

# PERANGKAT LUNAK PENJADWALAN UNTUK MATA KULIAH, DOSEN, KELAS DAN MAHASISWA DI STMIK LIKMI

Rachmat Selamat

Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

Jl. Ir. H. Juanda 96 Bandung 40132

E-mail : [if25005@students.itb.ac.id](mailto:if25005@students.itb.ac.id)

---

## Abstrak

Penulis mencoba meneliti tentang penerapan algoritma *ant* untuk memecahkan permasalahan penjadwalan dengan batasan mata kuliah, dosen, kelas dan mahasiswa di STMIK-LIKMI dan membuat perangkat lunak yang mudah digunakan. Algoritma *ant* yang digunakan adalah RBAS yang menerapkan ranking untuk batasan yang digunakan. Untuk membuat perangkat lunak mudah digunakan, maka perlu dibuat antar muka yang cocok digunakan oleh pemakai, memberikan pesan kesalahan yang sesuai, dan menampilkan informasi yang berguna bagi pemakai dalam pemakaian perangkat lunak.

**Kata-kata kunci:** jadwal, algoritma *ant*, RBAS, dan antar muka.

---

## 1. PENDAHULUAN

Seiring dengan pesatnya laju teknologi ini, dituntut adanya informasi yang cepat, tepat, akurat serta dapat dipertanggungjawabkan. STMIK-LIKMI adalah sekolah tinggi yang bergerak di bidang komputer yang peduli dengan teknologi. Banyak teknologi yang sudah diterapkan, tetapi ada juga yang belum dimanfaatkan, di antaranya adalah masalah penjadwalan. Penjadwalan ini melibatkan beberapa macam data, yaitu: data waktu yang dapat diisi oleh dosen, data jumlah sesi atau kelas dan mata kuliah dari dosen, data mata kuliah, data ruang kelas yang tersedia, dan data daftar ulang dari siswa.

Sebagai perbandingan, ada program umum yang dapat menangani masalah penjadwalan, tetapi program ini tidak mengatasi semua masalah. Adapun masalah yang tidak diatasi oleh program ini adalah tampilan jadwal yang dihasilkan sulit dibaca karena menggunakan struktur hirarki, waktu untuk 1 sks tidak dapat ditetapkan karena program ini mengisi waktu secara langsung (berapa lama), tampilan program kurang baik, sehingga agak sulit dalam pengisian data karena program ini dirancang untuk kebutuhan umum, dan

harga program ini cukup mahal (\$ 899.00)[16].

## 2. PENJADWALAN

Istilah penjadwalan yang dimaksud oleh penulis adalah masalah penentuan beberapa kejadian ke beberapa slot waktu (biasa disebut juga *timetabling*). Masalah ini juga biasa diterapkan dalam penentuan ujian, seminar, proyek, pertemuan, kelas dan sebagainya.

Dalam masalah *timetable* memiliki beberapa macam batasan yang dapat menyebabkan *output* yang dihasilkan menjadi salah. Dalam menerapkan batasan ini pada suatu masalah, biasanya tidak selalu sama untuk setiap masalah. Adapun macam-macam batasan tersebut adalah[4] :

### 1. *Edge constraint*

*Edge constraint* adalah batasan yang mengatur dua kejadian tidak boleh menempati satu slot waktu yang sama. Contoh pada hari senin jam 07.30 sampai 09.10 tidak mungkin dosen A mengajar mata kuliah A di ruang 204 dan mengajar mata kuliah B di ruang 205.

### 2. *Ordering constraint*

*Ordering constraint* adalah batasan yang menjaga urutan kejadian dalam timetable. Contoh mata kuliah A harus dilakukan sebelum mata kuliah B. Biasanya masalah batasan ini jarang digunakan karena dapat menyebabkan penjadwalan menjadi lebih rumit. Contoh sesi 1 diisi dengan A dan sesi 2 diisi dengan B sudah memenuhi kondisi, tetapi karena adanya *ordering constraint*, maka jadwal ini tidak memenuhi kondisi.

### 3. *Event-spread constraint*

*Event-spread constraint* adalah batasan yang mengatur penyebaran kejadian pada suatu timetable. Contoh dalam menjalani ujian, seorang siswa diperbolehkan max 3 mata kuliah yang diujikan dalam 1 hari.

### 4. *Preset specification and exclusion*

*Preset specification and exclusion* adalah menentukan terlebih dahulu slot waktu yang akan digunakan oleh suatu kejadian sebelum proses pencarian solusi dilakukan. Contoh mata kuliah A harus dimasukkan ke sesi 1 atau sesi 3 hari jumat (jam 11.00 sampai 12.40) tidak boleh diisi kuliah karena untuk yang beragama Islam ini adalah jam untuk beribadah.

### 5. *Capacity constraint*

*Capacity constraint* adalah batasan yang berhubungan dengan kapasitas suatu ruangan. Contoh untuk ruang 204 max diisi oleh 33 orang siswa.

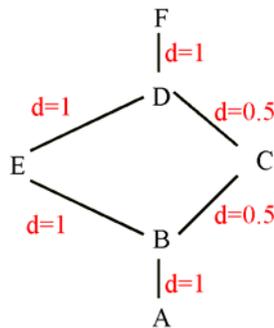
6. *Hard and soft constraint*

*Hard constraint* adalah batasan yang sama sekali tidak boleh dilanggar, sedangkan *soft constraint* adalah batasan yang diusahakan semaksimal mungkin tidak dilanggar namun jika dilanggar, hal tersebut masih dapat diterima. Untuk menentukan *hard constraint* dan *soft constraint* merupakan hal yang sulit karena pada saat pencarian solusi, keduanya dapat berubah. Contoh karena alasan tertentu, ruang kuliah 204 untuk kuliah A diubah ke ruang 205.

### 3. TEORI ALGORITMA ANT

Algoritma *Ant* ini ditemukan oleh Marco Dorigo pada tahun 1996 dengan mempelajari cara kerja semut dalam membangun jalan untuk mengambil makanan. Jalan yang dibangun oleh semut ini adalah jalan yang terpendek, padahal semut itu buta. Maka setelah dipelajari didapat kesimpulan, bahwa pengambilan jalur ini didasarkan pada sejumlah perjalanan tiap semut. Jika jalur yang diambil lebih pendek, maka jejak semut yang melalui jalur ini akan lebih banyak. Maka jejak yang terbanyak ini ditandai sebagai *pheromones*. *Pheromones* ini terus dibuat sampai dapat menuju semua tempat yang diharapkan[6]. Algoritma dasar dari semut ini dinamakan Ant Colony Optimization.

Komponen utama yang ada pada algoritma ini adalah *colony*, *ant*, *route* dan *pheromone*[6]. *Colony* adalah sekumpulan tempat yang merupakan tempat asal dan tempat tujuan dari semut. *Ant* adalah semut yang akan mencari jalur dari tempat asal menuju tempat tujuan. *Route* adalah jalur dari tempat asal yang mungkin dapat dituju oleh semut untuk sampai ke tempat tujuan. *Pheromone* adalah jalur yang dianggap sebagai jalur optimal dari tempat asal menuju tujuan.



Gambar 3.1 Contoh jalur semut

Berdasarkan gambar III.1, dapat dijelaskan sebagai berikut : A dan F adalah *colony*, sedangkan F-D, D-E, E-B, D-C, C-B, B-A adalah *route*.  $d$  adalah jarak yang ditempuh oleh jalur tersebut. Sebagai contoh, di colony A dan F akan ditempatkan 16 semut untuk masing-masing colony. Langkah-langkah dalam menentukan jalur dapat digambarkan sebagai berikut:

1. 16 semut F akan berjalan ke D dan 16 semut A berjalan ke B.
2. Semut F akan dibagi 2 dengan 8 menuju E dan 8 menuju C, begitu juga untuk semut B.  
Sehingga pada jalur E dan C masing-masing akan terdapat 16 semut.
3. Dilihat dari intensitas setiap cabangnya, maka dapat dilihat bahwa intensitas kemungkinan di jalur E adalah 8 dan di jalur C adalah 16.

Setelah dikirimkan semut untuk mencari jalur selama beberapa waktu, maka dapat disimpulkan yang menjadi *pheromone* adalah F-D, D-C, C-B dan B-A karena pada akhirnya semut-semut yang tadi dikirimkan hanya melewati jalur ini[14].

Berdasarkan penelitian yang ada, algoritma *Ant* ini sudah digunakan untuk beberapa masalah seperti *traveling salesman problem*, *graph coloring problem*, *quadratic assignment problem*, *vehicle routing problem* dan masih banyak lagi [3]. Algoritma *Ant* dasar diperlihatkan pada gambar III.2[12]:

```

For each colony do
  For each ant do
    Generate route
    Evaluate route
    Evaporate pheromone in trails
    Deposit pheromone on trails
  End for
End for

```

**Gambar 3.2 Algoritma *ant* dasar**

### VARIASI ALGORITMA ANT

Seiring dengan perkembangan teknologi, algoritma *Ant* ini juga ikut berkembang dengan bermacam-macam variasi. Kinerja variasi ini lebih baik dari algoritma dasar *Ant*. Beberapa variasi yang akan dibahas adalah *max min* (yang membatasi jalur yang akan

ditempuh) dan *rank based* (yang memberikan ranking pada setiap *constraint*).

Algoritma Max-Min Ant System (MMAS) ini dikembangkan oleh Stutzle and Hoos (1996)[5] dengan melihat perbedaan langkah pada saat memeriksa jalur yang akan dijadikan *pheromone* (*evaporate pheromone in trails*) dan menyimpan jalur *pheromone* (*deposit pheromone on trails*). Dalam algoritma ini, semua *pheromone* akan disimpan ke dalam *best global route*. Untuk pemilihan *route*, cara yang digunakan masih sama dengan algoritma dasar ant. MMAS diperlihatkan pada gambar 3.3[1] :

```
For each colony do
  For each ant do
    Generate route
    Evaluate route
  End for
  If Verify for global or local best then
    Evaporate pheromone in all trails
    Deposit pheromone on best global route
  End For
```

**Gambar 3.3 Algoritma MMAS**

Rank Based Ant System (RBAS) ditemukan oleh Bullnheimer pada tahun 1997. RBAS ini mengkombinasikan algoritma *Ant* dengan pemberian ranking pada setiap solusi yang ada. Ranking ini didasarkan pada kualitas dari solusi tersebut. Setiap iterasi pada algoritma ini akan mengerjakan sesuai dengan urutan ranking. Solusi terbaik akan menghasilkan perubahan terbesar pada tingkat *pheromone*, ketika solusi berikutnya mengupdate *pheromone* dan menurunkan jumlahnya secara linier menurut ranking mereka [8]. RBAS diperlihatkan pada gambar III.4[1]:

```

For each colony do
  For each ant do
    Generate route
    Evaluate route
  End for
  If Verify for global or local optimum then
    Evaporate pheromone in all trails
    Generate an elite from current colony
    Deposit a Y amount of pheromone on elite trails
    Deposit an X amount pheromone on best global route
  End for

```

**Gambar 3.4 Algoritma RBAS**

#### 4. TEORI *USER INTEFACE*

Membuat perangkat lunak yang mudah digunakan berhubungan dengan interaksi antara user dengan aplikasi tersebut, yang berupa *user interface*. Pada bagian ini akan dijelaskan pengaruh *user interface* kepada pemakai dan kriteria mengenai *user interface* yang mudah digunakan.

Dengan adanya *user interface*, dapat mempengaruhi emosi dari pemakai dalam menggunakan perangkat lunak. Pengaruh yang dihasilkan bisa membuat pemakai bahagia, tetapi bisa juga membuat pemakai frustrasi. Berikut ini beberapa alasan yang dapat membuat frustrasi[9] :

1. Ketika perangkat lunak tidak bekerja dengan semestinya atau crash.
2. Ketika sistem tidak mengerjakan yang diminta untuk dikerjakan.
3. Ketika sistem tidak menyediakan informasi yang cukup untuk membuat pemakai tahu apa yang harus dikerjakan.
4. Ketika sistem membutuhkan pemakai untuk mengerjakan banyak langkah dalam mengerjakan pekerjaan sehingga bila ada kesalahan maka harus dimulai dari awal lagi.
5. Ketika pesan kesalahan menampilkan pesan yang tidak jelas dan membingungkan.

Bagaimana mengatasi hal tersebut ? Untuk alasan tersebut dapat dihindari dengan

cara sebagai berikut[9] :

1. Untuk alasan 1, 2 dan 3 dapat dilakukan dengan memberikan pesan “*Under construction*” untuk memperjelas bahwa sistem belum selesai.
2. Untuk alasan 4, dapat dihindari dengan meningkatkan proses otomatis (*Upgrade*). Yang terpenting dalam meningkatkan ini adalah kebutuhan pemakai dalam belajar ulang menggunakannya. Selain itu bisa dengan menambahkan *Plug-ins* yang membuat sistem lebih baik dari sebelumnya.
3. Untuk alasan 5, pesan kesalahan sebaiknya dapat memberitahu kepada pemakai bagaimana memperbaiki kesalahan tersebut. Pesan tersebut dibuat supaya mudah dibaca dan dimengerti. Selain cara sebelumnya, cara lainnya dapat dilakukan dengan membuat pesan-pesan umum. Contoh :
  - a. SYNTAX ERROR : Terdapat masalah dalam menuliskan perintah.
  - b. INVALID FILENAME : Terdapat masalah penulisan nama file.
  - c. INVALID DATA : Terdapat masalah dalam data yang dimasukkan.
  - d. DRIVE ERROR:ABORT,RETRY OR FAIL? : Terdapat masalah dalam pembacaan disk.

Dalam sebuah penelitian, terdapat beberapa informasi bentuk *user interface* yang umum digunakan berdasarkan pemakainya[10] :

1. Mengetik baris perintah (*Command line*). Mengurangi ketakutan pemakaian komputer. Walaupun komputer sudah digunakan secara luas, tetapi banyak yang menghindari pemakaian komputer. Beberapa pemakai lama menghindari bantuan dari komputer berbasis alat seperti terminal bank atau pemroses kata, karena mereka takut bila merusak komputer atau membuat kesalahan yang memalukan.
2. Pemilihan menu. Untuk para pemula biasanya dimulai interaksi dengan komputer menggunakan pemilihan menu, mereka berharap meningkatkan mempercepat atau fasilitas yang lebih baik. Metode dibutuhkan untuk memperlancar perubahan pemakai pemula menjadi pemakai berpengetahuan dan menjadi pemakai ahli. Perbedaan kebutuhan dari pemula dan ahli adalah pesan kesalahan, tampilan yang kompleks, dan umpan balik informatif yang semua dibutuhkan pada pemeriksaan.
3. *Tools* untuk *interface*. Dengan menggunakan *tools* untuk membangun *user interface* dapat mengurangi waktu implementasi untuk tugas penting ketika mereka

menyelesaikan tugasnya. Tetapi untuk banyak situasi, pembuatan program secara luas masih harus ditambahkan.

4. *Direct manipulation*. Dengan tampilan secara visual, pemakai dapat mengoperasikan persentasi objek secara lebih atraktif.
5. *Input device* yang interaktif. Dengan banyaknya input device menunjukkan tantangan bagi para perancang sistem. Dengan menggunakan touchscreen, mouse, joystick, dan sebagainya dapat menyelesaikan benturan melalui percobaan secara luas dengan banyak tugas dan komunitas pemakai. Hal ini diukur dengan kecepatan, akurasi, kelelahan, perbaikan kesalahan dan kepuasan secara subyektif.
6. Asisten secara *online*. Walaupun banyak sistem menawarkan help atau informasi tutorial secara online, biasanya yang dimengerti hanyalah dasarnya saja. Untuk membantunya dibuatlah konsultan pemakai secara online yang menimbulkan kesuksesan dan kepuasan pemakai.
7. Penggalan informasi. Penggalan informasi seperti navigasi, browse, dan pencarian dari perpustakaan multimedia secara digital dan WWW menjadi lebih umum. Pemakai ingin bisa menyaring, memilih, dan menstruktur ulang informasi mereka secara cepat dan usaha yang minimum, tanpa takut salah orientasi atau kehilangan. Database besar untuk tulisan, gambar, suara, dan data ilmiah akan menjadi lebih mudah untuk digali dengan *tools* visual untuk memunculkan informasi.

## 5. PENERAPAN ALGORITMA ANT DALAM PENJADWALAN

Setelah dianalisis, untuk kasus yang sedang dihadapi maka akan digunakan metode *rank based ant system* / RBAS. Hal ini dikarenakan sistem penjadwalan yang lebih mudah jika menggunakan ranking untuk setiap konstrain yang akan dimasukkan ke dalam jadwal tersebut. Keunggulan dan kelemahan menggunakan *Rank based ant system* :

1. Dalam mencari solusi, setiap kendala harus diberikan ranking yang mana prioritas yang dikerjakan.
2. Metode ini sangat cocok untuk masalah penjadwalan (*timetable*), karena dapat menangani beberapa kendala yang memiliki nilai yang sama.
3. Karena yang didahulukan adalah prioritas yang terkecil, maka hasil yang dicapai belum tentu hasil terbaik.

Berdasarkan algoritma RBAS, maka untuk menyelesaikan masalah penjadwalan,

komponen dari RBAS dapat digambarkan sebagai berikut :

1. *Coloni* adalah waktu berdasarkan ruang tertentu.
2. *Ant* adalah dosen yang dikerjakan berdasarkan ranking.
3. *Route* adalah ruang untuk mata kuliah yang cocok dengan dosen yang memiliki waktu sesuai dengan *coloni*.
4. *Elite* adalah dosen yang memiliki waktu yang cocok dengan *coloni* dan tidak dapat dipindah yang dapat menyebabkan *local optimum*.
5. *Elite trails* adalah jalur yang sudah diplot sesudah ada *elite*.
6. *Best global route* adalah semua jalur terbaik yang sudah diplot.
7. Jalur terpendek yang dimaksud adalah ruang kuliah yang terpakai dengan efektif (ranking yang terendah akan terisi paling banyak) sesuai dengan jadwal yang diisi oleh dosen.

Berdasarkan komponen dari RBAS, maka algoritma RBAS dalam menyelesaikan penjadwalan dapat diterjemahkan sebagai berikut :

For setiap waktuR do

    For setiap dosen do

        If Ada waktuR=waktuP

            Ambil sekumpulan ruang untuk mata kuliah yang diajar oleh dosen

            Dari sekumpulan ruang yang sudah cocok dan diplot berdasarkan ranking terkecil dan sesuai dengan waktuP

        End for

    If Ada waktuP belum diisi dan tidak bisa dipindah (sisa 1) dan sesuai dengan waktuR

        Ubah ruang yang sudah diplot untuk dosen yang memiliki waktu kosong di jam lain dan memiliki waktu untuk noruang

End for

WaktuR adalah waktu untuk noruang tertentu

WaktuP adalah waktu yang diplot oleh dosen(yang bisa diisi oleh dosen)

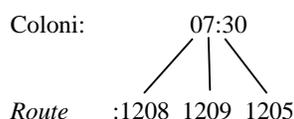
Dilihat dari terjemahan algoritma, perintah yang dikerjakan diawali dengan mengumpulkan *ant*, *route*, dan *coloni* yang digambarkan pada gambar V.1. Kemudian *Ant* yang cocok antara waktu plot dengan *coloni* disesuaikan yang tergambar pada gambar V.2. Setelah ditelusuri, maka didapatkan hasil 007 diplot pada *coloni* 07:30 di jalur 1208, 004

pada coloni 07:30 di jalur 1209, dan 005 pada coloni 09:15 di jalur 1208 yang tergambar pada gambar V.3. Untuk mata kuliah dengan kodemk KK101 dapat digunakan pada ruang 1208, 1209, dan 1205. Perjalanan ant ditentukan oleh ranking ant tersebut dengan urutan 007, 004, dan 005. Jalur yang ditempuh juga ditentukan oleh ranking route tersebut dengan urutan 1208, 1209, 1205. Berdasarkan ranking inilah maka 007 yang diprioritaskan pertama menempuh jalur 1208 yang diprioritaskan pertama. Untuk waktu juga diprioritaskan yang lebih awal.

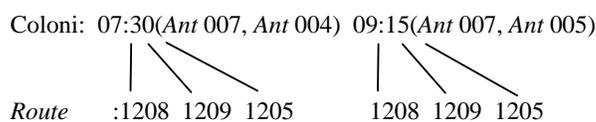
Ant 007(KodeMK:KK101 A,Waktu:07:30,09:15)

Ant 004(KodeMK:KK101 B,Waktu:07:30,11:00)

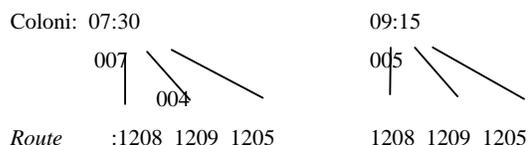
Ant 005(KodeMK:KK101 C,Waktu:09:15,11:00)



Gambar V.1 cara kerja ant pada kondisi awal



Gambar V.2 cara kerja ant disesuaikan dengan coloninya



**Gambar 5.1** cara kerja ant setelah ant memilih jalurnya

## 6. PERANCANGAN ANTAR MUKA

Untuk *interface* yang ditekankan adalah pemilihan menu, *direct manipulation*, *input device*, dan penggalan informasi. Pemilihan menu dibuat untuk mempermudah pengelompokkan perintah-perintah yang dapat dikerjakan. Sedangkan *direct manipulation* digunakan, karena pemakaian pemograman visual (Borland Delphi) yang objeknya sudah disediakan oleh program. *Input device* yang diutamakan dalam pengoperasian perangkat lunak adalah *keyboard*, karena dapat mengerjakan hal lebih cepat dan komputer harus ada *keyboard* untuk dapat mengerjakan sesuatu (komputer bisa juga tidak menggunakan *mouse*). Untuk penggalan informasi, perangkat lunak yang dibangun dapat menampilkan

informasi yang dibutuhkan untuk mempermudah serta mengurangi kesalahan pemasukan data.

Antar muka yang dirancang juga disesuaikan dengan kemudahan pemakaian terutama dalam menampilkan informasi tertentu. Antar muka pengisian waktu yang tidak boleh diisi, dirancang berdasarkan formulir sebenarnya yang telah digunakan untuk mengisi jadwal, sehingga pemakai tidak perlu beradaptasi. Antar muka pengisian waktu yang harus diisi, dirancang berdasarkan formulir sebenarnya dan tambahan informasi, menghitung total sks, total kelas, serta mata kuliah dan kelasnya, untuk mempermudah dosen dalam mengisi slot waktu yang digunakan untuk mengajar. Antar muka pengisian jadwal, dirancang berdasarkan papan pengisian jadwal secara manual dan tambahan informasi mata kuliah, kelas dan nomor dosen yang belum diisikan jadwalnya. Antar muka pengisian jadwal otomatis, dirancang berdasarkan pengisian jadwal secara langsung (tampilan grafik dirancang berdasarkan papan pengisian jadwal secara manual) dan informasi pengisian kelas dan mata kuliah pada slot waktu.

## **7. EVALUASI HASIL PENGUJIAN**

Hasil dari data sebenarnya dihasilkan dalam waktu yang lama (sekitar 1 sampai 2 minggu) karena masih menggunakan cara manual. Pada saat mengisi jadwal, setiap jadwal harus dicocokkan dengan jadwal yang diisi oleh dosen. Jika jadwal yang diisikan memang penuh pada jadwal yang sudah diisikan tersebut, jadwal yang diisi oleh dosen masih bisa dinegosiasikan. Untuk pengisian waktu oleh dosen disesuaikan dengan perkiraan kelas, sehingga ada kemungkinan perubahan pada saat proses pengisian jadwal dilakukan (jumlah kelas yang bisa bertambah atau berkurang). Setelah pendaftaran ulang, maka jumlah kelas dapat dihitung dan disesuaikan dengan dengan jumlah kelas sehingga jadwal yang diisi oleh dosen bisa ditambah. Proses pengisian jadwal ini dibutuhkan kerja keras, terutama pada saat menyesuaikan jadwal dengan jumlah ruang (ada saat waktu tertentu, jadwal terisi sangat padat dan ada juga saat jadwal kosong). Dilihat dari prosesnya yang cukup panjang, maka hasil dari data sebenarnya sudah mencerminkan kebutuhan.

Berdasarkan pemakaian perangkat lunak oleh pemakai, ada beberapa informasi yang sangat membantu dalam mengisi jadwal, yaitu :

1. Pada saat pemakai mengisi jadwal yang harus diisi oleh dosen, pemakai dapat melihat secara langsung berapa total sks yang dibutuhkan oleh dosen bersangkutan dalam mengisi jadwal.

2. Pada saat pemakai mengisi jadwal kuliah secara manual, terdapat informasi kelas, mata kuliah, dan nomor dosen yang belum diisikan pada jadwal.
3. Waktu pemakai mengisi jadwal secara manual, hasil pengisiannya dapat dilihat melalui grafik jadwal secara langsung.

Hasil penjadwalan otomatis didapatkan setelah semua waktu yang diisi oleh dosen, kelas yang diajar oleh dosen, dan ruang yang tersedia sudah benar-benar sesuai dengan kebutuhan. Prosesnya dikerjakan setelah jadwal dikonfirmasi karena ada penambahan atau pengurangan kelas dari daftar ulang siswa. Pada saat pengisian jadwal diproses, tidak dapat dilakukan perubahan jadwal (hal ini bisa dilakukan pada cara manual). Pengerjaannya tidak memerlukan kerja keras, tetapi dibutuhkan ketelitian pada saat mengisi data jadwal (waktu yang diisi oleh dosen, kelas yang diajar oleh dosen, dan ruang yang digunakan). Dilihat dari hasil penjadwalan otomatis, terdapat kekurangan pada mengisi mata kuliah yang diajar oleh dosen yang disesuaikan dengan urutan mata kuliah yang diambil. Khusus untuk mata kuliah yang lebih dari 2 sks, pengisian jadwal diisikan langsung untuk 2 pertemuan (pada kasus sebenarnya dimungkinkan tidak langsung 2 pertemuan).

Berdasarkan perbandingan hasil jadwal secara otomatis dan secara manual didapatkan kesimpulan bahwa masing-masing memiliki kelebihan dan kekurangan. Kelebihan menggunakan cara otomatis adalah pada masalah waktu memplot jadwal hanya dikerjakan dengan beberapa detik saja tetapi masih dibutuhkan beberapa penyesuaian pada mata kuliah yang ada pada jadwal tersebut, sedangkan pada cara manual dapat memakan waktu sehari-hari tetapi sudah sesuai dengan kebutuhan. Berikut ini kelebihan dari cara manual :

1. Dapat menambah atau mengubah jadwal pada saat proses pengisian jadwal berlangsung.
2. Dapat menyesuaikan jadwal dengan kriteria tertentu (misalnya jadwal ingin seimbang mata kuliahnya untuk tiap semester).
3. Waktu untuk mengajar bisa diatur dengan tidak mengikuti pembagian waktu (misalnya waktu mengajar dipadatkan agar tidak ada waktu sela / waktu menunggu jam berikutnya).

Selain itu, cara manual juga terdapat beberapa kekurangan, yaitu :

1. Waktu yang digunakan untuk pengisian jadwal, dari data yang sudah terkumpul sampai semua dosen sudah terisi jadwalnya, relatif lama (1 minggu sampai 2 minggu).
2. Pengisian jadwal memerlukan kerja keras, terutama pada saat menyesuaikan jadwal dengan jumlah ruang (ada saat waktu tertentu, jadwal terisi sangat padat dan ada juga saat jadwal kosong).

Sedangkan pada cara otomatis terdapat beberapa kelebihan, yaitu :

1. Pada saat pemakaian perangkat lunak terdapat beberapa informasi yang membantu dalam pengisian data (seperti informasi total sks, kelas yang belum diisi, dan grafik jadwal).
2. Pengisian jadwal tidak memerlukan kerja keras, tetapi memerlukan ketelitian pada saat mengisi data jadwal (waktu yang diisi oleh dosen, kelas yang diajar oleh dosen, dan ruang yang digunakan).
3. Waktu yang digunakan untuk pengisian jadwal, dari data yang sudah terkumpul sampai semua dosen sudah terisi jadwalnya, tidak lama (kurang dari 1 menit).
4. Untuk kasus *deadlock*, perangkat lunak tetap memberikan solusi berdasarkan nomor dosen terkecil. Untuk sisa jadwal yang belum dapat diisi, dapat diselesaikan dengan mengisi data secara manual dan mengkonfirmasi kepada dosen yang bersangkutan.

Dan kekurangan pada cara otomatis, yaitu :

1. Pada saat proses pengisian jadwal berlangsung, tidak dapat menambah atau mengubah data lain.
2. Proses bisa dilakukan jika semua data sudah terkumpul.
3. Mata kuliah yang diisikan pada jadwal disesuaikan dengan urutan mata kuliahnya (kelas yang diisikan pada jadwal pertama kali pasti kelas A, padahal dengan cara manual belum tentu kelas A yang diisikan pertama kali). Hal ini disebabkan karena perangkat lunak ini sangat tergantung pada batasan-batasan yang sudah ditentukan dan pada perangkat lunak ini belum terdapat batasan yang mampu mengatur kelas atau mata kuliah yang didahulukan.

## 8. KESIMPULAN

Kesimpulan yang didapatkan setelah melakukan penelitian :

1. Dengan menggunakan algoritma *ant* dengan metode berbasis ranking, mampu mengatasi masalah penjadwalan dengan baik.
2. Dengan menggunakan ranking sebagai standar, maka hasil yang didapat belum tentu hasil yang terbaik, tetapi hasil yang sesuai dengan ranking yang dapat diatur urutan ranking oleh pemakai.
3. Untuk masalah penjadwalan, batasan-batasan yang dibutuhkan adalah waktu yang bisa diisi untuk mengajar, ruangan yang dapat dipakai untuk mata kuliah tertentu, dan mata kuliah yang diajar oleh dosen (bisa berdasarkan kelas).
4. Dengan menggunakan ranking, dapat ditentukan beberapa batasan yang dapat didahulukan (misalnya untuk ruangan yang didahulukan adalah 1208, dapat diganti menjadi 1209 sehingga jadwal yang terisi 1209 lebih dahulu baru 1208).
5. Dengan menggunakan jadwal secara otomatis, mampu mengurangi banyak waktu dibandingkan dengan cara manual.
6. Pada saat pemakaian perangkat lunak terdapat beberapa informasi yang membantu dalam pengisian data (seperti informasi total sks, kelas yang belum diisi, dan grafik jadwal).
7. Pengisian jadwal tidak memerlukan kerja keras, tetapi memerlukan ketelitian pada saat mengisi data jadwal (waktu yang diisi oleh dosen, kelas yang diajar oleh dosen, dan ruang yang digunakan).
8. Perangkat lunak selalu memberikan solusi untuk batasan yang cocok dengan data jadwal yang ada.

## 9. DAFTAR PUSTAKA

1. Capriles, Priscila V. S. Z., Fonseca, Leonardo Goliatt da dan Barbosa, Helio J.C. *Ant Colony Algorithms Applied To Discrete Optimization Problems*. Download dari situs : 200.231.172.253/ cnmac/storal2/leonardo\_fonseca\_ST18.pdf didownload pada tanggal 29 Juni 2006.
2. Colorni, Alberto, Dorigo, Marco dan Maniezzo, Vittorio. *A genetic algorithm to solve the timetable problem*. Download dari situs : <http://citesser.comp.nus.edu.sg/colorni93genetic.html> pada tanggal 29 Juni 2006.
3. Ducatelle, Frederick.(2001). *Ant Colony Optimisation for Bin Packing and Cutting Stock Problems*. Master of Science School of Artificial Intelligence Division of Informatics University of Edinburgh.

4. Fang, Hsiao-Lan.(1994). *Genetic Algorithms in Timetabling and Scheduling*. Departement of Artificial Intelligence University of Edinburgh. Download dari situs : <http://citeseer.ist.psu.edu/fang94genetic.html> didownload pada tanggal 29 Juni 2006.
5. Gong, Wenrui, Wang, Gang dan Kastner, Ryan.(2005). *Instruction Scheduling Using MAX-MIN Ant System Optimization*. Download dari situs : [www.ece.ucsb.edu/~kastner/papers/glsvlsi05-instr\\_scheduling\\_as.pdf](http://www.ece.ucsb.edu/~kastner/papers/glsvlsi05-instr_scheduling_as.pdf) didownload pada tanggal 3 Mei 2006.
6. Jones, M Tim.(2003). *Ai Application Programming*. Charles river media.
7. Kohout, Peter. *Genetic and Ant Colony Optimization Algorithms*. Download dari situs : [www.codeproject.com/cpp/GeneticandAntAlgorithms.asp](http://www.codeproject.com/cpp/GeneticandAntAlgorithms.asp) didownload pada tanggal 22 Mei 2006.
8. Montgomery, Erin James.(2005). *Solution Biases and Pheromone Representation Selection In Ant Colony Optimisation*. Bond University.
9. Preece, Rogers dan Sharp.(2002). *Interaction Design – Beyond Human-Computer Interaction*. John Wiley & Sons, Inc.
10. Shneiderman, Ben.(1998). *Designing The User Interface – Strategies For Effective Human-Computer Interaction*. Addison-Wesley Longman, Inc.
11. Socha, Krzysztof, Sampels, Michael dan Manfrin, Max. *Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art*. Download dari situs : <http://citeseer.ist.psu.edu/socha03ant.html> di download pada tanggal 29 Juni 2006.
12. Solnon, Christine.(2002). *Ants Can Solve Constraint Satisfaction Problems*. Download dari situs : [www.cs.dartmouth.edu/~spl/Academic/AI/support/AntsCanSolveConstraintSatisfactionProblems.pdf](http://www.cs.dartmouth.edu/~spl/Academic/AI/support/AntsCanSolveConstraintSatisfactionProblems.pdf) didownload pada tanggal 17 Juli 2006.
13. WWW.asctimetables.com.
14. WWW.cs.nott.ac.uk/~gxx/courses/g5baim/005ants/powerpoint/antalgorithms.ppt.
15. WWW.timetabler.com.
16. WWW.timetablerplus.com.
17. WWW.wikipedia.org.