

**PEMBANGUNAN PERANGKAT LUNAK MENGGUNAKAN ALGORITMA
ANT COLONY OPTIMIZATION UNTUK OPTIMALISASI
PENJADWALAN KULIAH
(STUDI KASUS PENJADWALAN KULIAH JURUSAN TEKNIK KOMPUTER
UNIKOM)**

Selvia Lorena Br Ginting¹, Hayi Akbar²

^{1,2} Jurusan Teknik Komputer Unikom, Bandung

¹ selvia Lorena@yahoo.com, ² ayi.akbar@gmail.com

Abstrak

Salah satu kegiatan penjadwalan yang membutuhkan perhatian khusus dan ketelitian yang cukup tinggi adalah penjadwalan kuliah. Menciptakan suatu jadwal kuliah yang optimal cukup sulit dikarenakan banyak variabel yang saling terkait sehingga membutuhkan suatu penanganan yang signifikan. Penjadwalan kuliah merupakan pengaturan penempatan waktu dan ruangan berdasarkan jumlah kuliah dan akademik sejenis, dengan memperhatikan sejumlah aturan yang berhubungan dengan kapasitas dan lokasi dari ruangan yang tersedia, waktu bebas yang diperlukan dan sejumlah aturan lain yang berkaitan dengan toleransi untuk dosen dan hubungan antar mata kuliah khusus. Pengaturan penjadwalan kuliah dapat dilakukan secara manual maupun dengan otomatisasi (*software*) yang mana dalam menentukan jadwal-jadwal tersebut adalah sangat rumit dan membutuhkan waktu yang cukup lama. Tidak jarang terjadi jadwal yang bentrok satu sama lain baik untuk mahasiswa, dosen ataupun ruangan kelas. Untuk memudahkan pengaturan penjadwalan kuliah yang efektif dan optimal melalui suatu proses otomatisasi, penulis akan membangun perangkat lunak dengan mengimplementasikan algoritma *Ant Colony Optimization*, agar mahasiswa, dosen dan ruangan tidak mengalami jadwal yang bentrok antara satu mata kuliah dengan mata kuliah yang lain. Algoritma *Ant Colony Optimization* adalah salah satu jenis algoritma *meta-heuristic* yang sudah terbukti dapat menyelesaikan banyak sekali permasalahan *kombinatorial* yang sulit. Algoritma ini meniru tingkah laku semut ketika mereka berada di dalam sebuah koloni untuk mencari sebuah sumber makanan. Algoritma semut lebih menitikberatkan pada perilaku kebiasaan atau yang sering dilakukan (dilewati) oleh semut. Dengan kemampuan dan keunikan semut, yang memiliki kemampuan secara alami (*real ant*) untuk menemukan alur atau lintasan terpendek dari sarangnya ke suatu sumber makanan tanpa pengertian visual (penglihatan). Perangkat lunak yang dibangun menggunakan algoritma *Ant Colony Optimization* ini mampu menghasilkan penjadwalan kuliah dengan kesediaan waktu tertentu dari dosen dan kesediaan ruangan serta mampu memberikan informasi jadwal kuliah yang bentrok baik dari dosen maupun ruangan kelas.

Kata kunci: Penjadwalan Kuliah, Bentrok, Algoritma, *Ant Colony Optimization*

1. PENDAHULUAN

Kegiatan perkuliahan umumnya dilakukan dengan tatap muka antara pengajar (dosen) dengan mahasiswa dalam waktu yang terbatas. Lamanya waktu perkuliahan tergantung dari berapa SKS mata kuliah tersebut. SKS (Satuan Kredit Semester), adalah satuan yang digunakan dalam Sistem Kredit Semester yaitu suatu sistem penyelenggaraan perkuliahan di perguruan tinggi yang menyatakan beban studi mahasiswa, beban kerja dosen dan beban penyelenggaraan perkuliahan dengan menggunakan Satuan Kredit Semester (SKS). Besarnya satu satuan kredit semester (1 SKS) dalam kegiatan perkuliahan pada berbagai perguruan tinggi terkadang berbeda namun ada juga yang sama. Kebanyakan suatu Universitas menerapkan 1 SKS dalam kegiatan perkuliahan adalah setara dengan alokasi waktu lima puluh (50) menit kegiatan tatap muka terjadwal antara dosen dengan mahasiswa. Terbatasnya waktu tatap muka antara dosen dengan mahasiswa sering disertai dengan keterbatasan ruang juga kesediaan waktu dari dosen-dosen kemudian ada banyak mata kuliah yang diadakan dalam 1 semester, akan sulit bagi staff di suatu jurusan dan membutuhkan waktu yang lama untuk menjadwalkan kegiatan perkuliahan tersebut. Dalam proses penentuan jadwal kuliah ini terdapat banyak aspek yang harus dipertimbangkan yaitu waktu per SKS, waktu mahasiswa, ketersediaan dosen, matakuliah di masing-masing semester dan ruangan kelas, tidak jarang terdapat

jadwal bentrok satu sama lain baik dari sisi mahasiswanya, dosen ataupun ruangnya. Oleh karena itu dalam menentukan jadwal kuliah di setiap semesternya dibutuhkan pembangunan suatu perangkat lunak yang menerapkan suatu metode atau algoritma yang akan menghasilkan penjadwalan kuliah yang optimal. Algoritma yang diterapkan pada pembangunan perangkat lunak penjadwalan kuliah ini adalah algoritma dengan menggunakan algoritma *ant colony Optimization*. Algoritma ini lebih menitikberatkan pada perilaku semut, diinspirasi oleh koloni semut karena tingkah laku mereka yang menarik ketika mencari makanan. Semut-semut menemukan jarak terpendek antara sarang semut dan sumber makanannya. Ketika berjalan dari sumber makanan menuju sarang mereka, semut memberikan tanda dengan zat feromon sehingga akan tercipta jalur feromon. Feromon adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup untuk mengenali sesama jenis, individu lain, kelompok, dan untuk membantu proses reproduksi.

Dengan menggunakan algoritma ini maka sistem akan mencari cara tercepat untuk mencapai suatu penempatan jadwal yang optimal dengan kata lain, jadwal mahasiswa, dosen dan ruangan tidak akan mengalami bentrok untuk satu mata kuliah dengan mata kuliah yang lain. Tujuan dari pembangunan perangkat lunak ini adalah untuk mempermudah pengaturan jadwal kuliah pada Jurusan Teknik Komputer, FTIK Unikom. Namun

pada penelitian ini diasumsikan kapasitas ruangan sama dan mengabaikan mata kuliah laboratorium dikarenakan pada kondisi nyata, untuk masalah laboratorium dikelola oleh Koordinator Laboratorium.

2. Dasar Teori

2.1 Penjadwalan

Penjadwalan merupakan suatu proses pengorganisasian waktu untuk mendapatkan waktu yang efektif dan optimal. Penjadwalan Kuliah merupakan pengaturan penempatan waktu dan ruangan berdasarkan jumlah kuliah dan akademik sejenis, dengan memperhatikan sejumlah aturan yang berhubungan dengan kapasitas dan lokasi dari ruangan yang tersedia, waktu bebas yang diperlukan dan sejumlah aturan lain yang berkaitan dengan toleransi untuk dosen, dan hubungan antar mata kuliah khusus.

Menurut Kamus Besar Bahasa Indonesia, kuliah adalah pelajaran yg diberikan di perguruan tinggi. Terdapat berbagai istilah yang dikenal pada kegiatan kuliah (perkuliahan) di perguruan tinggi, diantaranya mahasiswa, dosen, mata kuliah, SKS, dan semester.

1. Mahasiswa adalah orang yang belajar di perguruan tinggi dan merupakan pelaku kegiatan kuliah di perguruan tinggi yang menerima pelajaran dari dosen.
2. Dosen adalah tenaga pengajar pada perguruan tinggi dan merupakan pelaku kegiatan kuliah di perguruan tinggi yang

memberi pelajaran kepada mahasiswa.

3. Mata kuliah adalah satuan pelajaran di tingkat perguruan tinggi yang diajarkan oleh dosen kepada mahasiswa.
4. SKS (Satuan Kredit Semester), adalah satuan yang digunakan dalam Sistem Kredit Semester. Sistem kredit semester adalah suatu sistem penyelenggaraan perkuliahan di perguruan tinggi yang menyatakan beban studi mahasiswa, beban kerja dosen dan beban penyelenggaraan perkuliahan dengan menggunakan Satuan Kredit Semester (SKS). Besarnya satu satuan kredit semester (1 SKS) dalam kegiatan perkuliahan pada berbagai perguruan tinggi terkadang berbeda namun ada juga yang sama. Di Universitas Pendidikan Indonesia, nilai 1 SKS dalam kegiatan perkuliahan adalah setara dengan alokasi waktu lima puluh (50) menit kegiatan tatap muka terjadwal antara dosen dengan mahasiswa.
5. Semester adalah satuan terkecil untuk lamanya masa program pendidikan. Jenjang pendidikan strata satu (S1) terdiri atas 8 semester. Satu tahun akademik (penyelenggaraan kegiatan perkuliahan dalam satu tahun) dibagi menjadi dua semester, yaitu semester ganjil dan semester genap

2.2 Algoritma Ant Colony

Algoritma adalah suatu urutan langkah atau prosedur yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan. Lima

komponen utama dalam algoritma yaitu *finiteness*, *definiteness*, *input*, *output* dan *effectiveness*. Sehingga dalam merancang sebuah algoritma ada 3 (tiga) komponen yang harus ada yaitu:

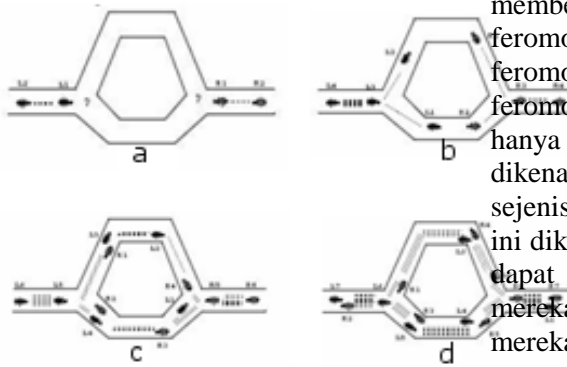
- a. Komponen masukan (*input*) terdiri dari pemilihan variabel, jenis variabel, tipe variabel, konstanta dan parameter (dalam fungsi).
- b. Komponen keluaran (*output*) merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah benar (menjawab) permasalahan dan tampilan yang ramah (*friendly*).
- c. Komponen proses (*processing*) 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 *Ant Colony* pertama kali dikenalkan oleh Marco Dorigo pada tesis Ph.D tahun 1992. Algoritma ini lebih menitik beratkan pada perilaku kebiasaan atau yang sering 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). Dengan menggunakan algoritma ini maka

sistem akan mencari cara tercepat untuk mencapai suatu penempatan jadwal.

Berikut adalah cara kerja *Ant Colony*:

1. Pada awalnya, semut berkeliling secara acak.
2. Ketika semut-semut menemukan jalur yang berbeda misalnya sampai pada persimpangan, mereka akan mulai menentukan arah jalan secara acak seperti pada Gambar 2.1.a.
3. Sebagian semut memilih berjalan ke atas dan sebagian lagi akan memilih berjalan ke bawah seperti pada Gambar 2.1.b.
4. Ketika menemukan makanan mereka kembali ke koloninya sambil memberikan tanda dengan jejak feromon.
5. Karena jalur yang ditempuh lewat jalur bawah lebih pendek, maka semut yang bawah akan tiba lebih dulu dengan asumsi kecepatan semua semut adalah sama seperti pada Gambar 2.1.c.
6. Feromon yang ditinggalkan oleh semut di jalur yang lebih pendek aromanya akan lebih kuat dibandingkan feromon di jalur yang lebih panjang seperti pada Gambar 2.1.d.
7. Semut-semut lain akan lebih tertarik mengikuti.



Gambar 2.1 Cara Kerja Semut

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 feromon yang tinggi.
2. Alur yang lebih pendek cenderung untuk suatu tingkat pertumbuhan nilai feromon yang lebih tinggi.
3. Dengan menggunakan suatu sistem komunikasi tak langsung melalui *phromonea edge*.

2.3 Algoritma Ant Colony Optimization

Algoritma *Ant colony Optimization* merupakan teknik probabilistik untuk menjawab masalah komputasi yang bisa dikurangi dengan menemukan jalur yang baik dengan graf. ACO di inspirasi oleh koloni semut karena tingkah laku semut yang menarik ketika mencari makanan. Semut-semut menemukan jarak terpendek antara sarang semut dan sumber makanannya. Ketika berjalan dari sumber makanan menuju sarang mereka, semut

memberikan tanda dengan zat feromon sehingga akan tercipta jalur feromon. Berbeda dengan hormon, feromon menyebar ke luar tubuh dan hanya dapat mempengaruhi dan dikenali oleh individu lain yang sejenis, proses peninggalan feromon ini dikenal sebagai *stigmergy*. Semut dapat mencium feromon dan ketika mereka memilih jalur mereka, mereka cenderung memilih jalur yang ditandai oleh feromon dengan konsentrasi yang tinggi. Apabila semut telah menemukan jalur yang terpendek maka semut-semut akan terus melalui jalur tersebut. Jalur lain yang ditandai oleh feromon lama akan memudar atau menguap, seiring berjalannya waktu. Jalur-jalur yang pendek akan mempunyai ketebalan feromon dengan probabilitik yang tinggi dan membuat jalur tersebut akan dipilih dan jalur yang panjang akan ditinggalkan. Jalur feromon membuat semut dapat menemukan jalan kembali ke sumber makanan atau sarang mereka.

Koloni semut yang nyata dan *artificial* terdapat banyak kemiripan. Keduanya terbentuk dari sebuah populasi yang terdiri dari individu-individu yang berkerja sama untuk mencapai tujuan. Semut *artificial* hidup di dunia virtual, karenanya mereka hanya memodifikasi nilai numerik (disebut analogi *artificial pheromones*) yang berhubungan dengan keadaan-keadaan permasalahan yang berbeda. Sebuah rangkaian dari nilai-nilai feromon yang berhubungan dengan keadaan permasalahan disebut *pheromone trail* atau jejak feromon. Mekanisme untuk evaporation atau penguapan

feromon pada koloni semut nyata yang membuat semut *artificial* dapat melupakan sejarah (jalur-jalur yang pernah diambil) dan fokus pada arah pencarian baru yang menjanjikan. Seperti semut-semut nyata, semut-semut *artificial* membuat solusi secara berurutan dengan bergerak dari satu keadaan permasalahan ke lainnya. Semut-semut nyata hanya berjalan, memilih arah berdasarkan konsentrasi feromon lokal dan kebijakan keputusan stokastik. Semut *artificial* membuat solusi sedikit demi sedikit, dan bergerak dari keadaan permasalahan yang tersedia dan membuat keputusan stokastik setiap langkah. Meskipun begitu, terdapat perbedaan antara yang nyata dan semut *artificial* sebagai berikut.

- a. Semut *artificial* hidup di dunia dan pada waktu diskrit, mereka berpindah secara sekuen melewati set batasan dari permasalahan.
- b. *Update* feromon (penumpukan dan penguapan feromon) tidak dilakukan dengan jalan yang sama pada semut yang nyata dan semut *artificial*. *Update* feromon dilakukan oleh beberapa dari semut *artificial* dan terkadang dilakukan saat solusi telah dibangun.
- c. Beberapa implementasi dari semut *artificial* menggunakan mekanisme tambahan yang tidak ada pada semut-semut nyata, seperti *local search*, *backtracking*, dan lain-lain.

Dalam Algoritma *Ant Colony Optimization*, 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 (τ_{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 (α), nilai $\alpha \geq 0$.
6. Tetapan pengendali visibilitas (β), nilai $\beta \geq 0$.
7. Visibilitas antar wilayah = $1/d_{ij}$ (η_{ij}).
8. Banyak semut (m).
9. Tetapan penguapan jejak semut (ρ), nilai ρ harus > 0 dan < 1 untuk mencegah jejak *pheromone* yang tak terhingga.
10. Jumlah siklus maksimum (NC_{max}) bersifat tetap selama algoritma dijalankan, sedangkan τ_{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 τ_{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 (1) 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 sebagai titik tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua titik satu persatu dikunjungi atau telah menempati tabu. Jika s menyatakan indeks urutan kunjungan, titik asal dinyatakan sebagai tabu(s) dan titik-titik lainnya dinyatakan sebagai {N-tabu}, maka untuk menentukan titik tujuan digunakan persamaan probabilitas titik untuk dikunjungi sebagai berikut:

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k' \in (N-tabu_k)} [\tau_{ik'}]^\alpha [\eta_{ik'}]^\beta}$$

Untuk j {N-tabu} (1)

$$P_{ij}^k = 0$$

Untuk j lainnya (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 Lk setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini dilakukan berdasarkan tabu masing-masing dengan persamaan berikut:

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

dengan dij adalah jarak antara titik i ke titik j yang dihitung berdasarkan persamaan:

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

b. Pencarian rute terpendek.

Setelah Lk setiap semut dihitung, akan didapat harga minimal panjang rute tertutup setiap siklus atau LminNC dan harga minimal panjang rute tertutup secara keseluruhan adalah atau Lmin.

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:

$$\Delta\tau_{ij}^k = \frac{Q}{L_k},$$

untuk (i,j) kota asal dan titik tujuan dalam tabu
(6)

$$\Delta\tau_{ij}^k$$

untuk (i,j) lainnya
 (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:

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

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 *5b* 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: α , β , lintasan penting yang dilalui serta menarik, ρ , 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 τ_{ij} and η_{ij} , $\forall (ij)$.
2. {Contruction}
 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.
 until ant k has completed its solution.
 end for

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

3. Analisis Sistem

Perangkat lunak yang dibangun fungsinya adalah untuk mengatur penjadwalan kuliah yang efektif sehingga tidak terdapat jadwal yang bentrok satu sama lain. Algoritma *Ant Colony Optimization* yang diterapkan pada perangkat lunak ini akan memasang *event* yang berisi komponen mata kuliah, kelas dan dosen pada *time slot* berisi komponen waktu dan ruang. Dalam pengaturan penjadwalan kuliah sendiri terdapat berbagai *constraint* yang berkaitan dalam penjadwalan yang harus dilibatkan sebagai pertimbangan yang dapat menyebabkan persoalan. *Constraint* sendiri merupakan kendala yang didefinisikan agar jadwal yang dihasilkan merupakan jadwal yang baik. *Constraint* dibagi menjadi dua kategori yaitu *hard constraint* dan *soft constraint*. *Hard constraint* adalah suatu syarat dimana tidak boleh terjadi pelanggaran terhadap kendala yang ditetapkan, yaitu:

1. Satu kelas tidak dapat belajar pada ruangan yang berbeda dalam satu waktu.
2. Satu kelas tidak dapat belajar lebih dari satu mata kuliah dalam satu waktu.
3. Dosen tidak dapat mengajar lebih dari satu mata kuliah dalam satu waktu.

4. Dosen tidak dapat mengajar lebih dari satu kelas dalam satu waktu.
5. Ruang tidak dapat dipakai lebih dari satu mata kuliah dalam satu waktu.

Soft constraint adalah kendala-kendala yang mungkin dilanggar, karena pelanggaran *soft constraint* tidak akan mempengaruhi keberhasilan penjadwalan, namun walaupun terjadi pelanggaran *soft constraint* maka jadwal yang dihasilkan bukan merupakan hasil yang nyaman untuk digunakan. Penerapan *soft constraint* ini biasanya digunakan untuk membuat penjadwalan lebih baik, karena semakin sedikit jumlah pelanggaran *soft constraint* maka hasil penjadwalan sesuai dengan harapan. Penentuan *soft constraint* itu akan berbeda sesuai dengan kebutuhan dan kebijakan instansi dalam hal ini Universitas yang bersangkutan. Maka sesuai dengan penelitian yang dilakukan di Jurusan Teknik Komputer tahun 2012 sampai 2013 semester ganjil dan genap, *Soft constraint* yang ada dan sebaiknya terpenuhi, yaitu:

1. Distribusi perkuliahan diharapkan dapat merata tiap harinya untuk setiap kelas.
2. Tidak boleh adanya jadwal kuliah yang berisian dengan jadwal kuliah angkatan sebelumnya maupun sesudahnya, sehingga mahasiswa dapat mengambil mata kuliah angkatan sebelumnya maupun sesudahnya.
3. Dari banyak hari perkuliahan dalam satu minggu diharapkan

setiap kelas mendapatkan satu atau dua hari perkuliahan yang kosong.

4. Mata kuliah yang dijadwalkan diharapkan memiliki rentang waktu yang cukup untuk mengikuti kuliah selanjutnya,

User dapat menggunakan semua fitur pada aplikasi penjadwalan *ant colony* seperti menambah, menghapus, mengedit data baik untuk data dosen, data matakuliah, data ruang, data kelas, data matakuliah tahun, data jurusan, data fakultas, data kesediaan dosen, data kesediaan ruangan, penjadwalan metoda *ant colony* sehingga mengupdate database yang diinginkan dan user dapat menjadwalkan matakuliah menggunakan algoritma *ant colony* berdasarkan database yang telah ada.

Data yang akan digunakan dalam penjadwalan kuliah Algoritma *Ant Colony Optimaliation* berdasarkan pada jadwal perkuliahan semester 1-8 tahun akademik 2012-2013 pada Jurusan Teknik Komputer Universitas Komputer Indonesia. Dimana meliputi matakuliah, ruangan dan dosen sedangkan untuk laboratorium dan kerja praktek diatur oleh Koordinator Lab (diluar pihak jurusan).

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

1. Intensitas jejak semut antar titik (τ_{ij}) dan perubahannya ($\Delta\tau_{ij}$). τ_{ij} harus diinisialisai sebelum memulai siklus. τ_{ij} digunakan

dalam persamaan probabilitas titik yang akan dikunjungi. $\Delta\tau_{ij}$ diinisialisasi setelah selesai satu siklus. $\Delta\tau_{ij}$ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya.

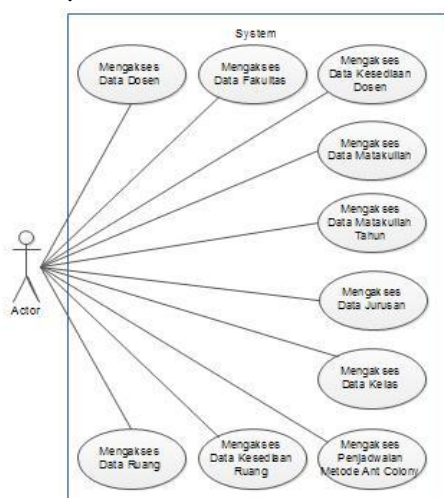
2. Tetapan siklus semut (Q). Q merupakan konstanta yang digunakan dalam persamaan untuk menentukan $\Delta\tau_{ij}$. Nilai Q ditentukan oleh pengguna.
3. Tetapan pengendali intensitas jejak semut (α). α digunakan dalam persamaan probabilitas titik yang akan dikunjungi yang berfungsi sebagai pengendali intensitas jejak semut. Nilai α ditentukan oleh pengguna.
4. Tetapan pengendali visibilitas (β). β digunakan dalam persamaan probabilitas titik yang akan dikunjungi dan berfungsi sebagai pengendali visibilitas. Nilai β ditentukan oleh pengguna.
5. Visibilitas antar titik (η_{ij}). η digunakan dalam persamaan probabilitas titik yang akan dikunjungi. Nilai η merupakan hasil dari $1/d_{ij}$ (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 (ρ). ρ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya. Nilai ρ 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 NCmax ditentukan oleh pengguna.

Data output yang diperoleh user dari proses aplikasi penjadwalan matakuliah dengan menggunakan Algoritma *Ant Colony* ini yaitu kombinasi antara kode matakuliah, nama matakuliah, nama dosen, hari, jam, jumlah SKS, dan ruang yang digunakan.

3.1 Diagram Use Case

Diagram *use case* adalah sebuah diagram yang digunakan untuk menunjukkan tampilan grafis dari fungsionalitas yang diberikan oleh sistem dilihat dari sisi aktor, tujuan aktor, dan hal yang berkaitan dengan *use case* yang ada



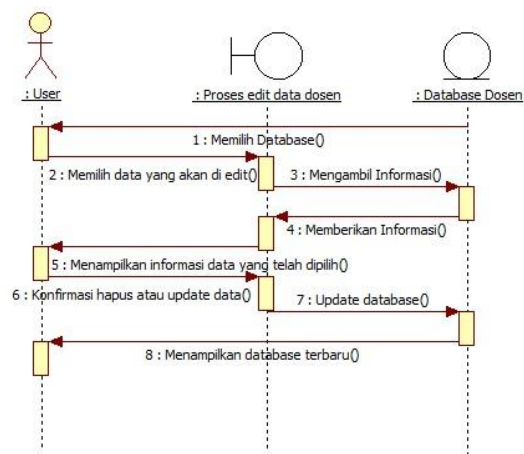
Gambar 3.1 Diagram Use Case

Diagram *Use Case* ini menggambarkan user dapat menggunakan seluruh fitur aplikasi penjadwalan dengan menggunakan Algoritma *Ant Colony Optimization* seperti penambahan, pengurangan dan pemrosesan data dosen, data matakuliah, data ruang, data kelas,

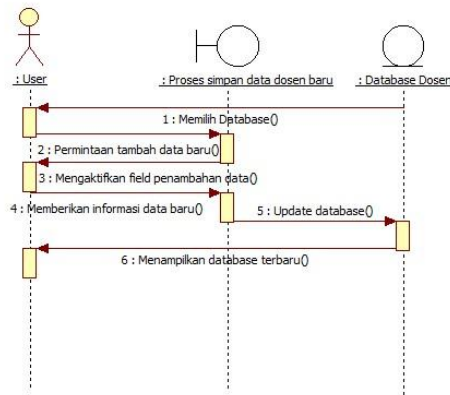
data matakuliah tahun, data jurusan, data fakultas, data kesiediaan dosen, data kesiediaan ruangan dan penjadwalan *Ant Colony*.

3.2 Diagram Sequence

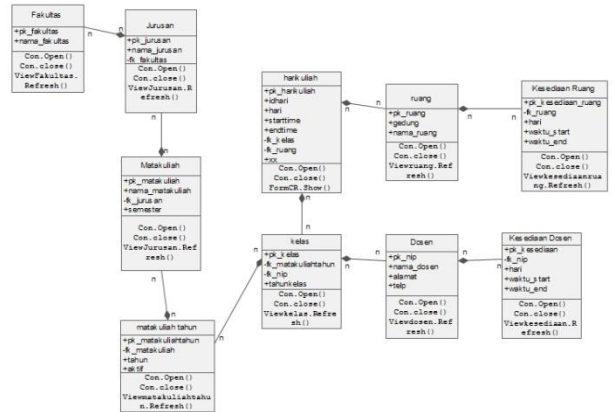
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.



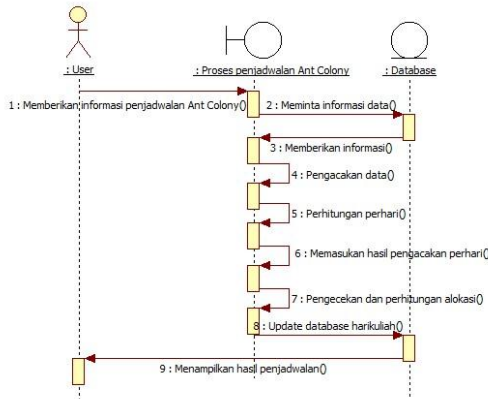
Gambar 3.2 Diagram Sequence Proses Edit



Gambar 3.3 Diagram Sequence Proses Simpan



Gambar 3.5 Diagram Kelas



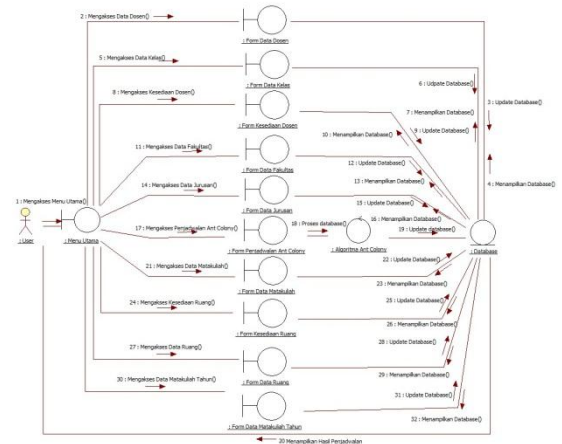
Gambar 3.4 Diagram Sequence Proses Penjadwalan Ant Colony

3.3 Diagram Kelas

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

3.4 Collaboration Diagram

Collaboration Diagram menekankan pada organisasi objek-objek yang berpartisipasi pada interaksi. Collaboration Diagram member pemahaman yang jelas pada pembaca tentang aliran kendali (flow of control) dalam konteks organisasi structural objek-objek yang berkolaborasi

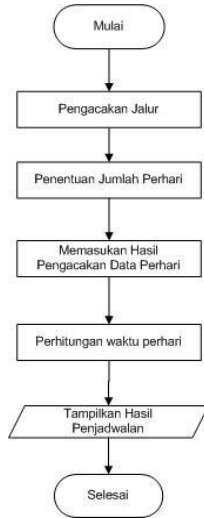


Gambar 3.6 Collaboration Diagram

3.5 Flowchart

Flowchart merupakan bagan yang menunjukkan alir di dalam program

atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

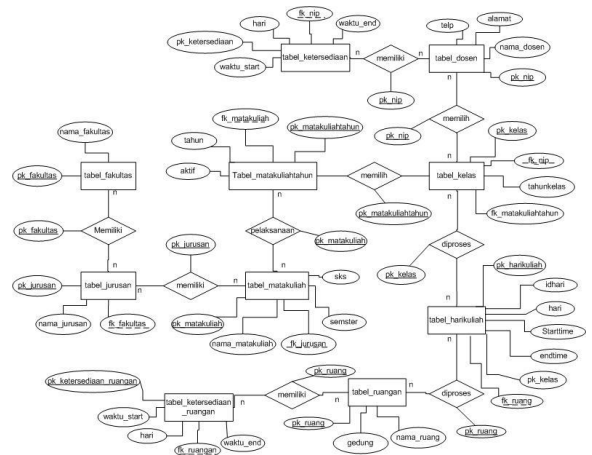


Gambar 3.7 Flowchart Algoritma Ant Colony Optimization

4. Perancangan Sistem

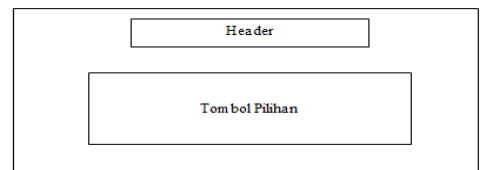
4.1 Entity Relationship Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi yang memodelkan struktur data dan hubungan antar data, untuk meng gambarkannya digunakan beberapa notasi dan simbol.



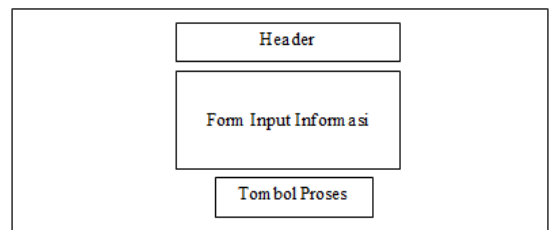
Gambar 4.1 ERD

4.2. Perancangan Tampilan Awal



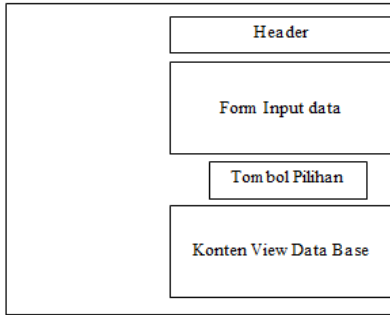
Gambar 4.2 Tampilan Awal

4.3 Perancangan Tampilan Penjadwalan



Gambar 4.3 Tampilan Penjadwalan

4.4 Perancangan Tampilan Submenu



Gambar 4.4 Tampilan Submenu

menggunakan Algoritma *Ant Colony Optimization* di mana terdapat beberapa pilihan seperti data dosen, data matakuliah, data ruang, data kelas, data matakuliah tahun, data jurusan, data fakultas, data kesediaan dosen, data kesediaan ruangan, penjadwalan metoda *Ant Colony* yang akan ditambahkan atau mengedit *database*-nya.

5 Implementasi dan Pengujian Sistem

5.1 Implementasi

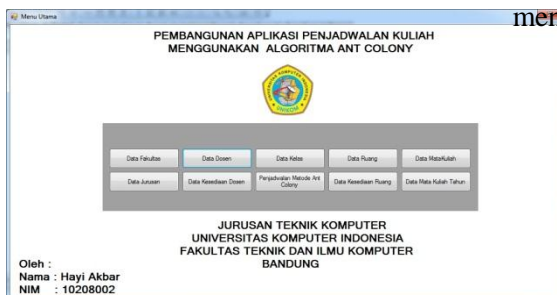
Tahap implementasi dilakukan dengan mentransformasikan fungsionalitas algoritma ke dalam kode-kode program kemudian diperiksa kesesuaian program yang dihasilkan dengan perancangan yang telah dilakukan sebelumnya yang tentunya sesuai dengan spesifikasi kebutuhan



Gambar 5.2 Data Dosen

Berikut adalah tampilan bagian-bagian perangkat lunak yang telah dibangun.

Gambar 5.2 merupakan informasi dosen dimana terdapat NIP dosen, nama dosen, alamat dan telepon yang dapat ditambahkan ataupun mengedit *database* dosen.



Gambar 5.1 Tampilan Menu Utama

Gambar 5.1 merupakan menu utama aplikasi penjadwalan kuliah

penambahan atau pengeditan database.



Gambar 5.3 Data Kelas

Gambar 5.3 merupakan informasi mengenai hubungan dosen dengan matakuliah yang akan diajarkan untuk setiap kelasnya baik untuk semester ganjil maupun genap dimana terdapat penambahan atau pengeditan database.



Gambar 5.5 Data Ruang

Gambar 5.5 merupakan informasi ruangan yang telah terdaftar dimana terdapat tombol penambahan ataupun pengeditan data ruangan.



Gambar 5.4 Data Jurusan

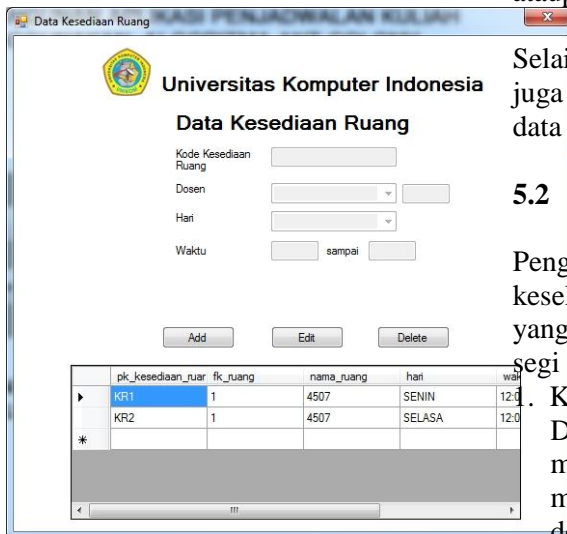
Gambar 5.44 merupakan informasi hubungan jurusan dengan fakultasnya dimana terdapat



Gambar 5.6 Data Mata Kuliah

Gambar 5.6 merupakan informasi semua matakuliah beserta untuk

jurusan apa dimana terdapat tombol penambahan ataupun pengeditan data matakuliah.



Gambar 5.7 Data Kesiadaan Ruang

Gambar 5.7 merupakan informasi kesiadaan ruangan yang tidak selalu dapat dipakai setiap harinya dimana terdapat tombol penambahan ataupun pengeditan data fakultas.

Gambar 5.8 merupakan informasi kesiadaan dosen yang akan mengajar terdapat tombol penambahan ataupun pengeditan data fakultas.

Selain tampilan-tampilan di atas, juga ada tampilan data fakultas dan data mata kuliah tahun.

5.2 Pengujian Sistem

Pengujian ini dilakukan untuk keseluruhan sistem dari aplikasi yang dibangun ditinjau dari berbagai segi sebagai berikut:

1. Kebutuhan

Dari segi kebutuhan dalam menangani kasus penjadwalan matakuliah ini seperti adanya dosen, matakuliah, semester, ruang, hari, dan waktu.

2. Kegunaan

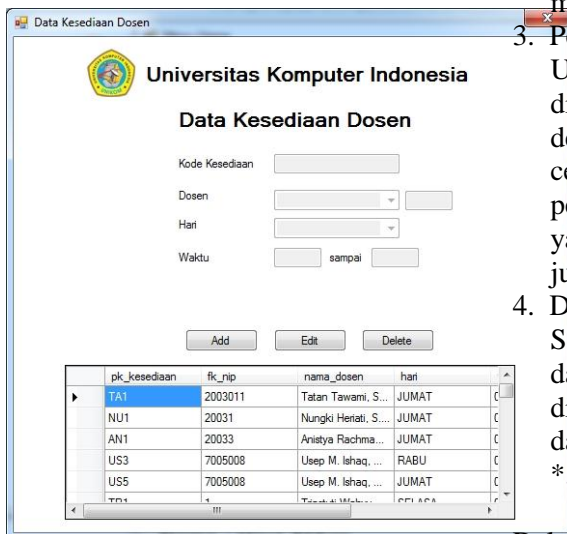
Kegunaan dari sistem aplikasi ini dari awal bertujuan untuk mendapatkan solusi penjadwalan matakuliah yang optimal secara otomatis dengan menggunakan metoda *Ant Colony*.

3. Performa

Untuk performa sistem sendiri diinginkan penjadwalan otomatis dengan penemuan solusi yang cepat, tetapi setelah dilakukan penganalisaan terdapat waktu yang cukup lama dikarenakan jumlah data yang banyak

4. Dokumentasi

Sebuah dokumentasi agar mudah dalam penggunaan dapat disimpan dan digunakan file dalam beberapa format seperti *.xls, .pdf, .doc



Gambar 5.8 Data Kesiadaan Dosen

Dalam pengujian perangkat lunak Penjadwalan Kuliah diperlukan beberapa informasi seperti jam awal

sebagai awal perkuliahan dimulai, jam akhir sebagai akhir perkuliahan setiap harinya, sks/menit untuk mengatur berapa lama per sksnya dan pemilihan semester ganjil atau genap untuk melakukan penjadwalannya pemrosesan sendiri membutuhkan waktu yang cukup lama dikarenakan banyaknya data yang diproses. Setelah memasukan kelas dan jadwal dosen dapat melakukan proses penjadwalan dengan inputan jam awal dan jam akhir, berapa menit per sks, tahun ajaran seperti yang ditunjukan Gambar 5.9 Form Penjadwalan.

4. Dari total misal ada 20 kelas akan di bagi hari (6) dirata2 menjadi 3-4 kelas per hari.
5. Menyesuaikan kelas dengan dosennya.
6. Mengurutkan jalur yang sudah diacak sebelumnya.
7. Perhitungan waktu perhari nya dari batas waktu awal sampai akhir perkuliahan perhari nya.
8. Melakukan pengacakan ruang yang disesuaikan dengan kelas yang ada.
9. Menentukan ruang sesuai dengan jam digunakannya ruang.
10. Setelah proses selesai, maka akan di tampilkan jadwalnya.



Gambar 5.9 Form Penjadwalan

Pada saat pemrosesan, tahapan yang dilakukan :

1. Mapping antara tabel_kelas dengan tabel_kesediaan dosen maka akan membentuk tabel_kelas_dosen.
2. Perhitungan jarak antara waktu perhari.
3. Menentukan jumlah perhari nya dari hasil pengacakan jalur.

Setelah diproses menggunakan Algoritma *Ant Colony* maka akan memasuki *Cristal Report* yang berguna untuk media pengolahan atau penyimpanan hasil penjadwalan yang lebih lanjut dalam bentuk format microsoft excel, microsoft word ataupun portable document format.

Kelas Mata Kuliah	Nama Mata Kuliah	Nama Dosen	Ruang	Kelas	EndTime
SENIN 08:00	TC30201 Teknik Informatika Dasar	Wendy Zamani, M.Si.	4501	08N1	10:00
08:45	TC30205 Antarmuka Komputer	I. Sulistiyo, MT	4517	09N2	11:00
11:00	TC30207 Komunikasi Data	Suzana Indriani, L. MT	4519	11N4	13:00
11:15	TC30203 Praktek 8	Umar M. Sidiq, M.Si.	4517	07N3	12:30
SELASA 08:00	TC30208 Kalkulus II	Sumardi Satriyo Utomo, M.	4519	07N4	09:00
TC30206 Agama dan Etika I	Prof. Dr. H. Sutopo Satrio,	4517	07N1	09:30	
08:15	TC30209 Bahasa Sastra	Sri Rahmawati, S.Si., MT	4507	11N2	11:30

Gambar 5.10 Cristal Report

Berikut merupakan hasil penyimpanan berformat microsoft excel setelah melalui tahap *Cristal Report*.

	Kode Mata	Nama Mata Kuliah	Nama Dosen	Ruang	Kelas	EndTime
SENIN 09:15:00	TK36201	Teknik Menulis Inisiah	Wandi Zaman, M.Si.	4021	10TK1	10:45:00
	TK36205	Antarmuka Komputer	I. Syahrial, MT.	4017	10TK2	13:00:00
	TK34207	Komunikasi Data	Suamni Indrani L., MT.	4019	11TKA	15:15:00
	TK32103	Praktik II	Ungap M. Inahaj, M.Si.	4017	12TK1	17:30:00
SELASA 07:00:00	TK32104	Kalkulus II	Triandah Widiyanti Utami, MT.	4019	12TKA	08:15:00
	TK32106	Agama dan Etika I	Prof. Dr. H. Suljani Saiful, M.	4017	12TK1	08:30:00
	TK34205	Bahasa Rabbani	Uti Nurhayati, S. Pd., M. Pd.	4007	11TK2	11:30:00

Gambar 5.11 Tampilan .xls

Berikut merupakan hasil penyimpanan berformat microsoft word setelah melalui tahap *Cristal Report*.

	Kode Mata Kuliah	Nama Mata Kuliah	Nama Dosen	Ruang	Kelas	EndTime
SENIN 09:15	TK36201	Teknik Menulis Inisiah	Wandi Zaman, M.Si.	4021	10TK1	10:45
	TK36205	Antarmuka Komputer	I. Syahrial, MT.	4017	10TK2	13:00
	TK34207	Komunikasi Data	Suamni Indrani, MT.	4019	11TKA	15:15
	TK32103	Praktik II	Ungap M. Inahaj, M.Si.	4017	10TK1	17:30
SELASA 07:00	TK32104	Kalkulus II	Triandah Widiyanti Utami, MT.	4019	12TKA	08:15

Gambar 5.12 Tampilan .doc

Berikut merupakan hasil penyimpanan berformat portable document format setelah melalui tahap *Cristal Report*.

	Kode Mata Kuliah	Nama Mata Kuliah	Nama Dosen	Ruang	Kelas	EndTime
SENIN 09:15:	TK36201	Teknik Menulis Inisiah	Wandi Zaman, M.Si.	4021	10TK1	10:45:
	TK36205	Antarmuka Komputer	I. Syahrial, MT.	4017	10TK2	13:00:
	TK34207	Komunikasi Data	Suamni Indrani L., MT.	4019	11TKA	15:15:
	TK32103	Praktik II	Ungap M. Inahaj, M.Si.	4017	12TK1	17:30:
SELASA 07:00:	TK32104	Kalkulus II	Triandah Widiyanti Utami, MT.	4019	12TKA	08:15:
	TK32106	Agama dan Etika I	Prof. Dr. H. Suljani Saiful, M.	4017	12TK1	08:30:

Gambar 5.13 Tampilan .pdf

6 Penutup

6.1 Kesimpulan

Perangkat lunak Penjadwalan Kuliah dengan menerapkan metoda *Ant Colony* dapat melakukan pengaturan penjadwalan kuliah secara otomatis dan relatif cepat yang di-*create* berdasarkan kesediaan waktu dari dosen dan kesediaan ruangan. Disamping itu perangkat lunak ini juga mampu memberikan informasi jadwal mata kuliah yang bentrok baik dari sisi dosen maupun ruangan kelas. Kemudian perangkat lunak ini akan memperbaikinya secara otomatis. Aplikasi ini dapat diimplementasikan di Jurusan Teknik Komputer FTIK Unikom, namun ada bagian yang belum rampung dibangun yaitu otomatisasi Tabel Hari. Tabel Hari yang dihasilkan oleh perangkat lunak ini dalam format .xls (file excel) belum sepenuhnya otomatis dikarenakan keterbatasan waktu dalam mendesain Tabel Hari seperti yang diinginkan jurusan. Tetapi secara keseluruhan perangkat ini dapat digunakan di Jurusan Teknik Komputer. Diharapkan peneliti berikutnya dapat merampungkan bagian-bagian yang belum sempurna sehingga aplikasi ini sepenuhnya menjadi suatu *software* aplikasi yang berkualitas.

6.2 Saran

Berikut merupakan saran berdasarkan perangkat lunak yang telah dibangun:

1. Diharapkan dalam pengembangan ditambahkan pengaturan kapasitas ruangan jadi pengalokasian jumlah mahasiswanya lebih jelas.
 2. Untuk mendapatkan hasil penjadwalan kuliah yang lebih cepat dan berkualitas hendaknya dilakukan pembelajaran teknik-teknik algoritma semut yang lebih spesifik dan mendalam serta mengkombinasikan dengan algoritma lain.
- [8]. Munawar. *Pemodelan Visual Dengan UML*, Graha ilmu, Yogyakarta, 2005.

7 Daftar Pustaka

- [1]. Dorigo, M and Thomas Stutzle, *Ant colony Optimization*, MIT Press, 2004.
- [2]. Dorigo, M dan Gambardella, L.M., *Ant colony System: A Cooperative Learning Approach to the Traveling*, 1996.
- [3]. M. Dorigo, *Optimization, Learning and Natural Algorithms*, PhD thesis, Milan, Italy: politecnico di Milano, 1992.
- [4]. Abdullah, Salwani. *Heuristic Approaches for University Timetabling Problems*. Doctoral Dissertation. University of Nottingham, England, 2006.
- [5]. Yustina, Eva. *desain dan implementasi jadwal kuliah dengan menggunakan algoritma semut berbasis web*, UIN, Malang, 2008.
- [6]. Hidayatullah, Priyanto. *Visual basic .NET membuat aplikasi database dan program kreatif*
- [7]. Kurniawan, Erick dan Rully Yulian MF. *Migrasi Visual Basic 6 ke Visual Basic .NET*, 2009.

