Penjadwalan Perkuliahan dengan Metode Metaheuristic Ant Colony Optimization Studi Kasus Politeknik Negeri Bali

Komang Ayu Triana Indah, Putu Gede Sukarata
Jurusan Teknik Elektro, Politeknik Negeri Bali
triana_indah@pnb.ac.id

Abstrak: Masalah penjadwalan kuliah merupakan masalah yang sangat kompleks, dimana inti dari penjadwalan tersebut adalah bagaimana menjadwalkan beberapa komponen yang terdiri dari mahasiswa, dosen, ruang, waktu, dan matakuliah dengan memperhatikan sejumlah batasan dan syarat (*constraint*) tertentu. Tujuan yang ingin dicapai adalah menerapkan metode Metaheuristik Ant Colony Optimization (ACO) untuk optimalisasi sistem penjadwalan perkuliahan. Setiap langkah yang dilakukan oleh ACO menggunakan algoritma yang diadaptasi dari prilaku semut untuk menyelesaikan permasalahan kombinatorial. Sistem ini dibuat dengan menggunakan pemrograman Microsoft Visual Basic dengan memasukkan beberapa parameter yaitu Data Dosen, Mata Kuliah, Ruangan, dan beberapa variabel dari masing-masing parameter yang kemudian diproses sehingga menghasilkan penjadwalan perkuliahan. Adapun hasil yang diperoleh dari implementasi ACO untuk memecahkan masalah penjadwalan kuliah di perguruan tinggi dilihat dari jumlah *constraint* yang terlanggar serta lamanya waktu yang diperlukan dari masing-masing iterasi pada tiap metode sampai mendapatkan jadwal kuliah sehingga memperoleh jadwal dengan tingkat kebenaran 89 persen yang merupakan perbandingan pelanggaran *constraint* terhadap jumlah keseluruhan *timeslot* yang digunakan untuk penjadwalan.

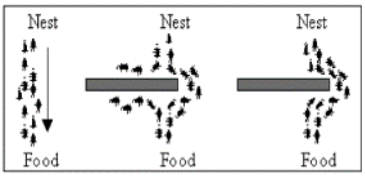
Kata kunci: *algoritma ACO, metaheuristic, optimisasi.*

Abstract: *Lecture scheduling problems are a very complex problem, where the core of the scheduling is how to schedule several components consisting of students, lecturers, space, time, and subject matter with due regard to certain constraint and constraint. The goal to be achieved is to apply the Ant Colony Optimization (ACO) Metaheuristic method to optimize the lecture scheduling system. Every step taken by ACO uses an algorithm adapted from ant behavior to solve combinatorial problems. This system is created using Microsoft Visual Basic programming by entering several parameters, namely Lecturer, Course, Room, and several variables from each parameter which are then processed to produce lecture scheduling. The results obtained from the ACO implementation to solve college scheduling problems are seen from the number of constrainted constraint and the length of time required from each iteration in each method to get a class schedule so that they get a schedule with a 89 percent truth level which is a comparison of constraint violations to the total number of timeslot used for scheduling.*

Keywords: *ACO algorithm, metaheuristic, optimization.*

I. PENDAHULUAN

Heuristik berasal dari kata Yunani "*heuriskein*" yang berarti seni untuk menemukan strategi dalam menyelesaikan persoalan sedangkan meta berarti metodologi tingkat tinggi atau lanjut [1]. Pada ilmu komputer, metode heuristik merupakan suatu teknik untuk penyelesaian permasalahan yang tidak menekankan pada pembuktian apakah solusi yang didapatkan adalah benar karena pembuktian apakah suatu solusi benar merupakan fokus dari metode penyelesaian analitik. Metode Heuristik merupakan suatu metode penyelesaian yang menggunakan konsep pendekatan [2]. ACO adalah sistem agen yang mensimulasikan tingkah laku natural dari kelompok semut (*ants*) yang mengandung mekanisma kerjasama dan adaptasi. Ide dasar dari proses ini dapat dilihat pada Gambar 1. Algoritma ini terinspirasi dari tingkah laku koloni semut dalam mencari makan, Di dunia nyata, semut (awalnya) berjalan secara acak, dan ketika menemukan makanan kembali ke koloni mereka sambil meletakkan pheromone jejak. Semut-semut tersebut meninggalkan zat (*pheromone*) di jalan yang mereka lalui.



Gambar 1. Ants Menemukan Jalur Terpendek

Jika semut lain menemukan jalur tersebut, mereka tidak cenderung untuk menjaga bepergian secara acak, tapi malah mengikuti jejak, kembali dan menguatkannya jika pada akhirnya mereka menemukan makanan. Gagasan awalnya berasal dari mengamati makanan eksploitasi sumber daya di antara semut, di mana semut secara individual memiliki kemampuan kognitif terbatas secara kolektif mampu

menemukan jalur terpendek antara sumber makanan dan sarang. Semut pertama menemukan sumber makanan (F), melalui cara apapun (a), kemudian kembali ke sarang (N), meninggalkan jejak *pheromone* (b). Semut tanpa pandang bulu mengikuti empat kemungkinan, tapi penguatan landasan membuatnya lebih menarik sebagai rute terpendek [3].

Masalah penjadwalan secara umum adalah aktifitas penugasan yang berhubungan dengan sejumlah kendala, sejumlah kejadian yang dapat terjadi pada suatu periode waktu dan tempat/lokasi tertentu sehingga fungsi objektif sedekat mungkin terpenuhi. Masalah ini muncul di berbagai bidang kegiatan maupun instansi seperti rumah sakit, universitas, penerbangan, pabrik, dan lain-lain. Desain model masalah penjadwalan bervariasi sesuai dengan kebutuhan serta keadaan di lapangan [4].

Penyampaian informasi pada lembaga akademik merupakan hal yang sangat penting terutama informasi yang berkaitan dengan kegiatan perkuliahan. Salah satunya adalah informasi jadwal kuliah. Penjadwalan kuliah merupakan pekerjaan yang tidak mudah, dimana inti dari masalah ini adalah bagaimana menjadwalkan berbagai komponen yang terdiri dari mahasiswa, dosen, ruang, dan waktu dengan memperhatikan sejumlah batasan dan syarat tertentu.

Terdapat beberapa permasalahan dalam penjadwalan mata kuliah di antaranya:

1. Pada bagian Jurusan / Program Studi, yaitu bagaimana mengidentifikasi Mata Kuliah yang ditawarkan sesuai kurikulum pada semester yang akan dibuat penjadwalan.
2. Bagaimana menyusun jadwal perkuliahan bagi mahasiswa maupun dosen meliputi penentuan mata kuliah, waktu, dan tempat perkuliahan serta penentuan dosen pengampu setiap mata kuliah.
3. Bagaimana melakukan penjadwalan kuliah yang dilakukan oleh jurusan yang berisi nama hari dan jam kuliah, mata kuliah, ruangan dan nama dosen secara sistematis sehingga tidak terjadi bentrok.
4. Menentukan syarat minimal 16 kali tatap muka antara dosen dan mahasiswa selama satu mata kuliah dalam setiap semester, atau minimal 75% perkuliahan sebelum mengikuti ujian.
5. Bagaimana menentukan penjadwalan perkuliahan sehingga beban sks setiap dosen minimal 12 sks per semester.

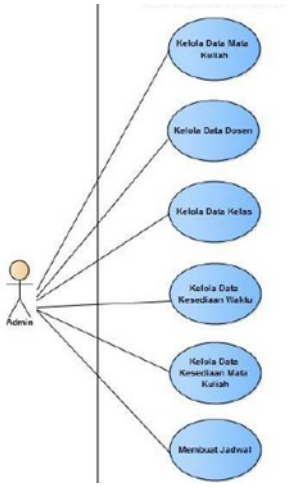
Syarat-syarat dalam penjadwalan kuliah terbagi dalam dua kelompok sesuai dengan tingkat kewajiban syarat tersebut terpenuhi, yaitu *hard constraint* (harus terpenuhi) dan *soft constraint* (diupayakan untuk terpenuhi). Sebuah solusi hanya dapat dikatakan sah dan valid apabila dalam solusi tersebut sama sekali tidak ada *hard constraint* yang terlanggar. Berbeda dengan hard constraint, kendala yang termasuk dalam *soft constraint* adalah kendala yang tidak selalu dapat terpenuhi dalam proses pembentukan jadwal, akan tetapi meskipun tidak harus terpenuhi, jadwal yang dihasilkan harus semaksimal mungkin berusaha memenuhi ketentuan *soft constraint* [5]. Masalah penjadwalan kuliah dapat diselesaikan dengan beberapa metode heuristik, seperti *tabu search*, *simulated annealing*, dan algoritma genetika [5]. Dalam penelitian ini, permasalahan penjadwalan kuliah akan diselesaikan dengan menggunakan algoritma algoritma *Ant Colony Optimation* (ACO) sehingga dengan digunakannya algoritma ini diharapkan akan diperoleh hasil penjadwalan yang lebih optimal. Pada penelitian ini dirancang sebuah aplikasi untuk mengatur jadwal kuliah secara otomatis agar menghasilkan keluaran berupa jadwal kuliah, jadwal ini masih memungkinkan untuk diubah secara manual sesuai dengan keinginan dari *user*. Untuk pembuatan aplikasi akan digunakan sebuah algoritma metaheuristic yaitu *Ant Colony* atau yang lebih dikenal dengan nama algoritma semut untuk mendapatkan hasil yang lebih optimal.

II. METODE PENELITIAN

Pada perancangan awal ditentukan terlebih dahulu diagram suatu penjadwalan untuk memudahkan dalam penyusunan *flowchart* algoritma programnya, di antaranya:

A. Diagram Use Case

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



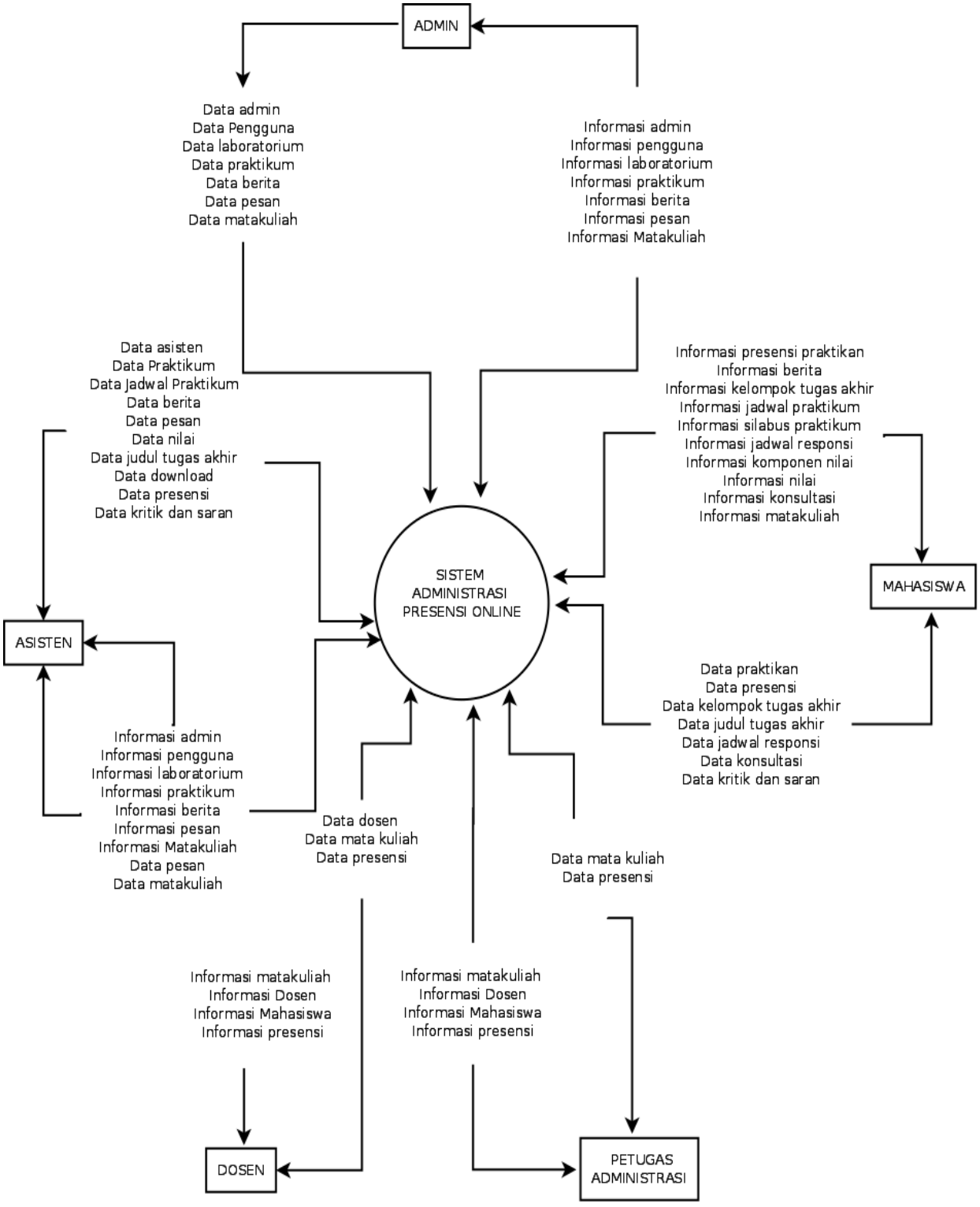
Gambar 2. Diagram *use case* perkuliahan [6].

B. Diagram Aktivitas

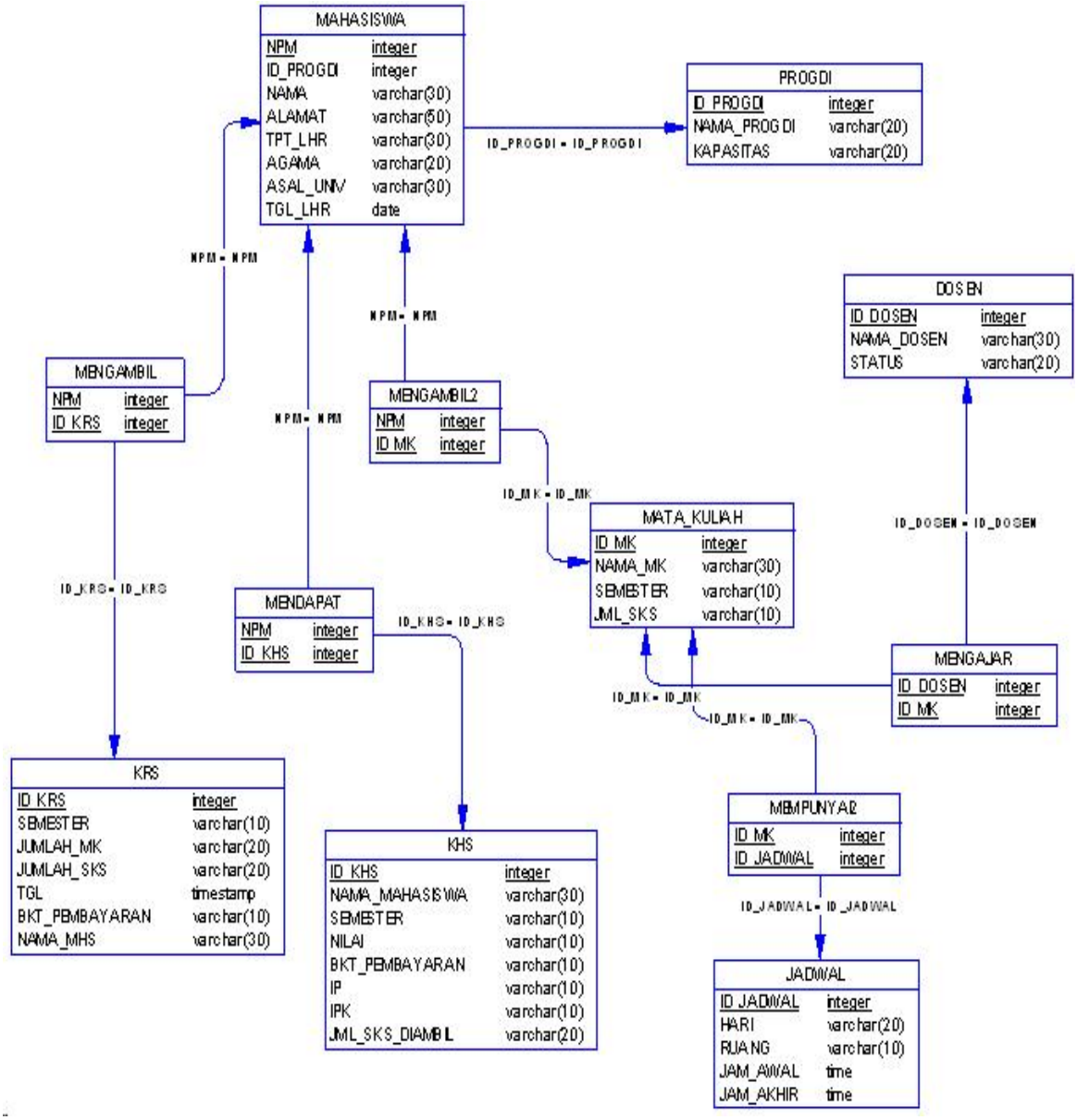
Diagram aktivitas adalah sebuah teknik penjelasan diagram yang bebas menunjukkan aliran aktivitas dan kegiatan langkah demi langkah. Diagram pada Gambar 3 digunakan untuk menjelaskan aliran bisnis dan operasi sebuah komponen dari sistem [6].

C. Diagram Struktural

Diagram pada Gambar 4 diwakilkan oleh diagram kelas yang merupakan sebuah tampilan statis struktur program. Diagram kelas menggambarkan struktur dengan memperlihatkan kelas dari sistem, baik atribut maupun relasi dengan kelas lain.



Gambar 3. Diagram aktivitas administrasi perkuliahan [7].



Gambar 4. Diagram struktural dari proses administrasi perkuliahan [7].

D. Penjadwalan dengan Algoritma ACO

Algoritma ACO didasarkan pada ide-ide berikut: Masing-masing jalur (*path*) yang diikuti oleh semut diasosiasikan sebagai kandidat solusi. Ketika seekor semut melalui sebuah jalur, sejumlah dijatuhkan pada jalur sesuai dengan kualitas *term* (hubungan) kandidat solusi untuk "*target problem*". Kaidah yang terbaik yang dibangun oleh seluruh semut dianggap sebagai kaidah yang dicari. Kaidah yang lain dibuang. Hal ini melengkapi sebuah iterasi dari sistem itu. Algoritma ini menggunakan kontruksi kaidah dan peningkatan pheromone. Misalkan *termijn* kondisi kaidah (*rule*) dalam format $A_i(t) = V_{ij}$, di mana A_i adalah atribut ke-*i* dan V_{ij} adalah nilai ke-*j* dari domain Ai. Probabilitas *term ij* terpilih untuk ditambahkan ke *current partial rule* diberikan oleh persamaan berikut [8].

$$P_{ij}(t) = \frac{\tau_{ij}(t).\eta_{ij}(t)}{\sum_i^a \sum_i^{b_i} \tau_{ij}(t).\eta_{ij}(t), \forall i \in I} \qquad (1)$$

dengan

$\eta_{ij}(t)$ adalah nilai dari *problem-dependent heuristic function* untuk *termij*;

$\tau_{ij}(t)$ adalah jumlah *pheromone* yang tersedia saat itu (*at time* t) dalam posisi i,j dari *trail* yang diikuti oleh semut;

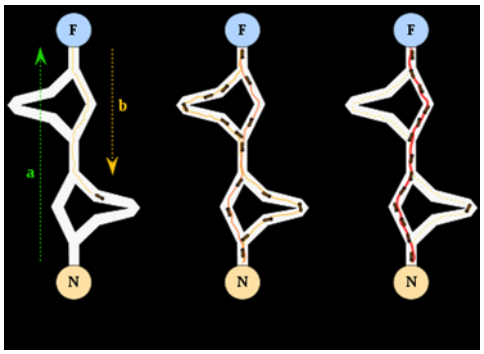
a adalah jumlah total atribut;

b_i adalah jumlah total nilai pada domain atribut *I*;

I adalah atribut yang belum digunakan oleh semut.

ACO itu sendiri terinspirasi oleh koloni-koloni semut dalam mencari makan. Semut-semut tersebut meninggalkan zat (*pheromone*) di jalan yang mereka lalui. Algoritma ACO ini merupakan algoritma pencarian berdasarkan probabilistik, di mana

probabilistik yang digunakan merupakan probabilistik dengan bobot sehingga butir pencarian dengan bobot yang lebih besar akan berakibat memiliki kemungkinan terpilih yang lebih besar pula [9] dengan asumsi yang terlihat pada Gambar 5 berikut.



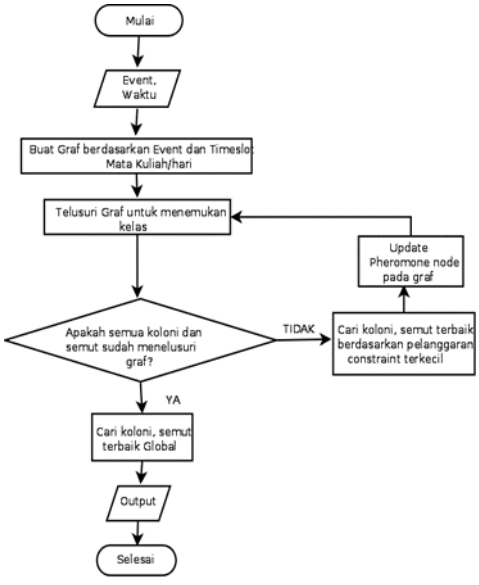
Gambar 5. Perilaku nyata semut mencari makanan [4]

Dari Gambar 5, maka *pheromone* ACB > *pheromone* ADB.

Adapun algoritma umum ACO adalah sebagai berikut [9]:

Masukan data permasalahan
while not selesai do
bangkitkan semut
bangkitkan solusi
update pheromone
hancurkan semut
end while
keluarkan solusi

Diagram alir algoritma ACO dapat dilihat pada Gambar 6.



Gambar 6. Diagram alir algoritma ACO [9].

Perhitungan menggunakan algoritma ACO yaitu dengan asumsi parameter yang digunakan pada penjadwalan ini adalah [8]:

1. Hari yang digunakan adalah lima hari dan masing-masing hari terdiri atas 15 *timeslot*, dimana 1 *timeslot* = 45 menit, sehingga total *timeslot* selama 1 minggu yaitu: 15 × 5 = 75 *timeslot*.
2. Mata kuliah yang dijadwalkan sebanyak 28 dan ruangan sebanyak 16 kelas.
3. *Node* diasumsikan ruangan kelas yang akan digunakan untuk menjadwalkan mata kuliah berdasarkan *timeslot* yang sudah ditentukan.
4. Pada penelusuran graf bertujuan untuk menentukan matakuliah yang sesuai dengan kelas agar tidak bentrok. Jumlah ruangan yang dipergunakan yaitu 16 kelas, jumlah hari perkuliahan yaitu 5 hari, dan *timeslot* sebanyak 75.
5. Hari yang digunakan untuk penjadwalan adalah Senin s/d Jumat dengan *timeslot*/hari = 15, satu *timeslot* = 45 menit, sehingga 45 × 15 = 7 jam 15 menit.

Diasumsikan penjadwalan disusun dari pukul 08.00 hingga pukul 15.15.

E. Data Masukan

Untuk proses masukan data, terdiri dari proses pemasukan data yang berupa data dosen, data mata kuliah, data ruang, dan data waktu kuliah dan ujian. Data dosen diisi oleh dosen dengan mengisi formulir data mengajar, sedangkan ruangannya ditentukan oleh Sekretariat Program. Data ini disimpan ke *database*. Proses pemasukan data matakuliah yang ditawarkan tiap semesternya, termasuk penentuan jumlah kelas per-matakuliah yang ditawarkan ini disesuaikan dengan kesanggupan dosen mengajar.

F. Data Keluaran

Data keluaran yang dihasilkan berupa laporan (*print out*) jadwal kuliah, yang berisi data mata kuliah perjurusan yang diadakan tiap semester. Laporan ini selanjutnya digunakan mahasiswa untuk *key-in* kuliah. Selain itu juga terdapat laporan (*print out*) jadwal dosen, merupakan laporan mengajar dosen yang diserahkan kepada dosen yang bersangkutan agar mengetahui jadwal mengajarnya.

G. Instrumen Penelitian

Instrumen yang dipergunakan dalam penelitian ini yaitu dengan metode ACO melalui program simulasi dengan Bahasa pemrograman yang akan digunakan dalam program simulasi yaitu Microsoft Visual Basic dengan metode algoritma *ACO* dan secara umum sebagai pencari solusi optimal menggunakan Komputer Intel Core i5-2270 processor untuk pemrosesan data.

H. Metode dan Teknik Pengumpulan Data

Pada penelitian ini, menggunakan metode dan teknik pengumpulan literatur yang berhubungan dengan perencanaan dan analisa metode ACO, melakukan observasi terhadap objek penelitian di antaranya bidang administrasi pada Politeknik Negeri Bali, Jurusan Teknik Elektro, perancangan program simulasi berdasarkan aplikasi yang dilakukan dengan metode optimasinya serta metode pengujian dan analisa system untuk mengetahui unjuk kerja algoritma ACO.

I. Metode dan Teknik Analisis Data

Metode dan teknik analisis data dilakukan dengan pengujian sistem menggunakan program simulasi dengan bahasa pemrograman Microsoft Visual Basic. Pengujian tersebut dilakukan untuk mengetahui optimasi penjadwalan dengan metode ACO untuk membedakan efektivitas penggunaan dalam kasus penjadwalan perkuliahan. Pengujian ini dilakukan untuk mata kuliah semester ganjil dan semester genap yang ada di Jurusan Teknik Elektro Politeknik Negeri Bali dalam perbandingan dengan waktu. Untuk kasus penjadwalan termasuk ke dalam kasus kompleks *metaheuristic* yang tidak mungkin dapat ditangani oleh algoritma standard biasa, dilakukan penujian algoritma. Penjadwalan memerlukan faktor *guessing* dalam penyelesaian masalahnya, kemudian masuk ke algoritma untuk diolah dan diperoleh hasil optimal. Untuk pengujian perangkat lunak dan struktur data**,** algoritma semut memerlukan sebuah kondisi awal ketika pembangkitan solusi hingga proses *update pheromone*. Semua tipe data ini memerlukan *range* yang standard seperti pada program umumnya. Untuk itulah dipilih tipe data integer (di dalam Net dikenal sebagai Int32) dengan ukuran 4 bytes dengan *range* -2,147,483,648 .. 2,147,483,647, untuk tipe angka desimal akan dipilih *double* sebagai tipe datanya dengan ukuran 8 *bytes* untuk presisi data 2.79769313486232e30 sampai dengan 1.79769313486232e308 [10].

III. HASIL DAN PEMBAHASAN

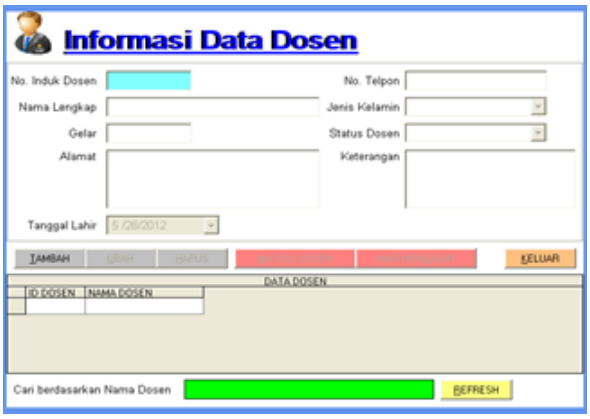
A. Rancangan Form Menu Utama

Rancangan antarmuka menu utama pada Gambar 7 berfungsi sebagai form awal ketika menjalankan program.

Gambar 7. Antarmuka menu utama.

B. Rancangan Form Data Dosen

Rancangan antar muka informasi dosen pada Gambar 8 berfungsi untuk mengisi data dosen yang akan mengajar pada setiap semester.



Gambar 8. Antarmuka informasi data dosen.

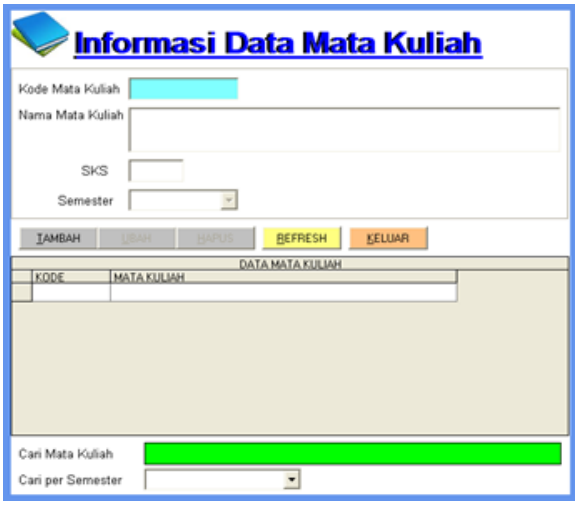
C. Rancangan Form Data Matakuliah

Rancangan antar muka matakuliah Gambar 9 berfungsi untuk mengisi data matakuliah yang akan diajarkan pada setiap semester.

Gambar 9. Antarmuka informasi data ruangan.

D. Rancangan Form Data Matakuliah

Rancangan antar muka matakuliah seperti pada Gambar 10 berfungsi untuk mengisi data matakuliah yang akan diajarkan pada setiap semesternya.



Gambar 10. Antarmuka informasi data matakuliah.

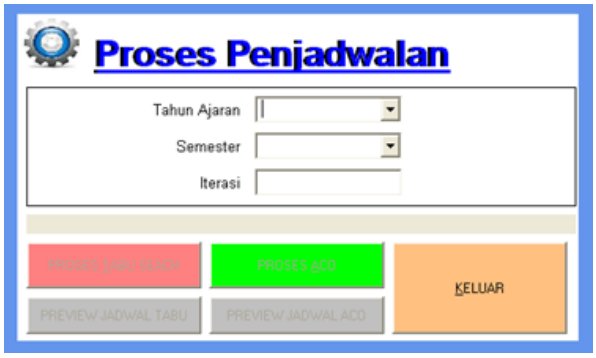
E. Rancangan Form Data Pencocokan

Rancangan antar muka pencocokan pada Gambar 11 berfungsi untuk mencocokan data yang telah diinput sebelumnya.

Gambar 11. Antarmuka informasi data pencocokan.

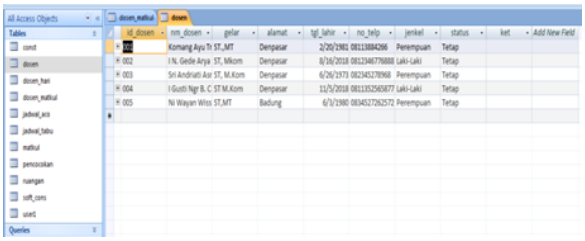
F. Rancangan Form Proses Penjadwalan

Rancangan antar muka pencocokan pada Gambar 12 berfungsi untuk menampilkan hasil penjadwalan kuliah yang telah dilakukan.



Gambar 12. Antarmuka proses penjadwalan.

Perangkat lunak (*software*) metode *ant colony optimization* pada penjadwalan kuliah diimplementasikan dengan Microsoft Visual Basic 6.0. Implementasi sistem merupakan tahap dimana sistem siap untuk dioperasikan pada keadaan yang sebenarnya, sehingga diketahui bahwa sistem yang dibuat telah dapat menghasilkan tujuan yang diinginkan. Sebelum program diterapkan dan diimplementasikan, maka program harus bebas dari kesalahan (*error free*). Gambar 13 merupakan hasil implemantasi basis data dari sistem penjadwalan perkuliahan yang dibuat.

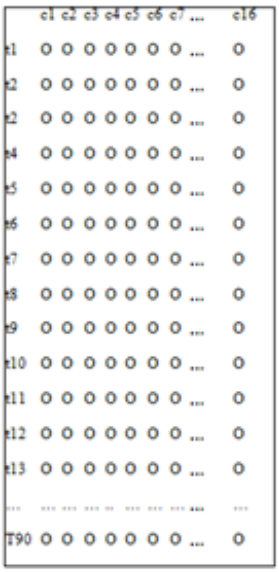


Gambar 13. Basis data dari implementasi program.

Pada Gambar 13 ditunjukkan terdapat 6 (enam) tabel yang saling berelasi. Tabel matakuliah (kd_matakuliah) berelasi *one to many* dengan tabel pencocokan (kd_matakuliah). Tabel ruangan (id_ruangan) berelasi *one to many* dengan tabel pencocokan (id_ruangan). Tabel dosen (id_dosen) berelasi *one to many* dengan tabel pencocokan (id_dosen). Tabel jadwal berelasi *one to one* dengan tabel pencocokan. Kesalahan program yang mungkin terjadi antara lain kesalahan penulisan bahasa, kesalahan sewaktu proses atau kesalahan logika. Setelah program bebas dari kesalahan, program diuji coba dengan memasukkan data untuk diolah.

Perhitungan dengan menggunakan algoritma ACO dengan asumsi parameter yang digunakan pada penjadwalan ini adalah sebagai berikut:

1. Hari yang digunakan adalah lima hari dan masing-masing hari terdiri atas 15 *timeslot*, dengan 1 *timeslot* = 45 menit, sehingga total *timeslot* selama 1 minggu yaitu: 15 × 7 = 75 timeslot.
2. Mata kuliah yang dijadwalkan sebanyak 28 dan ruangan sebanyak 16 kelas.
3. *Node* diasumsikan ruangan kelas yang akan digunakan untuk menjadwalkan mata kuliah berdasarkan *timeslot* yang sudah ditentukan dan diaplikasikan pada *node* graf sebagai berikut:



Gambar 14. *Node* untuk kelas = 16, timeslot = 75.

Penelusuran graf bertujuan untuk menentukan matakuliah yang sesuai dengan kelas agar tidak bentrok. Jumlah ruangan yang dipergunakan yaitu 16 kelas, jumlah hari perkuliahan yaitu 5 hari, dan *timeslot* 75. Dari hasil program aplikasi didapatkan seperti pada Gambar 15 dan 16 yang merupakan hasil proses penjadwalan dengan menggunakan metode ACO. Penjadwalan efektif pada semester genap yaitu 9 penjadwalan dibandingkan pada semester ganjil sejumlah 8 penjadwalan.

G. Analisis Nilai Constraint dan Durasi Waktu/Timeslot

Total keseluruhan *timeslot* berjumlah 2789, sehingga untuk memudahkan analisa digunakan 4 *timeslot* dari keseluruhan penjadwalan untuk perhitungan dari komponen mata kuliah, dosen pengampu, ruangan, durasi jam, *constraint* yang dilanggar (*hard* dan *soft constraint*) sehingga dapat menentukan persentase tingkat kebenaran penjadwalan, jumlah pelanggaran serta waktu penelusuran/proses penjadwalan [11]. Gambar 16 merupakan sampel atau potongan output program dari total 2789 *timeslot*.

Bahwa iterasi terbaik adalah iterasi dari koloni 3, Kelas DH-203, nilai pelanggaran *constraint* = 0, *timeslot* = 032, dengan *timeslot* terkecil diasumsikan durasi waktu yang diaplikasikan paling pendek. Hasil pelanggaran *constraint* terkecil tersebut cenderung diperoleh bukan pada iterasi pertama, karena prinsip dari ACO, update *pheromone* terjadi pada iterasi-iterasi besar sehingga constraint bisa terdeteksi tidak pada saat iterasi pertama.

Total pelanggaran *constraint* dari penjadwalan ACO yaitu 249, dan total *timeslot* yang digunakan dalam keseluruhan penjadwalan adalah 2786 [12].

Sistem perangkat lunak yang akan dibuat adalah sebuah perangkat lunak yang berisi banyak data input, perhitungan menggunakan ACO, peraturan yang ada, pola desain dan pemakaian pada lingkup kerjanya, dan fleksibilitas. Karena itu dibutuhkan sebuah bahasa pemrograman yang terstruktur, handal, praktis, mudah digunakan dan juga mendukung batasan sistem operasi secara umum. *Visual Basic* adalah salah satu bahasa pemrograman yang memiliki semua persyaratan yang dibutuhkan dalam proses pembuatan perangkat lunak ini.

jadwal_aco

thn_ajaran	hari	kd_matakuliah	id_dosen	semester	kd_ruangan
2011/2012	1	TE056123	011669	Genap	203
2011/2012	2	TE055101	011669	Ganjil	101
2011/2012	5	TE661899	011669	Genap	101
2011/2012	3	TE 2211	048207	Genap	203
2011/2012	4	TE 2212	048207	Genap	203
2011/2012	1	TE053041	048207	Ganjil	204
2011/2012	5	TE 6423	048207	Genap	201
2011/2012	1	TE2209	055928	Genap	202
2011/2012	3	TE 55203	058241	Ganjil	203
2011/2012	1	TE 4232	058241	Genap	203
2011/2012	5	TE057567	104298	Ganjil	201
2011/2012	4	TE057326	104298	Ganjil	204
2011/2012	2	TE 6609	104298	Genap	201
2011/2012	2	TE6904	104298	Genap	204
2011/2012	5	TE 2213	125699	Genap	203
2011/2012	1	TE055023	130362	Ganjil	102
2011/2012	2	TE056327	130482	Ganjil	205
2011/2012	5	TE056329	130482	Genap	204
2011/2012	3	TE6528	130789	Genap	201
2011/2012	1	TE6510	130789	Genap	101
2011/2012	5	TE053067	130803	Ganjil	101
2011/2012	5	TE053987	130803	Ganjil	102
2011/2012	2	TE 4231	130803	Genap	202

Gambar 15. Hasil pemrograman.

penjadwalan_aco

thn_ajaran	hari	kd_matkuliah	nm_matakuliah	sks	id_dosen	nm_dosen	gelar	semester	kd_ruangan	jam_ke	cons	Timeslot
2017/2018	2	TE 05300	Pengantar Sistem Informasi	2	257768	Sri Andriati	ST.,MT.	Ganjil	202	5	0	025
2017/2018	2	TE55641	Bahasa Pemrograman	2	163774	Eddy Indrayana	ST.,MT.	Ganjil	202	7	0	027
2017/2018	2	TE 05100	(i) Aljabar Vektor Kompleks (A)	2	958235	Indah Ciptayani	SKom.,MT	Ganjil	204	1	0	021
2017/2018	2	TE 05531	Jaringan Komputer	2	132163	Made Ari Dwi Suta	ST.,M.Sc., PhD.	Ganjil	203	7	2	227
2017/2018	2	TE55311 2	(vii)Komputasi Cerdas Pada Sistem Tenaga	2	958233	IB Putra Manuaba	SKom.,MT	Ganjil	204	5	2	225
2017/2018	2	TE05101 1	Project Sistem Informasi	2	175619	Komang Arya Astawa	ST.MT.	Ganjil	205	1	2	221
2017/2018	2	TE05652 7	Sistem Informasi Manajemen	2	130482	Wiswani	ST.,MT	Ganjil	205	5	0	025
2017/2018	2	TE 05742	Bahasa Inggris	2	297253	IGst Nym Sukerti	SSos,MHum	Ganjil	205	7	1	127
2017/2018	3	TE05584 0	Pemrograman Berbasis Web	3	163774	Budi Sentana	ST.,MT.	Ganjil	101	1	3	331
2017/2018	5	TE05795 6	(vii) Pemrograman Java dan Teknologi Bergerak	3	257772	Eddy Indrayana	ST.,MT.	Ganjil	102	1	3	351
2017/2018	5	TE05756 7	Pemrograman Dasar	2	104298	Indah Ciptayanai	S.Kom.,MT	Ganjil	201	3	0	053
2017/2018	3	TE 55218	(v) Pemrograman Internet	2	257772	Manik Prihartini	ST.,MT.	Ganjil	101	5	1	135
2017/2018	3	TE05102 1	(i) Konsep Pemrograman Komputer (B)	3	257772	Candra Winerta	SKom., MKom.	Ganjil	102	3	2	233
2017/2018	3	TE55401	Instalasi Komputer	2	236127	Ari Dwi Suta	ST.,MT.	Ganjil	102	7	2	237
2017/2018	3	TE 55201	(v) Analisa Desain Sistem Informasi	2	132240	Ayu Harry	ST.,MT.	Ganjil	201	1	0	031
2017/2018	3	TE05531 4	Pancasiladan Kewarganegaraan	2	843101	Elina Rudiastuti	SH.,MHum	Ganjil	202	3	2	233
2017/2018	3	TE 56406	(v) Jaringan Telekomunikasi	3	132135	Gede Sukarata	ST.,MSc.	Ganjil	201	5	2	235

Gambar 16. Hasil pemrograman.

IV. KESIMPULAN

Hasil penelitian untuk menganalisa unjuk kerja metode *Ant Colony Optimization* dilihat dari jumlah *constraint* yang terlanggar serta lamanya waktu yang diperlukan dari masing-masing iterasi sampai mendapatkan jadwal kuliah, dimana hasilnya terdapat total pelanggaran *constraint* yaitu 249 dari 2786 *timeslot*, hasil ini dapat dilihat dari output program pada Gambar 16. Sedangkan pelanggaran constraint terkecil terdapat pada iterasi dari koloni 3, kelas DH—203, dengan pelanggaran *constraint*=0, pada *timeslot* 032, pelanggaran terkecil maksudnya tidak ada jadwal yang bentrok/*constraint* terlanggar dari komponen-komponen yang dijadwalkan melalui proses pencocokan pada program aplikasi.

Dari analisis unjuk kerja Metode ACO tersebut, yang diasumsikan untuk memperoleh jadwal dengan tingkat kebenaran kurang lebih sebesar 89%, diperoleh perbandingan total pelanggaran constraint terhadap jumlah *timeslot* yaitu 249 pelanggaran dari total 2789 timeslot. Selain itu pada ACO *variabel local solution* dilakukan secara probabilistik, sehingga sulit untuk memperoleh penjadwalan dengan tingkat kebenaran 100%.

Pada ACO, hasil yang baik cenderung diperoleh pada iterasi-iterasi besar, hal ini membuktikan bahwa *update* terhadap *pheromone* berpengaruh terhadap penelusuran semut dalam menghasilkan solusi dan proses ini dipengaruhi oleh jumlah komponen yang dijadwalkan. Semakin besar kasus, maka waktu yang diperlukan akan semakin lama pula.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada semua pihak di Politeknik Negeri Bali atas segala dukungan dan kerjasamanya untuk dapat membantu peneliti dalam menyelesaikan penelitian ini.

DAFTAR PUSTAKA

[1] I. Berlianty and A. Miftahol, *Teknik-teknik optimasi heuristik*. Yogyakarta: Graha Ilmu, 2010.

[2] M. Gendreau and J. Y. Potvin, *Handbook of metaheuristics*. New York: Springer Science+Business Media, 2010.

[3] K. F. Doerner, D. Merkle and T. Stuzle, "Special issue on ant colony optimization," *Swarm Intelligent*, vol. 3, no. 1, 2009.

[4] A. Jain, S. Jain and P. K. Chande, "Formulation of genetic algorithm to generate good quality course timetable," *International Journal of Innovation, Management and Technology*, vol. 1, no. 3 pp. 248-251, 2010.

[5] R. Arifudin, "Optimasi penjadwalan proyek dengan penyeimbangan biaya menggunakan kombinasi CPM dan algoritma genetika," *Jurnal Masyarakat Informatika*, vol. 2, no. 4, 2012.

[6] A. Fernandez, E. Handoyo, and M. Somantri, *Production and operations management* (4th ed.). Upper Saddle River, NJ: PrenticeHall, 2011.

[7] H. Babaei, J. Karimpour and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers & Industrial Engineering*, vol. 86, pp. 43-59, 2015.

[8] N. M. A. Al Salami, "Ant colony optimization algorithm," *UbiCC Journal*, vol. 4, no. 3, pp. 823-826, 2009.

[9] I. Berlianty and A. Miftahol, "Ant colony optimization," *Computational Intelligence Magazine*, IEEE, vol.1, no. 4, pp. 28-39, 2010.

[10] R. P. Badoni, D. K. Gupta, and P. Mishra, 2014. "A new hybrid algorithm for university course timetabling problem using events based on groupings of students," *Computers & Industrial Engineering*, vol. 78, pp. 12-25, 2014.

[11] P. H. Chen and H. H. Cheng, *IRT-based automated test assembly: A sampling and stratification perspective*. Texas: The University of Texas at Austin, August 2005.

[12] K. Socha and M. Dorigo, "Ant colony for continuous domains," *European Journal of Operation Research*, vol. 186, issue 3, pp. 1155-1173, 2008.