-MIK4207 | 21 |
| MMM1202 | ALJABAR LINEAR ELEMENTER | MAT-A-MMM1202 | 53 |
| MMM1202 | ALJABAR LINEAR ELEMENTER | MAT-B-MMM1202 | 53 |
| MMM3204 | ALJABAR LINEAR NUMERIK | MAT-MMM3204 | 57 |
| MMM1207 | ALJABAR VEKTOR DAN MATRIKS | MAT-A-MMM1207 | 75 |
| MMM1207 | ALJABAR VEKTOR DAN MATRIKS | MAT-B-MMM1207 | 74 |
| MIK2203 | ANALISIS DAN DESAIN ALGORITMA I | ILKOM-A-MIK2203 | 54 |
| MIK2203 | ANALISIS DAN DESAIN ALGORITMA I | ILKOM-B-MIK2203 | 54 |
| MKK3501 | ANALISIS INSTRUMENTAL II | KIM-A-MKK3501 | 74 |
| MKK3501 | ANALISIS INSTRUMENTAL II | KIM-B-MKK3501 | 74 |
| MKK3841 | ANALISIS KIMIA LINGKUNGAN | KIM-MKK3841 | 71 |
| MIE2806 | ANALISIS PENGUKURAN FISIS | ELINS-A-MIE2806 | 57 |
| MIE2806 | ANALISIS PENGUKURAN FISIS | ELINS-B-MIE2806 | 57 |
| MMS3402 | ANALISIS REGRESI TERAPAN | STAT-A-MMS3402 | 53 |
| MMS3402 | ANALISIS REGRESI TERAPAN | STAT-B-MMS3402 | 52 |
| MFG3948 | ANALISIS SPEKTROMSINYAL DIGITAL | GEOFIS-A-MFG3948 | 49 |
| MFG3948 | ANALISIS SPEKTROMSINYAL DIGITAL | GEOFIS-B-MFG3948 | 49 |
| MMS3417 | ANALISIS VARIANSI TERAPAN | STAT-MMS3417 | 78 |

6.2.3 Penjadwalan mata kuliah

Penjadwalan mata kuliah merupakan proses untuk menempatkan kelas-kelas mata kuliah yang telah dibentuk ke ruang dan waktu tertentu sehingga memenuhi aturan-aturan penjadwalan yang sudah ditetapkan oleh fakultas MIPA. Penjadwalan mata kuliah diimplementasikan menggunakan algoritma genetika. Hasil jadwal yang diperoleh harus memenuhi aturan wajib dan dikatakan semakin baik apabila semakin terpenuhi aturan khusus yang ditentukan. Dari percobaan untuk proses penjadwalan menggunakan kelas yang telah dibangkitkan hasilnya sejumlah 324 kelas berhasil terjadwalkan. Ini berarti tidak ada data hilang selama proses penjadwalan karena kelas dapat terjadwalkan secara penuh. Representasi hasil penjadwalan kelas-kelas mata kuliah pada semester ganjil tahun 2014 ditunjukkan pada tabel 6.8.

Tabel 6.8 Gambaran hasil jadwal kelas mata kuliah untuk semester ganjil tahun 2014 pada 29 data pertama

| Mata Kuliah | | Paket smt | Nama kelas | SKS | Dosen | Ruang | Jumlah Peserta | Jadwal Tatap Muka | | | | |
|-------------|----------------------------------|-----------|------------------|-----|--|---------|----------------|-------------------|-------------|-------------|-------------|-------------|
| Kode | Nama | | | | | | | senin | selasa | rabu | kamis | jumat |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-1 | 2 | ROTO, Drs., M.Eng., Ph.D. | G3 | 59 | | | 15:30-17:10 | | |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-2 | 2 | SUNARTA, Drs., M.S. | U2.02 | 58 | | | 15:30-17:10 | | |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-3 | 2 | MOH. ALI JOKO WASONO, M.S., Dr. | U2.06 | 58 | | | | 14:30-16:10 | |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-4 | 2 | SUTARNO, M.Si., Dr. | M2.12 | 58 | | | | 14:30-16:10 | |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-5 | 2 | SUNARTA, Drs., M.S. | S2.02 | 58 | | | | | 15:30-17:10 |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-6 | 2 | INDRIANA KARTINI, S.Si., M.Si., Ph.D. | U2.01 | 58 | | | | | 15:30-17:10 |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-7 | 2 | EDI SUHARYADI, S.Si., M.Eng., Dr.Eng. | S2.08 | 58 | | | 09:30-11:10 | | |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-8 | 2 | ADHITYA RONNIE EFFENDIE, M.Si., M.Sc., | G3 | 58 | | | 09:30-11:10 | | |
| UNU1000 | AGAMA ISLAM | 1 | UNU1000-9 | 2 | GUNTUR MARUTO, Drs., S.U., Dr. | U2.04 | 58 | | | | 15:30-17:10 | |
| MIK2201 | ALGORITMA DAN STRUKTUR DATA II | 3 | ILKOM-A-MIK2201 | 3 | YOHANES SUYANTO, Drs., M.I.Kom. | S2.01 | 61 | | | | | 14:30-17:00 |
| MIK2201 | ALGORITMA DAN STRUKTUR DATA II | 3 | ILKOM-B-MIK2201 | 3 | JANOE HENDARTO, Drs., M.I.Kom. | S2.08 | 60 | | | | | 14:30-17:00 |
| MIK4207 | ALGORITMA GENETIKA | 5 | ILKOM-MIK4207 | 3 | FAIZAH, S.Kom., M.Sc. | S2.06 | 21 | | | 13:30-16:00 | | |
| MMM1202 | ALJABAR LINEAR ELEMENTER | 1 | MAT-A-MMM1202 | 3 | ARI SUPARWANTO, M.Si., Dr.rer.nat. | U2.06 | 53 | | 10:30-13:00 | | | |
| MMM1202 | ALJABAR LINEAR ELEMENTER | 1 | MAT-B-MMM1202 | 3 | BUDI SURODJO, M.Si., Dr. | M2.10 | 53 | | 10:30-13:00 | | | |
| MMM3204 | ALJABAR LINEAR NUMERIK | 5 | MAT-MMM3204 | 2 | YENI SUSANTI, S.Si., M.Si. | U2.06 | 57 | | 07:30-09:10 | | | |
| MMM1207 | ALJABAR VEKTOR DAN MATRIKS | 1 | MAT-A-MMM1207 | 2 | YENI SUSANTI, S.Si., M.Si. | S2.03 | 75 | | 15:30-17:10 | | | |
| MMM1207 | ALJABAR VEKTOR DAN MATRIKS | 1 | MAT-B-MMM1207 | 2 | ARI SUPARWANTO, M.Si., Dr.rer.nat. | S2.505 | 74 | | 15:30-17:10 | | | |
| MIK2203 | ANALISIS DAN DESAIN ALGORITMA I | 3 | ILKOM-A-MIK2203 | 3 | ANNY KARTIKA SARI, M.Sc., Dr | M2.10 | 54 | | | 14:30-17:00 | | |
| MIK2203 | ANALISIS DAN DESAIN ALGORITMA I | 3 | ILKOM-B-MIK2203 | 3 | JANOE HENDARTO, Drs., M.I.Kom. | U2.06 | 54 | | | 14:30-17:00 | | |
| MKK3501 | ANALISIS INSTRUMENTAL II | 5 | KIM-A-MKK3501 | 2 | AGUS KUNCAKA, Dr., DEA. | Lab.Das | 74 | | | | 11:30-13:10 | |
| MKK3501 | ANALISIS INSTRUMENTAL II | 5 | KIM-B-MKK3501 | 2 | DWI SISWANTO, Drs., M.Eng., Ph.D. | M2.12 | 74 | | | | 11:30-13:10 | |
| MKK3841 | ANALISIS KIMIA LINGKUNGAN | 1 | KIM-MKK3841 | 2 | ENDANG TRI WAHYUNI, M.S., Dr., Prof. | S2.505 | 71 | | | 07:30-09:10 | | |
| MIE2806 | ANALISIS PENGUKURAN FISIS | 3 | ELINS-A-MIE2806 | 2 | MASIRAN, M.Si. | M2.10 | 57 | | | | 07:30-09:10 | |
| MIE2806 | ANALISIS PENGUKURAN FISIS | 3 | ELINS-B-MIE2806 | 2 | LINA ARYATI, Dra., M.S., Dr.rer.nat. | K1 | 57 | | | | 07:30-09:10 | |
| MMS3402 | ANALISIS REGRESI TERAPAN | 5 | STAT-A-MMS3402 | 2 | HERNI UTAMI, S.Si., M.Si., | M2.10 | 53 | | | | 13:30-15:10 | |
| MMS3402 | ANALISIS REGRESI TERAPAN | 5 | STAT-B-MMS3402 | 2 | SUGENG, A.Md. | S2.505 | 52 | | | | 13:30-15:10 | |
| MFG3948 | ANALISIS SPEKTRUM SINYAL DIGITAL | 5 | GEOFIS-A-MFG3948 | 2 | BUDI EKAWALURCAHWA, Drs., M.Si. | S2.08 | 49 | 10:30-12:10 | | | | |
| MFG3948 | ANALISIS SPEKTRUM SINYAL DIGITAL | 5 | GEOFIS-B-MFG3948 | 2 | MARDANI, S.E., M.T. | U2.03 | 49 | 10:30-12:10 | | | | |
| MMS3417 | ANALISIS VARIANSI TERAPAN | 5 | STAT-MMS3417 | 2 | SRI HARYATMI, M.Sc., Dr., Prof. | U2.01 | 78 | | 08:30-10:10 | | | |

6.3 Pengujian

Pengujian dilakukan dalam 2 proses yang berbeda, yakni pengujian pada prediksi peserta mata kuliah yang mengimplementasikan jaringan syaraf tiruan dan pengujian pada penjadwalan mata kuliah yang mengimplementasikan algoritma genetika. Pengujian dilakukan pada media komputer yang memiliki *hardware* dengan spesifikasi prosesor 4x Intel(R) Core(TM) i3-2120 CPU @3.30GHz, dan memori 4029MB. Untuk perangkat lunaknya menggunakan spesifikasi sistem operasi Linux Mint 15 Olivia, *server* Apache 2.2.22, php 5.4.9, mysql 5.5.34 , dan *browser* Chromium 30.0.1599.114.

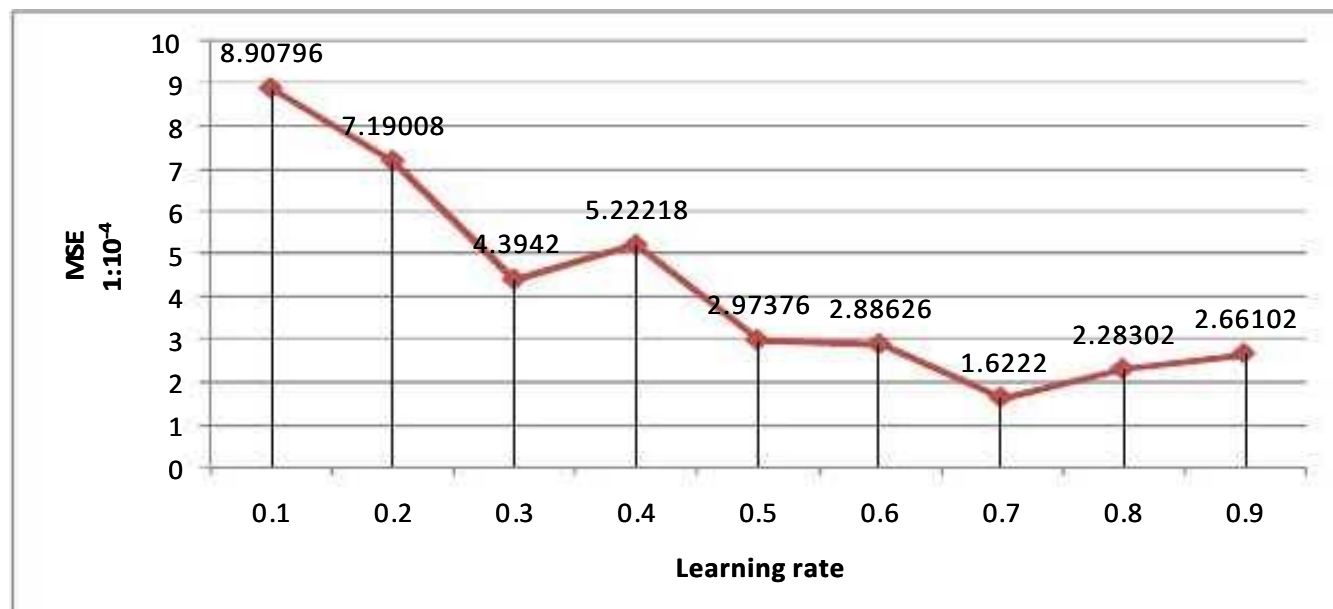
6.3.1 Pengujian prediksi peserta mata kuliah

Pengujian prediksi peserta mata kuliah dilakukan untuk mengetahui jumlah peserta mata kuliah pada periode tahun tertentu dengan penerapan jaringan syaraf tiruan. Proses pengujian dilakukan dengan mengkombinasikan parameter - parameter jaringan syaraf seperti *learning rate*, *threshold*, dan jumlah *epoch*. Tujuan dari uji coba adalah untuk mengetahui pengaruh parameter jaringan syaraf terhadap jumlah peserta yang dihasilkan, dalam hal ini adalah nilai MSE-nya. Terdapat 3 tahap dalam pengujian prediksi jumlah peserta mata kuliah untuk mendapatkan parameter optimal pada sistem dan masing-masing parameter akan diujikan terhadap nilai MSE yang dihasilkan. Tahapan tersebut antara lain adalah pengujian *learning rate*, pengujian jumlah *epoch*, dan pengujian besar *threshold*. Pengujian pertama adalah pengujian *learning rate* terhadap nilai MSE. Pada pengujian *learning rate* dilakukan pada rentang 10% hingga 90% dengan masing-masing perubahan sebesar 10%. Percobaan dilakukan sebanyak 5 kali untuk mendapatkan rata-rata nilai MSE. Hasil rata-rata nilai MSE dari uji coba besar *learning rate* disajikan pada tabel 6.9.

Tabel 6.9 Nilai rata-rata MSE pada pengujian *learning rate*

| Learning rate | MSE ($1:10^{-4}$) | | | | | |
|---------------|---------------------|--------|--------|--------|--------|-----------|
| | coba-1 | coba-2 | coba-3 | coba-4 | coba-5 | rata-rata |
| 0.1 | 9.9882 | 9.8781 | 7.6017 | 8.601 | 8.4708 | 8.90796 |
| 0.2 | 9.6758 | 7.6218 | 5.2337 | 6.5173 | 6.9018 | 7.19008 |
| 0.3 | 5.4248 | 2.1506 | 4.4773 | 6.7159 | 3.2024 | 4.3942 |
| 0.4 | 1.1395 | 6.9708 | 4.5252 | 5.1108 | 8.3646 | 5.22218 |
| 0.5 | 2.5012 | 3.5407 | 2.06 | 0.3667 | 6.4002 | 2.97376 |
| 0.6 | 2.3612 | 0.007 | 3.1161 | 4.1931 | 4.7539 | 2.88626 |
| 0.7 | 0.0621 | 4.8088 | 2.7825 | 0.0043 | 0.4533 | 1.6222 |
| 0.8 | 1.3645 | 1.5933 | 0.1123 | 7.7422 | 0.6028 | 2.28302 |
| 0.9 | 0.0187 | 8.1243 | 2.0612 | 1.0087 | 2.0922 | 2.66102 |

Pada tabel 6.9 terlihat hasil rata-rata nilai MSE skala $1:10^{-4}$ dari 5 percobaan yang dilakukan. Nilai MSE terkecil terletak pada *learning rate* 0,7 dengan nilai MSE 1,6222. Untuk nilai MSE terbesar terletak pada *learning rate* 0,1 dengan nilai MSE 8,90796. Sehingga *learning rate* yang akan digunakan dalam tahap pengujian selanjutnya yaitu 0,7. Adapun grafik pengaruh besar *learning rate* terhadap nilai MSE ditunjukkan oleh grafik pada gambar 6.1.



Gambar 6.1 Perubahan MSE pada pengujian *learning rate*

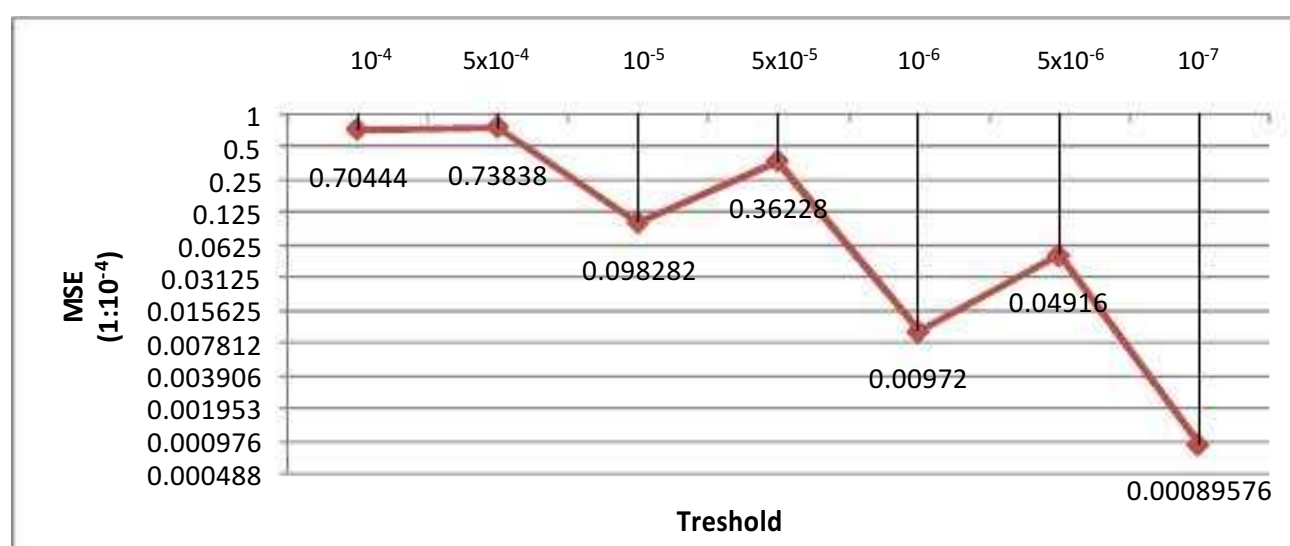
Berdasarkan hasil yang ditunjukkan oleh grafik pada gambar 6.1 terlihat bahwa nilai MSE yang dihasilkan memiliki kecenderungan semakin turun. Nilai MSE sempat mengalami kenaikan ketika *learning rate* bernilai 0,4 dan konsisten naik setelah *learning rate* bernilai 0,7.

Pengujian kedua yaitu pengujian *threshold* terhadap nilai MSE. Pada pengujian *threshold* dilakukan dengan menggunakan 7 nilai uji yaitu 10^{-4} , 5×10^{-4} , 10^{-5} , 5×10^{-5} , 10^{-6} , 5×10^{-6} , dan 10^{-7} . Percobaan dilakukan sebanyak 5 kali untuk mendapatkan rata-rata nilai MSE. Hasil rata-rata nilai MSE dari uji coba besar *threshold* disajikan pada gambar 6.10.

Tabel 6.10 Nilai rata-rata MSE pada pengujian *threshold*

| Threshold | MSE ($1:10^{-4}$) | | | | | |
|--------------------|---------------------|--------|----------|----------|-----------|------------|
| | coba-1 | coba-2 | coba-3 | coba-4 | coba-5 | rata-rata |
| 10^{-4} | 0.8689 | 0.9879 | 0.0012 | 0.9942 | 0.67 | 0.70444 |
| 5×10^{-4} | 1.2776 | 0.2505 | 0.2597 | 1.5836 | 0.3205 | 0.73838 |
| 10^{-5} | 0.0971 | 0.0991 | 9.79E-02 | 0.0988 | 0.09851 | 0.098282 |
| 5×10^{-5} | 0.3905 | 0.3773 | 0.4923 | 0.4935 | 0.0578 | 0.36228 |
| 10^{-6} | 0.0099 | 0.0098 | 0.0094 | 0.0099 | 0.0096 | 0.00972 |
| 5×10^{-6} | 0.0499 | 0.0489 | 0.0481 | 0.0495 | 0.0494 | 0.04916 |
| 10^{-7} | 0.0008 | 0.0009 | 0.0009 | 0.000979 | 0.0008998 | 0.00089576 |

Pada tabel 6.10 terlihat hasil rata-rata nilai MSE skala $1:10^{-4}$ dari 5 percobaan yang dilakukan. Nilai MSE terendah terletak pada saat *threshold* bernilai 10^{-7} yaitu 0.000896. Untuk nilai MSE tertinggi diperoleh pada saat *threshold* bernilai 5×10^{-4} yaitu 0,73838. Sehingga untuk tahap pengujian selanjutnya digunakan *threshold* 10^{-7} . Adapun grafik pengaruh besar *threshold* terhadap nilai MSE ditunjukkan oleh grafik pada gambar 6.2.



Gambar 6.2 Perubahan MSE pada pengujian *threshold*

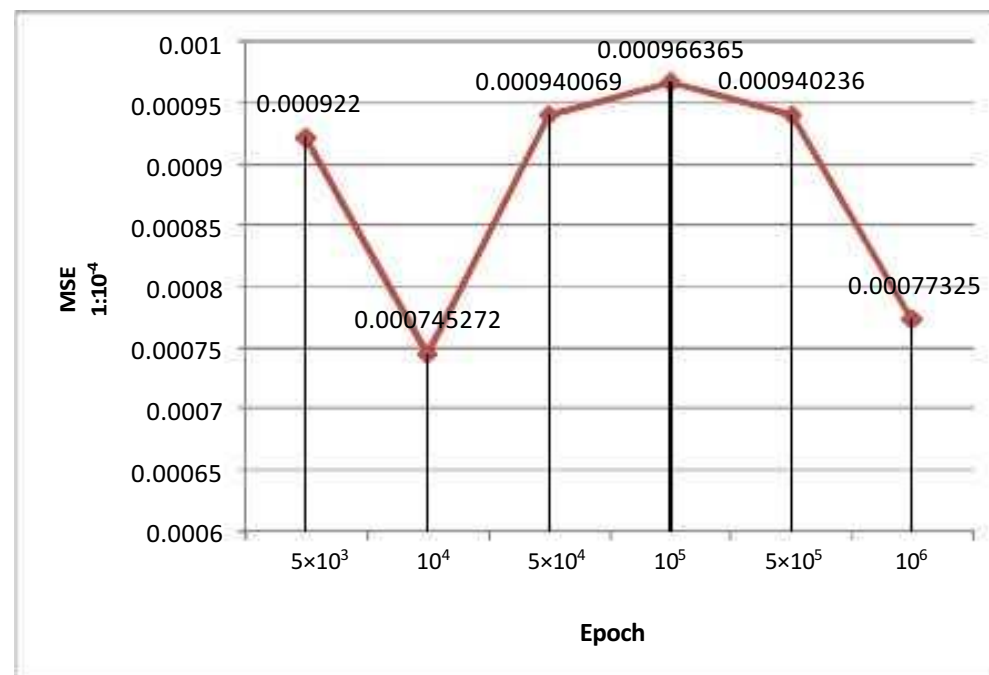
Berdasarkan hasil yang ditunjukkan oleh grafik pada gambar 6.2 terlihat bahwa nilai MSE yang dihasilkan bersifat fluktuatif dan memiliki kecenderungan semakin turun. Nilai MSE beberapa kali sempat mengalami kenaikan ketika *threshold* bernilai 5×10^{-4} , 5×10^{-5} , dan 5×10^{-6} .

Pengujian terakhir untuk prediksi peserta mata kuliah yaitu pengujian jumlah *epoch* terhadap nilai MSE. Pada pengujian ini menggunakan 6 nilai *epoch* yaitu 5000, 10000, 50000, 100000, 500000, dan 1000000. Percobaan dilakukan sebanyak 5 kali untuk mendapatkan rata-rata nilai MSE. Hasil rata-rata nilai MSE dari uji coba besar *threshold* disajikan pada tabel 6.11.

Tabel 6.11 Nilai rata-rata MSE pada pengujian jumlah *epoch*

| Epoch | MSE ($1:10^{-4}$) | | | | | |
|---------|---------------------|------------|------------|------------|------------|------------|
| | coba-1 | coba-2 | coba-3 | coba-4 | coba-5 | rata-rata |
| 5000 | 0.0009841 | 0.0009334 | 0.000909 | 0.0008936 | 0.0008899 | 0.000922 |
| 10000 | 0.00095407 | 0.00086557 | 0.00092722 | 0.00097942 | 8.6694E-08 | 0.00074527 |
| 50000 | 0.0008765 | 0.00095646 | 0.00096835 | 0.00095424 | 0.00094479 | 0.00094007 |
| 100000 | 0.00095938 | 0.00098786 | 0.00097848 | 0.00091254 | 0.00099356 | 0.00096636 |
| 500000 | 0.00093696 | 0.00098249 | 0.00099714 | 0.00080078 | 0.00098381 | 0.00094024 |
| 1000000 | 7.50E-04 | 0.00099036 | 0.00096295 | 0.00094866 | 0.00096428 | 0.00077325 |

Pada tabel 6.11 terlihat hasil rata-rata nilai MSE skala $1:10^{-4}$ dari 5 kali percobaan yang dilakukan. Nilai MSE terendah terletak ketika jumlah *epoch* bernilai 10000 yaitu 0,00074527. Untuk nilai MSE tertinggi diperoleh pada saat jumlah *epoch* bernilai 100000 yaitu 0,00096636. Adapun grafik pengaruh jumlah *epoch* terhadap nilai MSE ditunjukkan oleh grafik pada gambar 6.3.



Gambar 6.3 Perubahan MSE pada pengujian jumlah *epoch*

Secara teori semakin besar jumlah *epoch* maka semakin kecil nilai MSE yang dihasilkan, namun berdasarkan hasil yang ditunjukkan oleh grafik pada gambar 6.3 terlihat bahwa nilai MSE yang dihasilkan mengalami kenaikan pada 5×10^4 dan 10^5 kemudian nilainya turun setelahnya.

Dari tahap-tahap pengujian untuk prediksi peserta mata kuliah yang telah dilakukan maka didapatkan parameter-parameter yang optimal untuk sistem yakni *learning rate* 7×10^{-1} , *threshold* 10^{-7} , dan jumlah *epoch* 10^5 .

6.3.2 Pengujian penjadwalan mata kuliah

Pengujian penjadwalan mata kuliah dilakukan untuk mengetahui optimasi jadwal yang memenuhi aturan dengan penerapan algoritma genetika. Proses pengujian dilakukan dengan mengkombinasikan parameter-parameter genetika seperti probabilitas *crossover*, probabilitas mutasi, jumlah populasi, dan jumlah generasi. Tujuan dari uji coba adalah untuk mengetahui pengaruh parameter genetika terhadap jadwal yang dihasilkan, dalam hal ini adalah nilai *fitness*-nya. Dalam pengujian penjadwalan mata kuliah dibagi dalam 4 tahap, yakni pengujian

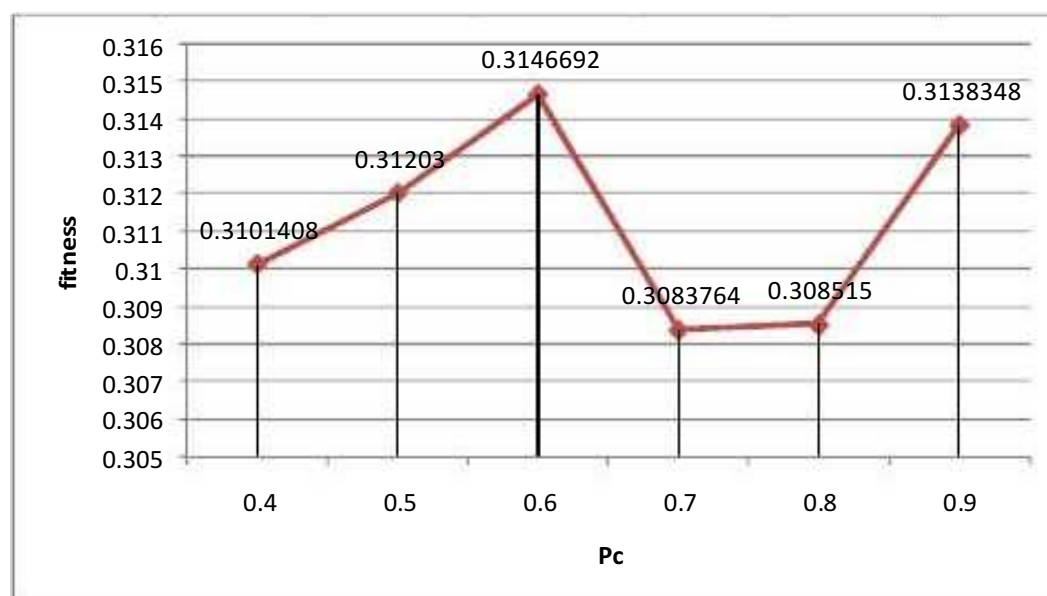
probabilitas *crossover*, pengujian probabilitas mutasi, pengujian jumlah populasi, dan terakhir pengujian jumlah generasi.

Pengujian pertama adalah pengujian pengaruh probabilitas *crossover* terhadap nilai *fitness* yang dilakukan pada 6 nilai yaitu 0,4, 0,5, 0,6, 0,7, 0,8 dan 0,9. Untuk setiap nilai uji dilakukan 5 kali percobaan untuk mendapatkan rata-rata nilai *fitness*-nya. Hasil rata-rata nilai *fitness* dari uji coba yang dilakukan disajikan pada tabel 6.12.

Tabel 6.12 Nilai rata-rata *fitness* pada pengujian probabilitas *crossover*

| Pc | fitness | | | | | |
|-----|----------|----------|----------|----------|----------|-----------|
| | coba-1 | coba-2 | coba-3 | coba-4 | coba-5 | rata-rata |
| 0.4 | 0.325239 | 0.300764 | 0.305123 | 0.300904 | 0.318674 | 0.310141 |
| 0.5 | 0.317075 | 0.316897 | 0.310678 | 0.305393 | 0.310107 | 0.31203 |
| 0.6 | 0.320413 | 0.312791 | 0.314979 | 0.311343 | 0.31382 | 0.314669 |
| 0.7 | 0.320254 | 0.310658 | 0.311688 | 0.304944 | 0.294338 | 0.308376 |
| 0.8 | 0.303915 | 0.315924 | 0.327999 | 0.298818 | 0.295919 | 0.308515 |
| 0.9 | 0.321994 | 0.322555 | 0.307685 | 0.312435 | 0.304505 | 0.313835 |

Berdasarkan tabel 6.12 terlihat bahwa nilai rata-rata *fitness* tertinggi yaitu 0,314669 yang terletak pada probabilitas *crossover* 0,6. Untuk nilai rata-rata *fitness* terendah adalah 0,308376 yang terletak ketika probabilitas *crossover* bernilai 0,7. Untuk mengetahui lebih jelas mengenai perubahan rata-rata nilai *fitness* dapat dilihat grafik pada gambar 6.4.



Gambar 6.4 Perubahan nilai *fitness* pada pengujian probabilitas *crossover*

DAFTAR PUSTAKA

- Adamanti, J., 2002, Penyelesaian Penjadwalan Mata Kuliah di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada dengan Menggunakan Algoritma Genetika, *Skripsi*, Jurusan Ilmu Komputer dan Elektronika FMIPA UGM, Yogyakarta.
- Anggamardika, P., 2008, Penyelesaian Masalah Penjadwalan Matakuliah di Fakultas MIPA Universitas Gadjah Mada Dengan Artificial Immune System, *Skripsi*, Jurusan Ilmu Komputer dan Elektronika FMIPA UGM, Yogyakarta.
- Arfandi, N., 2013, Implementasi Algoritma Genetika Untuk Proses Penempatan (Plotting) Peserta Kuliah Kerja Nyata di Universitas Gadjah Mada, *Skripsi*, Jurusan Ilmu Komputer dan Elektronika FMIPA UGM, Yogyakarta.
- Aydin, M.A., 2008, Solving University Course Timetabling Problem Using Genetic Algorithm, *Tesis*, Institute Of Sciences Industrial Engineering Bahcesehir University, Istanbul.
- Bagui, S., and Earp, R., 2003, *Database Design Using Entity-Relationship Diagrams*, Auerbach Publications, Florida.
- Burke, E.K., Elliman, D.G., and Weare, R.F., 1995, *A Genetic Algorithm for University Timetabling*, Baywood Publishing Company, San Fransisco.
- Converse, T., Park, J., and Morgan, C., 2004, *PHP5 and MySQL Bible*, Wiley Publishing, Inc., Indiana.
- Davis, M.E., and Phillips, J.A., 2007, *Learning PHP and MySQL*, Second Edition, O'Reilly Media, Inc., California.
- Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Alabama.
- Graupe, D., 1997, *Principles of Artificial Neural Networks*, Vol. 3, World Scientific Pub Co, Inc., Chicago.

- Griffiths, A., 2010, *Codeigniter 1.7 Professional Development*, Packt Publishing Ltd., Birmingham.
- Hajek, M., 2005, *Neural Networks*, University of KwaZulu-natal, KwaZulu-natal.
- Kawaguchi, K., 2000, A Multithreaded Software Model for Backpropagation Neural Network Applications, *Thesis*, Department of Electrical and Computer Engineering, University of Texas, Austin.
- Kendall, K.E. and Kendall, J.E., 2011, *Systems Analysis and Design*, Eighth Edition, Prentice Hall , New Jersey.
- Kusumadewi, S., 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta.
- Naramore, E., Gerner, J., Le-Scouarnec, Y., Stolz, J., and Glass, M.K., 2005, *Beginning PHP5, Apache, and MySQL Web Development*, Wiley Publishing, Inc., Indiana.
- Negnevitsky, M., 2005, *Artificial Intelligence*, Pearson Education, England.
- Pallaw, V.K., 2010, *Database Management Systems*, Second Edition, Asian Books, New Delhi.
- Puspitaningrum, D., 2006, *Pengantar Jaringan Syaraf Tiruan*, Penerbit Andi, Yogyakarta.
- Rahayu, D.D., 2010, Implementasi Algoritma Genetika Dalam Penjadwalan Praktikum, *Skripsi*, Jurusan Ilmu Komputer dan Elektronika FMIPA UGM, Yogyakarta.
- Siang, J.J., 2004, Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB, Penerbit Andi, Yogyakarta.
- Sivanandam, S.N., and Deepa, S.N., 2007, *Introduction to Genetic Algorithms*, Springer, New York.
- Suyanto, 2011, *Artificial Intelligence: Searching - Reasoning - Planning - Learning*, Edisi Revisi, Informatika Bandung, Bandung.