

PEMBUATAN APLIKASI PENJADWALAN MATA KULIAH MENGUNAKAN ALGORITMA ANT COLONY OPTIMIZATION (Studi Kasus: Program Studi Informatika Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta)

Andreas Silalahi¹, Mayanda Mega Santoni^{2*}, Anita Muliawati³
Program Studi Informatika, Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
Email: andreassilalahi@gmail.com¹, megasantoni@upnvj.ac.id²

Abstrak. Penjadwalan dibuat agar kegiatan perkuliahan dapat berjalan secara teratur. Salah satu penerapan penjadwalan ada pada penjadwalan mata kuliah. Pada beberapa instansi pendidikan seperti Program Studi Informatika Universitas Pembangunan Nasional Veteran Jakarta, penjadwalan mata kuliah masih dilakukan secara manual. Banyak data yang dapat mempengaruhi penjadwalan seperti dosen, mahasiswa, jumlah ruang, dan waktu yang tersedia membuat proses pembuatan jadwal menjadi tidak mudah dan membutuhkan banyak waktu. Masalah ini dapat dibantu dengan menggunakan algoritma yang baik. Pada jurnal ini, aplikasi penjadwalan mata kuliah yang menggunakan algoritma *Ant Colony Optimization* akan diuji untuk pembuatan jadwal mata kuliah. Pada riset yang telah dilakukan oleh peneliti lain ditemukan bahwa bahwa penjadwalan mata kuliah menggunakan *Ant Colony Optimization* dapat berjalan dengan efektif dan dapat membantu pengurangan masalah pada waktu proses penjadwalan.

Kata Kunci : Penjadwalan, Universitas, *Ant Colony Optimization*.

1 Pendahuluan

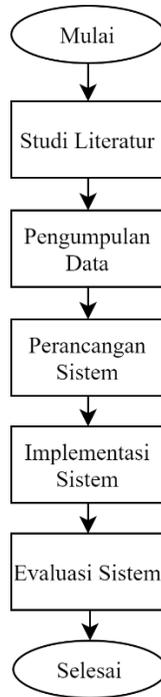
Penjadwalan mempunyai fungsi yang penting di sebuah universitas dan penjadwalan kegiatan perkuliahan. Penjadwalan juga termasuk hal yang rumit dapat dilihat oleh banyak faktor yang mempengaruhi seperti jadwal dosen, mata kuliah, dan ketersediaan ruang kelas sehingga cukup merepotkan apabila harus dikerjakan secara manual. Penempatan penjadwalan mata kuliah juga sering terdapat masalah seperti bentrokan jadwal dosen maupun ruang yang ditempati untuk perkuliahan sehingga mengakibatkan tidak efektifnya proses kegiatan dalam perkuliahan. Untuk mengatasi masalah ini dibutuhkan sebuah metode optimasi yang dapat diterapkan guna penyusunan jadwal perkuliahan yang disertai dengan berbagai batasan dan aturan yang dapat ditentukan oleh pembuat jadwal. Dari berbagai penelitian, algoritma genetika dan algoritma koloni semut adalah algoritma yang dirasa cocok untuk penelitian tentang metode optimasi penjadwalan mata kuliah. Permasalahan penjadwalan mata kuliah dipengaruhi oleh beberapa macam faktor dan batasan yang ditentukan.

Penelitian untuk permasalahan penjadwalan otomatis telah dilakukan oleh beberapa peneliti. Riset yang dilakukan oleh [1] menunjukkan bahwa algoritma penjadwalan akan menghasilkan hasil optimal apabila data penjadwalan sudah dialokasikan dengan tepat. Dengan menggunakan *Ant Colony Optimization*, penjadwalan sudah dapat dilakukan secara optimal karena setiap kegiatan perkuliahan dari setiap mata kuliah dapat dialokasikan tanpa ada masalah dan waktu yang diperlukan tidak banyak. Namun masih ada pelanggaran terhadap batasan yang ditentukan seperti penanganan kelas paralel. Riset yang dilakukan oleh [2] menunjukkan bahwa penjadwalan mata kuliah menggunakan *Ant Colony Optimization* dapat berjalan dengan efektif dan dapat membantu pengurangan masalah pada waktu proses penjadwalan. Riset yang dilakukan oleh [3] menunjukkan bahwa penjadwalan mata pelajaran yang menggunakan algoritma koloni semut dapat diterapkan untuk pemecahan masalah penjadwalan mata pelajaran di SMU, dengan hasil yang baik yaitu tidak terjadi pelanggaran antara komponen yang telah masuk ke penjadwalan. Oleh karena studi literatur yang sudah dilakukan pada tiga jurnal sebelum nya, penelitian ini akan menggunakan algoritma *ant colony*

optimization pada pembuatan aplikasi penjadwalan mata kuliah di Program Studi Informatika Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta.

2 Metodologi Penelitian

Pada tahap ini akan dijelaskan alur penelitian yang akan dilakukan. Susunan alurnya dapat dilihat pada Gambar. 1.



Gambar. 1. Kerangka Berpikir

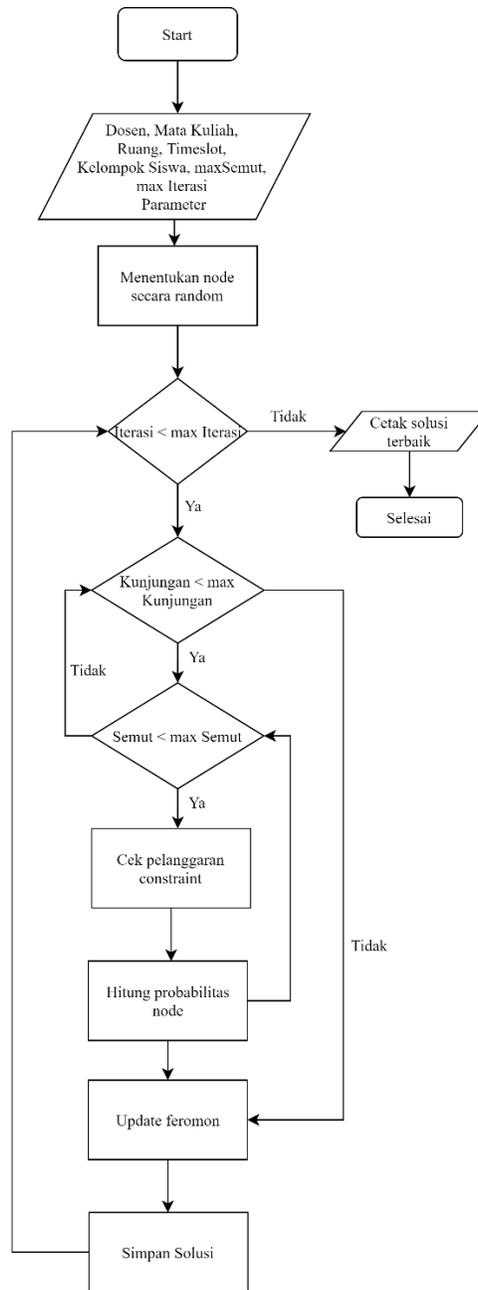
2.1 Studi Literatur

Dalam penelitian ini dilakukan riset berdasarkan jurnal yang berhubungan mengenai penjadwalan mata kuliah pada universitas dan penjadwalan terkait waktu dengan algoritma *Ant Colony Optimization*.

2.2 Pengumpulan Data

Data untuk keperluan penelitian didapatkan dari bagian pendidikan dan pengajaran Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta.

2.3 Perancangan Sistem



Gambar. 2. Perancangan Sistem

Pada **Gambar. 2** menjelaskan gambaran sistem yang akan dibuat bekerja. Proses dimulai dengan proses input data kegiatan mata kuliah (dosen, matakuliah, ruang, dan parameter untuk proses algoritma). Proses dilanjutkan dengan proses pengecekan apakah iterasi, kunjungan, dan semut yang sedang berjalan melebihi batas maksimal yang sudah di input sebelumnya. Setelah pengecekan, akan ada batasan atau *constraint* untuk penjadwalan. Batasan atau *constraint*

adalah peraturan yang diperoleh dari penjadwalan secara manual yang telah dilakukan sebelumnya oleh bagian akademik fakultas. Batasan atau *constraint* terbagi menjadi dua, yaitu *hard constraint* dan *soft constraint*[4]. Contoh *hard constraint* yang digunakan pada penelitian ini adalah:

- 1) Dalam satu hari perkuliahan, mahasiswa tidak diperbolehkan mengambil kelas perkuliahan lebih dari 2 kelas untuk mata kuliah yang memiliki 3 SKS dan 3 kelas untuk mata kuliah 2 SKS.
- 2) Dalam satu hari perkuliahan, dosen diperbolehkan mengajar maksimal 2 kelas untuk mata kuliah 3 SKS dan 3 kelas beban 2 SKS.

Untuk *soft constraint* yang digunakan adalah preferensi jadwal mengajar dari masing-masing dosen seperti kapan dosen tidak dapat hadir untuk mengajar di fakultas karena bentrokan jadwal di luar kampus. Langkah selanjutnya adalah penghitungan probabilitas dari setiap node yang dilewati dan pembaharuan feromon di jalur yang telah dilewati. Selanjutnya, solusi perjalanan akan disimpan dan pengecekan apabila $\max \text{Iterasi}$ sudah tidak melebihi iterasi maka akan dilanjutkan dengan ekstraksi solusi terbaik. Pada bagian ini algoritma *ant colony optimization* akan bekerja untuk membantu optimasi saat menghasilkan jadwal pengajaran dari data yang telah di *input* oleh pengguna sehingga membantu mengurangi adanya kesalahan dalam penjadwalan secara otomatis.

2.4 Implementasi Sistem

Penelitian ini akan menggunakan bahasa pemrograman Java untuk membantu implementasi sistem dalam penelitian ini.

2.4.1. Pengujian Data

Setiap kandidat solusi yang dihasilkan akan dievaluasi untuk menentukan solusi terbaik. Solusi paling baik dari seluruh semut dalam koloni akan disebut dengan $C_{\text{globabest}}$. Proses evaluasi akan dilakukan menggunakan perhitungan dari nilai antikalitas, yang nilainya adalah acuan daripada pada jumlah *event* yang tidak mendapat tempat di *timetable* atau yang disebut dengan kondisi *unplace* dan nilai pelanggaran untuk *soft constraint* yang ada. Kandidat solusi yang baik mempunyai nilai antikalitas yang kecil. Hal ini berarti, semakin kecil nilai antikalitas, semakin besar peluang suatu kandidat solusi yang dihasilkan untuk menjadi suatu $C_{\text{globabest}}$. Proses algoritma akan diulang sampai menemui solusi paling optimal yang diharapkan, yaitu saat *unplace* = 0 dan nilai pelanggaran *soft constraint* = 0 [5].

2.4.2. Pengujian dan Evaluasi Sistem

Pengujian sistem yang telah dirancang, berupa pengujian keseluruhan program dari input data hingga mendapatkan output yang dibutuhkan. Proses pengujian sistem pada fungsi internal berfokus pada algoritma sistem, memastikan bahwa semua tahapan sudah dilalui dengan baik. Fungsi eksternal adalah pengarahan uji coba untuk menemukan kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil yang baik serta sesuai dengan hasil yang dibutuhkan. Setelah pengujian akan dilakukan evaluasi hasil dari data yang sudah diimplementasi menggunakan sistem.

2.5 Evaluasi Sistem

Setelah mendapatkan hasil pengujian, pada bagian ini akan dianalisa data hasil uji untuk melakukan cek ulang apakah sistem sudah bekerja sesuai dengan yang diharapkan.

3 HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Pada bagian ini dijelaskan mengenai implementasi data yang telah dikumpulkan dan telah dirancang sedemikian rupa dan proses implementasi atau pengujian data ke aplikasi penjadwalan mata kuliah otomatis dengan algoritma Ant Colony Optimization. Aplikasi menggunakan bahasa pemrograman Java dan berjalan pada platform desktop menggunakan bantuan *software* NetBeans IDE dan DBMS PHPMyAdmin. Menggunakan aplikasi yang di uji coba maka setelah melalui proses penjadwalan dengan algoritma *Ant Colony Optimization* akan didapatkan *output* yaitu nilai pelanggaran *hard constraint*, pelanggaran *soft constraint*, dan jadwal perkuliahan. Input yang dapat dimasukkan pada aplikasi sebelum proses penjadwalan berjalan adalah jumlah semut dan iterasi yang diinginkan. Pada percobaan ini akan dimasukkan beberapa nilai berbeda untuk setiap parameter agar dapat ditemukan hasil penjadwalan yang optimal, dimana akan dicari nilai pelanggaran *soft constraint* terkecil. Nilai awal parameter yang dipakai dalam uji coba pada penjadwalan ini tersedia pada **Tabel 1**.

Tabel 1. Inisialisasi Parameter Sistem

| Parameter | Nilai |
|------------------|-------|
| Alfa | 3 |
| Beta | 2 |
| Rho | 0,01 |
| Q (Konstanta) | 2 |
| m (Jumlah semut) | 5 |
| Iterasi | 5 |

3.2 Pengujian Data

Untuk uji coba akan dilakukan penggantian masing-masing nilai parameter. Penggantian nilai parameter dilakukan agar didapatkan hasil terbaik dari percobaan yang nilai nya akan ditentukan secara random. Berikut adalah hasil terbaik yang didapatkan pada pengujian

Tabel 2. Percobaan 1 dengan nilai alpha 3

| No. | Semut Terbaik | <i>Soft Constraint</i> | <i>Hard Constraint</i> |
|-----|---------------|------------------------|------------------------|
| 1. | 4, Iterasi 1 | 5 | 0 |
| 2. | 4, Iterasi 3 | 7 | 0 |
| 3. | 3, Iterasi 5 | 8 | 0 |
| | Total | 20 | 0 |

Tabel 3. Percobaan 3 dengan nilai alpha 7

| No. | Semut Terbaik | Soft Constraint | Hard Constraint |
|--------------|---------------|-----------------|-----------------|
| 1. | 4, Iterasi 3 | 6 | 0 |
| Total | | 6 | 0 |

Tabel 4. Percobaan 4 dengan nilai beta 4

| No. | Semut Terbaik | Soft Constraint | Hard Constraint |
|--------------|---------------|-----------------|-----------------|
| 1. | 2, Iterasi 2 | 4 | 0 |
| Total | | 4 | 0 |

Tabel 5. Percobaan 8 dengan nilai rho 0,50

| No. | Semut Terbaik | Soft Constraint | Hard Constraint |
|--------------|---------------|-----------------|-----------------|
| 1. | 5, Iterasi 1 | 5 | 0 |
| 2. | 4, Iterasi 2 | 7 | 0 |
| Total | | 12 | 0 |

Tabel 6. Percobaan 11 dengan nilai Q 5

| No. | Semut Terbaik | Soft Constraint | Hard Constraint |
|--------------|---------------|-----------------|-----------------|
| 1. | 2, Iterasi 1 | 6 | 0 |
| 2. | 4, Iterasi 3 | 5 | 0 |
| Total | | 11 | 0 |

Tabel 7. Percobaan 13 dengan nilai semut 10

| No. | Semut Terbaik | Soft Constraint | Hard Constraint |
|--------------|---------------|-----------------|-----------------|
| 1. | 7, Iterasi 4 | 5 | 0 |
| Total | | 5 | 0 |

Tabel 8. Percobaan 16 dengan nilai iterasi 7

| No. | Semut Terbaik | Soft Constraint | Hard Constraint |
|--------------|---------------|-----------------|-----------------|
| 1. | 3, Iterasi 3 | 9 | 0 |
| Total | | 9 | 0 |

Percobaan pertama pada Tabel. 2 dilakukan dengan menggunakan parameter *default*. Pada uji data ini akan dilihat semut semu beberapa yang menghasilkan nilai pelanggaran *soft constraint* paling kecil. Contoh pada percobaan pertama menghasilkan 3 semut terbaik. Contoh pertama adalah semut ke 4 pada iterasi pertama dengan jumlah 5 pelanggaran *soft constraint* dan total dari ketiga semut adalah 20 pelanggaran *soft constraint*. Tabel selanjutnya adalah hasil terbaik dari penggantian nilai setiap parameter. Pada percobaan dalam **Tabel. 3** menunjukkan bahwa hasil terbaik didapat dengan mengganti nilai parameter alpha menjadi 7 dengan 6 pelanggaran *soft constraint*. Pada percobaan dalam **Tabel. 4** menunjukkan bahwa hasil terbaik didapat dengan mengganti nilai parameter beta menjadi 4 dengan 4 pelanggaran *soft constraint*. Pada percobaan dalam **Tabel. 5** menunjukkan bahwa hasil terbaik didapat dengan mengganti nilai parameter rho menjadi 0,50 dengan total 12 pelanggaran *soft constraint*. Pada percobaan dalam **Tabel. 6** menunjukkan bahwa hasil terbaik didapat dengan mengganti nilai parameter Q menjadi 5 dengan total 11 pelanggaran *soft constraint*. Pada percobaan dalam **Tabel. 7** menunjukkan bahwa hasil terbaik didapat dengan mengganti nilai parameter semut menjadi 10 dengan 5 pelanggaran *soft constraint*. Pada percobaan dalam **Tabel. 8** menunjukkan bahwa hasil terbaik didapat dengan mengganti nilai parameter iterasi menjadi 7 dengan total 9 pelanggaran *soft constraint*. Setelah proses data dengan masing-masing nilai parameter yang diubah serta data mata kuliah sebanyak 112 jadwal yang akan dijadwalkan dengan sistem, didapatkan hasil yaitu:

- 1) Hasil terbaik untuk percobaan yang diukur dari nilai pelanggaran *soft constraint* terkecil ada pada **Tabel. 2** yaitu percobaan keempat dengan pengubahan nilai beta = 4. Nilai pelanggaran *soft constraint* setelah dilakukan penjadwalan adalah 4 dengan semut terbaik yang didapatkan adalah semut kedua pada iterasi kedua serta pelanggaran *hard constraint* adalah 0.
- 2) Berikut adalah hasil terbaik untuk nilai dari setiap parameter pada uji coba yang telah dilakukan:
 - a) Alpha = 7
 - b) Beta = 4
 - c) Rho = 0,50
 - d) Q = 5
 - e) Semut = 10
 - f) Iterasi = 7
- 3) Pelanggaran *hard constraint* sudah mencapai 0 pada setiap generasi dimana nilai ini sudah sangat baik karena *hard constraint* bersifat tidak dapat dilanggar.
- 4) *Unplace* atau kondisi dimana permintaan jadwal tidak masuk ke dalam timetable penjadwalan adalah 0. Artinya, semua jadwal dapat dijadwalkan dengan baik meskipun masih ada beberapa pelanggaran pada *soft constraint*.

4 KESIMPULAN

Dari observasi hasil pengimplementasian data penjadwalan mata kuliah secara manual ke dalam sistem penjadwalan otomatis menggunakan algoritma Ant Colony Optimization, dapat diambil beberapa kesimpulan yaitu:

1. Algoritma Ant Colony Optimization dapat digunakan untuk pembuatan jadwal mata kuliah pada Prodi Informatika Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta.
2. Dari 18 percobaan yang dilakukan dengan nilai parameter yang diubah, algoritma berhasil menghindari pelanggaran *hard constraint* dan kondisi *unplace* dimana semua mata kuliah yang direncanakan dapat masuk ke dalam penjadwalan otomatis.
3. Terdapat pelanggaran *soft constraint* pada setiap percobaan dengan nilai parameter yang berbeda.
4. Waktu untuk penjadwalan otomatis akan semakin lama apabila nilai semut dan iterasi dinaikkan.

5 REFERENSI

- [1] T. Simanjuntak, E. A. Simatupang, B. Sibarani, N. K. A. Ziliwu, P. I. Del, and J. Sisingamangaraja, "UNIVERSITY COURSE TIMETABLING USING ANT COLONY OPTIMIZATION," pp. 1–13, 2012.
- [2] M. Mazlan, M. Makhtar, A. F. K. A. Khairi, M. A. Mohamed, and M. N. A. Rahman, "Ant colony optimisation for solving university course timetabling problems," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 139–141, 2018, doi: 10.14419/ijet.v7i2.15.11371.
- [3] D. R. Anamisa, S. Kom, M. Kom, D. Jurusan, T. Multimedia, and D. J. Teknik, "Implementasi Alokasi Jadwal Mata Pelajaran SMU menggunakan Algoritma Semut," *J. Ilm. NERO*, vol. 1, no. 1, pp. 33–41, 2014.
- [4] R. Sidik, M. Fitriawati, S. Mauluddin, and A. Nursikuwagus, "Model Penerapan Algoritma Ant Colony Optimization (Aco) Untuk Optimasi Sistem Informasi Penjadwalan Kuliah," *J. Teknol. dan Inf.*, vol. 8, no. 2, 2018, doi: 10.34010/jati.v8i2.1257.
- [5] W. Aprilius, L. Augustino, and O. Yeremia M. H., "Implementasi Algoritma MAX-MIN Ant System pada Penjadwalan Mata Kuliah," *J. Ultim.*, vol. 5, no. 2, pp. 48–53, 2013, doi: 10.31937/ti.v5i2.320.