



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Pengambilan Sampel

Pengambilan sampel adalah proses pemilihan sejumlah elemen dari populasi, untuk mendapatkan pemahaman tentang sifat atau karakteristik dari suatu populasi. Pengambilan sampel dilakukan untuk menyederhanakan pengumpulan data ataupun pengujian data, seperti pada penelitian yang melibatkan ratusan bahkan ribuan elemen mengingat waktu, biaya, dan sumber daya manusia yang tersedia (Sekaran, 2003). Menurut Gay yang dikutip oleh Mahmud dalam bukunya yang berjudul “Metode Penelitian Pendidikan”, ukuran minimum sampel yang diambil adalah 10% dari populasi.

2.2 Technology Acceptance Model

Menurut Simarmata (2015), Technology Acceptance Model (TAM) adalah teori yang mampu menjelaskan perilaku pengguna terhadap teknologi, penerimaan teknologi disebabkan oleh faktor *perceived ease of use*, *perceived usefulness*, dan *actual use*, yang diusulkan oleh Davis pada tahun 1986.

Perceived usefulness didefinisikan sebagai sejauh mana seseorang percaya bahwa dengan menggunakan suatu teknologi akan meningkatkan performanya.

Perceived ease of use didefinisikan sebagai sejauh mana seseorang percaya bahwa menggunakan suatu teknologi akan terbebas dari usaha fisik maupun mental (Simarmata, 2015). Pertanyaan untuk *perceived usefulness* mengacu pada pertanyaan yang meyakinkan seseorang apakah teknologi yang digunakan akan memberikan manfaat dan pertanyaan untuk *perceived ease of use* mengacu pada

pertanyaan yang meyakinkan seseorang apakah teknologi tersebut mudah digunakan (Rahmah, 2017).

2.3 Penjadwalan

Pengertian jadwal menurut Kamus Besar Bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci (Kamus Besar Bahasa Indonesia, 2012). Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan ke dalam jadwal.

Menurut Chamber (1995) bahwa jadwal didefinisikan sebagai sesuatu yang menjelaskan di mana dan kapan orang-orang dan sumber daya berada pada suatu waktu. Setiap jangka waktu memiliki daftar mata kuliah yang sedang diajarkan, oleh siapa dan di mana. Jadwal dapat dinyatakan dalam sejumlah cara yang berbeda dan masing-masing mahasiswa harus memiliki jadwal sendiri tergantung pada mata pelajaran, begitu juga masing-masing dosen.

Di perguruan tinggi, program penjadwalan merupakan salah satu hal penting dalam proses belajar mengajar, karena semua kegiatan dosen dan mahasiswa bergantung pada jadwal yang ada, sehingga harus disusun optimasi penjadwalan sehingga tidak ada jadwal dalam waktu yang bersamaan antara dosen dan mahasiswa (Ariani, dkk, 2012). Dikarenakan dalam penyusunan jadwal memiliki tantangan tersendiri yang cukup rumit agar tidak ada kegiatan pada waktu yang bersamaan (Fernandez, dkk, 2009).

2.4 Kuliah Pengganti

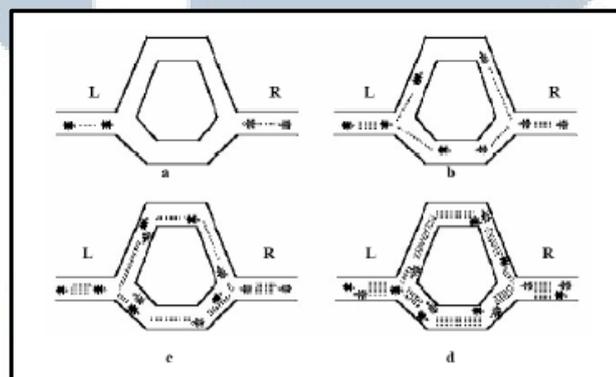
Kuliah pengganti diartikan bahwa sebelumnya sudah ada jadwal pasti pada mahasiswa dan dosen sehingga harus diatur kembali jadwal pengganti untuk menggantikan jadwal kuliah yang ditiadakan (Winatha, 2010). Dosen yang berhalangan hadir karena alasan yang dibenarkan atau alasan tertentu menurut peraturan diharuskan melapor kepada penanggung jawab mata kuliah dan menentukan solusi yaitu dosen yang berhalangan hadir memberikan jadwal kuliah pengganti pada waktu lain yang telah disepakati bersama oleh dosen dan mahasiswa (TIM SPMI UNDIP, 2011).

2.5 Ant Colony

Ant Colony pertama kali dikenalkan oleh Marco Dorigo pada tesis PhD tahun 1992. Algoritma semut lebih menitikberatkan pada perilaku kebiasaan atau yang sering dilakukan (dilewati) oleh semut. Dengan kemampuan dan keunikan semut, yang memiliki kemampuan secara alami untuk menemukan alur atau 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 algoritma *Ant Colony* (Lorena & Akbar, 2011).

1. Pada awalnya, semut berkeliling secara acak.
2. Ketika semut-semut menemukan jalur yang berbeda misalnya sampai pada persimpangan, semut akan mulai menentukan arah jalan secara acak seperti pada bagian a di Gambar 2.1.

3. Sebagian semut memilih berjalan ke atas dan sebagian lagi akan memilih berjalan ke bawah seperti pada bagian b di Gambar 2.1
4. Ketika menemukan makanan semut 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 bagian c di Gambar 2.1
6. Feromon yang ditinggalkan oleh semut di jalur yang lebih pendek aromanya akan lebih kuat dibandingkan feromon di jalur yang lebih panjang seperti pada bagian d di Gambar 2.1
7. Semut-semut lain akan lebih tertarik mengikuti semut di jalur yang lebih pendek.



Gambar 2.1 Cara kerja semut (Lorena & Akbar, 2011)

Terdapat tiga gagasan utama yang dapat diambil oleh algoritma Semut yang berasal dari *real ant* (Lorena & Akbar, 2011).

1. Semut-semut (*ants*) mempunyai suatu pilihan probabilitas atau kemungkinan untuk memilih suatu lintasan atau 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 *pheromonea edge*.

2.6 Algoritma Ant Colony Optimization

Algoritma *Ant Colony Optimization* (ACO) merupakan teknik probabilistik untuk menjawab masalah komputasi yang bisa dikurangi dengan menemukan jalur yang baik dengan graf. ACO pertama kali dikembangkan oleh Marco Dorigo pada tahun 1992. Sesuai dengan nama algoritmanya, ACO diinspirasi 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 semut, 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. Berbeda dengan hormon, feromon menyebar ke luar tubuh dan hanya dapat dikenali oleh individu lain yang sejenis. Semut dapat mencium feromon dan ketika semut memilih jalur semut, semut 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 probabilitas 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 semut. Algoritma ACO telah banyak digunakan untuk

menghasilkan penyelesaian yang mendekati optimal (Bullnheimer, dkk, 1997).

Aplikasi algoritma semut dalam kehidupan sehari-hari mencakup beberapa persoalan sebagai berikut (Leksono, 2009).

- a. *Traveling Salesman Problem* (TSP), yaitu mencari jalur terpendek dalam sebuah graf menggunakan jalur Hamilton.
- b. *Quadratic Assignment Problem* (QAP) yang berusaha menempatkan sejumlah sumber n pada sejumlah m lokasi dengan meminimalkan biaya *assignment*.
- c. *Job-shop Scheduling Problem* (JSP), juga salah satu contoh aplikasi algoritma semut untuk menjadwalkan sejumlah j pekerjaan menggunakan sejumlah m mesin sehingga seluruh pekerjaan diselesaikan dalam waktu yang minimal.

Pewarnaan graf koloni semut yang nyata dan *artificial* terdapat banyak kemiripan (Dorigo & Socha, 2006). Keduanya terbentuk dari sebuah populasi yang terdiri dari individu-individu yang berkerja sama untuk mencapai tujuan. Semut *artificial* hidup di dunia virtual, karenanya semut 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 berurut 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, semut berpindah secara sekuen melewati setiap 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 semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jalur terpendek (Lorena & Akbar, 2011), 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.
3. titik berangkat dan titik tujuan.
4. Banyak semut.
5. Tetapan penguapan jejak semut (ρ), nilai ρ dimulai dari 0 sampai dengan 1 untuk mencegah jejak feromon yang tak terhingga.

6. 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 pada poin pertama dilakukan, kemudian m semut ditempatkan pada titik pertama.

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 $tabu_k(1)$ 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 $tabu_k$ sebagai titik tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua titik satu persatu dikunjungi atau telah menempati $tabu_k$.

Langkah 4 :

a. Memperbarui 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. 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 (2.1).

$$\tau_{ij} = \rho + \tau_{ij} \quad \dots(2.1)$$

Dimana :

τ_{ij} = Intensitas jejak semut antar titik

ρ = Tetapan penguapan jejak semut

Langkah 5 :

Pengosongan *tabu list*, dan ulangi langkah 2 jika diperlukan. *Tabu list* 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. Algoritma tersebut mengacu pada seperti Gambar 2.2.

```

1. {Initialization}
   Initialize  $\tau_{ij}$ 
2. {Construction}
   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-k
   until ant k has completed its solution.
   end for
3. {Trail update}
   For each ant move (ij) do
   compute  $\tau_{ij}$ 
   update the trail matrix.
   end for
4. {Terminating condition}
   If not (end test) go to step 2

```

Gambar 2.2 Pseudocode algoritma semut (Lorena & Akbar, 2011)