

**DESAIN DAN IMPLEMENTASI JADWAL KULIAH  
DENGAN MENGGUNAKAN ALGORITMA SEMUT  
BERBASIS WEB**

**SKRIPSI**

Oleh:

**EVA YUSTINA**  
**NIM : 04550045**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI (UIN) MALANG  
MALANG  
2008**

**DESAIN DAN IMPLEMENTASI JADWAL KULIAH  
DENGAN MENGGUNAKAN ALGORITMA SEMUT  
BERBASIS WEB**

**SKRIPSI**

**Diajukan Kepada :  
Universitas Islam Negeri Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:**

**Eva Yustina  
NIM.04550045**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI (UIN) MALANG  
MALANG  
2008**

## HALAMAN PERSETUJUAN

# DESAIN DAN IMPLEMENTASI JADWAL KULIAH DENGAN MENGGUNAKAN ALGORITMA SEMUT BERBASIS WEB

## SKRIPSI

Oleh:

Eva Yustina  
NIM. 04550045

Disetujui oleh :

Pembimbing I

Pembimbing II

Syahiduzzaman, M.Kom  
NIP. 150 368 777

A. Nasichuddin, M.A  
NIP. 150 302 531

Mengetahui

Ketua Jurusan Teknik Informatika

Fakultas Sains Dan Teknologi Universitas Islam Negeri Malang

Suhartono, S.Si, M.Kom.  
NIP. 150 327 241

## HALAMAN PENGESAHAN

# DESAIN DAN IMPLEMENTASI JADWAL KULIAH DENGAN MENGGUNAKAN ALGORITMA SEMUT BERBASIS WEB

## SKRIPSI

Oleh:

Eva Yustina

NIM. 04550045

Telah Dipertahankan di Depan Dewan Penguji dan  
Dinyatakan Lulus Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal : 29 Juli 2008

Susunan Dewan Penguji :

Tanda Tangan

1. Penguji Utama : Suhartono, M.Kom  
(.....)  
NIP. 150 327 241
2. Ketua Penguji : Ririen Kusumawati, M.Kom  
(.....)  
NIP. 150 368 775
3. Sekr. Penguji : Syahiduzzaman, M.Kom (.....)  
NIP. 150 368 777
4. Anggota Penguji : A. Nasichuddin, M.A (.....)  
NIP. 150 302 531

Mengetahui dan Mengesahkan  
Ketua Jurusan Teknik Informatika

**Suhartono, S.Si, M.Kom.**  
**NIP. 150 327 241**  
**HALAMAN PERSEMBAHAN**

*Seiring do'a dan dukungannya skripsi ini kupersembahkan untuk :*

*Ayah dan ibu yang telah memberikan dorongan materi dan spiritual yang tidak terhingga sehingga penulis dapat menyelesaikan skripsi ini.*

*Mas Andik terima kasih mau ngalah buat berbagi komputer dan tinta printnya.*

*Mas Dita terima kasih banget buat semua yang sudah mas berikan, kesabaran dan perhatiannya.*

*Mame' dan mas Yoni selaku tim sukses yang membuat programku nggak error-error terus.*

*Seluruh Dosen Teknik Informatika UIN Malang.*

*Laboran Teknik Informatika : Mas Gun, Mas Deni , Mas Wawan thanks sudah bantuin.*

*Temen-temen Teknik Informatika angkatan perdana yang tidak bisa saya sebutkan satu persatu terima kasih kita sudah berjuang bersama-sama selama 4 tahun. Semoga Sukses...!!!*

*Dan semua pihak yang tidak dapat saya sebutkan satu persatu saya minta maaf dan terima kasih.*

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PERSETUJUAN</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iv</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vi</b>
<b>DAFTAR ISI</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xi</b>
<b>ABSTRAK</b> .....	<b>xiii</b>
<b>BAB I. PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Masalah .....	5
1.4 Tujuan dan Manfaat Penelitian .....	6
1.5 Metodologi .....	6
1.6 Sistematika Penulisan .....	7
<b>BAB II. TINJAUAN PUSTAKA</b>	
2.1 Penjadwalan Kuliah .....	9
2.1.1 Jadwal Secara Umum .....	9
2.1.2 Penjadwalan Kuliah .....	10
2.2 Algoritma Semut .....	10
2.2.1 Algoritma .....	10
2.2.2 Algoritma Semut .....	12
2.3 PHP .....	27
2.4 MySQL .....	28
<b>BAB III. DESAIN SISTEM</b>	
3.1 Analisis Kebutuhan Perangkat Lunak .....	29
3.1.1 Metode Analisis .....	29
3.1.2 Hasil Analisis .....	29
3.1.2.1 Analisis Kebutuhan Input .....	30
3.1.2.2 Analisis Kebutuhan Proses .....	32
3.1.2.3 Analisis Kebutuhan Output .....	34
3.1.3 Kebutuhan Antar Muka .....	34
3.2 Perancangan Perangkat Lunak .....	35
3.2.1 Metode Perancangan .....	35
3.2.2 Perancangan .....	35
3.2.2.1 Perancangan Desain Sistem .....	35
3.2.2.2 Algoritma .....	41
3.3 Desain Tampilan .....	42
3.3.1 Halaman Administrator .....	42
3.3.2 Halaman User .....	43
3.4 Desain Database	
4.4.1 Tabel Utama .....	44

4.4.2 Tabel Pendukung .....	46
<b>BAB IV. IMPLEMENTASI DAN PEMBAHASAN</b>	
4.1 Algoritma Program .....	50
4.1.1 Koneksi .....	50
4.1.2 Save .....	51
4.1.3 Update .....	52
4.1.4 Delete .....	53
4.2 Algoritma Semut .....	53
4.2.1 Ant.Class .....	53
4.2.2 AntAlgorithm.class .....	57
4.3 Pengujian Program .....	59
4.3.1 Halaman Admin .....	59
4.3.2 Halaman User .....	70
<b>BAB V. KESIMPULAN DAN SARAN</b>	
5.1 Kesimpulan .....	72
5.2 Saran .....	72
<b>DAFTAR PUSTAKA</b>	

## DAFTAR GAMBAR

Gambar 2.1 Alur semut ke sumber makanan dan sarangnya .....	14
Gambar 2.2 Alur suatu semut ketika akan melewati rintangan .....	14
Gambar 2.3 Alur semut yang melewati dua jalur .....	15
Gambar 2.4 Alur ant yang melewati jalur terpendek.....	15
Gambar 2.5 Ilustrasi skrip PHP dieksekusi .....	27
Gambar 3.1 Context Diagram .....	36
Gambar 3.2 Data Flow Diagram .....	37
Gambar 3.3 ER-Diagram .....	38
Gambar 3.4 Flow Chart Enty .....	39
Gambar 3.5 Flow Chart Edit .....	40
Gambar 3.6 Flow Chart Delete .....	41
Gambar 3.7 Desain Halaman Administrator .....	42
Gambar 3.8 Desain Halaman User .....	43
Gambar 4.1 Halaman Login .....	60
Gambar 4.2 Halaman Index .....	60
Gambar 4.3 Halaman Fakultas .....	61
Gambar 4.4 Halaman Jurusan .....	62
Gambar 4.5 Halaman Program Studi .....	62
Gambar 4.6 Halaman Dosen .....	63
Gambar 4.7 Halaman Matakuliah .....	63
Gambar 4.8 Halaman Ruang .....	64
Gambar 4.9 Halaman Matakuliah Tahun .....	64
Gambar 4.10 Halaman Klesediaan Dosen .....	65
Gambar 4.11 Halaman Kelas Mengajar .....	65
Gambar 4.12 Halaman Introduction Ant Algorithm .....	66
Gambar 4.13 Halaman Pre-Requistic .....	67
Gambar 4.14 Halaman Generate .....	67
Gambar 4.15 Halaman Hasil Generate .....	68
Gambar 4.16 Halaman Berita .....	69
Gambar 4.17 Halaman Artikel .....	69
Gambar 4.18 Halaman Depan .....	70
Gambar 4.19 Halaman Berita .....	71
Gambar 4.20 Halaman Daftar Tamu.....	71
Gambar 4.21 Halaman Form Buku Tamu .....	72



## ABSTRAKSI

**Yustina. Eva, 2008. Desain dan Implementasi Jadwal Kuliah Dengan Menggunakan Algoritma Semut Berbasis Web**, Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Malang.  
Pembimbing : Syahiduzzaman, M.Kom, Ach. Nasichuddin, M.A

---

**Kata Kunci** : *Jadwal kuliah, Algoritma semut.*

Penjadwalan kuliah dengan menggunakan algoritma semut mampu memberikan solusi alternatif pada penjadwalan kuliah yang menginginkan waktu tertentu secara optimal.

Pada penelitian ini metode yang digunakan adalah studi pustaka. Studi pustaka ini dilakukan dengan membaca buku-buku serta literatur yang berhubungan dengan penjadwalan serta algoritma semut. Selain dengan menggunakan studi pustaka peneliti juga menggunakan studi lapangan. Hal ini dimaksudkan untuk mendapatkan gambaran yang sebenarnya tentang keadaan instansi dengan mengamati secara langsung pada instansi tersebut, dengan cara pengamatan atau metode pengumpulan data dengan melakukan interview pada bagian-bagian yang terlibat dalam proses penyusunan jadwal kuliah.

Dari hasil analisis didapatkan: penjadwalan kuliah dengan menggunakan algoritma semut mampu memberikan solusi dalam penyusunan jadwal *on-line*, algoritma semut dapat diterapkan untuk membuat jadwal kuliah dengan hasil yang optimal, algoritma semut mampu memberikan solusi dalam penyusunan jadwal kuliah yang menginginkan waktu tertentu.

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Pengaturan jadwal apapun tentunya sangat rumit dan merepotkan, tentu banyak menghabiskan waktu dan tenaga dalam menyusunnya. Penjadwalan diperlukan untuk mengatur waktu kerja sehingga didapatkan jadwal yang seefisien mungkin. Sebuah jadwal akan mudah disusun apabila hanya terdapat sedikit komponen yang dijadwalkan, namun akan menjadi rumit apabila komponen yang dijadwalkan berjumlah banyak.

Beberapa contoh penjadwalan antara lain penjadwalan produksi, penjadwalan kerja karyawan, penjadwalan pertandingan, penjadwalan kuliah serta masih banyak lagi. Pada setiap penjadwalan diharapkan akan mendapatkan jadwal yang optimal dan efektif serta dapat mengurangi permasalahan dan kesalahan yang ada.

Penjadwalan kuliah berfungsi untuk mengorganisir (*manage*) waktu agar lebih efektif dan efisien karena waktu merupakan sesuatu yang sangat berharga dan penting, hal ini dapat dilihat dalam ayat-ayat Al-Quran yang menerangkan tentang waktu, seperti dalam Al-Quran Surat Al-Lukman ayat 29.

أَلَمْ تَرَ أَنَّ اللَّهَ يُولِجُ اللَّيْلَ فِي النَّهَارِ وَيُولِجُ النَّهَارَ فِي اللَّيْلِ وَسَخَّرَ

الشَّمْسَ وَالْقَمَرَ كُلٌّ يَجْرِي إِلَىٰ أَجَلٍ مُّسَمًّى وَأَنَّ اللَّهَ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿٢٩﴾

*Tidakkah kamu memperhatikan, bahwa sesungguhnya Allah memasukkan malam ke dalam siang dan memasukkan siang ke dalam*

*malam dan Dia tundukkan matahari dan bulan masing-masing berjalan sampai kepada waktu yang ditentukan, dan sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan.*(QS. Al-Lukman: 29)

Pembahasan skripsi ini mengenai penjadwalan kuliah, secara umum penjadwalan kuliah memiliki beberapa komponen, antara lain : matakuliah yang ditawarkan, dosen, jam kuliah, ruangan kelas serta jumlah SKS (BAK UIN).

- Matakuliah yang ditawarkan merupakan satuan pelajaran yang diajarkan, tiap semester program matakuliah yang ditawarkan berbeda, dimana dibedakan dengan aturan semester genap atau ganjil. Dalam pengambilan (pemrograman) matakuliah tergantung dari mahasiswa tersebut ingin mengambil matakuliah pilihannya, namun tentu saja terdapat syarat yang harus dipenuhi terlebih dahulu.
- Dosen adalah seorang pengajar dari matakuliah yang ditawarkan (disajikan), biasanya dosen dapat mengajar lebih dari satu matakuliah, tetapi sering terjadi dalam penyusunan jadwal kuliah jam mengajar seorang dosen bersamaan untuk matakuliah yang berbeda, sehingga terjadi bentrok antara matakuliah yang satu dengan yang lainnya.
- Jam kuliah merupakan waktu yang dibutuhkan untuk menyampaikan materi pelajaran, umumnya jam kuliah tergantung pada beban SKS untuk tiap satu SKS lamanya berkisar antara 45 menit hingga 50 menit.
- Ruangan (kelas) komponen yang sangat penting karena kegiatan perkuliahan tidak akan berjalan maksimum jika ruangan yang tersedia tidak memenuhi kapasitas.

- Jumlah SKS dalam penjadwalan digunakan sebagai tolak ukur lamanya kegiatan kuliah dilaksanakan.

Saat ini penjadwalan kuliah dilakukan secara manual dengan menyusun tabel, tetapi dalam penyusunan jadwal kuliah terdapat beberapa kesalahan yang baik kesalahan pada ruang, waktu, serta dosen pengajar.

Berdasarkan uraian tersebut penyusunan jadwal kuliah akan lebih baik (efektif) bila dilakukan secara *on line (computerize)*, penyusunan jadwal secara *on-line* memerlukan metode yang bisa diterapkan agar memberikan hasil yang baik, untuk itu ada beberapa metode penjadwalan yang digunakan, misalnya *linier programming, genetic algorithms* maupun *operation research*. Pada penelitian ini digunakan metode algoritma semut dengan harapan dapat dihasilkan penjadwalan yang lebih optimum dari metode yang pernah digunakan.

Algoritma semut (*Ant Algorithms*) pertama kali diperkenalkan oleh Marco Dorigo pada tesis PhD tahun 1992. Algoritma semut lebih menitik beratkan pada perilaku kebiasaan atau yang sering dilakukan (dilewati) oleh semut. Dengan kemampuan dan keunikan semut, yang memiliki kemampuan secara alami (*riil ant*) untuk menemukan alur/lintasan yang terpendek dari sarangnya ke suatu sumber makanan tanpa pengertian visual (penglihatan), sehingga diperjelas dalam Al-Quran yang diturunkan oleh Allah swt terdapat surat An-Naml yang berarti semut. Di dalam surat An-Naml ayat 18-19 diceritakan mengenai kepandaian semut menilai keadaan pada zaman Nabi Sulaiman as:

حَسْبِيَ إِذَا أْتَوْا عَلَيَّ وَإِذِ اللَّيْلُ قَالَتْ نَمَلَةٌ  
يَتَأَيُّهَا النَّمْلُ ادْخُلُوا مَسَكِنَكُمْ لَا يَحْطِمَنَّكُمْ سُلَيْمَانُ وَجُنُودُهُ وَهُمْ لَا يَشْعُرُونَ ﴿١٨﴾

*Hingga apabila mereka sampai di lembah semut berkatalah seekor semut: "Hai semut-semut, masuklah ke dalam sarang-sarangmu, agar kamu tidak diinjak oleh Sulaiman dan tentaranya, sedangkan mereka tidak menyadari." (QS. An-Naml: 18)*

فَتَبَسَّمَ ضَاحِكًا مِّن قَوْلِهَا وَقَالَ رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ  
الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَىٰ وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ وَأَدْخِلْنِي  
بِرَحْمَتِكَ فِي عِبَادِكَ الصَّالِحِينَ ﴿١٩﴾

*Maka dia tersenyum dengan tertawa Karena (mendengar) perkataan semut itu. dan dia berdoa: "Ya Tuhanku berilah Aku ilham untuk tetap mensyukuri nikmat mu yang Telah Engkau anugerahkan kepadaku dan kepada dua orang ibu bapakku dan untuk mengerjakan amal saleh yang Engkau ridhai dan masukkanlah Aku dengan rahmat-Mu ke dalam golongan hamba-hamba-Mu yang saleh". ( Q.S. An-Naml :19)*

Dalam tafsir Ibnu Katsir dijelaskan lebih detail mengenai surat An-Naml ayat 18 dan 19 sebagai berikut:

Ketika Sulaiman dan tentaranya mendekati lembah semut, Sulaiman pun mendengar suara raja semut yang memerintahkan semut-semut lain supaya segera masuk ke liang masing-masing sehingga tidak diinjak oleh Sulaiman dan tentaranya dengan tanpa disadari. Tentang dimana letaknya lembah semut, pendapat para ahli masih simpang siur.

Mendengar ucapan raja semut tersebut, Sulaiman tersenyum dan tertawa. sebab raja semut itu mengatakan bahwa tentara sulaiman tidak bermaksud untuk membuat kejahatan. Hanya saja mereka tidak sadar akan menginjak dan membinasakan semut-semut itu. Untuk melahirkan kegembiraanya, maka sulaiman mensyukuri Allah dan memuji-Nya. Sulaiman berdoa: " wahai Tuhanku, jadikanlah aku sebagai orang yang terus menerus mensyukuri nikmat-Mu dan

memelihara tugas syukur itu, agar aku tetap mensukuri nikmat-Mu yang telah engkau curahkan padaku serta bapak dan ibuku. (Ibnu Katsir – 1976).

Salah satu bentuk dari algoritma semut yang sangat berhasil adalah optimisasi koloni semut (*Ant Colony Optimization* atau yang lebih populer dengan ACO). Metode ACO ditawarkan guna menemukan solusi yang lebih baik dengan memberikan berbagai bentuk kombinasi dalam menyusun jadwal perkuliahan (M. Dorigo - 1992).

## 1.2 Rumusan Masalah

Dari permasalahan yang ada didapatkan rumusan masalah sebagai berikut :

1. Apakah desain dan implementasi jadwal kuliah dengan menggunakan algoritma semut mampu memberikan solusi dalam penyusunan jadwal kuliah?
2. Bagaimana cara membuat jadwal kuliah yang menginginkan waktu tertentu?
3. Bagaimana algoritma semut dapat diterapkan untuk membuat jadwal kuliah dengan hasil yang optimal?

## 1.3 Batasan Masalah

Adapun batasan masalah yang akan dibahas antara lain :

1. Studi kasus dilakukan pada Universitas Islam Negeri (UIN) Malang.
2. Mata kuliah praktikum disertakan dalam penjadwalan.
3. Satu SKS diberikan waktu selama 50 menit.

## 1.4 Tujuan dan Manfaat Penelitian

### a. Tujuan Penelitian

#### 1. Tujuan Institusional

Sebagai syarat kelulusan program sarjana (S-1) pada Jurusan Teknik Informatika fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Malang.

#### 2. Tujuan Pokok

1. Menerapkan algoritma semut dalam pencarian solusi alternatif pada penjadwalan kuliah yang optimal.
2. Menerapkan algoritma semut dalam penjadwalan kuliah dengan menginginkan waktu tertentu.

### b. Manfaat Penelitian

Manfaat yang akan dicapai dari skripsi ini adalah untuk membantu menyelesaikan masalah penjadwalan kuliah sehingga didapatkan jadwal kuliah yang optimal.

## 1.5 Metodologi

Dalam penyusunan skripsi ini, metode yang digunakan oleh penyusun antara lain:

### b. Studi Pustaka

Studi pustaka ini dilakukan dengan membaca buku-buku yang berhubungan dengan penjadwalan serta algoritma semut. Selain itu penulis juga mencari jurnal serta *e-book* dari internet, dengan

maksud agar diperoleh dasar teoritis dan gambaran algoritma semut pada proses penjadwalan.

c. Studi Lapangan

Studi lapangan dilakukan untuk mendapatkan gambaran yang sebenarnya tentang keadaan instansi dengan mengamati secara langsung pada instansi tersebut, dengan cara pengamatan atau metode pengumpulan data dengan melakukan *interview* pada bagian-bagian yang terlibat dalam proses penyusunan jadwal kuliah.

## 1.6 Sistematika Penulisan

Secara garis besar keseluruhan isi bahasan dari skripsi ini terbagi dalam bab-bab, yaitu:

### BAB I : PENDAHULUAN

Dalam bab ini membahas tentang:

- Latar Belakang
- Rumusan Masalah
- Batasan Masalah
- Tujuan dan Manfaat Penelitian
- Metodologi
- Sistematika Penulisan

### BAB II : TINJAUAN PUSTAKA

Dalam Bab II ini berisi tentang:



- Definisi serta uraian jadwal kuliah
- Teori-teori algoritma semut

### BAB III : DESAIN

Dalam bab ini akan diuraikan tentang analisis kebutuhan perangkat lunak serta desain sistem yang akan dibuat.

### BAB IV : IMPLEMENTASI

Pada bab ini akan dibahas mengenai desain program, *database* program serta operasional program dengan membuat program aplikasi dengan PHP, CSS dan javasript.

### BAB V : PENUTUP

Bab ke V ini merupakan bab penutup yang berisi kesimpulan yang diperoleh setelah aplikasi program selesai dibuat dan diuji coba serta saran.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penjadwalan Kuliah

##### 2.1.1 Jadwal Secara Umum

Jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar (tabel kegiatan) atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terinci.

Penjadwalan memiliki arti proses (cara) pembuatan jadwal atau memasukkan rencana kegiatan ke dalam jadwal, selain itu penjadwalan juga merupakan proses penugasan pada satu set (bagian) sumberdaya. Penjadwalan suatu konsep yang penting yang bisa diterapkan pada berbagai bidang, misalkan komputasi dan proses produksi.

Penjadwalan maupun perencanaan yang dilakukan secara otomatis (terkomputerisasi) merupakan cabang dari kecerdasan buatan yang mengacu pada realisasi dari strategi atau urutan pekerjaan, khususnya untuk pelaksanaan agen cerdas, robot otomatis dan kendaraan tanpa awak. Tidak seperti pengendalian klasik dan masalah klasifikasi, solusinya kompleks, tidak diketahui dan harus ditemukan dan dioptimalkan pada ruang multidimensi.

### 2.1.2 Penjadwalan Kuliah

Penjadwalan kuliah (*Lecture Timetabling*) adalah masalah menempatkan waktu dan ruangan kepada sejumlah matakuliah, tutorial dan kegiatan akademik sejenis dengan memperhatikan sejumlah aturan yang berhubungan dengan kapasitas dan lokasi dari ruangan yang tersedia, waktu yang diperlukan dan sejumlah aturan lain yang berkaitan dengan toleransi untuk dosen, dan hubungan antara matakuliah pilihan (Ross P., Dave C, Hasiao L.F - 1994).

Inti dari penjadwalan kuliah adalah bagaimana menjadwalkan sejumlah komponen yang terdiri atas matakuliah, mahasiswa, dosen, ruangan, dan waktu dengan sejumlah aturan dan batasan (*constraint*) tertentu.

Permasalahan penyusunan jadwal kuliah merupakan hal yang sangat kompleks, karena melibatkan banyak dosen, pembagian waktu, kapasitas serta jumlah ruangan serta banyak matakuliah yang ditawarkan tiap semester, semakin banyak komponen yang ada maka akan semakin banyak pula kombinasi dari komponen yang mungkin terjadi.

## 2.2 Algoritma Semut

### 2.2.1 Algoritma

Menurut kamus bahasa Indonesia algoritma adalah urutan logis pengambilan putusan untuk pemecahan masalah, dalam pengertian yang lain algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah, perintah-perintah ini dapat diterjemahkan secara bertahap dari

awal hingga akhir. Masalah tersebut dapat berupa apa saja dengan catatan untuk setiap masalah ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma, algoritma dapat berakhir untuk semua kondisi awal yang memenuhi kriteria.

Algoritma adalah kumpulan instruksi yang dibuat secara jelas untuk menunjukkan langkah-langkah penyelesaian suatu masalah. Algoritma memegang peranan penting dalam bidang pemrograman.

Dalam algoritma terdapat 5 komponen utama yaitu *finiteness*, *definiteness*, *input*, *output* dan *effectiveness*, sehingga dalam merancang sebuah algoritma ada 3 (tiga) komponen yang harus ada yaitu:

1. Komponen masukan (*input*)

Komponen ini biasanya terdiri dari pemilihan variable, jenis variable, tipe variable, konstanta dan parameter (dalam fungsi).

2. Komponen keluaran (*output*)

Komponen ini merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran., karakteristik keluaran yang baik adalah benar (menjawab) permasalahan dan tampilan yang mudah/ramah (*friendly*).

3. Komponen proses (*processing*)

Komponen ini merupakan bagian utama dan terpenting dalam merancang sebuah algoritma. Dalam bagian ini terdapat logika masalah, logika algoritma (sintaksis dan semantik), rumusan, metode (rekursi, perbandingan, penggabungan, pengurangan dan lain-lain).

Algoritma dalam bidang komputer, misalnya EDP (*Elektronik Data Processing*) atau MIS (*Management Information System*), algoritma sering dimanfaatkan untuk menyelesaikan suatu masalah atau untuk proses pengambilan keputusan. Sistem analisis (*analisis system*) tentunya menggunakan algoritma untuk merancang/membangun suatu sistem. Bagi programmer, algoritma digunakan untuk membuat modul-modul program.

### 2.2.2 Algoritma Semut (*Ant Algorithm*)

Semut merupakan jenis serangga sosial yang hidupnya berkoloni, secara individu tidak begitu berguna. Semut dapat bekerja sama dengan sesamanya secara efektif untuk melaksanakan sejumlah pekerjaan. Sebagai contoh, semut mampu untuk menemukan jalur terpendek dari suatu sumber makanan ke sarang mereka tanpa menggunakan petunjuk yang nyata dan kembali lagi ke sumber makanan tersebut. Mereka juga mampu untuk beradaptasi dengan perubahan yang terjadi di dalam lingkungan mereka, sebagai contoh menemukan jalur terpendek yang baru ketika yang lama sudah tidak memungkinkan lagi karena munculnya rintangan. (R.Beckers, J.L Deneubourg, S.Goss - 1992)

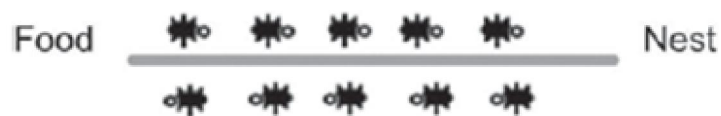
Algoritma semut yang diilhami oleh perilaku kerja sama yang diperlihatkan oleh *rill ant*, dalam melakukan berbagai aktifitas sehari-hari seperti pencarian makanan. Individual *rill ant* tidak mampu untuk melakukan suatu tugas terstruktur, tetapi suatu koloni semut mempunyai

potensi yang baik dalam menyelesaikan suatu aktifitas secara terkordinasi (M. Dorigo – 1992).

Media komunikasi utama dari *ant* adalah melalui alur yang dibuat dengan suatu bahan kimia unsur tiruan yang disebut *pheromone*, metode komunikasi tak langsung ini dikenal sebagai *stigmergy*. Kapan saja semut meninggalkan sarangnya untuk mencari makanan, semut tersebut akan meletakkan suatu *pheromone* di *trail* lintasan pada alur perjalanannya.

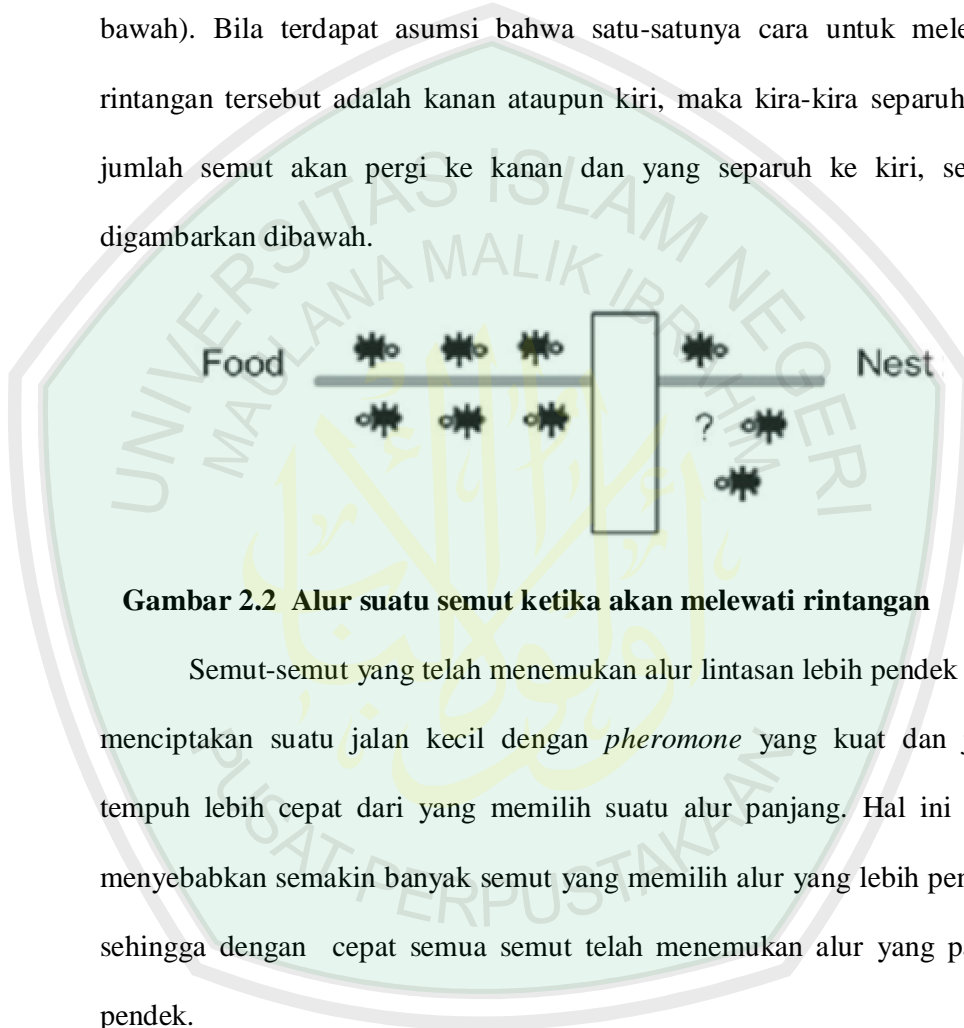
Banyaknya semut yang telah menempuh perjalanan pada alur, akan menentukan kekuatan dari *pheromone* yang ada pada *trail* tersebut. Setiap semut yang melewati lintasan di alur akan meninggalkan *pheromone* dan semakin menguatkan *pheromone* yang ada dan hal tersebut dapat membantu semut yang lain untuk mengikuti. Sebagai awal perjalanan semut melakukan randomisasi, semut yang tiba dengan alur terpendek merupakan suatu alur yang paling diikuti. Perilaku ini dikenal sebagai perilaku *autocatalytic* atau mekanisme umpan balik positif dimana penguatan menyangkut rute sebelumnya yang paling diikuti, menjadi rute yang paling diinginkan untuk pencarian masa depan.

Berikut adalah ilustrasi terhadap apa yang terjadi manakala suatu koloni semut yang mengikuti suatu alur paling pendek antara suatu sumber makanan dan sarang.



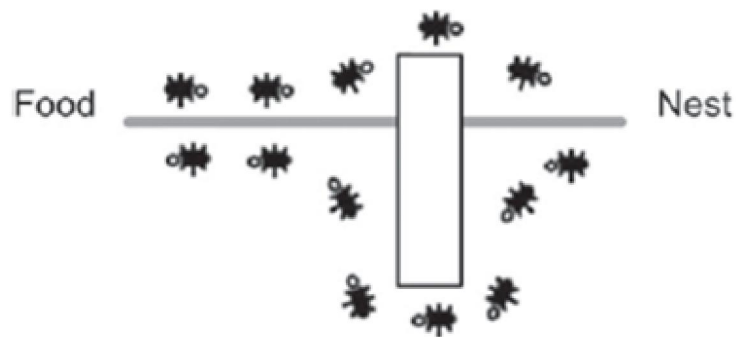
### Gambar 2.1 Alur semut ke sumber makanan dan sarangnya

Munculnya suatu rintangan yang terdapat dalam alur lintasan, semut akan secara acak memilih beberapa cara di sekitar itu (kanan, kiri, atas, atau bawah). Bila terdapat asumsi bahwa satu-satunya cara untuk melewati rintangan tersebut adalah kanan ataupun kiri, maka kira-kira separuh dari jumlah semut akan pergi ke kanan dan yang separuh ke kiri, seperti digambarkan dibawah.



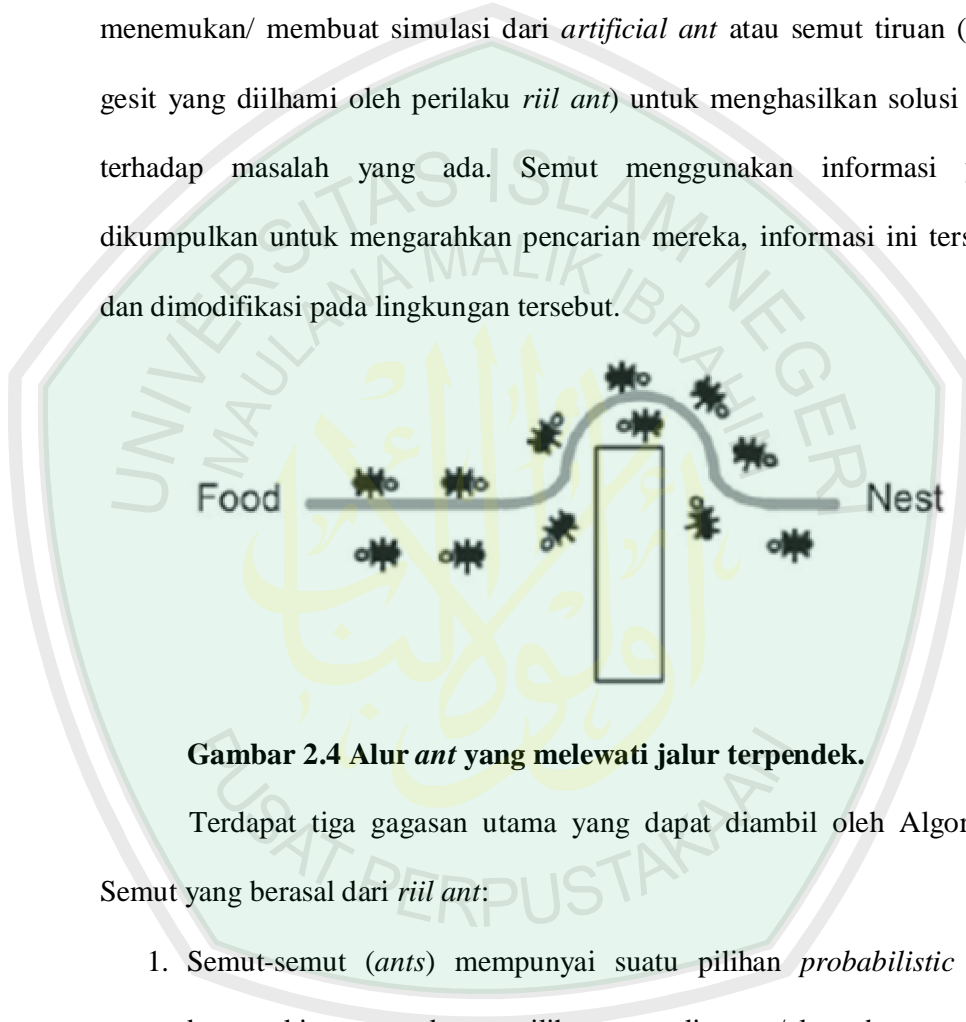
### Gambar 2.2 Alur suatu semut ketika akan melewati rintangan

Semut-semut yang telah menemukan alur lintasan lebih pendek akan menciptakan suatu jalan kecil dengan *pheromone* yang kuat dan jarak tempuh lebih cepat dari yang memilih suatu alur panjang. Hal ini akan menyebabkan semakin banyak semut yang memilih alur yang lebih pendek, sehingga dengan cepat semua semut telah menemukan alur yang paling pendek.



### Gambar 2.3 Alur semut yang melewati dua jalur.

Algoritma semut mencoba untuk menerapkan teknik serupa dalam rangka memecahkan permasalahan hidup nyata. Gagasan utamanya adalah menemukan/ membuat simulasi dari *artificial ant* atau semut tiruan (agen gesit yang diilhami oleh perilaku *riil ant*) untuk menghasilkan solusi baru terhadap masalah yang ada. Semut menggunakan informasi yang dikumpulkan untuk mengarahkan pencarian mereka, informasi ini tersedia dan dimodifikasi pada lingkungan tersebut.



### Gambar 2.4 Alur ant yang melewati jalur terpendek.

Terdapat tiga gagasan utama yang dapat diambil oleh Algoritma Semut yang berasal dari *riil ant*:

1. Semut-semut (*ants*) mempunyai suatu pilihan *probabilistic* atau kemungkinan untuk memilih suatu lintasan/alur dengan nilai *pheromone* yang tinggi.
2. Alur yang lebih pendek cenderung untuk suatu tingkat pertumbuhan nilai *pheromone* yang lebih tinggi.
3. Dengan menggunakan suatu system komunikasi tak langsung melalui *pheromonea edge*.



Perilaku koloni semut telah menginspirasi munculnya sebuah metodologi baru yang di dalamnya terdapat sekumpulan semut buatan, yang dinamai dengan *ants*, yang saling bekerja sama dalam mencari solusi terhadap suatu masalah optimisasi kombinatorial dengan cara bertukar informasi melalui *pheromone* yang diletakkan pada ruas-ruas sebuah graf. Oleh karena itu, sistem ini dinamai dengan *ant system* (AS) dan algoritma yang diperkenalkan ini disebut algoritma *ant*. (M. Dorigo, V. Maniezzo, A. Coloni - 1996)

*Ants* yang digunakan dalam sistem ini mempunyai perbedaan besar dengan hewan semut yang asli, antara lain *ants* akan memiliki memori, mereka tidak sepenuhnya buta, dan mereka akan berada pada lingkungan dimana waktunya adalah diskrit.

Berdasarkan perilaku *probabilistic* dari *rill ant*, algoritma semut adalah agen perangkat lunak yang mengkoordinir suatu jalur, dengan *update* informasi dari suatu memori umum, yang berupa *pheromone* di *trail* dari *rill ant* ketika sejumlah besar *artificial agent* sederhana ini mengkoordinir suatu jalur berdasarkan pada *updating* memori, maka semut-semut bisa membangun sebuah solusi terbaik untuk permasalahan optimisasi kombinatorial yang sulit.

*Pheromone* yang memiliki tingkat penguapan yang tak dikehendaki atau solusi mutu yang lemah akan dipindahkan dari ruang solusi. Ketika suatu memori kolektif bisa dibentuk, maka diijinkan *artificial ant* untuk menemukan solusi terbaik dalam periode waktu yang singkat atau pendek.

Setiap semut memiliki sebuah memori, yang dinamai *tabu list*, yang berisi semua tempat yang telah dikunjunginya pada setiap tur. *Tabu list* ini mencegah *ants* untuk mengunjungi tempat-tempat yang sebelumnya telah dikunjungi selama tur tersebut berlangsung, yang membuat solusinya menjadi mungkin.

Peranan utama dari penguapan *pheromone* tadi adalah untuk mencegah *stagnasi*, yaitu situasi dimana semua *ants* berakhir dengan melakukan tur yang sama. Proses di atas kemudian diulangi sampai tur-tur yang dilakukan mencapai jumlah maksimum (berdasarkan *user*) atau sistem ini menghasilkan perilaku *stagnasi* dimana sistem ini berhenti untuk mencari solusi alternatif.

Aturan transisi status yang digunakan oleh *ant system* dinamai *random-proportional rule*. Dari aturan yang terdapat dalam *ant system* dapat disimpulkan sebagai berikut:

- Didalam cakupan parameter keoptimalan, algoritma ini selalu menemukan solusi yang sangat bagus untuk semua permasalahan yang dicoba.
- Algoritma ini dengan cepat menemukan solusi yang bagus, meskipun demikian *ants* tidak memperlihatkan perilaku *stagnasi*, maksudnya *ants* terus mencari kemungkinan adanya tur baru yang lebih baik.
- Dengan meningkatkan dimensi masalah, kepekaan nilai-nilai parameter pada dimensi masalah tersebut diketahui menjadi sangat rendah.

Berdasarkan kesimpulan yang diperoleh dari *ant system*, maka dibuat *ant colony system* (ACS) dengan maksud untuk meningkatkan efisiensinya ketika diterapkan pada permasalahan yang lebih kompleks. Akan tetapi, ACS memiliki perbedaan dengan *ant system* yang sebelumnya karena tiga aspek utama. (M. Dorigo, L. M. Gambardella - 1997)

1. Aturan transisi status pada sistem ini memberikan suatu cara langsung untuk menyeimbangkan antara penjelajahan (*exploration*) ruas-ruas yang baru dengan eksploitasi (*exploitation*) dari sebuah *priori* dan pengetahuan yang dihimpun mengenai masalah tersebut.
2. Aturan pembaruan *pheromone* global hanya dilakukan pada ruas-ruas yang merupakan bagian dari tur terbaik.

Pembaruan *pheromone* secara global hanya dilakukan oleh semut yang membuat tur terpendek sejak permulaan percobaan. Pada akhir sebuah iterasi, setelah semua *ants* menyelesaikan tur mereka, sejumlah *pheromone* ditaruh pada ruas-ruas yang dilewati oleh seekor semut yang telah menemukan tur terbaik (ruas-ruas yang lain tidak diubah). Tingkat *pheromone* itu diperbarui dengan menerapkan aturan pembaruan *pheromone* global.

3. Disaat *ants* membangun sebuah solusi, diterapkan suatu aturan pembaruan *pheromone* lokal (*local pheromone updating rule*).

Peranan dari aturan pembaruan *pheromone* lokal ini adalah untuk mengacak arah tur-tur yang sedang dibangun, sehingga jalur-

jalur yang telah dilewati sebelumnya oleh tur seekor semut mungkin akan dilewati kemudian oleh tur *ants* yang lain. Dengan kata lain, pengaruh dari pembaruan lokal ini adalah untuk membuat tingkat ketertarikan ruas-ruas yang ada berubah secara dinamis: setiap kali seekor semut menggunakan sebuah ruas maka ruas ini dengan segera akan berkurang tingkat ketertarikannya (karena ruas tersebut kehilangan sejumlah *pheromone*-nya), secara tidak langsung *ants* yang lain akan memilih ruas-ruas lain yang belum dikunjungi. Konsekuensinya, *ants* tidak akan memiliki kecenderungan untuk berkumpul pada jalur yang sama.

Salah satu contoh yang paling berhasil dari algoritma semut adalah *Ant Colony Optimization* (ACO). ACO digunakan untuk merancang algoritma metaheuristik pada permasalahan pengoptimuman kombinasi.

Algoritma metaheuristik merupakan algoritma yang digunakan untuk memecahkan masalah pada optimum lokal, yang merujuk pada beberapa dasar heuristic antara lain: pembangunan heuristic diawali dari solusi *null* dan menambahkan beberapa element untuk mendapatkan kelengkapan yang baik, atau pencarian heuristic lokal yang diawali dari solusi yang lengkap dan memodifikasi beberapa element untuk mendapatkan yang lebih baik.

Ciri penting dari algoritma ACO adalah kombinasi informasi sebuah struktur *priori* dari suatu peluang solusi terhadap informasi *posteriori* sebelum didapatkan solusi yang baik, serta penggunaannya secara eksplisit sebelum didapatkan solusi dari element-element yang ada.

Algoritma semut mempunyai keistimewaan antara lain:

1. Algoritma ini merupakan suatu algoritma alami karena didasarkan pada perilaku *ants* dalam menetapkan alur dari koloni mereka, untuk membawa makanan ke sumber tinggal mereka dan kembali lagi sampai makanan di koloni mereka habis.
2. Sifat *parallel* dan *distributed* karena berhubungan dengan suatu populasi agen yang bergerak secara serempak, dengan bebas dan tanpa adanya penyelia.
3. Bekerja sama karena masing-masing agen memilih suatu alur atas dasar informasi *pheromone trail* yang diletakkan oleh agen yang lain, dimana sebelumnya telah memilih alur yang sama. Perilaku kerja sama ini juga *autocatalytic* yaitu menyediakan suatu umpan balik positif, karena kemungkinan suatu agen yang memilih suatu alur itu meningkat dengan banyaknya agen-agen yang sebelumnya memilih alur itu.
4. Dapat diterapkan secara sempurna dengan perubahan minimal ke permasalahan optimasi kombinasi lain seperti *Quadratic Assignment Problem (QAP)* dan *Job-Shop Scheduling Problem (JSP)*. (Hariyadi. M.Amin - 2007)

Dalam algoritma semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek, yaitu:

**Langkah 1 :**

- a. Inisialisasi harga parameter-parameter algoritma.

Parameter-parameter yang diinisialisasikan adalah :

1. Intensitas jejak semut antar titik dan perubahannya ( $\tau_{ij}$ ).
  2. Banyak titik ( $n$ ) termasuk koordinat ( $x,y$ ) atau jarak antar titik ( $d_{ij}$ ).
  3. titik berangkat dan titik tujuan.
  4. Tetapan siklus semut ( $Q$ ).
  5. Tetapan pengendali intensitas jejak semut ( $\alpha$ ), nilai  $0 < \alpha < 1$ .
  6. Tetapan pengendali visibilitas ( $\beta$ ), nilai  $0 < \beta < 1$ .
  7. Visibilitas antar wilayah =  $1/d_{ij}$  ( $\eta_{ij}$ ).
  8. Banyak semut ( $m$ ).
  9. Tetapan penguapan jejak semut ( $\rho$ ), nilai  $\rho$  harus  $> 0$  dan  $< 1$  untuk mencegah jejak pheromone yang tak terhingga.
  10. Jumlah siklus maksimum ( $NC_{max}$ ) bersifat tetap selama algoritma dijalankan, sedangkan  $\tau_{ij}$  akan selalu diperbaharui nilainya pada setiap siklus algoritma mulai dari siklus pertama ( $NC=1$ ) sampai tercapai jumlah siklus maksimum ( $NC=NC_{max}$ ) atau sampai terjadi konvergensi.
- b. Inisialisasi titik pertama setiap semut.

Setelah inisialisasi  $\tau_{ij}$  dilakukan, kemudian  $m$  semut ditempatkan pada titik pertama tertentu secara acak.

## Langkah 2 :

Pengisian titik pertama ke dalam *tabu list*. Hasil inisialisasi titik pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen

pertama *tabu list*. Hasil dari langkah ini adalah terisinya elemen pertama *tabu list* setiap semut dengan indeks titik tertentu, yang berarti bahwa setiap  $tabu_k (I)$  bisa berisi indeks titik antara 1 sampai  $n$  sebagaimana hasil inisialisasi pada langkah 1.

**Langkah 3 :**

Penyusunan rute kunjungan setiap semut ke setiap titik. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap titik, akan mulai melakukan perjalanan dari titik pertama, masing-masing sebagai titik asal dan salah satu titik lainnya sebagai titik tujuan, kemudian dari titik kedua masing-masing, koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari wilayah yang tidak terdapat pada  $tabu_k$  sebagai titik tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua titik satu persatu dikunjungi atau telah menempati  $tabu_k$ . Jika  $s$  menyatakan indeks urutan kunjungan, titik asal dinyatakan sebagai  $tabu_k(s)$  dan titik-titik lainnya dinyatakan sebagai  $\{N-tabu_k\}$ , maka untuk menentukan titik tujuan digunakan persamaan probabilitas titik untuk dikunjungi sebagai berikut :

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k' \in \{N-tabu_k\}} [\tau_{ik'}]^\alpha \cdot [\eta_{ik'}]^\beta} \quad \text{Untuk } j \in \{N-tabu_k\} \dots\dots\dots (1)$$

$$p_{ij}^k = 0, \text{ Untuk } j \text{ lainnya} \dots\dots\dots (2)$$

dengan i sebagai indeks titik asal dan j sebagai indeks titik tujuan.

**Langkah 4 :**

a. Perhitungan panjang rute setiap semut.

Perhitungan panjang rute tertutup (*length closed tour*) atau  $L_k$  setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini dilakukan berdasarkan  $tabu_k$  masing-masing dengan persamaan berikut :

$$L_k = d_{tabu_k(n), tabu_k(1)} + \sum_{s=1}^{n-1} d_{tabu_k(s), tabu_k(s-1)} \dots\dots\dots (3)$$

dengan  $d_{ij}$  adalah jarak antara titik i ke titik j yang dihitung berdasarkan persamaan :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \dots\dots\dots (4)$$

b. Pencarian rute terpendek.

Setelah  $L_k$  setiap semut dihitung, akan didapat harga minimal panjang rute tertutup setiap siklus atau  $L_{minNC}$  dan harga minimal panjang rute tertutup secara keseluruhan adalah atau  $L_{min}$ .

c. Perhitungan perubahan harga intensitas jejak kaki semut antar titik.



Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar titik yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan nilai intensitas jejak kaki semut antar titik. Persamaan perubahan ini adalah :

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \dots\dots\dots (5)$$

dengan  $\Delta\tau_{ij}^k$  adalah perubahan harga intensitas jejak kaki semut antar titik setiap semut yang dihitung berdasarkan persamaan:

$$\Delta\tau_{ij}^k = \frac{Q}{L_k}, \text{ untuk } (i,j) \in \text{kota asal dan titik tujuan dalam } tabu_k \dots\dots (6)$$

$$\Delta\tau_{ij}^k, \text{ untuk } (i,j) \text{ lainnya} \dots\dots\dots (7)$$

**Langkah 5 :**

- a. Perhitungan harga intensitas jejak kaki semut antar titik untuk siklus selanjutnya. Harga intensitas jejak kaki semut antar titik pada semua lintasan antar titik ada kemungkinan berubah karena adanya penguapan dan perbedaan

jumlah semut yang melewati. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah.

Harga intensitas jejak kaki semut antar titik untuk siklus selanjutnya dihitung dengan persamaan :

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij} \dots\dots\dots (8)$$

- b. Atur ulang harga perubahan intensitas jejak kaki semut antar titik. Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar titik perlu diatur kembali agar memiliki nilai sama dengan nol.

**Langkah 6 :**

Pengosongan *tabu list*, dan ulangi langkah 2 jika diperlukan. *Tabu list* perlu dikosongkan untuk diisi lagi dengan urutan titik yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi. Algoritma diulang lagi dari langkah 2 dengan harga parameter intensitas jejak kaki semut antar titik yang sudah diperbaharui.

Penampilan algoritma semut disesuaikan pada beberapa parameter yaitu:  $\alpha$ ,  $\beta$ , lintasan penting yang dilalui serta menarik,  $\rho$ , persistensi lintasan,  $\tau_{ij}(0)$ , tingkat identitas lintasan,  $m$ , banyaknya jumlah semut, dan  $Q$ . Parameter-parameter tersebut digunakan untuk memperjelas solusi-solusi yang baik serta dengan perhitungan yang minimum.

Algoritma tersebut mengacu pada:

1. {Initialization}

Initialize  $\tau_{ij}$  and  $\tau_{ij}, \forall (ij)$ .

2. {Construction}

For each ant  $k$  ( current in state  $i$ ) do

repeat

choose in probability the state to move into.

append the chosen move to the  $k$ -th ant's set

$tabu_k$ .

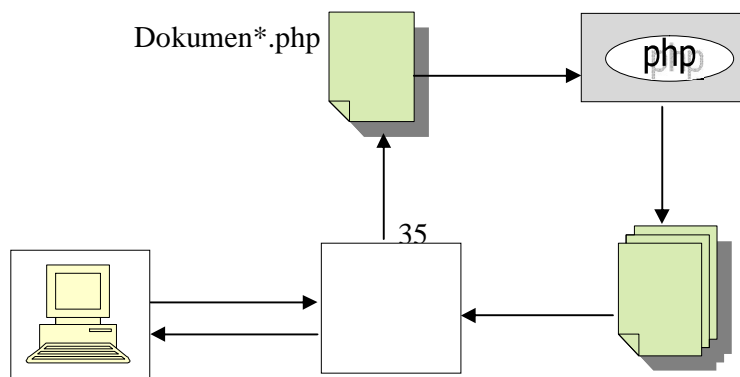
until ant  $k$  has completed its solution.

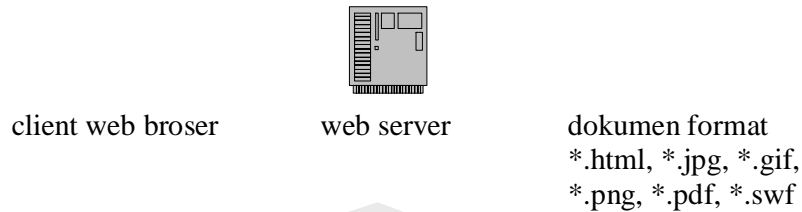
```
end for  
3. {Trail update}  
For each ant move (ij) do  
    compute  $\tau_{ij}$   
    update the trail matrix.  
end for  
4. {Terminating condition}  
If not(end test)go to step 2
```

### 2.3 PHP

PHP (*Hypertext Preprocessor*) dikembangkan pertama kali tahun 1995 oleh Rasmus Lerdorf yang merupakan salah satu anggota group Apache. PHP pertama kali didesain sebagai alat *tracking* penunjang web site Lerdorf, yang kemudian fungsinya diperlebar dan dihubungkan dengan Apache. PHP sepenuhnya dikembangkan untuk bahasa skrip *side-server programming*. PHP bersifat *open source* dan dapat digabungkan dengan berbagai server yang berbeda-beda.

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan berada di server (*server-side HTML embedded scripting*). Artinya sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada halaman HTML biasa. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi-aplikasi yang dijalankan diatas teknologi web. Dalam hal ini, aplikasi pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan di web server. (Sunnyoto. Andi - 2007)





**Gambar 2.5 Ilustrasi skrip PHP dieksekusi**

Skrip PHP memiliki beberapa kelebihan, antara lain:

- Mudah dibuat dan dijalankan.
- Mampu berjalan pada web server dengan sistem operasi yang berbeda-beda (UNIX, Windows dan Macintosh).
- PHP bisa didapatkan dengan gratis.
- Dapat berjalan pada web server yang berbeda seperti Microsoft personal Web server, Apache, IIS, Xitami, dan lain-lain.
- Dapat di-*embedded* (diletakkan pada tag HTML).

## 2.4 MySQL

MySQL adalah salah satu database server *open source* yang populer. MySQL merupakan perangkat lunak yang digunakan untuk membangun database yang sering digunakan di lingkungan linux maupun windows.

Sebuah database mendefinisikan struktur untuk menyimpan informasi. Dalam database ada beberapa tabel dan table pada HTML, sebuah tabel berisi baris, kolom (*field*) dan *cell*. Sebuah database biasanya berisi lebih dari satu tabel dan mempunyai nama yang berbeda. Masing-masing tabel berisi baris dan data.

(Sunyoto. Andi - 2007)

## BAB III

### DESAIN SISTEM

#### 3.1 Analisis Kebutuhan Perangkat Lunak

##### 3.1.1 Metode Analisis

Aplikasi untuk menentukan jalur terpendek ini dirancang dengan menggunakan algoritma semut. Untuk melihat proses aplikasi yang mencakup proses input dan proses output dinyatakan dengan diagram sistem yang diperjelas dengan diagram alir (*flow chart*). Pada tahap ini digunakan notasi-notasi untuk menggambarkan arus data dimana akan sangat membantu dalam proses komunikasi dengan pemakai.

Diagram alir (*flow chart*) digunakan untuk menggambarkan sistem baru yang akan dikembangkan secara logis tanpa mempertimbangkan terlebih dahulu lingkungan fisik dimana sistem ini akan digunakan.

##### 3.1.2 Hasil Analisis

Dari data yang diperoleh melalui survey selama penelitian dan setelah dilakukan proses analisis yang terdiri dari kebutuhan input, kebutuhan analisis, dan kebutuhan output, yaitu :

##### 3.1.2.1 Analisis Kebutuhan Input

Input dari aplikasi penentuan jalur terpendek ini, berupa parameter-parameter yang diperlukan dalam penjadwalan kuliah dan algoritma semut yaitu :

1. Fakultas, yang meliputi:

- Kode Fakultas
- Nama Fakultas

2. Jurusan, yang meliputi:

- Kode Jurusan
- Nama Jurusan

3. Program Studi

- Kode Prodi
- Kode Fakultas
- Nama Prodi

4. Data Dosen, yang meliputi:

- Nomor Induk Pegawai (Dosen)
- Nama Dosen
- Alamat
- Nomor Telpon

5. Data Mata Kuliah, yang meliputi:

- Kode Matakuliah
- Nama Matakuliah
- ID Prodi
- Jumlah SKS

- Semester
6. Data Ruang, yang meliputi:
    - Kode Ruang
    - Nama Gedung
    - Nama Ruang
  7. Data Semester (tahun) yang meliputi:
    - Kode Matakuliah
    - Tahun
  8. Data Kelas Mengajar meliputi:
    - Nomor Induk Pegawai (Dosen)
    - ID Matakuliah Tahun
  9. Data Kesiediaan Mengajar meliputi:
    - Nomor Induk Pegawai (Dosen)
    - Hari Bersedia Mengajar
    - Jam Awal Mengajar
    - Jam Akhir Mengajar

Selain input yang berkaitan dengan penjadwalan kuliah, dalam *page* untuk administrator juga terdapat form untuk mengolah data artikel serta berita seputar kampus dan anggota, antara lain:

1. Form berita, yang meliputi:
  - ID Berita
  - Judul Berita
  - Isi Berita

- Editor
2. Form Artikel, yang meliputi:

- ID Artikel
- Judul Artikel
- Isi Artikel
- Pengirim

3. Anggota, meliputi:

- Nomor Induk Anggota
- Nama
- Username
- Password
- E-mail

### 3.1.2.2 Analisis Kebutuhan Proses

Untuk mendapatkan hasil penjadwalan kuliah yang optimal dengan menggunakan algoritma semut dibutuhkan parameter-parameter antara lain:

1. Intensitas jejak semut antar titik ( $\tau_{ij}$ ) dan perubahannya ( $\Delta\tau_{ij}$ )  
 $\tau_{ij}$  harus diinisialisasi sebelum memulai siklus.  $\tau_{ij}$  digunakan dalam persamaan probabilitas titik yang akan dikunjungi.  $\tau_{ij}$  diinisialisasi setelah selesai satu siklus.  $\tau_{ij}$  digunakan untuk menentukan  $\tau_{ij}$  untuk siklus selanjutnya.
2. Tetapan siklus semut (Q)



Q merupakan konstanta yang digunakan dalam persamaan untuk menentukan  $ij$ . Nilai Q ditentukan oleh pengguna.

3. Tetapan pengendali intensitas jejak semut ( )

digunakan dalam persamaan probabilitas titik yang akan dikunjungi yang berfungsi sebagai pengendali intensitas jejak semut. Nilai  $ij$  ditentukan oleh pengguna.

4. Tetapan pengendali visibilitas ( )

digunakan dalam persamaan probabilitas titik yang akan dikunjungi dan berfungsi sebagai pengendali visibilitas. Nilai  $ij$  ditentukan oleh pengguna.

5. Visibilitas antar titik (  $ij$  )

$ij$  digunakan dalam persamaan probabilitas titik yang akan dikunjungi. Nilai  $ij$  merupakan hasil dari  $1/dij$  (jarak antar titik).

6. Banyak semut (m)

m merupakan banyak semut yang akan melakukan siklus dalam algoritma semut. Nilai m ditentukan oleh pengguna.

7. Tetapan penguapan jejak semut ( )

digunakan untuk menentukan  $ij$  untuk siklus selanjutnya. Nilai  $ij$  ditentukan oleh pengguna.

8. Jumlah siklus maksimum ( $NC_{max}$ )

$NC_{max}$  adalah jumlah maksimum siklus yang akan berlangsung. Siklus akan berhenti sesuai dengan  $NC_{max}$  yang telah ditentukan atau telah konvergen. Nilai  $NC_{max}$  ditentukan oleh pengguna.

### 3.1.2.3 Analisis Kebutuhan Output

Data output yang diperoleh dari proses aplikasi penentuan jalur terpendek ini adalah kombinasi antara kode matakuliah, nama matakuliah, nama dosen, hari, waktu (jam kuliah dilaksanakan), jumlah SKS, ruang yang digunakan dan kelas sehingga dihasilkan jadwal kuliah yang optimum. Selain itu sebagai informasi juga disajikan berita seputar kampus serta artikel-artikel yang bermanfaat bagi *user*.

### 3.1.3 Kebutuhan Antar Muka

Perancangan antar muka dengan menggunakan PHP, javascript serta css merupakan pilihan yang tepat untuk mengimplementasikan pencarian jalur terpendek berbasis web. Kelebihan lain dari tampilan antar muka berbasis web ini adalah tampilan yang memudahkan bagi administrator serta penggunaannya untuk menggunakan sistem ini.

## 3.2 Perancangan Perangkat Lunak

### 3.2.1 Metode Perancangan

Metode perancangan yang dikembangkan untuk membangun pencarian jalur terpendek dengan menggunakan *context diagram* dan *data*

*flow diagram* (DFD) yang kemudian diperjelas dengan *flow chart* atau perancangan terstruktur (*structure design method*). *Flow chart* pada dasarnya merupakan konsep perancangan yang mudah dengan penekanan pada sistem modular (*Top Down Design*) dan pemrograman terstruktur (*structure programming*).

### **3.2.2 Perancangan**

Berdasarkan analisis yang telah dilakukan maka dapat diketahui apa saja yang menjadi masukan sistem, keluaran sistem, metode yang digunakan sistem, serta sistem antar muka (*interface*) yang dibuat, sehingga sistem yang dibuat nantinya sesuai dengan yang diharapkan.

Perancangan sistem ini akan dibagi menjadi beberapa subsistem yaitu :

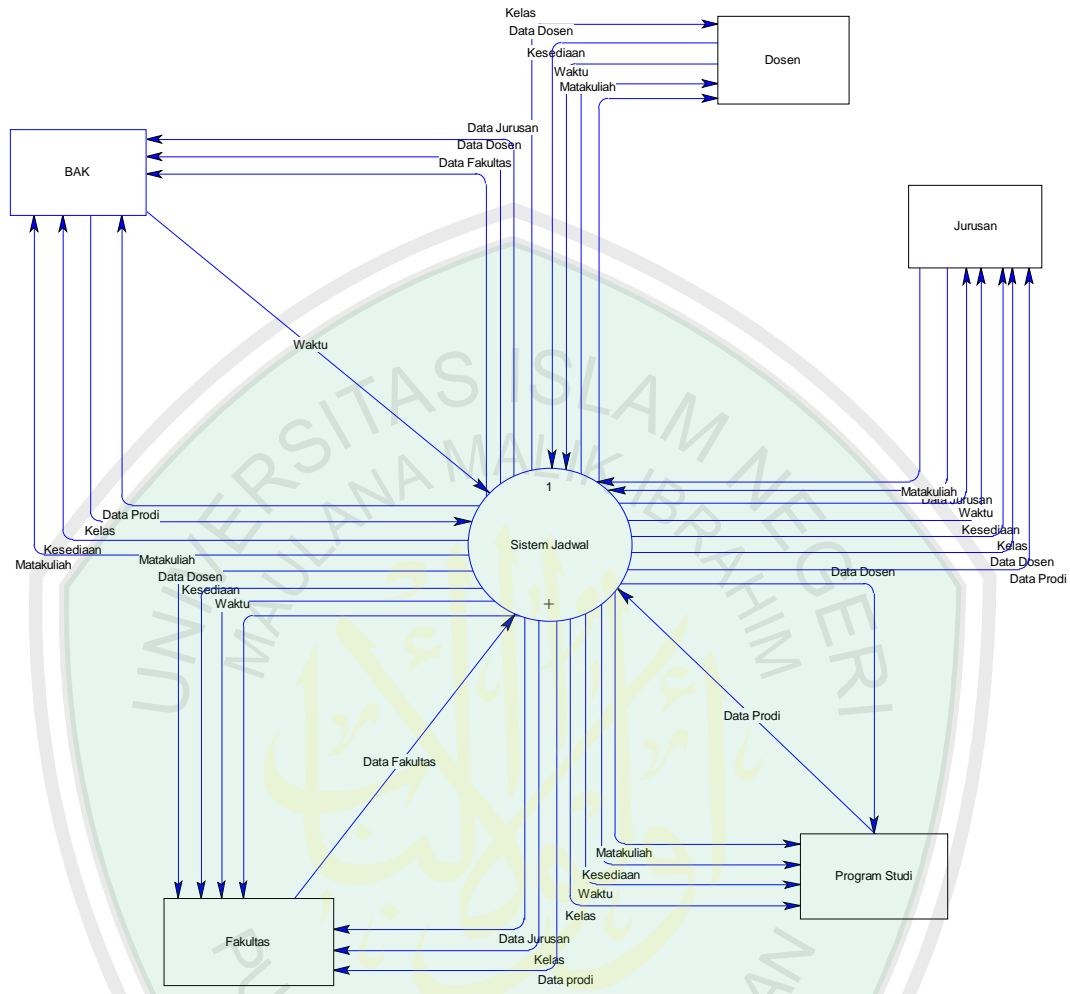
#### **3.2.2.1 Perancangan Desain Sistem**

Untuk lebih menjelaskan perancangan sistem yang dibangun maka dipergunakan context diagram dan data flow diagram.

##### **a. Context Diagram**

Context diagram dibuat untuk mengetahui entity external dan mengetahui aliran data dari dan menuju sistem, tetapi tidak memperlihatkan proses secara mendetail dan penyimpanan data.

Untuk menggambarkan secara umum sistem penjadwalan kuliah context diagramnya sebagai berikut:



**Gambar 3.1 Context Diagram**

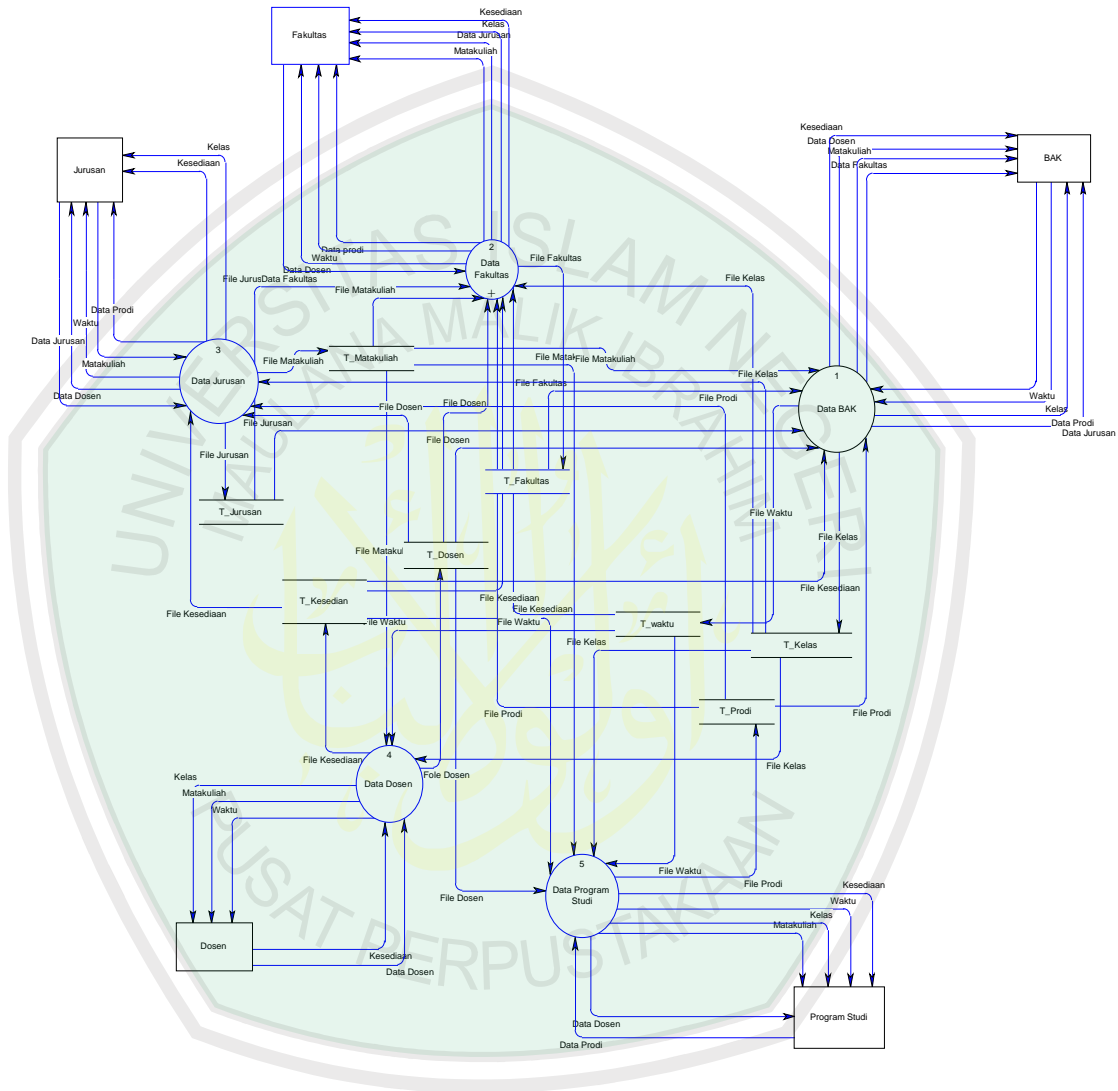
**b. Data Flow Diagram (DFD)**

Data flow diagram merupakan diagram yang menggambarkan aliran data dan proses-proses yang berkaitan dengan perubahan atau informasi data secara keseluruhan pada sebuah sistem.

Dari pengertian diatas dapat diketahui pengertian DFD adalah mendesai programdan database.

DFD pada sistem penjadwalan kuliah hanya terdiri dari satu

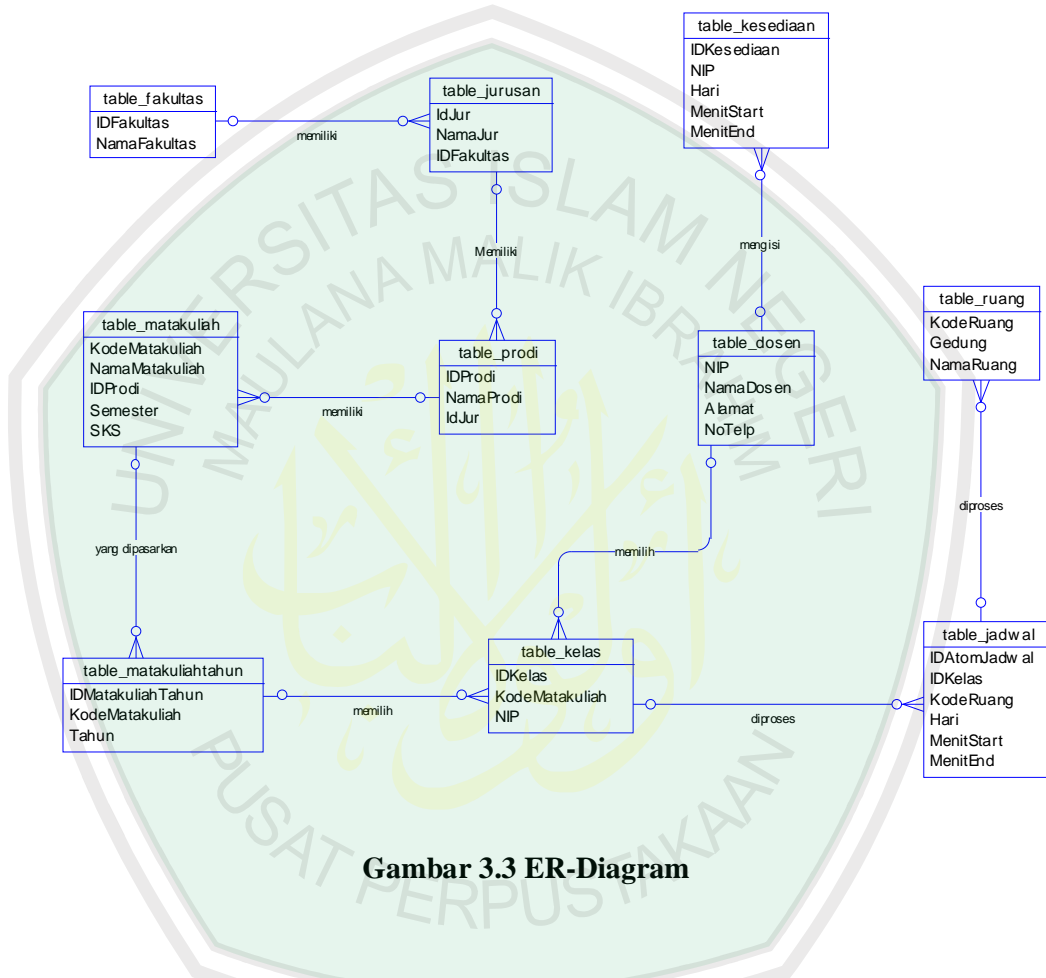
level yaitu:



Gambar 3.2 Data Flow Diagram

### c. Entity Relationship Model (ER)

Pembuatan ER dimaksudkan agar dapat memodelkan suatu database yang didalamnya terdapat relasi yang dihubungkan oleh *field* kunci.

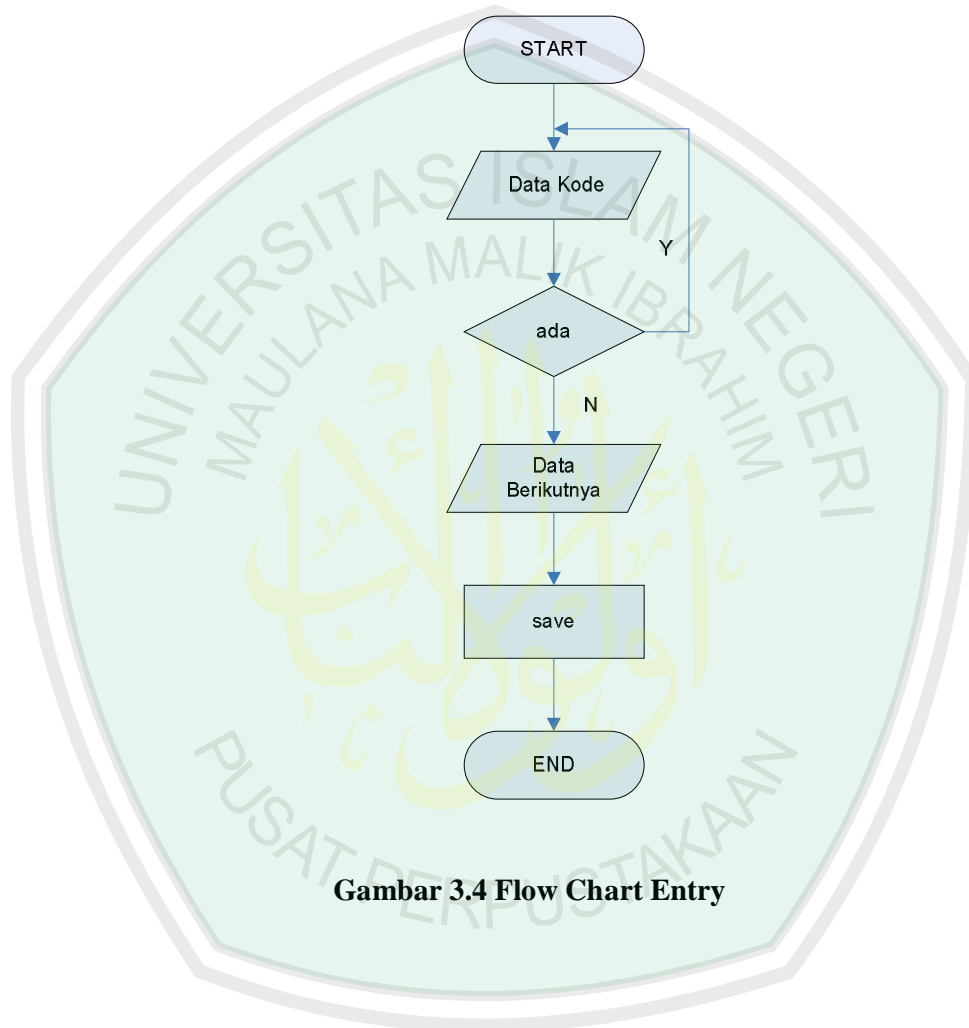


**Gambar 3.3 ER-Diagram**

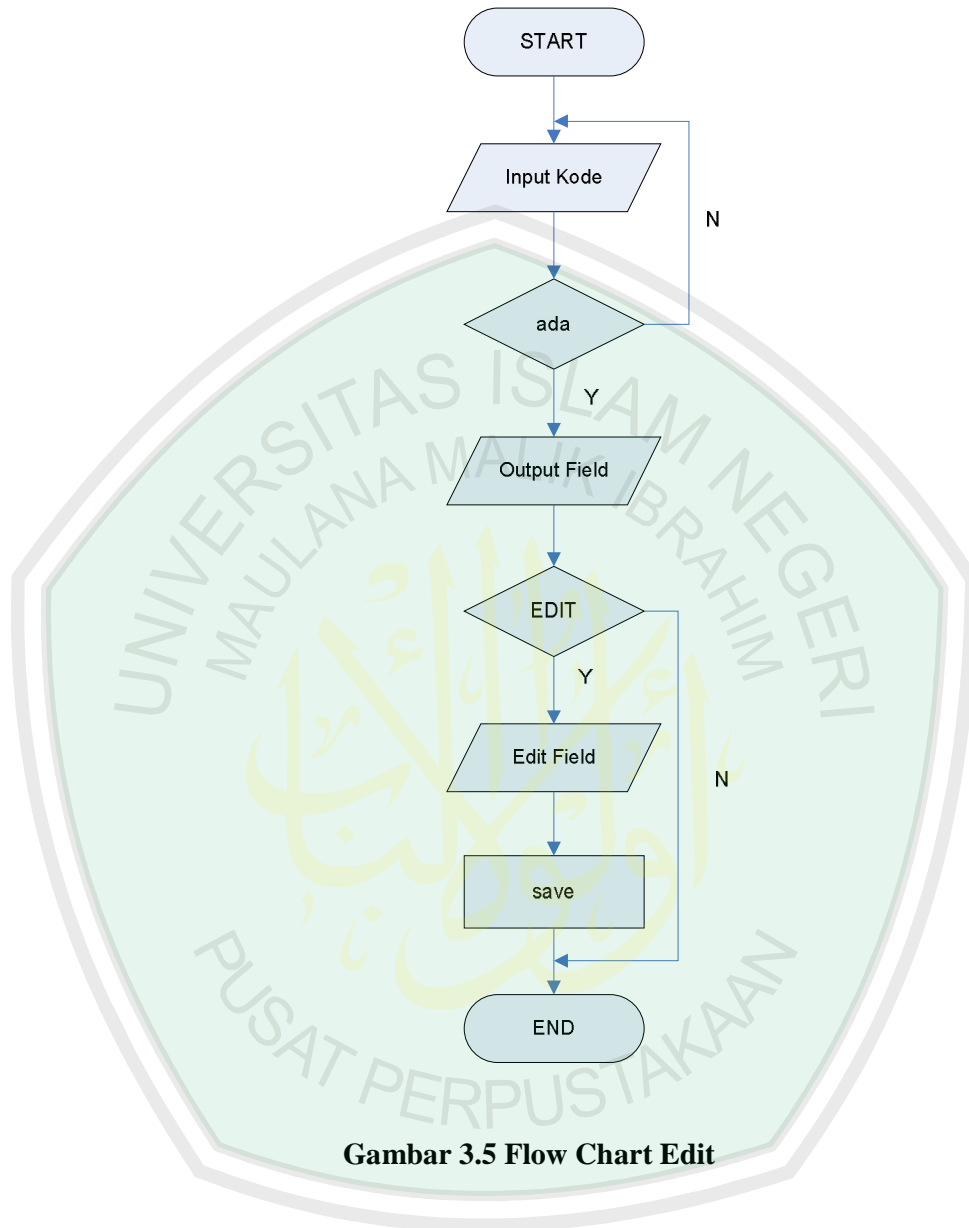
**d. Flow Chart**

Pada halaman admin terdapat beberapa form untuk memasukkan (*entry*) data yang dibutuhkan, *update* data yang sudah ada pada database dan menghapus data yang tidak dibutuhkan lagi dalam proses penjadwalan.

Pembuatan *Flow Chart* ditujukan untuk memperjelas urutan sistem program secara umum, untuk lebih jelasnya dapat dilihat pada gambar berikut:

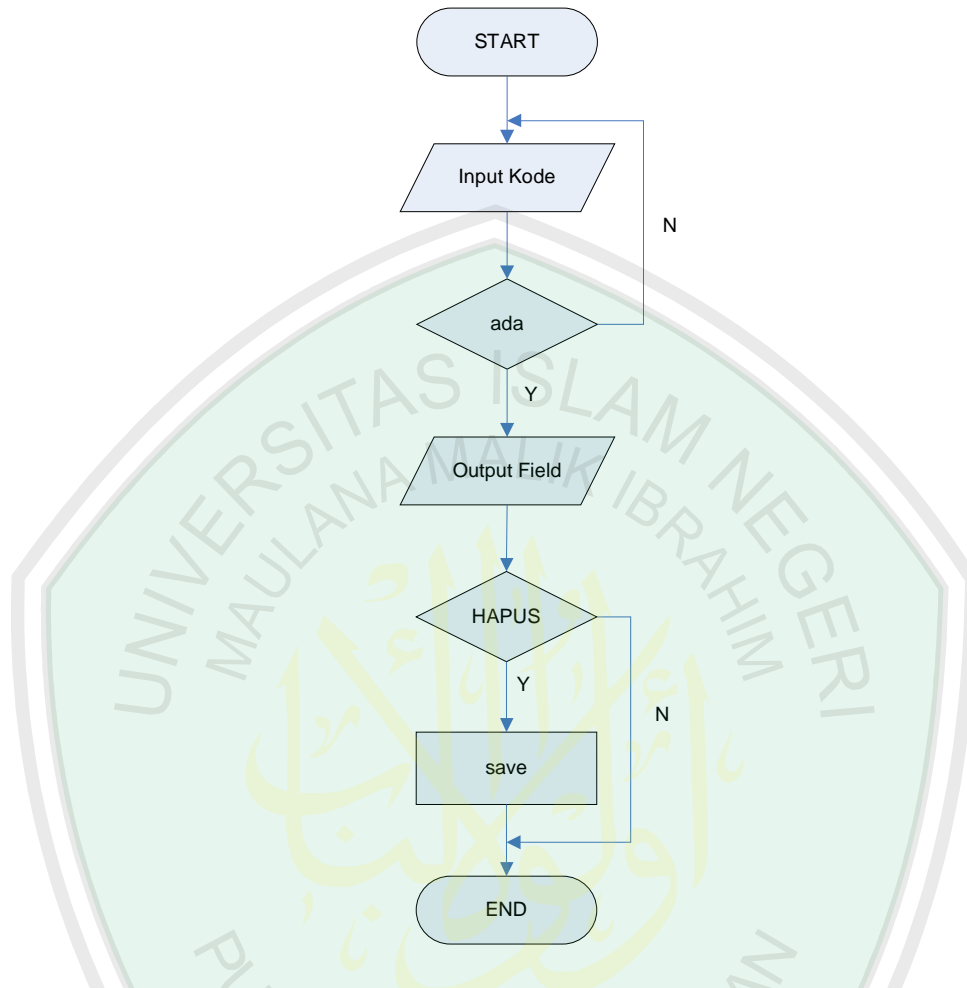


**Gambar 3.4 Flow Chart Entry**



Gambar 3.5 Flow Chart Edit





**Gambar 3.6 Flow Chart Delete**

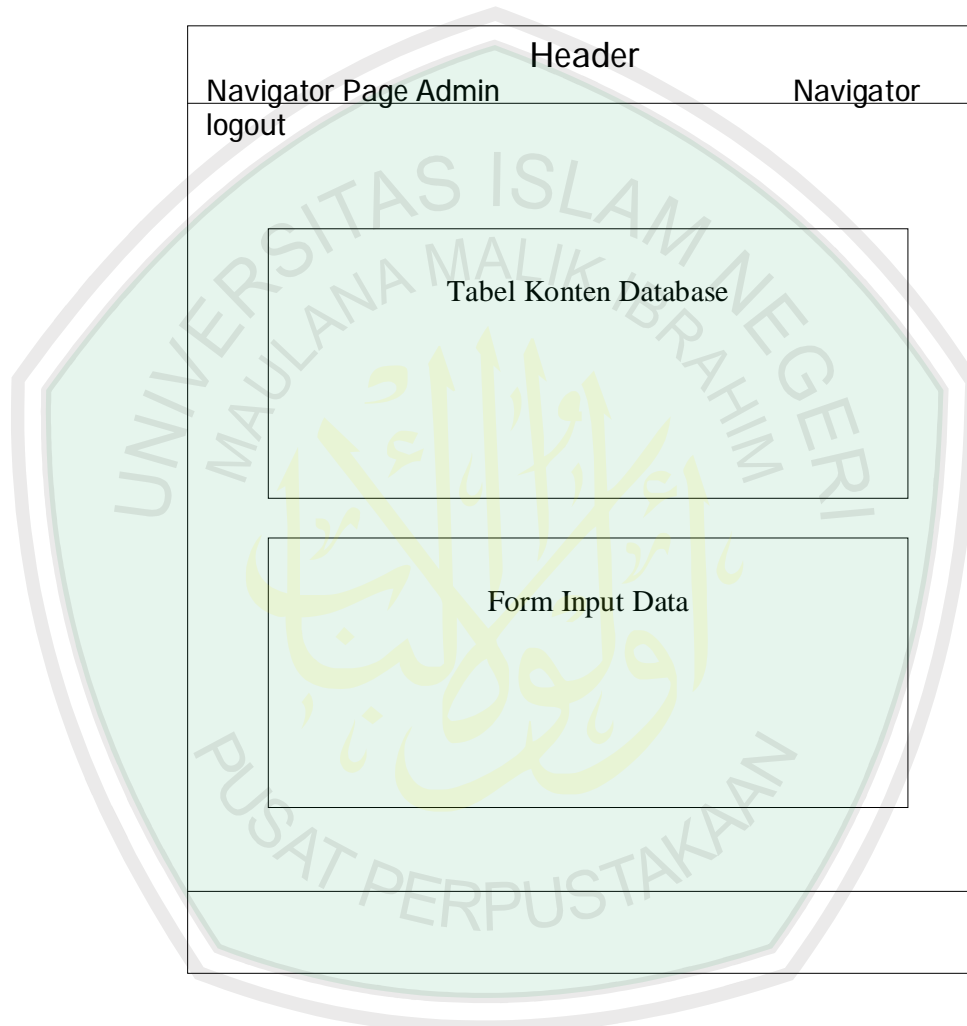
### 3.2.2.2 Algoritma

Algoritma semut mempunyai urutan kerja secara global sebagai berikut:

```
procedure ACO_Metaheuristic
  while (not_termination)
    generateSolution ()
    pheromoneUpdate ()
    daemonActions ()
  end while
end prosedure
```

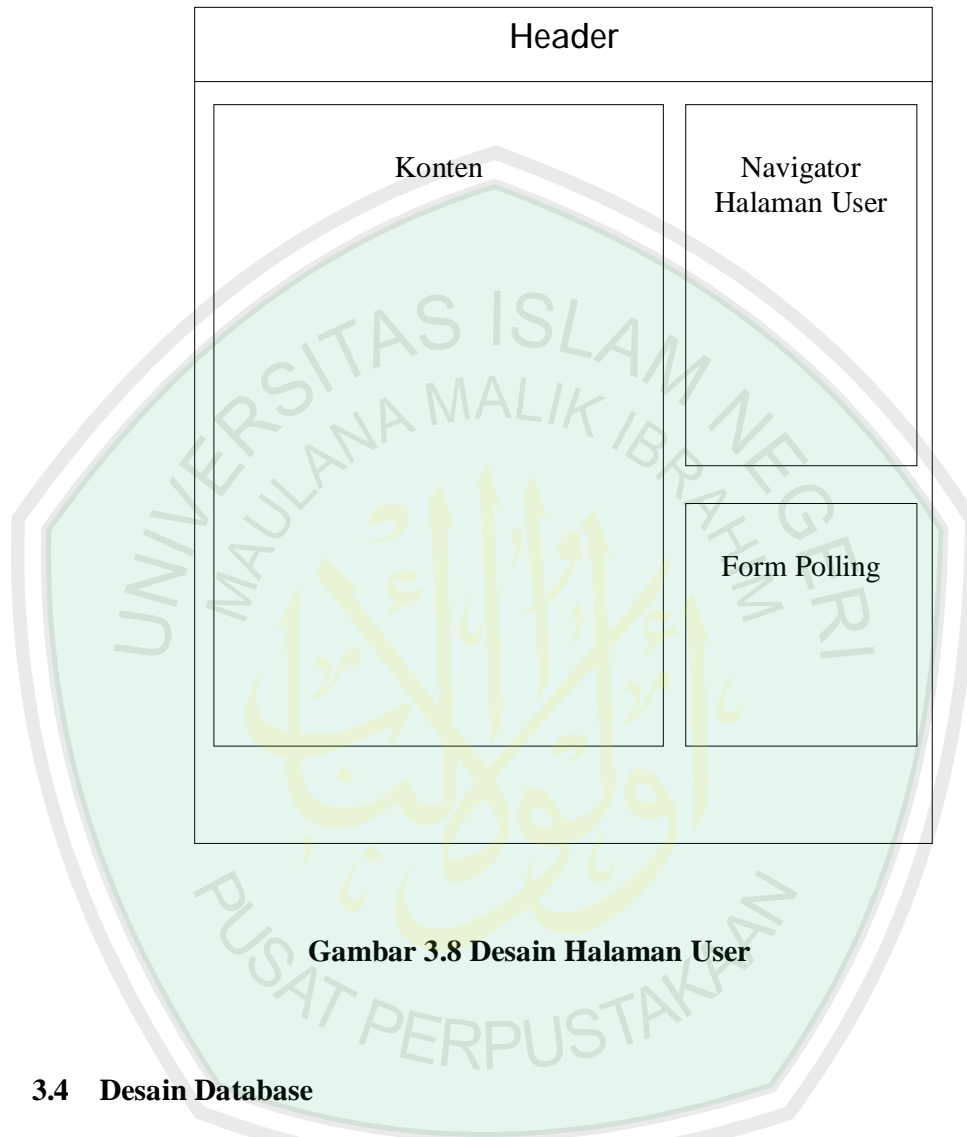
### 3.3 Desain Tampilan (*Interface*)

#### 3.3.1 Halaman Administrator



**Gambar 3. 7 Desain Halaman Administrator**

#### 3.3.2 Halaman User



**Gambar 3.8 Desain Halaman User**

### **3.4 Desain Database**

Pada pengerjaan skripsi ini terdapat satu database yang di dalamnya terdapat 10 tabel utama dan 6 tabel pendukung, antara lain:

#### **4.4.1 Tabel Utama**

Ø Fakultas

Nama	Type	Size	Keterangan
IDFakultas	Varchar	2	ID Fakultas ( <b>Primary Key</b> )
NamaFakultas	Varchar	20	Nama Fakultas

Ø Jurusan

Nama	Type	Size	Keterangan
IdJur	Varchar	4	ID Jurusan ( <b>Primary Key</b> )
NamaJur	Varchar	25	Nama Jurusan

Ø Program Studi

Nama	Type	Size	Keterangan
IDProdi	Varchar	4	ID Jurusan ( <b>Primary Key</b> )
IDFakultas	Varchar	2	ID Fakultas ( <b>Foreign Key</b> )
NamaProdi	Varchar	25	Nama Program Studi

Ø Dosen

Nama	Type	Size	Keterangan
NIP	Varchar	12	Nomor Induk Pegawai ( <b>Primary Key</b> )
NamaDosen	Varchar	25	Nama Dosen
Alamat	Varchar	40	Alamat Dosen
NoTelp	Varchar	20	Nomor Telpon

Ø Matakuliah

Nama	Type	Size	Keterangan
KodeMatakuliah	Varchar	6	Kode Matakuliah ( <b>Primary Key</b> )
IDProdi	Varchar	6	Id Program Studi ( <b>Foreign Key</b> )
NamaMatakuliah	Varchar	30	Nama Matakuliah
SKS	Smallint	6	Jumlah semester
Semester	Smallint	6	Semester

Ø Matakuliah Tahun

Nama	Type	Size	Keterangan
IDMatakuliahTahun	Int	11	ID Matakuliah Tahun( <b>Primary Key</b> )

IDMatakuliah Tahun	Varchar Int	6 11	ID Matakuliah ( <b>Foreign Key</b> ) Tahun Ajaran
-----------------------	----------------	---------	--

Ø Kelas

Nama	Type	Size	Keterangan
IDKelas	Int	11	Kode Ruang ( <b>Primary Key</b> )
IDMatakuliahTahun	Int	11	Matakuliah ( <b>Foreign Key</b> )
NIP	Varchar	12	Dosen ( <b>Foreign Key</b> )

Ø Ruang

Nama	Type	Size	Keterangan
KodeRuang	Varchar	5	Kode Ruang ( <b>Primary Key</b> )
Gedung	Varchar	20	Nama Gedung
NamaRuang	Varchar	20	Nama Ruang

Ø Kesediaan

Nama	Type	Size	Keterangan
IDKesediaan	Int	11	Id Kesediaan ( <b>Primary Key</b> )
NIP	Varchar	12	NIP Dosen ( <b>Foreign Key</b> )
Hari	Smallint	6	Hari Aktif
MenitStart	Int	11	Waktu Awal
MenitEnd	Int	11	Waktu Akhir

Ø Jadwal

Nama	Type	Size	Keterangan
------	------	------	------------

IDAtomJadwal	Int	11	Id Jadwal ( <b>Primary Key</b> )
IDKelas	Int	11	Id Kelas ( <b>Foreign Key</b> )
KodeRuang	Varchar	5	Kode Ruang
Hari	Smallint	6	Hari
MenitStart	Int	11	Waktu Awal
MenitEnd	Int	11	Waktu Akhir

### 3.4.1 Tabel Pendukung

#### Ø Anggota

Nama	Type	Size	Keterangan
NIA	Char	10	Nomor Induk Anggota ( <b>Primary Key</b> )
Username_anggota	Varchar	25	Username (inisial)
Nama_anggota	Varchar	25	Nama
Password_anggota	Text		Password
Level_anggota	Char	1	Level Pengguna

#### Ø Level

Nama	Type	Size	Keterangan
Level	Int	2	Anggota ( <b>Primary Key</b> )
Level_name	Varchar	20	Nama

#### Ø Buku Tamu

Nama	Type	Size	Keterangan
Nama	Varchar	20	Nama
Email	Varchar	30	E-mail
Pesan	Text		Komentar
Tanggal	Date		Tanggal Mengisi

#### Ø Artikel

Nama	Type	Size	Keterangan
------	------	------	------------

Id	Int	10	Id Artikel (Primary Key)
Judul	Varchar	100	Artikel
Konten	Text		Isi Artikel
Tanggal	Date		Dimuat
Pengirim	Varchar	50	Sumber

Ø Berita

Nama	Type	Size	Keterangan
Id	Varchar	2	Id Berita (Primary Key)
Judul	Varchar	40	berita
Isi	Text		Isi Berita
Tanggal	Date		Tanggal dimuat
Penulis	Varchar	10	Editor

Ø Polling

Nama	Type	Size	Keterangan
Pilihan_1	Int	4	-
Pilihan_2	Int	4	-
Pilihan_3	Int	4	-
Pilihan_4	Int	4	-

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN

#### 4.1 Algoritma Program

##### 4.1.1 Koneksi (Connection)

Dalam pembuatan sistem ini terdapat 2 skrip penghubung (koneksi) ke database (MySQL) yang digunakan antara lain dbConn.php dan koneksi.php.

##### **Koneksi.php**

```
mysql_connect("localhost","root","root");  
mysql_select_db("jadwal");
```

##### **dbConn.php**

```
include "include/config.inc.php";  
$db=mysql_connect($CFG_DB_HOST,$CFG_DB_USER,$CFG_DB_PASS)  
or die(mysql_error());  
$mysqlDb=mysql_select_db($CFG_DB_NAME)or  
die(mysql_error());
```

Dalam dbConn.php terdapat skip “include/config.inc.php” yang berisi:

```
$CFG_DB_HOST = "localhost";  
$CFG_DB_USER = "root";  
$CFG_DB_PASS = "root";  
$CFG_DB_NAME = "jadwal";
```

Penjelasan:

- `mysql_connect` digunakan untuk melakukan pengujian dan koneksi pada server database MySQL.
- `mysql _select_db` digunakan untuk melakukan koneksi pada database di dalam server MySQL yang berhasil terkoneksi.



- `$CFG_DB_HOST` berisi nama host (*hostname*) atau alamat server database MySQL. Pada program ini dipergunakan localhost.
- `$CFG_DB_USER` berisi nama *user* yang telah diberi hak untuk dapat mengakses server database. Dalam `include/config.inc.php` root sebagai *username*.
- `$CFG_DB_PASS` berisi *password* atau kata sandi untuk dapat masuk ke dalam database.
- `$CFG_DB_NAME` berisi nama database yang dipilih untuk diproses.

#### 4.1.2 Save

```
$Query = "";  
if($QueryType == "new")  
{  
    $Query = "insert into $tableName  
    (" . implode($tableFieldName, ',') . ") values  
    (" . implode($tableFieldValue, ',') . ")";  
};
```

Penjelasan:

- `$Query` digunakan untuk melakukan eksekusi perintah SQL untuk memanipulasi database yang berhasil terkoneksi.
- `$QueryType == "new"` digunakan sebagai variable untuk memasukkan ke dalam database.
- `$Query = "insert into $tableName, perintah insert` digunakan untuk menyisipkan data pada table dalam database.

- `$tableName` merupakan nama tabel yang digunakan sebagai tempat untuk memanipulasi data.

### 4.1.3 Update

```
($QueryType == "edit")
{
    $primaryKey = $_REQUEST["pkf"];
    $Query = "update $tableName set ";
    foreach($tableFieldName as $fieldName)
    {
        if($primaryKey != str_replace("`", "", $fieldName))
        {
            $parameterEdit[] = $fieldName
             .= " . $tableFieldValue[str_replace("`", "",
             $fieldName)];
        }
    }
    $Query .= implode($parameterEdit, ",");
    $Query .= " where ` $primaryKey`
     = " . $tableFieldValue[$primaryKey];
    $result = mysql_query($Query)
}
```

#### Penjelasan:

- `$QueryType == "edit"` digunakan sebagai variable untuk mengedit atau memperbaiki data yang ada dalam database.
- `$Query = "update $tableName, perintah update` digunakan untuk memperbaiki nilai suatu data di dalam suatu sistem database.
- `$Query .= " where ` $primaryKey` , perintah where` menunjukkan penggantian nilai dimana *primaryKey* sebagai tolak ukur. Dan pada *primaryKey* tidak dapat diganti.

#### 4.1.4 Delete

```
($QueryType == "delete")
{
    $primaryKey = $_REQUEST["pkf"];
    $Query = "delete from $tableName where
    $primaryKey = ".$tableFieldValue[$primaryKey];
    $result = mysql_query($Query);
    $ProcessResult = mysql_error();
}
```

Penjelasan;

- `$QueryType == "delete"` digunakan sebagai variable untuk menghapus data yang ada dalam database.
- `$primaryKey = $_REQUEST["pkf"]`, sebagai variabel *request* yang menunjukkan *input name primaryKey* yang diberi nama pkf.
- `$Query = "delete from $tableName, perintah delete"` digunakan untuk menghapus sebuah *record* pada tabel suatu sistem database.

## 4.2 Algoritma Semut (Ant Algorithm)

### 4.2.1 Ant.class

```
class Ant
{
    private $the_food;
    private $the_pheronome;
    private $startPoint;
    private $lastPoint;
    private $currentPoint;
    private $food_collected;
    private $food_storage;
    private $fitness_constraint;
    private $fitness_cost;
    public function __construct($food)
    {
```

```
$this->the_food = $food;
$this->startPoint = rand(0, count($this->the_food));
$this->currentPoint = $this->startPoint;
$this->last_point = $this->startPoint;
$this->food_collected = array();
}
public function setPheronome($pheronome)
{
    $this->the_pheronome = $pheronome;
}
function collectThatFood()
{
    $eat = true;
    $eatCount = 0;
    while($eat)
    {
        $collect = false;
        foreach($this->food_collected as $key)
        {
            if ($key == $this->currentPoint)
            {
                $collect = true;
            }
        }
        if(!$collect)
        {
            $this->food_collected[] = $this->currentPoint;
            $this->the_pheronome->set($this->last_point,
            $this->currentPoint);
            $this->last_point = $this->currentPoint;
        }
        Else
        {
            $collectCount++;
        }
        if($collectCount >= count($this->the_food))
        {
            $eat = false;
        }
        $this->currentPoint = rand(0, count($this
        >the_food));
    }
}
function putTheFoodOnStorage()
{
    foreach($this->food_collected as $food)
    {
        $size = $this->the_food[$food][2];
        list($day, $time) = $this->getEmptyStorage($size);
        if($day > -1 && $time > -1)
        {
            for($sizeCheck=0; $sizeCheck < $size; $sizeCheck++)
            {
                $this->food_storage[$day]
                [$time+$sizeCheck] = $food;
            }
        }
    }
}
```



- `public function __construct($food)` , fungsi yang digunakan untuk memasukkan *food* (komponen yang dipergunakan untuk penjadwalan) pada fungsi-fungsi berikutnya.
- `public function setPheronome($pheronome)` fungsi untuk mencari nilai *pheromone*.
- `function collectThatFood` fungsi yang berguna untuk menentukan letak makanan dan merujuk pada fungsi `setPheronome`.
- `function putTheFoodOnStorage` fungsi yang digunakan sebagai penunjuk jalan dari tempat makanan ke sarang.
- `function getEmptyStorage` fungsi untuk mencari tempat penyimpanan makanan yang masih kosong pada sarang semut.
- `function set_storage` fungsi yang digunakan untuk menyusun *array*.
- `function get_storage` fungsi untuk mengambil *array* yang sudah disimpan oleh fungsi `set_storage`.
- `function getTheFood` yang digunakan untuk menyusun makanan.
- `function set_constraint` fungsi yang digunakan untuk mencari nilai *pheromone* yang tertinggi.

#### 4.2.2 AntAlgorithm.class

Ant algorithm.class merupakan program yang utama (*main program*) dalam proses penjadwalan kuliah dengan menggunakan algoritma semut.

```
class AntAlgorithm
{
    private $ant_size;
    private $gen_limit;
    private $ant_food;
    private $ant_constraint;
    private $ant_colony;
    private $pheronome;
    private $food_storage;

    public function __construct()
    {
        $this->pheronome = new Pheronome();
    }
    public function set_antCount($antCount = 1)
    {
        $this->ant_size = $antCount;
    }
    public function set_genLimit($genLimit = 1)
    {
        $this->gen_limit = $genLimit;
        echo "Generasi di set " . $genLimit;
    }
    public function set_food($arrFood = array())
    {
        $this->ant_food = $arrFood;
    }
    public function set_storage($arrStorage = array())
    {
        $this->food_storage = $arrStorage;
    }
    public function set_constraint($arrConstraint = array())
    {
        $this->ant_constraint = $arrConstraint;
    }
    public function tellAntToCollectTheFood()
    {
        $this->orderTheAnt();
        for($gen=0;$gen<$this->gen_limit;$gen++)
        {
            foreach($this->ant_colony as $antId => $ant)
            {
                $ant->set_storage($this->food_storage
                    [$gen][$antId]);
                $ant->collectThatFood();
                $ant->putTheFoodOnStorage();
                $this->food_storage[$gen][$antId] =
                    $ant->get_storage();
            }
        }
    }
}
```

```
public function getCollectedFood()
{
    return $this->food_storage;
}
private function orderTheAnt()
{
    for($i=0;$i<$this->ant_size;$i++)
    {
        $myAnt = new Ant($this->ant_food);
        $myAnt->setPheronome($this->pheronome);
        $myAnt->set_constraint($this->ant_constraint);
        $this->ant_colony[] = $myAnt;
    }
}
public function fitnessCount()
{
    foreach($this->food_storage as $gen => $antResult)
    {
        foreach($antResult as $antId => $food_result)
        {
            foreach($food_result as $dayId => $timeResult)
            {
                foreach($timeResult as $timeId)
                {}
            }
        }
    }
}
```

Penjelasan :

- public function \_\_construct fungsi untuk memberikan nilai awal.
- public function set\_antCount fungsi yang menuntukan banyaknya jumlah *ants*.
- public function set\_genLimit fungsi yang memberikan batasan pada nilai generate.
- public function set\_food berfungsi untuk menentukan *array* dari makanan (*food*).
- function set\_storage fungsi yang digunakan untuk menyusun *array*.



- public function `getCollectedFood` berfungsi untuk menentukan *crash value* dari makanan (*food*).
- public function `fitnessCount` berfungsi untuk menghitung besarnya *fitness value*.

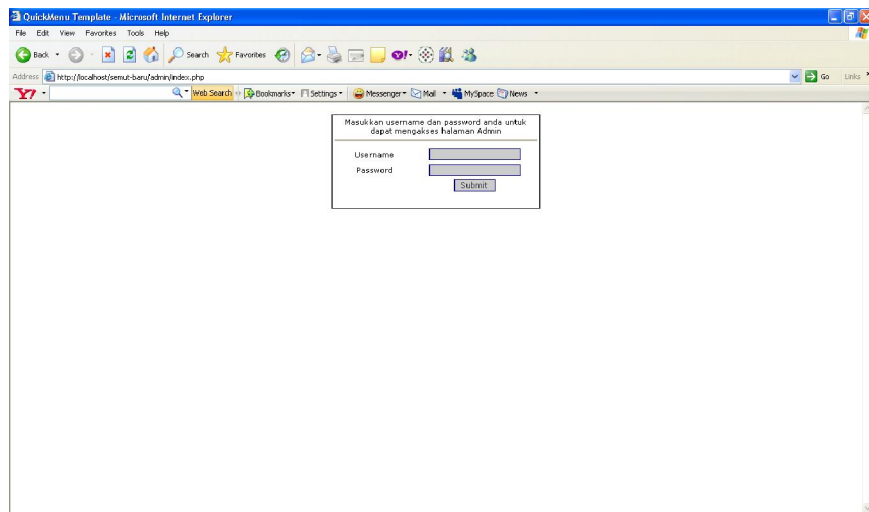
### 4.3 Pengujian Program

Untuk pengujian menggunakan data sementara (data sampel), yang diambil hanya pada jurusan Teknik Informatika.

#### 4.3.1 Halaman Admin

- **Halaman Login**

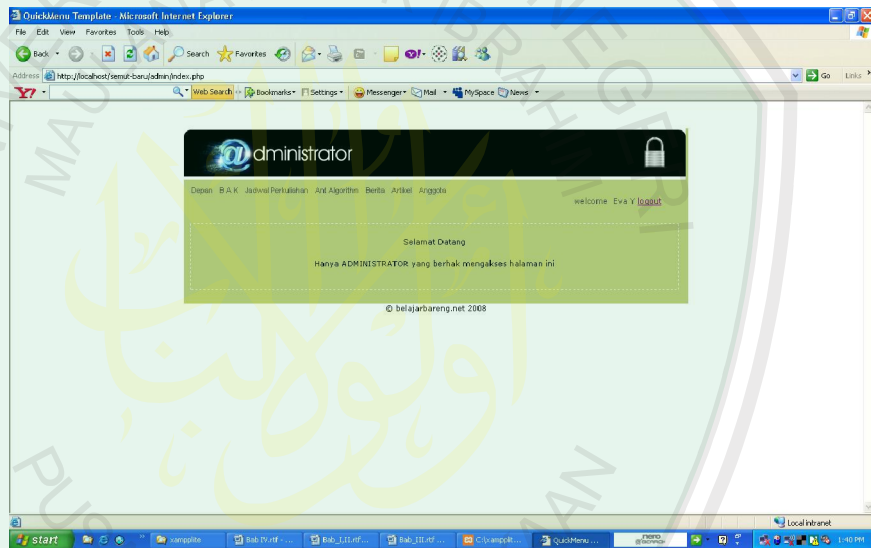
Pada halaman ini berisi form login untuk admin, dimana pada form tersebut berfungsi untuk memberikan hak ases kepada administrator, BAK, Fakultas, Jurusan dan Dosen



**Gambar 4.1 Halaman Login**

- **Halaman depan (*home*)**

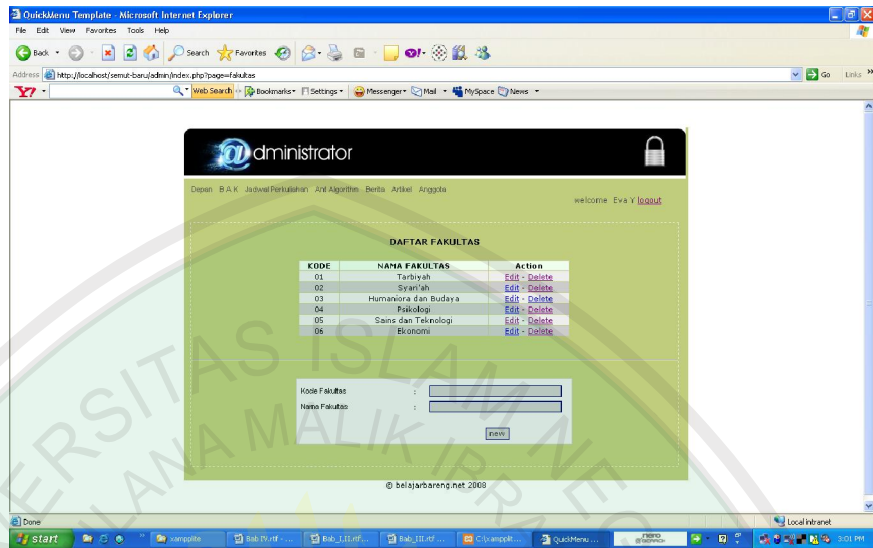
Halaman ini merupakan halaman yang tampil setelah berhasil login.



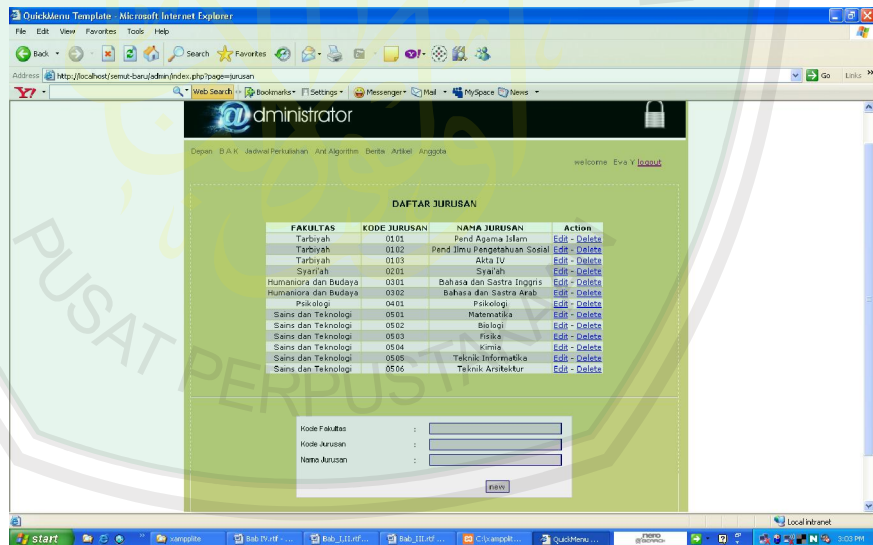
**Gambar 4.2 Halaman Index (*home*) Admin**

- **BAK**

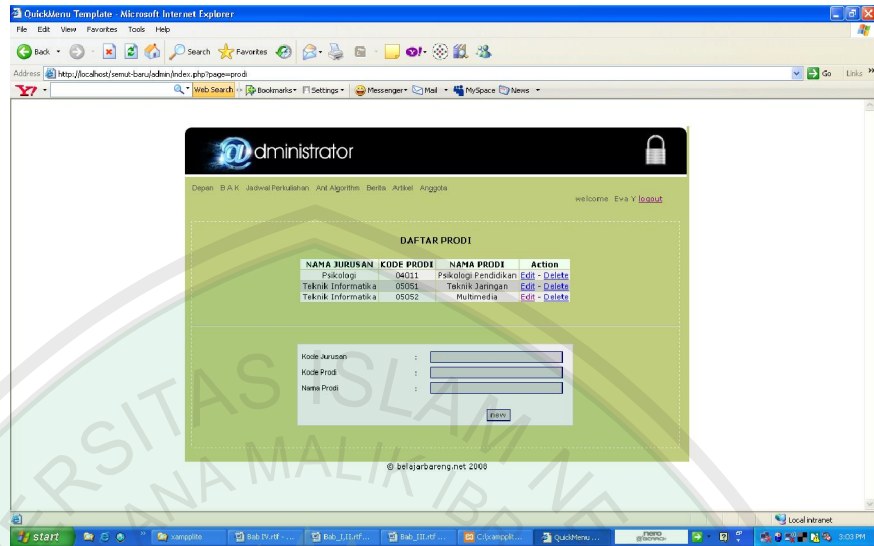
Pada *link* BAK terdapat 9 sub *link* antara lain: halaman fakultas, halaman jurusan, halaman prodi, halaman dosen, halaman matakuliah, halaman ruang, halaman semester tahun, halaman keediaan mengajar dan halaman kelas mengajar. Pada tiap-tiap halaman terdapat form untuk mengolah data baik *entry*, *update* dan *delete* serta tabel yang berisi data yang ada dalam database.



Gambar 4.3 Halaman Fakultas



Gambar 4.4 Halaman Jurusan



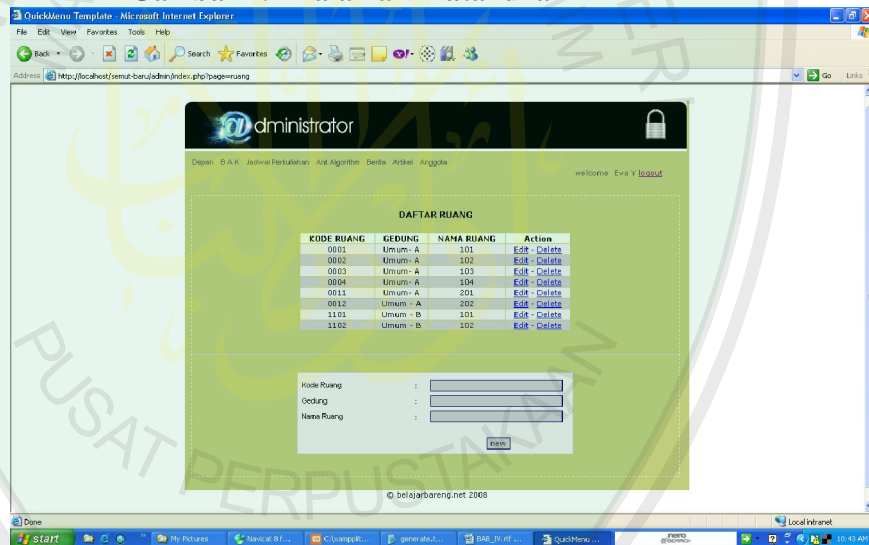
Gambar 4.5 Halaman Program Studi



Gambar 4.6 Halaman Dosen



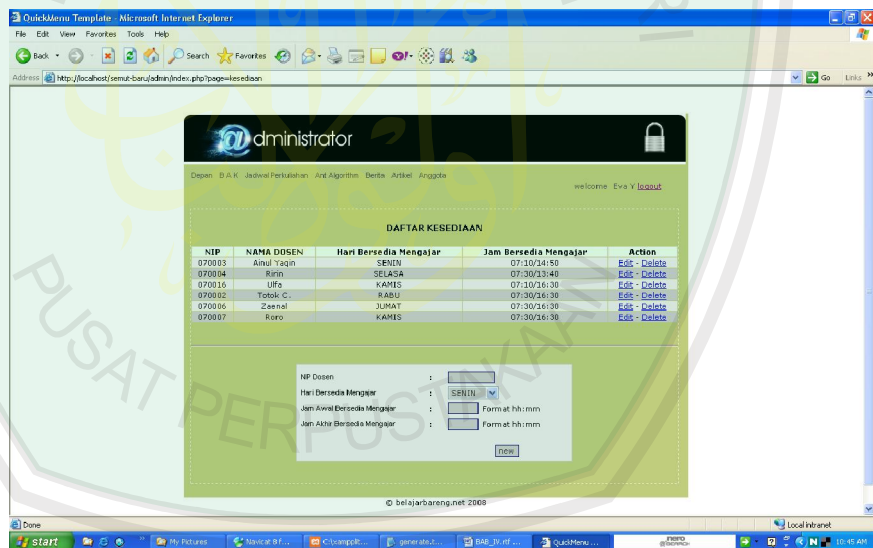
Gambar 4.7 Halaman Matakuliah



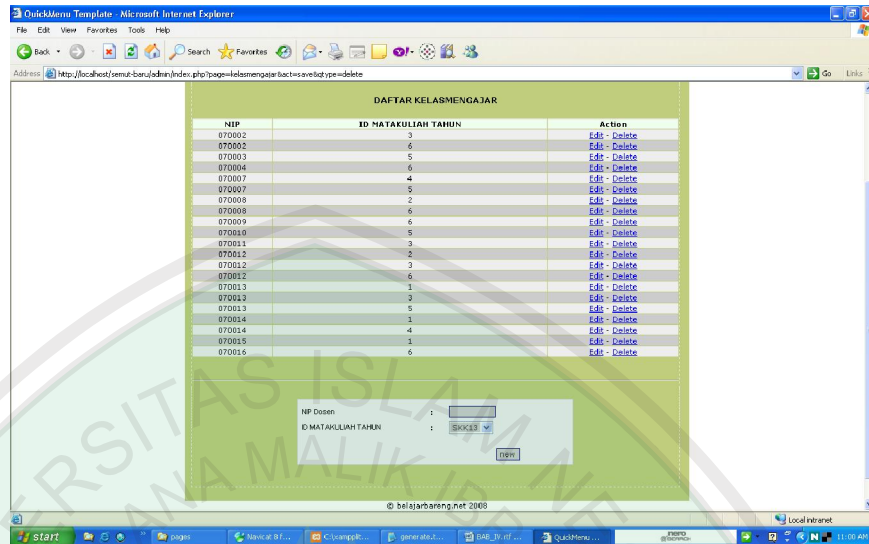
Gambar 4.8 Halaman Ruang



Gambar 4.9 Halaman Matakuliah Tahun



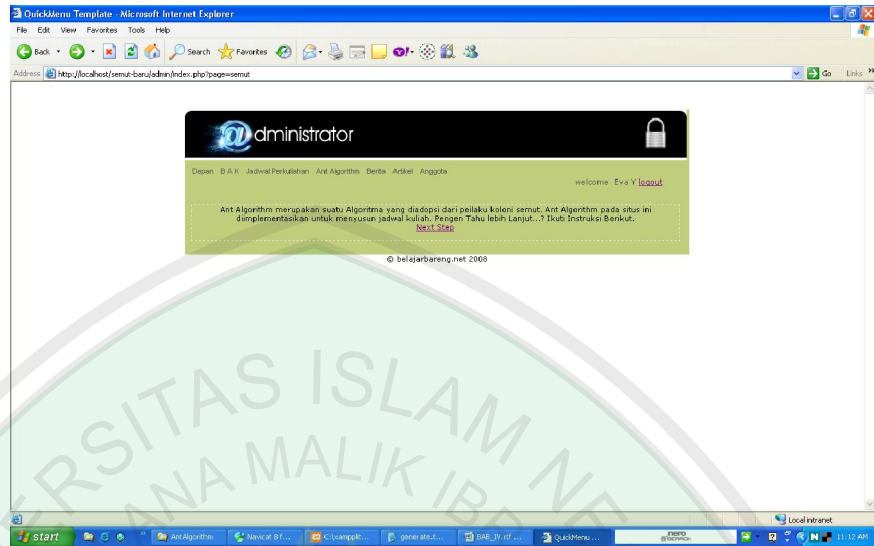
Gambar 4.10 Halaman Kesiediaan Dosen



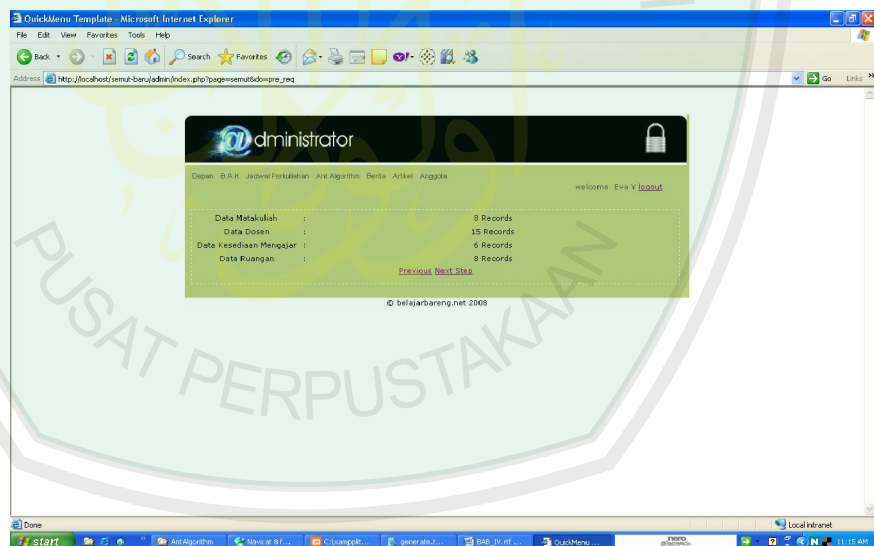
**Gambar 4.11 Halaman Kelas Mengajar**

- **Ant Algorithm**

Pada link ant algorithm terdapat 3 sublink yang terdiri dari halaman introduction ant algorithm, halaman pre-requistic, dan halaman generate



**Gambar 4.12** Halaman *Introduction Ant Algorithm*



**Gambar 4.13** Halaman *Pre-Reqistic*





**Gambar 4. 14 Halaman Generate**

- **Jadwal Perkuliahan**

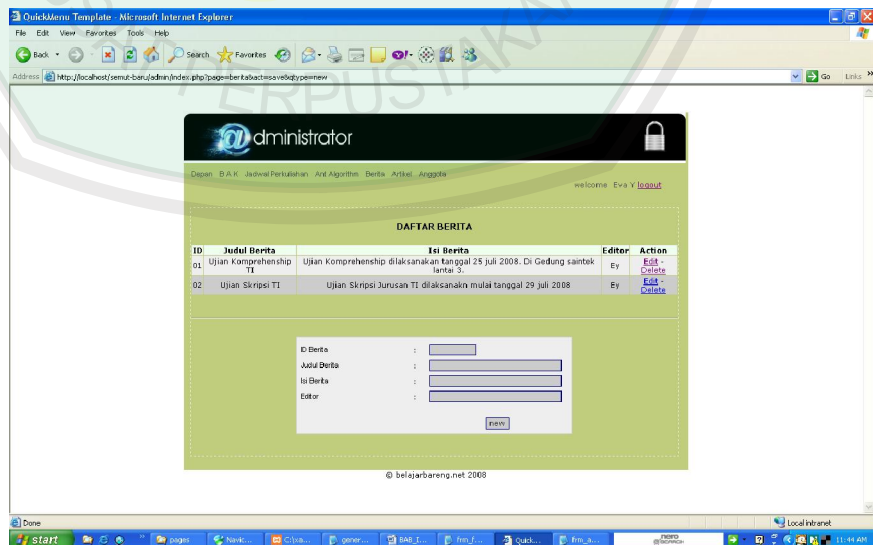
Halaman ini menampilkan jadwal dari hasil hasil generate pada halaman generate.



Gambar 4.15 Halaman Hasil Generate

- **Halaman Berita**

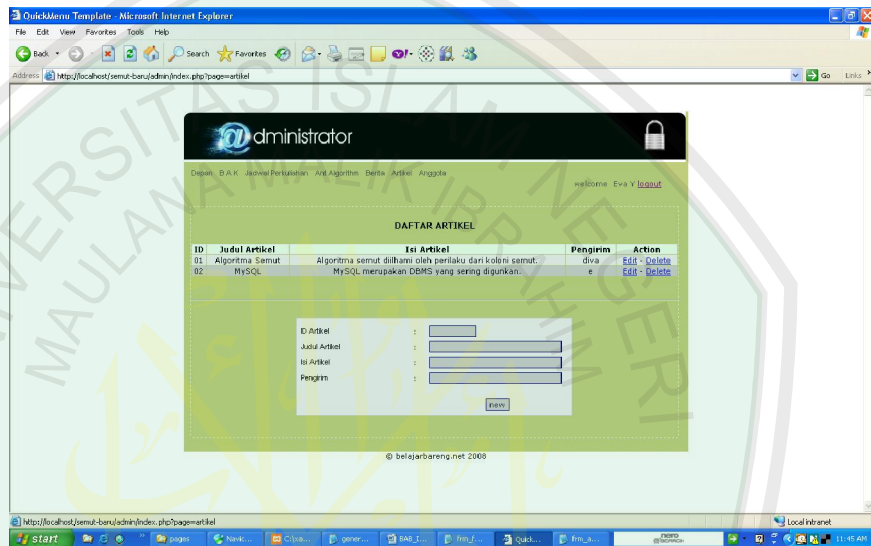
Halaman ini dipergunakan untuk *mensetting* berita yang akan di *upload* pada halaman user.



Gambar 4.16 Halaman Berita

- **Halaman Artikel**

Halaman artikel dipergunakan untuk mensetting berita yang akan di *upload* pada halaman user.



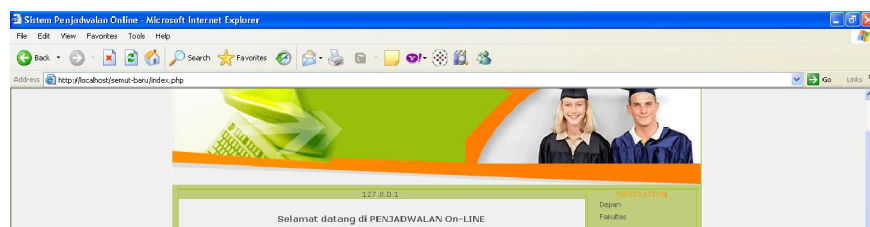
Gambar 4.17 Halaman Artikel

### 4.3.2 Halaman User

Halaman user merupakan halaman yang dapat diakses oleh siapa saja tanpa harus *login* terlebih dahulu. Pada halaman ini terdapat navigator (*link*) serta form polling pada sisi kanan halaman. Tampilan pada halaman yang lain sama, hanya saja isi (*contain*) yang berbeda.

Link yang ada pada halaman user berbentuk menu *drop down* antara lain : depan, Fakultas dengan sublink jurusan-jurusan serta mahasiswa dengan sublink berita, form buku tamu dan daftar buku tamu.

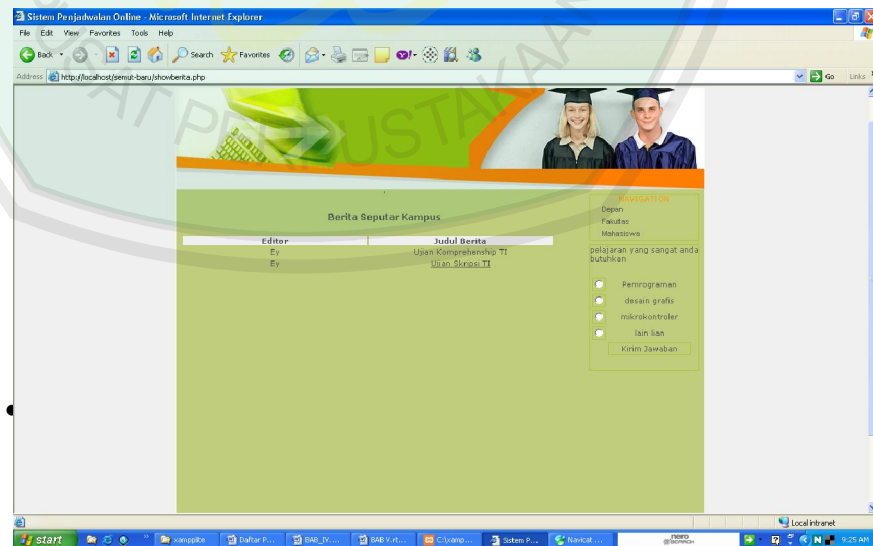
- **Halaman Depan (*home*)**

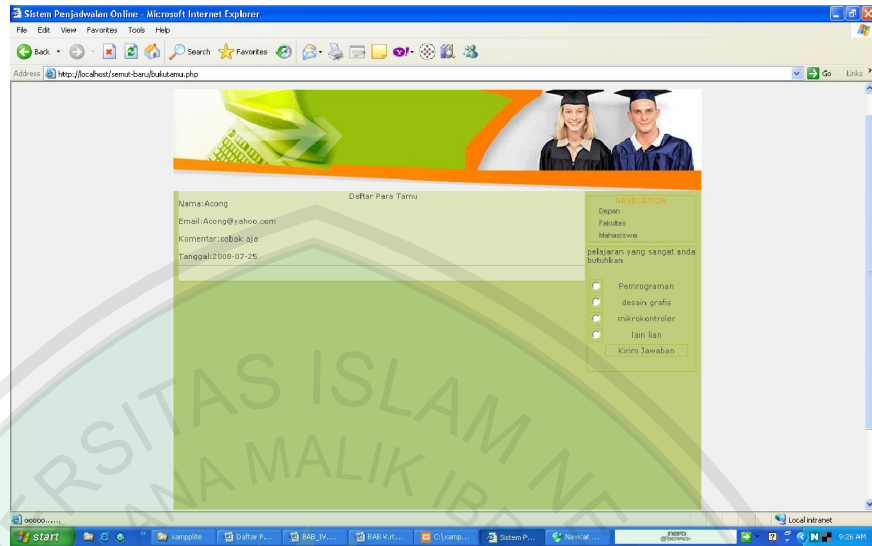


**Gambar 4.18 Halaman Depan**

**Gambar 4. 18 Halaman Depan**

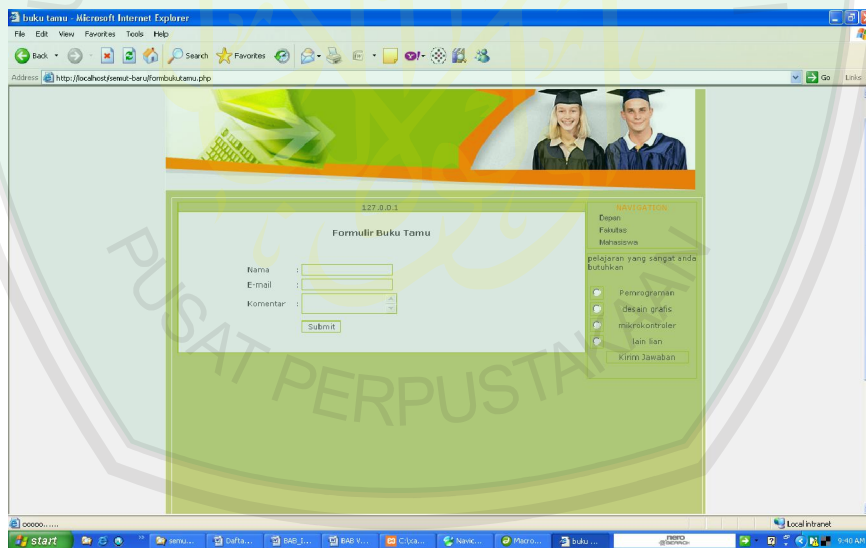
• **Halaman Berita**





**Gambar 4.20 Halaman Daftar Tamu**

- **Halaman Form Buku Tamu**



**Gambar 4.21 Halaman Form Buku Tamu**

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

1. Penjadwalan kuliah dengan menggunakan algoritma semut mampu memberikan solusi dalam penyusunan jadwal online.
2. Algoritma semut dapat diterapkan untuk membuat jadwal kuliah dengan hasil yang optimal.
3. Algoritma semut mampu memberikan solusi dalam penyusunan jadwal kuliah yang menginginkan waktu tertentu.

#### 5.2 Saran

1. Penyusunan jadwal kuliah dengan menggunakan algoritma semut, diperlukan pengamatan dalam menentukan jumlah koloni dan berapa kali generate yang harus dilakukan.
2. Untuk mendapatkan hasil penjadwalan kuliah yang lebih baik perlu dilakukan pengkajian dari teknik-teknik algoritma semut yang lebih mendalam.

## DAFTAR PUSTAKA

- Hariyadi. M.Amin.MT, 2007, *Al-Quran dan Semut: Inspirasi Al-Quran dalam Membangun Algoritma Ant*, Malang: UIN-Malang Press.
- M. Dorigo and Thomas Stutzle, 2004, *Ant Colony Optimization*, MIT Press.
- M. Dorigo, 1992, *Optimization, Learning and Natural Algorithms*, PhD thesis, Milan, Italy: politecnico di Milano.
- M. Dorigo, V. Maniezzo, dan A.Colorni, 1996, *Positive Feedback As A Search Startegy*, (Teach. Rep. 91-016), Milan, Italy: Politecnico di Milano, Dipartimento di Elettronica.
- M. Dorigo dan L. M. Gambardella, 1997, *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*.
- Purbo. W.Onno, 2001, *Buku Pintar Internet membangun web E-Commerce*, Jakarta: Elex Media Komputindo.
- R. Beckers, J.L Deneubourg, dan S.Goss, 1992, *Trail and U-turns in the Selection of the shortest Path by the Ant Lasius Niger*, Journal of theoretical biologi.
- Ross.P., Dave C, Hasiao L.F, 1994, *Succesful Lecture Timetabling With Evolutionary Algorithms*, U.K: Departement of Artificial Intelegence University of Edinburgh.
- Sunyoto. Andi, 2007, *AJAX membangun Web dengan Teknologi Asynchroneuse JavaScript dan XML*, Yogyakarta: Andi.