

IMPLEMENTASI ALOKASI JADWAL MATA PELAJARAN SMU MENGUNAKAN ALGORITMA KOLONI SEMUT (AKS)

Devie Rosa Anamisa, S.Kom, M.Kom

Jurusan D3 Teknik Multimedia Dan Jaringan-Fakultas Teknik

Universitas Trunojoyo Madura Indonesia 60111

email:devros_gress@yahoo.com

ABSTRAK

Dalam merancang penjadwalan mata pelajaran di SMU (Sekolah Menengah Umum) membutuhkan waktu, tenaga dan ketelitian. Karena dalam perancangan jadwal mata pelajaran harus memperhatikan aturan-aturan penjadwalan serta faktor-faktor yang mempengaruhi seperti guru, kelas, waktu dan mata pelajaran serta batasan-batasan baik batasan yang boleh dilanggar (*soft constraint*) maupun batasan yang harus dipenuhi (*hard constraint*) dalam mengalokasikan jadwal. Beberapa solusi terhadap penyelesaian penjadwalan mata pelajaran menunjukkan bahwa semakin besar volume batasan penjadwalan maka alokasi jadwal semakin kompleks sehingga diperlukan algoritma optimasi untuk menyelesaikan permasalahan tersebut. Penelitian ini menyelesaikan permasalahan alokasi jadwal mata pelajaran menggunakan algoritma koloni semut (AKS). AKS merupakan suatu algoritma berbasis *agent* dengan meminimalkan terjadinya bentrok jadwal sehingga mampu menyelesaikan permasalahan penjadwalan mata pelajaran di SMU dengan cepat dan mudah. Hasil uji coba menunjukkan bahwa AKS yang dikembangkan dalam penelitian ini mampu menyelesaikan persoalan penjadwalan mata pelajaran dengan hasil memuaskan. Kualitas keberhasilan tersebut diperoleh menggunakan parameter nilai alfa dan beta masing-masing sebesar satu dan penguapan feromon sebesar 0,3.

Kata Kunci : *Implementasi, Penjadwalan mata pelajaran, Algoritma, Optimasi, Algoritma Koloni Semut.*

1. Pendahuluan

Alokasi jadwal mata pelajaran di SMU (Sekolah Menengah Umum) membutuhkan waktu, tenaga dan ketelitian untuk membuatnya. Alokasi jadwal harus dibuat dengan memperhatikan aturan-aturan penjadwalan yang cukup rumit serta *constraint-constraint* yang berlaku di SMU. Selama ini penjadwalan pelajaran di SMU terkait jadwal mata pelajaran dan pembagian guru di setiap kelas yang ada masih menggunakan cara manual. Hal ini memicu sering muncul masalah bentrok jadwal yang menyebabkan seorang guru berada di dua kelas yang berbeda. Untuk mengatasi permasalahan ini, diperlukan perancangan dalam pembuatan jadwal mata pelajaran di SMU. Cukup banyak teknik yang telah dipergunakan untuk menyelesaikan alokasi jadwal mata pelajaran SMU. Tetapi untuk menyelesaikan sebuah permasalahan penjadwalan mata pelajaran di SMU, sebuah solusi yang optimum sangat tidak memungkinkan untuk dicari. Karena itulah pencarian solusi yang mendekati optimum sudah dirasakan cukup.

Beberapa teknik yang bisa digunakan untuk mendapatkan solusi tersebut yaitu menggunakan algoritma berbasis *agent* atau algoritma pencarian konvensional. Dua algoritma yang dapat digunakan untuk melakukan proses penjadwalan mata pelajaran di antaranya adalah pencarian melebar pertama (*breadth-first search*), pencarian mendalam pertama (*depth-first search*). Akan tetapi, kedua metode ini tidak memiliki kecenderungan untuk memberikan hasil

dengan solusi yang baik. Hal ini disebabkan kedua metode di atas merupakan bagian dari metode pencarian buta (*blind search*). Salah satu algoritma berbasis *agent* yang bisa digunakan untuk menyelesaikan permasalahan optimasi kombinatorial seperti permasalahan penjadwalan mata pelajaran yaitu Algoritma Koloni Semut (AKS). Algoritma Koloni Semut (AKS), termasuk dalam kelompok *Swarm Intelligence*, merupakan salah satu jenis pengembangan paradigma yang digunakan untuk memecahkan masalah tersebut dengan meniru perilaku kumpulan atau kawanan (*Swarm*) serangga dan terinspirasi dari tingkah laku koloni semut dalam mencari makan. Algoritma Koloni Semut diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut [1]. Beberapa penelitian dalam penyelesaian penjadwalan menggunakan algoritma koloni semut menyebutkan bahwa AKS merupakan algoritma optimasi yang memiliki akurasi yang tinggi dibandingkan dengan menggunakan algoritma optimasi lainnya. Seperti penelitian sebelumnya mengenai penjadwalan ujian pada lembaga pendidikan tinggi yang memiliki dua batasan, dikenal dengan istilah *first order-second order conflicts*. *First-order conflict* terjadi ketika siswa mengambil dua ujian yang dijadwalkan pada waktu yang sama. *Second-order conflict* terjadi ketika siswa mengambil dua ujian di waktu yang berdekatan. Untuk menyelesaikan permasalahan tersebut, penelitian ini menerapkan teknik SA (*simulasi annealing*), AG (Algoritma Genetika) dan ACS (*Ant Colony System*). Empat teknik tersebut menghasilkan optimasi berturut-turut ACS (43,3%), AG (26,6%) dan SA (3,3%) [2].

Dalam penelitian ini diharapkan dapat membuat implementasi alokasi jadwal mata pelajaran SMU sehingga mampu menyelesaikan permasalahan yang dapat mengganggu keefektifan waktu belajar siswa dengan memenuhi sejumlah *constraint* dan aturan aturan penjadwalan yang sudah ditentukan dengan memastikan seorang guru mengajar mata pelajaran sesuai dengan kurikulum yang ditentukan untuk menghindari terjadinya bentrok antar komponen-komponen yang dijadwalkan dalam mencapai penjadwalan yang optimal.

2. Tinjauan Pustaka

Dasar teori merupakan sebuah teori penunjang untuk melaksanakan penelitian ini. Dalam penelitian ini teori penunjang tersebut diambil berdasarkan konsep yang ekivalen dengan pembuatan sistem ini.

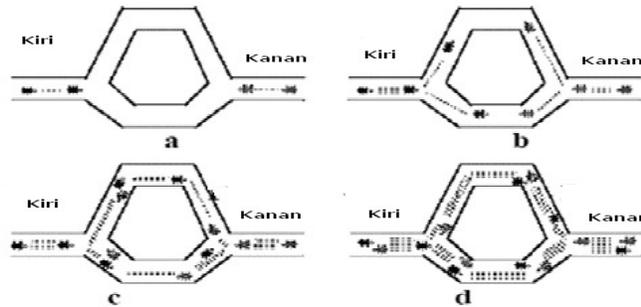
Alokasi jadwal merupakan proses untuk menyusun suatu jadwal (alokasi sumber daya pada objek-objek yang ada pada ruang waktu dan bergantung pada kendala-kendala sedemikian rupa sehingga terpenuhi sekumpulan sasaran yang diinginkan) atau urutan proses yang diperlukan dalam sebuah persoalan.

Persoalan penjadwalan biasanya berhubungan dengan penjadwalan mata pelajaran. Permasalahan jadwal mata pelajaran (*teacher timetable*) adalah permasalahan pengalokasian waktu dan tempat untuk suatu kegiatan di instansi pendidikan, seminar, dan lain-lain untuk memenuhi beberapa kendala yang berhubungan dengan kapasitas dan lokasi dari ruang, waktu, dan hal lain yang berhubungan dengan sistem belajar mengajar.

Batasan-batasan dalam penjadwalan dibagi ke dalam dua kategori yaitu "*hard*" dan "*soft*" *constraints*. Jadwal yang melanggar "*hard constraints*" adalah solusi yang tidak mungkin, dan harus diperbaiki atau di buang dengan algoritma penjadwalan [3]. *Soft constraints* tidak begitu penting dibandingkan dengan *hard constraints*, dan biasanya tidak mungkin untuk menghindari pelanggaran. Ketika metode penjadwalan diaplikasikan, jadwal biasanya ditingkatkan dengan fungsi penalti, yang menghitung tingkat pelanggaran jadwal yang telah disusun. Beberapa *soft constraints* lebih penting dibandingkan dengan *hard constraints* dan ini sering di tentukan dengan nilai prioritas.

Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Gambar 1. menunjukkan perjalanan semut menemukan sumber makanan. Koloni semut dapat menemukan jalur terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilewati. Semakin banyak semut yang melewati suatu lintasan maka semakin jelas bekas jejak kakinya.

Hal ini menyebabkan lintasan yang dilalui semut dalam jumlah sedikit semakin lama semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya lintasan yang dilalui semut dalam jumlah banyak semakin lama akan semakin bertambah kepadatan semut yang melewatinya atau bahkan semua semut melewati lintasan tersebut.

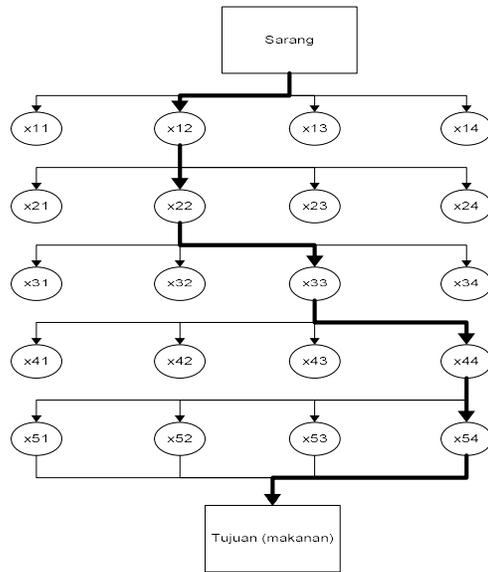


Gambar 1. Perjalanan Semut Menemukan Sumber Makanan [4].

Gambar 1.a menunjukkan perjalanan semut dalam menemukan jalur terpendek dari sarang ke sumber makanan, terdapat dua kelompok semut yang melakukan perjalanan. Kelompok semut kiri berangkat dari arah kiri ke kanan dan kelompok semut kanan berangkat dari kanan ke kiri. Kedua kelompok berangkat dari titik yang sama dan dalam posisi pengambilan keputusan jalan sebelah mana yang akan diambil. Kelompok kiri membagi dua kelompok lagi. Sebagian melewati jalan atas dan sebagian melewati jalan bawah. Hal ini juga berlaku pada kelompok kanan. Gambar 1.b dan Gambar 1.c menunjukkan bahwa kelompok semut berjalan pada kecepatan yang sama dengan meninggalkan feromon atau jejak kaki di jalan yang telah dilalui.

Feromon yang ditinggalkan oleh kumpulan semut yang melewati jalan atas telah mengalami banyak penguapan karena semut yang melewati jalan atas berjumlah lebih sedikit dibandingkan jalan yang di bawah. Hal ini disebabkan jarak yang ditempuh lebih panjang dibandingkan jalan bawah. Sedangkan feromon yang berada pada bagian bawah penguapannya cenderung lebih lama. Karena semut yang melewati jalan bawah lebih banyak daripada semut yang melewati jalan atas. Gambar 1.d menunjukkan bahwa semut-semut yang lain pada akhirnya memutuskan untuk melewati jalan bawah karena feromon yang ditinggalkan masih banyak, sedangkan feromon pada jalan atas sudah banyak menguap sehingga semut-semut tidak memilih jalan atas.

Semakin banyak semut yang melewati jalan maka semakin banyak semut yang mengikutinya, semakin sedikit semut yang melewati jalan, maka feromon yang ditinggalkan semakin berkurang bahkan hilang. Dari sinilah kemudian terpilihilah jalur terpendek antara sarang dan sumber makanan. Proses AKS dapat dijelaskan dengan membuat grafik *multilayer* dari permasalahan optimasi yang dihadapi. Seperti ditunjukkan pada Gambar 2. dimana jumlah *layer* sama dengan jumlah *variable* dari permasalahan yang dihadapi dan jumlah simpul (*node*) dalam suatu *layer* tertentu sama dengan jumlah nilai diskret yang diijinkan untuk *variable* yang berkaitan untuk suatu *variable*. Gambar 2. menunjukan suatu masalah dengan lima *variable* dan empat kemungkinan nilai untuk setiap *variable*.



Gambar 2. Proses AKS dalam Bentuk *Network Multi-layer*

Proses dalam AKS bisa dijelaskan sebagai berikut, misalkan ada N semut dalam satu koloni. Semut-semut itu akan memulai gerakan dari sarang mereka menuju tujuan akhir melalui beberapa lapis, dari lapisan pertama menuju lapisan terakhir dan berakhir pada simpul tujuan di akhir setiap siklus atau iterasi. Setiap semut hanya bisa memilih satu simpul dalam setiap *layer* dengan prosedur tertentu untuk memilihnya seperti Persamaan 1.

Simpul-simpul yang dipilih sepanjang lintasan ini merupakan kandidat solusi. Sekali lintasan terpilih, semut-semut akan menyimpan sejumlah feromon pada lintasan tersebut menggunakan rumus. Seekor semut k pada simpul i akan memilih simpul j yang dituju pada *layer* berikutnya dengan probabilitas pada Persamaan 1.

$$P_{i,j} = \begin{cases} \frac{\tau_{i,j}^\alpha}{\sum \tau_{i,j}^\alpha} & , \text{jika } j \in N_i^{(k)} \\ 0 & , \text{jika } j \notin N_i^{(k)} \end{cases}, \quad (1)$$

di mana α menunjukkan derajat kepentingan feromon dan $N_i^{(k)}$ adalah pilihan yang dipunyai semut k (*neighborhood*) pada saat semut berada pada simpul i .

Ketetangaan dari semut k pada simpul i akan mengandung semua simpul yang bisa dituju dan tersambung secara langsung ke simpul i , kecuali simpul yang sudah dikunjungi sebelumnya.

Seekor semut k ketika melewati ruas akan meninggalkan feromon $\tau_{i,j}$. Jumlah feromon yang terdapat pada ruas i,j setelah dilewati semut k diberikan dengan Persamaan 2.

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta\tau^k, \quad (2)$$

Dengan meningkatnya nilai feromon pada ruas $i-j$, maka kemungkinan ruas ini akan dipilih lagi pada iterasi berikutnya semakin besar. setelah sejumlah simpul dilewati maka akan terjadi penguapan feromon dengan aturan pada Persamaan 3.

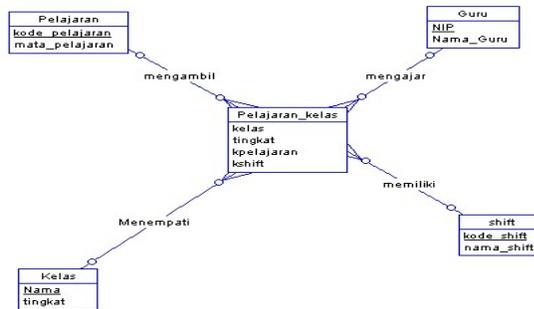
$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \rho \cdot \Delta\tau_{i,j} \in A, \quad (3)$$

di mana $\rho \in (0,1)$ adalah parameter tingkat penguapan dan A menyatakan segmen atau ruas yang sudah dilalui oleh semut k sebagai bagian dari lintasan dari sarangnya menuju ke makanan.

Penurunan jumlah feromon memungkinkan semut untuk mengeksplorasi lintasan berbeda selama proses pencarian. Ini juga akan menghilangkan kemungkinan lintasan yang kurang bagus.

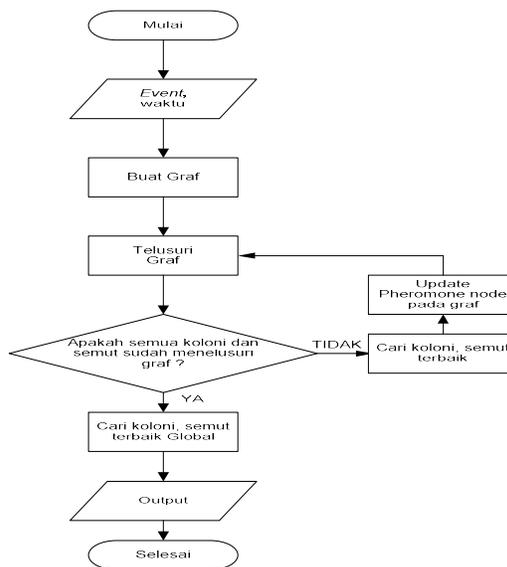
3. Perancangan dan Implementasi

Perancangan sistem digunakan untuk memberikan gambaran sehingga diharapkan dapat mempermudah dalam membuat aplikasi penjadwalan. Untuk itu perlu dilakukan implementasi data dan proses serta implementasi antar muka. Implementasi data merupakan berupa data masukan, data proses dan data keluaran. Data masukan berupa data yang digunakan perangkat lunak untuk diproses dengan AKS dan disimpan dalam *database* MySQL. Digunakan sebagai masukan pada implementasi data dalam penjadwalan tersebut. Data-data untuk membuat penjadwalan berasal dari data akademik di SMU, seperti entitas guru, entitas mata pelajaran dan entitas kelas. Sedangkan data-data entitas lain, dibuat untuk membangun sistem penjadwalan ini. Data masukan yang diolah dalam sistem ditunjukkan dalam Gambar 3.



Gambar 3. Conceptual Data Model Alokasi Jadwal SMU

Sedangkan data proses adalah data-data yang digunakan selama proses pembentukan jadwal. Data ini diolah dari data masukan dan digunakan untuk menghasilkan data keluaran. Terdapat beberapa jenis data proses dalam penjadwalan dengan AKS, dapat ditunjukkan dalam Gambar 4.



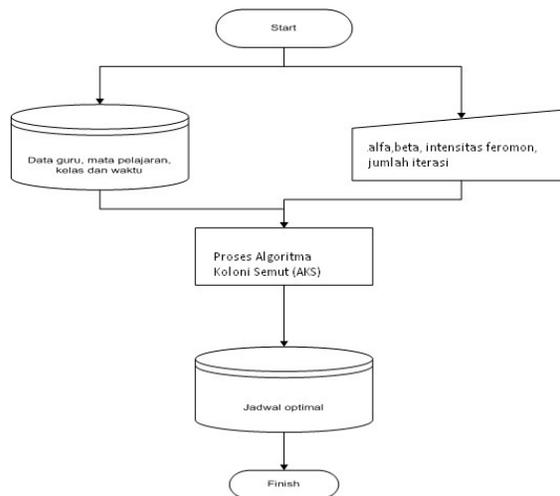
Gambar 4. AKS untuk penjadwalan Mata Pelajaran SMU

Sedangkan implementasi antar-muka atau implementasi keluaran berupa implementasi jadwal dengan pelanggaran kecil merupakan hasil akhir yang didapatkan dari penyelesaian permasalahan penjadwalan mata pelajaran menggunakan algoritma AKS. Jadwal tersebut digunakan sebagai jadwal mata pelajaran dan tidak memperhatikan jadwal belajar mengajar di luar sekolah (*ekstra kurikuler*).

Penelitian ini menggunakan batasan-batasan, asumsi dan *rule* sebagai berikut:

1. Ada sejumlah mata pelajaran di SMU dari kelas X sampai kelas XII yang ingin dibuat jadwal.
2. Penjadwalan dibuat untuk menjadwalkan mata pelajaran dengan pengulangan selama satu minggu.
3. Dalam satu waktu tidak boleh ada guru yang mengajar di dua tempat.
4. Sebuah kelas tidak boleh dipakai oleh lebih dari satu mata pelajaran pada waktu yang bersamaan.
5. Mata pelajaran yang memiliki kelas yang sama tidak boleh berada pada suatu hari dan jam yang sama kecuali mata pelajaran tersebut memiliki kode pelajaran yang sama (berbeda kelas).
6. Jadwal sampai hari sabtu.
7. Jadwal berada dalam batas waktu dan ruang yang ditentukan.

Dalam implementasi penjadwalan ini digunakan untuk memberikan informasi dalam penyusunan jadwal belajar siswa dalam satu minggu untuk kelas kelas X, XI dan XII dengan kriteria-kriteria penjadwalan baik *hard-constraint* maupun *soft constraint* sehingga dapat mendesain sistem penjadwalan dengan mudah dan cepat. Diagram alir data yang terjadi dalam algoritma yang dikembangkan untuk menghasilkan suatu output. Penentuan desain aplikasi pada penelitian ini memberikan gambaran mengenai apa dan bagaimana algoritma yang diusulkan akan diterapkan. Tahapan proses penjadwalan dapat dilihat pada Gambar 5.



Gambar 5. Tahapan Proses Penjadwalan Mata Pelajaran Di SMU dengan AKS

4. Hasil dan Pembahasan

Penjadwalan mata pelajaran adalah proses penempatan mata pelajaran pada waktu dan kelas yang tersedia dengan memenuhi syarat-syarat tertentu. Sebelum melakukan proses penempatan mata pelajaran, yang perlu dilakukan adalah perancangan jadwal mata pelajaran di SMU.

Untuk menghasilkan jadwal yang optimal, juga perlu dilakukan perancangan proses menggunakan AKS. Dalam menghasilkan penjadwalan yang optimal sangat dipengaruhi oleh beberapa faktor-faktor yang mendukung penjadwalan tersebut diantaranya guru, mata pelajaran, kelas dan shift. Selain beberapa faktor pendukung, penjadwalan juga memiliki beberapa constraint yang tidak boleh dilanggar dalam mendapatkan hasil jadwal yang optimal sehingga tidak mengganggu keefektifan waktu belajar siswa.

Parameter-parameter AKS digunakan untuk menghasilkan jadwal optimal perlu melakukan pencarian jalur jadwal yang memiliki pelanggaran terkecil, sehingga dipengaruhi parameter-parameter AKS, diantaranya nilai heuristik, intensitas feromon, alfa dan beta. Nilai heuristik merupakan nilai yang digunakan semut dalam mencari informasi heuristik dari simpul i ke simpul j . Informasi heuristik bagi semut adalah informasi dalam memilih jalur dengan kadar feromon yang tinggi.

Jalur dengan kadar feromon yang tinggi akan mendapat perhatian lebih sehingga semut-semut lainnya akan menggunakan jalan tersebut. Akan tetapi jalur yang sering dilewati semut, memiliki kadar feromon yang semakin menipis, ini dikarenakan feromon mengalami penguapan. Nilai heuristik juga tergantung pada setiap permasalahan yang berbeda. Sedangkan alfa dan beta adalah parameter pengontrol bobot nilai heuristik dan intensitas feromon atau jejak feromon.

Implementasi *interface* dari parameter seperti yang ditunjukkan dengan $timeslot=6$ hari (senin-sabtu) $\times 15 timeslot/hari = 90 timeslot$, P (derajat penguapan feromone)=0.3 sehingga terbentuk $T_{max}=1/p=1/0,3=3,3$, $T_{min}=0,19$. Dari parameter parameter tersebut dilakukan penelusuran graf untuk menentukan simpul-simpul kemungkinan yang memiliki penguapan feromon yang kuat, dan dengan parameter-parameter tersebut menghasilkan tiga kemungkinan pemilihan *node* dan kandungan feromone-nya, yaitu: $node (e_2,t_1) =0,2$, $node (e_2,t_2)=0,3$, dan $node (e_2,t_3)=0,4$, peluang $(e_2,t_1) =2/9$, $(e_2,t_2) =3/9$ dan $(e_2,t_3) =4/9$, seperti yang ditunjukkan pada Gambar 6.

	e1	e2	e3	e4	e5	e6	e7	...	e193
t1	○	○	○	○	○	○	○	...	○
t2	○	○	○	○	○	○	○	...	○
t2	○	○	○	○	○	○	○	...	○
t4	○	○	○	○	○	○	○	...	○
t5	○	○	○	○	○	○	○	...	○
t6	○	○	○	○	○	○	○	...	○
t7	○	○	○	○	○	○	○	...	○
t8	○	○	○	○	○	○	○	...	○
t9	○	○	○	○	○	○	○	...	○
t10	○	○	○	○	○	○	○	...	○
t11	○	○	○	○	○	○	○	...	○
t12	○	○	○	○	○	○	○	...	○
t13	○	○	○	○	○	○	○	...	○
...
t90	○	○	○	○	○	○	○	...	○

Gambar 6. Node yang terbentuk ($event=193, timeslot=90$)

Setelah setiap simpul yang berisikan koloni semut mempunyai produk feromon, langkah selanjutnya adalah semut melintasi semua jalur jadwal. Jalur jadwal terdiri dari beberapa *layer* dan setiap *layer* terdapat beberapa simpul dan di dalam simpul terdapat beberapa semut. Setiap simpul yang memiliki kadar feromon yang kuat akan disimpan dalam *tabulist*.

Kemudian dilakukan evaluasi probabilitas feromon dalam mencari lintasan yang jarang dilewati oleh semut atau probabilitasnya kecil. Dengan kata lain mencari lintasan dengan pelanggaran yang terkecil. Hasil penelusuran *node per node* disimpan pada ruang yang memungkinkan seperti pada Gambar 7.

t\Ruang	R 4	R 3	R 5
t1	(Event 7)		
t2			(Event 3)
t3		(Event 2)	
t4			
t5			
t6	(Event 1)		
t7			
t8			(Event 4)
t9			
t10			
t11	(Event 6)		(Event 4)
t12			
t13		(Event 5)	(Event 5)
t14			
t15			

Gambar 7. Pemasangan penelusuran *node* ke ruangan

Pencarian koloni dan semut terbaik (nilai pelanggaran *hard constraint* terkecil), maka didapatkan pula *node-node* yang dilalui koloni dan semut tersebut, seperti pada Tabel 1.

Tabel 1. Koloni, semut dan nilai pelanggaran *hard constraint*-nya

koloni	semut	Nilai
1	1	13.0
1	2	14.0
1	3	18.0
1	4	10.0
1	5	13.0
2	1	9.0
2	2	13.0
2	3	13.0
2	4	10.0
2	5	13.0
3	1	18.0
3	2	14.0
3	3	8.0
3	4	14.0
3	5	13.0

Kemudian update feromone semua *node* dengan Persamaan 2. Sedangkan untuk *node* yang merupakan penelusuran koloni dan semut terbaik, maka feromone-nya di-*update* lagi dengan menambah 1, seperti pada Persamaan 4.

$$\tau_{(e,z)} \leftarrow \begin{cases} \tau_{\min}, & \text{jika } \tau_{(e,z)} < \tau_{\min} \\ \tau_{\max}, & \text{jika } \tau_{(e,z)} > \tau_{\max} \\ \tau_{(e,z)} & \end{cases} \quad (4)$$

Kemudian lakukan pengecekan untuk semua *node*. Untuk merepresentasikan kebutuhan itu, hasil implementasi aplikasi penjadwalan mata pelajaran SMU ditunjukkan pada Gambar 8.



Gambar 8. Antar muka Perancangan Penjadwalan Mata pelajaran di SMU

5. Kesimpulan

1. Dengan melihat hasil pengujian di atas dapat disimpulkan bahwa algoritma koloni semut dapat diterapkan dalam pemecahan masalah penjadwalan mata pelajaran di SMU, dengan hasil yang baik (tidak terjadi bentrok antar komponen-komponen yang dijadwalkan).
2. Hasil yang baik cenderung diperoleh pada iterasi-iterasi besar, hal ini membuktikan bahwa *update* feromone berpengaruh terhadap penelusuran semut dalam menghasilkan solusi.
3. Waktu proses penjadwalan dengan algoritma koloni semut ini sangat dipengaruhi oleh jumlah komponen yang dijadwalkan, semakin besar kasus maka waktu yang diperlukan akan semakin lama pula.

Daftar Pustaka

- [1]. P.Pongcharoen, W. P. (2007). Stochastic Optimasation Timetabling Tool For University Course Scheduling. *Int. J. Production Economics*. Vol.112 , Hal. 903-918.
- [2]. Kusumadewi, S. P. (2005). Penyelesaian Masalah Optimasi Menggunakan Teknik-Teknik Heuristik. Yogyakarta: Graha Ilmu.
- [3]. Dezhen Zhang, L. D. (2011). Hybrid Ant colony Optimization Based On Genetic Algorithm for Container Loading Problem. *IEEE Soft Computing and Patern Recognition (SoCPAR)* , 10-14.
- [4]. M.Dorigo, T. S. (2004). Ant Colony Optimazation. The MIT Press.