

Implementasi Algoritma *MAX-MIN Ant System* pada Penjadwalan Mata Kuliah

Studi Kasus: Program Studi Teknik Informatika, UMN

William Aprilius, Lorentzo Augustino, Ong Yeremia M. H.

Program Studi Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia
william.aprilius@yahoo.com, lorentjo@yahoo.co.id, xixoft@gmail.com

Diterima 18 Desember 2013

Disetujui 30 Desember 2013

Abstract—University Course Timetabling Problem is a problem faced by every university, one of which is Universitas Multimedia Nusantara. Timetabling process is done by allocating time and space so that the whole associated class and course can be implemented. In this paper, the problem will be solved by using MAX-MIN Ant System Algorithm. This algorithm is an alternative approach to ant colony optimization. This algorithm uses two tables of pheromones as stigmergy, i.e. timeslot pheromone table and room pheromone table. In addition, the selection of timeslot and room is done by using the standard deviation of the value of pheromones. Testing is carried out by using 105 events, 45 timeslots, and 3 types of categories based on the number of rooms provided, i.e. large, medium, and small. In each category, testing is performed 5 times and for each testing, the data recorded is the unplace and Soft Constraint Penalty. In general, the greater the number of rooms, the smaller the unplace.

Index Terms—ant colony optimization, max-min ant system, timetabling

I. PENDAHULUAN

Masalah penjadwalan mata kuliah merupakan masalah yang dihadapi oleh setiap universitas [1,2]. Masalah ini melibatkan sejumlah komponen, seperti waktu dan ruang dengan mempertimbangkan beberapa batasan (*constraint*) tertentu. Hal tersebut menjadikan masalah penjadwalan mata kuliah sebagai salah satu dari *NP Problem* [2-4], yaitu suatu masalah kombinatorial yang sulit diselesaikan.

Salah satu cara untuk menyelesaikan masalah penjadwalan adalah dengan menggunakan *metaheuristic network*. Terdapat lima *metaheuristic* yang telah dievaluasi untuk menyelesaikan masalah penjadwalan mata kuliah [5], yaitu Algoritma Genetik (GA), *Simulated Annealing* (SA), *Tabu Search* (TA), *Iterated Local Search* (ILS), dan *Ant Colony Optimization* (ACO). Hasil menunjukkan bahwa tidak terdapat *metaheuristic* terbaik yang dapat diterapkan

untuk semua bentuk penjadwalan.

Terdapat beberapa faktor yang memengaruhi proses penjadwalan mata kuliah di program studi Teknik Informatika Universitas Multimedia Nusantara (untuk selanjutnya disingkat UMN). Faktor-faktor tersebut adalah waktu perkuliahan, ruang kelas, dan mata kuliah. Waktu perkuliahan di UMN secara umum dimulai dari pukul 8 pagi hingga pukul 5 sore setiap harinya. Ruang kelas di UMN tidak bersifat homogen, tetapi memiliki fitur-fitur khusus untuk ruang tertentu. Hal ini menjadikan mata kuliah teori harus ditempatkan pada ruang kelas teori, demikian juga dengan mata kuliah praktik.

Proses penjadwalan secara mendasar dilakukan dengan mengalokasikan waktu dan ruang tersebut agar seluruh kelas mata kuliah dapat dilaksanakan [2]. Di UMN, hal ini dilakukan setiap awal semester dan menghasilkan jadwal yang akan digunakan pada semester tersebut. Dengan demikian, jadwal untuk suatu semester pada tahun tertentu akan berbeda dengan jadwal untuk semester yang sama di tahun yang berbeda.

Jadwal yang dihasilkan terkadang tidak seimbang, misalnya terdapat mahasiswa yang memiliki jadwal penuh untuk suatu hari, tetapi lenggang untuk hari lainnya. Oleh karena itu, penulis bermaksud membuat sebuah aplikasi yang dapat menghasilkan jadwal yang bersifat seimbang bagi mahasiswa di UMN.

II. ALGORITMA MAX-MIN ANT SYSTEM

Algoritma *MAX-MIN ant system* [6] merupakan algoritma yang dibangun sebagai pendekatan alternatif dan perbaikan untuk algoritma koloni semut (*Ant Colony Optimization*) [1,7]. Algoritma koloni semut menggunakan sebuah model yang dihasilkan dari pengamatan tingkah laku semut sesungguhnya, dan menggunakan model tersebut sebagai inspirasi untuk merancang sebuah algoritma untuk menyelesaikan

masalah. Ide utama dari algoritma ini adalah menerapkan prinsip pengorganisasian mandiri yang dilakukan semut sesungguhnya pada populasi *artificial agent* untuk menyelesaikan masalah komputasional [8].

Algoritma *MAX-MIN ant system* memiliki cara kerja yang hampir sama dengan algoritma koloni semut [1]. Pada setiap iterasi, algoritma melibatkan sejumlah *artificial ant* untuk membentuk solusi. Setiap *artificial ant* memilih jalur secara probabilistik untuk menuju solusi dengan menggunakan informasi *stigmergy* berupa jejak feromon. Feromon merupakan zat kimia yang disekresikan semut sesungguhnya untuk berkomunikasi dengan lingkungan dan semut lainnya [8].

Perbedaan algoritma *MAX-MIN ant system* dengan algoritma koloni semut terletak pada bagaimana kedua algoritma mengolah informasi jejak feromon, melakukan *local search* dan aturan meng-*update* jejak feromon. Selain itu, perbedaan lainnya adalah algoritma *max-min ant system* menggunakan batas atas dan batas bawah untuk nilai feromon [1,7].

III. MASALAH PENJADWALAN

Masalah penjadwalan dapat diselesaikan dengan menempatkan *event* (kombinasi unik dari kelas dan mata kuliah) pada suatu *timeslot* dan ruang, seperti pada [9], sehingga memenuhi *constraint* yang telah ditentukan.

Terdapat dua jenis *constraint* yang berfungsi sebagai pembatas suatu keluaran solusi, sekaligus sebagai fungsi evaluasi dari solusi yang dihasilkan. *Constraint* pertama disebut *hard constraint*. *Hard constraint* merupakan *constraint* yang bersifat tegas atau tidak boleh dilanggar. Berikut adalah *hard constraint* pada penjadwalan mata kuliah di Program Studi Teknik Informatika UMN, yang sebagian diadopsi dari [9].

- Tidak ada seorang mahasiswa yang menghadiri lebih dari satu *event* pada saat yang sama.
- Untuk setiap *event*, ruangan yang digunakan harus dapat menampung jumlah seluruh mahasiswa dari kelas tersebut dan memiliki fitur-fitur yang dibutuhkan.
- Hanya ada satu *event* yang menempati suatu ruang pada suatu waktu.
- Terdapat *event* yang pelaksanaannya harus mendahului suatu *event* tertentu lainnya. Misalnya, pelaksanaan mata kuliah teori harus mendahului mata kuliah praktik.

Constraint kedua disebut juga *soft constraint*. *Constraint* ini digunakan untuk menyatakan tingkat keseimbangan suatu solusi. Makin sedikit pelanggaran yang terjadi dalam suatu solusi, makin seimbang solusi tersebut. Berikut adalah *soft constraint* dari penjadwalan mata kuliah, yang dispesifikasi berdasarkan [9].

- Mahasiswa sebaiknya tidak hanya memiliki jadwal kelas pada *timeslot* terakhir di satu hari.
- Mahasiswa sebaiknya tidak memiliki jadwal kelas sebanyak lebih dari dua yang berturutan pada satu hari.
- Mahasiswa sebaiknya tidak hanya memiliki satu jadwal kelas di satu hari.

IV. IMPLEMENTASI ALGORITMA

A. Representasi Masalah

Terdapat *event* e , yaitu kombinasi kelas dan mata kuliah dalam array E dengan ukuran $|E|$, sehingga $e \in E$, yang akan menempati suatu *timeslot* t dan ruang r . Jumlah *timeslot* adalah $|T|$, yaitu hasil perkalian antara jumlah hari perkuliahan dan lama perkuliahan per hari dalam satuan jam. *Timeslot* t berada dalam array T berukuran $|T|$, sehingga $t \in T$. Jumlah ruangan adalah $|R|$, yang juga menyatakan ukuran dari array R , sehingga $r \in R$. Kombinasi *timeslot* dan ruang akan membentuk *timetable* yang direpresentasikan dalam bentuk matriks dua dimensi berukuran $|T| \times |R|$.

Terdapat dua tabel feromon, yaitu tabel feromon *timeslot* T_i dan tabel feromon ruang T_r . Tabel feromon *timeslot* direpresentasikan dalam bentuk matriks berukuran $|E| \times |T|$ dan memiliki elemen $\tau_{e,t}$, yaitu feromon untuk *event* e dan *timeslot* t . Tabel feromon ruang direpresentasikan dalam bentuk matriks berukuran $|E| \times |R|$ dan memiliki elemen $\tau_{e,r}$, yaitu feromon untuk *event* e dan ruang r . Nilai setiap elemen dalam kedua tabel feromon dinyatakan dalam bilangan real positif.

A. Rancangan Algoritma

B.1 Pengolahan Data Awal

Sebelum menerapkan algoritma untuk membentuk solusi, data awal yaitu data *event* diolah terlebih dahulu. Pengolahan data dilakukan sebagai berikut.

- Menentukan jumlah ruangan yang dapat digunakan oleh setiap *event*.
- Mengurutkan array E berdasarkan jumlah ruangan yang dapat digunakan secara menaik (*ascending*).

B.2 Inisialisasi Parameter

Algoritma *MAX-MIN ant system* yang digunakan pada permasalahan, memiliki beberapa parameter yang diadopsi dari [10], kecuali untuk jumlah semut dan tidak digunakannya *scaling parameter*. Berikut adalah parameter yang digunakan.

- Jumlah semut dalam setiap koloni adalah 10 ($m = 10$).
- Tingkat evaporasi feromon ρ . Nilai $\rho = 0,15$ untuk semua iterasi.
- Nilai feromon maksimal $\tau_{max} = 6,67$.
- Nilai feromon minimum $\tau_{min} = 2,5 \times 10^{-5}$.
- Laju eksplorasi $\gamma = 0,65$.

B.3 Inisialisasi Tabel Feromon

Seluruh elemen dari kedua tabel feromon yang berasosiasi dengan mata kuliah teori, diinisialisasi dengan nilai $(\tau_{max} + \tau_{min})/2$, yaitu 3,335. Namun, untuk elemen yang berasosiasi dengan mata kuliah praktik, elemen-elemen tersebut diinisialisasi dengan nilai $\tau_{max} = 6,67$.

B.4 Pembentukan Solusi

Setiap semut semu (untuk selanjutnya disebut semut) ke- i dalam koloni, dimana $i \leq m$, akan menempatkan setiap *event* e ke dalam *timetable*, dimana *timeslot* dan ruang dipilih secara probabilistik. Proses pemilihan dilakukan dengan memilih *timeslot* atau ruang untuk *event* e secara acak dengan nilai feromon yang berada dalam jangkauan $[\hat{\alpha}_{e,max} - s \hat{\alpha}_{e,max}]$, dimana $\tau_{e,max}$ adalah nilai feromon terbesar untuk *event* e dan s adalah nilai standar deviasi feromon untuk *event* e . Proses ini menghasilkan kandidat solusi yang ke- i , yaitu C_i .

Setiap kandidat solusi akan dievaluasi untuk menentukan solusi terbaik. Solusi terbaik dari seluruh semut dalam koloni disebut $C_{localbest}$. Proses evaluasi dilakukan dengan menghitung nilai antikalitas Q , yang nilainya bergantung pada jumlah *event* yang tidak dapat ditempatkan di *timetable* (*unplace*) dan nilai dari *Soft Constraint Penalty* (SCP). Solusi yang baik memiliki nilai antikalitas yang kecil. Hal ini berarti, makin kecil nilai antikalitas, makin besar peluang suatu kandidat solusi untuk menjadi $C_{localbest}$.

Pada iterasi koloni pertama, $C_{globalbest} = C_{localbest}$. $C_{globalbest}$ merupakan solusi terbaik dari seluruh solusi yang dihasilkan oleh seluruh koloni semut. Pada iterasi koloni selanjutnya, $C_{globalbest}$ adalah solusi dengan nilai antikalitas terkecil antara $C_{globalbest}$ saat itu dengan $C_{localbest}$ dari koloni terbaru.

Algoritma akan berjalan sampai batas waktu tertentu atau sampai menemukan solusi paling optimal, yaitu saat *unplace* = 0 dan SCP = 0.

B.5 Penentuan Nilai Antikalitas

Nilai antikalitas dari suatu kandidat solusi, yaitu $Q(s)$, bergantung pada nilai *unplace* dan SCP dari kandidat solusi tersebut. Persamaan (1) berikut adalah persamaan yang digunakan untuk menghitung nilai antikalitas.

$$Q(s) = unplace(s) \times 10000 + SCP(s) \quad (1)$$

Nilai SCP merupakan jumlah nilai yang ditentukan berdasarkan kejadian-kejadian sebagai berikut.

- Jika terdapat suatu kelas yang hanya memiliki sebuah *event* dalam satu hari, dikenakan SCP sebanyak 1 poin.
- Jika terdapat suatu kelas yang hanya memiliki sebuah *event* dalam satu hari dan *event* tersebut terjadi di *timeslot* terakhir pada hari yang bersangkutan, dikenakan SCP sebanyak 1 poin.
- Jika terdapat suatu kelas yang memiliki lebih dari atau sama dengan 3 *event* dalam satu hari secara berurutan, dikenakan SCP sebanyak 2 poin.

B.6 Pembaruan Jejak Feromon

Tabel feromon diperbarui berdasarkan solusi yang dihasilkan oleh semut, yaitu antara $C_{globalbest}$ saat itu atau $C_{localbest}$ dari koloni terbaru. Pemilihan dilakukan secara probabilistik, yaitu dengan menghitung nilai probabilitas $C_{localbest}$ dipilih untuk memperbarui tabel feromon. Secara matematis dapat dituliskan dalam (2) sebagai berikut.

$$p = \tilde{\alpha} \times \frac{Q(C_{globalbest})}{Q(C_{localbest})} \quad (2)$$

Jika nilai $p > 0,5$, tabel feromon akan diperbarui dengan $C_{localbest}$. Sebaliknya, jika $p \leq 0,5$, tabel feromon akan diperbarui dengan $C_{globalbest}$.

Dengan mengadopsi rumus pada [1], proses pembaruan nilai feromon dalam tabel feromon *timeslot* dapat dituliskan secara matematis dalam (3) sebagai berikut.

$$\hat{\alpha}_t = \begin{cases} \hat{\alpha}_{t+1} & \text{jika } t \text{ bagian dari solusi} \\ & \text{untuk event } e \\ \hat{\alpha}_t & \end{cases} \quad (3)$$

Proses pembaruan nilai feromon dalam tabel feromon ruang dapat dituliskan secara matematis dalam (4) sebagai berikut.

$$\hat{\alpha}_r = \begin{cases} \hat{\alpha}_{r+1} & \text{jika } r \text{ bagian dari solusi} \\ & \text{untuk event } e \\ \hat{\alpha}_r & \end{cases} \quad (4)$$

Selanjutnya, jejak feromon akan mengalami evaporasi. Persamaan (5) berikut adalah persamaan yang digunakan untuk menghitung nilai jejak feromon setelah evaporasi pada tabel feromon *timeslot*.

$$\hat{\alpha}_t = (1 - \tilde{n}) \times \hat{\alpha}_t \quad (5)$$

Nilai feromon dalam tabel feromon ruang juga mengalami evaporasi. Proses evaporasi tersebut dituliskan dalam (6) sebagai berikut.

$$\hat{\alpha}_r = (1 - \tilde{n}) \times \hat{\alpha}_r \quad (6)$$

Selanjutnya, seluruh nilai feromon dalam tabel feromon *timeslot* akan dihitung ulang dengan menggunakan (7) sebagai berikut.

$$\hat{\alpha}_t = \begin{cases} \hat{\alpha}_{min}, & \text{jika } \hat{\alpha}_t < \hat{\alpha}_{min} \\ \hat{\alpha}_{max}, & \text{jika } \hat{\alpha}_t > \hat{\alpha}_{max} \\ \hat{\alpha}_t & \end{cases} \quad (7)$$

Seperti pada tabel feromon *timeslot*, seluruh nilai feromon dalam tabel feromon ruang juga akan dihitung ulang dengan menggunakan (8) sebagai berikut.

$$\hat{\alpha}_r = \begin{cases} \hat{\alpha}_{min}, & \text{jika } \hat{\alpha}_r < \hat{\alpha}_{min} \\ \hat{\alpha}_{max}, & \text{jika } \hat{\alpha}_r > \hat{\alpha}_{max} \\ \hat{\alpha}_r & \end{cases} \quad (8)$$

B.7 Pseudocode

Berikut adalah *pseudocode* dari implementasi algoritma *MAX-MIN ant system*.

Prosedur PENJADWALAN-MATAKULIAH()

```
PengolahanDataAwal();
InisialisasiParameter();
InisialisasiTabelFeromonTimeslot();
InisialisasiTabelFeromonRuang();
DO
  FOR i = 1 TO jumlahSemut DO
    FOR j = 1 TO jumlahEvent DO
      cekTabelFeromonTimeslot(Event[j]);
      cekTabelFeromonRuang(Event[j]);
      tempatkanPadaWaktuRuang(Event[j]);
    ENDFOR
```

```
evaluasiTimetable();
ENDFOR
Clocalbest = solusiLokalTerbaikDalamKoloni();
Cglobalbest = terbaik(Cglobalbest, Clocalbest);
IF Cglobalbest = solusiOptimal() THEN break;
ENDIF;
updateTabelFeromonTimeslot();
updateTabelFeromonRuang();
WHILE (!MelewatiBatasWaktu);
EndProsedur
```

V. PENGUJIAN

Pengujian dilakukan berdasarkan penjadwalan yang dilakukan pada Program Studi Teknik Informatika UMN untuk semester genap, kurikulum 2010/2011. Pada penjadwalan tersebut, terdapat 105 *event* yang harus ditempatkan pada *timetable*, dengan menggunakan 45 *timeslot*, yaitu 5 hari perkuliahan dan 9 jam lama perkuliahan per hari.

Adapun pengujian yang dilakukan dibagi menjadi 3 kategori berdasarkan jumlah ruangan yang disediakan, yaitu kategori besar, sedang, dan kecil. Jumlah dan spesifikasi ruangan yang disediakan, yaitu ruangan untuk teori dan praktik, berdasarkan kategori ditunjukkan dalam Tabel 1 sebagai berikut.

Tabel 1. Jumlah Ruangan Berdasarkan Kategori

Kategori	Jumlah Ruangan		
	Teori	Praktik	Total
Besar	10	6	16
Sedang	5	4	9
Kecil	4	3	7

Pada setiap kategori, dilakukan 5 kali pengujian dan untuk setiap pengujian, data yang dicatat adalah nilai *unplace* dan SCP. Hasil pengujian untuk setiap kategori ditunjukkan pada Tabel 2 sebagai berikut.

Tabel 2. Hasil Pengujian

Kategori Besar		
Pengujian ke-	Nilai Unplace	Nilai SCP
1	0	30
2	0	23
3	0	25
4	0	28
5	0	32
Kategori Sedang		
Pengujian ke-	Nilai Unplace	Nilai SCP
1	2	27
2	0	27

3	2	22
4	1	31
5	1	21
Kategori Kecil		
Pengujian ke-	Nilai Unplace	Nilai SCP
1	9	25
2	8	27
3	5	27
4	6	22
5	6	25

Berdasarkan Tabel 2, dapat dihitung nilai rata-rata dari *unplace* dan SCP untuk setiap kategori. Hasil perhitungan rata-rata ditunjukkan pada Tabel 3.

Nilai persentase rata-rata *unplace* didefinisikan sebagai perbandingan antara nilai rata-rata *unplace* untuk kategori tertentu (terdapat dalam Tabel 3) dengan jumlah seluruh *event*, yaitu 105. Hal ini ditunjukkan pada Tabel 4.

Berdasarkan Tabel 4, dapat diketahui bahwa makin besar jumlah ruangan, makin kecil nilai *unplace*. Hal ini ditunjukkan dengan adanya kenaikan nilai persentase rata-rata *unplace* dari pengujian dengan jumlah ruangan kategori besar ke sedang, yaitu dari 0% menjadi 1,14% dan dari kategori sedang ke kecil, yaitu dari 1,14% menjadi 6,48%.

Tabel 3. Rata-rata Unplace dan SCP

Kategori	Nilai Unplace	Nilai SCP
Besar	0	27,6
Sedang	1,2	25,6
Kecil	6,8	25,2

Tabel 4. Persentase Rata-rata Unplace

Kategori	Rata-rata Unplace	Total Event	Persentase
Besar	0	105	0%
Sedang	1,2		1,14%
Kecil	6,8		6,48%

Tabel 5. Analisis Variansi Satu Faktor

Sumber Variasi	Derajat Kebebasan	Jumlah Kuadrat	Kuadrat Rata-rata	Statistik F
Antarkategori	2	16,53	8,267	0,723
D a l a m kategori	12	137,2	11,43	
Total	14	153,7	Nilai-p	0,51

Distribusi nilai SCP antarkategori dalam Tabel 2, tidak menunjukkan perbedaan yang signifikan. Hal ini juga ditunjukkan dalam Tabel 3, yaitu nilai rata-rata SCP untuk setiap kategori memiliki perbedaan yang kecil. Dengan demikian, dapat dikatakan bahwa perbedaan kategori pengujian tidak berdampak signifikan pada nilai SCP. Hal ini dapat dibuktikan dengan melakukan analisis variansi satu faktor, dengan hipotesis nihil (*null hypothesis*) adalah nilai rata-rata SCP untuk setiap kategori adalah sama. Hasil analisis variansi ditunjukkan pada Tabel 5.

Berdasarkan Tabel 5, dengan menggunakan nilai $\alpha = 0,05$, dapat diketahui bahwa nilai-p lebih besar daripada α (nilai-p > 0,05). Hal ini berarti tidak terdapat cukup keyakinan untuk menolak hipotesis nihil. Dengan demikian, tidak terdapat cukup keyakinan untuk menyatakan bahwa nilai rata-rata SCP antara kategori satu dengan kategori lainnya adalah berbeda. Hal ini membuktikan bahwa masih terdapat keyakinan untuk menyatakan bahwa perbedaan kategori pengujian tidak berdampak signifikan pada nilai SCP.

Gambar 1 berikut ini merupakan contoh jadwal yang dihasilkan pada pengujian pertama untuk kategori jumlah ruangan besar.

Gambar 1. Contoh jadwal pada pengujian ke-1 untuk kategori jumlah ruangan besar

VI. SIMPULAN

Algoritma *MAX-MIN ant system* dengan implementasi menggunakan dua tabel feromon, yaitu tabel feromon *timeslot* dan tabel feromon ruang, serta penggunaan informasi standar deviasi dari nilai feromon dapat digunakan untuk menyelesaikan masalah penjadwalan mata kuliah. Hal tersebut berdasarkan hasil pengujian, yaitu jika jumlah ruangan relatif besar, seluruh *event* dapat masuk dalam *timetable* dan tidak ada *hard constraint* yang dilanggar.

Implementasi algoritma yang dilakukan belum mampu menghasilkan solusi yang optimal, yaitu

saat nilai *unplace* dan SCP sama dengan nol. Hal ini berarti jadwal yang seimbang belum dapat dicapai. Berdasarkan hasil pengujian, walaupun dengan menggunakan jumlah ruangan yang besar, algoritma belum dapat menghasilkan solusi yang optimal.

Berdasarkan hasil pengujian, dapat disimpulkan pula bahwa perbedaan kategori pengujian tidak berdampak signifikan pada nilai SCP. Dengan demikian, penambahan jumlah ruangan tidak dapat memperbaiki tingkat keseimbangan suatu jadwal.

Algoritma yang digunakan pada *paper* ini dapat dikembangkan agar dapat memenuhi batasan (*constraint*) yang lebih luas, seperti adanya faktor kapasitas ruangan dan fitur ruangan yang lebih kompleks. Penelitian lebih lanjut dapat dilakukan untuk menemukan kombinasi jumlah *timeslot* dan ruangan untuk jumlah *event* tertentu, agar dapat menghasilkan solusi yang optimal. Hal ini juga dapat dilakukan untuk menemukan nilai inisial yang tepat untuk setiap tabel feromon. Selain itu, diperlukan pula pengujian lebih lanjut agar algoritma ini memungkinkan untuk diterapkan secara umum dalam berbagai masalah penjadwalan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Yustinus Eko Soelistio, dosen mata kuliah Intelegensia Semu UMN yang telah membimbing penulis dalam menyusun *paper* ini. Penulis juga mengucapkan terima kasih kepada keluarga dan rekan-rekan mahasiswa Teknik Informatika UMN yang telah memberi dukungan dan semangat selama penelitian dan penyusunan *paper* ini dilakukan.

DAFTAR PUSTAKA

- [1] Krzysztof Socha, Michael Sampels, dan Max Manfrin, "Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art," di dalam *EvoWorkshops '03 Proceedings of the 2003 international conference on Applications of evolutionary computing*, Berlin, 2003, hal. 334-345.
- [2] Krzysztof Socha, Joshua Knowles, and Michael Sampels, "A MAX-MIN Ant System for the University Course Timetabling Problem," di dalam *ANTS '02 Proceedings of the Third International Workshop on Ant Algorithms*, London, 2002, hal. 1-13.
- [3] Marco Dorigo dan Thomas Stutzle, "Ant Colony Optimization for NP-Hard Problem," di dalam *Ant Colony Optimization*. Massachusetts: MIT Press, 2003, bab 5, subbab 5.2, hal. 159-167.
- [4] Clemens Nothegger, Alfred Mayer, Andreas M. Chwatal, and Günther R. Raidl, "Solving The Post Enrolment Course Timetabling Problem By Ant Colony Optimization," *Annals of Operations Research*, vol. 194, no. 1, hal. 325-339, April 2012.
- [5] Olivia Rossi-Doria, dkk., "A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem," di dalam *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT 2002)*, Ghent, 2002, hal. 329-351.
- [6] Thomas Stützle dan Holger H. Hoos, "MAX-MIN Ant System," *Future Generation Computer Systems*, vol. 16, no. 9, hal. 889-914, Juni 2000.
- [7] Marco Dorigo. (2007). Ant Colony Optimization. Scholarpedia [Media online]. 2(3). Alamat situs: http://www.scholarpedia.org/article/Ant_colony_optimization
- [8] Marco Dorigo dan Thomas Stutzle, "From Real to Artificial Ants," di dalam *Ant Colony Optimization*. Massachusetts: MIT Press, 2003, bab 1, subbab 1.1, hal. 1-7.
- [9] Philipp Kostuch, "The University Course Timetabling Problem with a 3-phase approach," di dalam *PATAT'04 Proceedings of the 5th international conference on Practice and Theory of Automated Timetabling*, Berlin, 2005, hal. 109-125.
- [10] Socha, Krzysztof. (2003, Maret 31). MAX-MIN Ant System for International Timetabling Competition [Media online]. Alamat situs: <http://www.idsia.ch/Files/ttcomp2002/socha.pdf>