BAB II
DASAR TEORI

2.1 Penjadwalan Kuliah

Penjadwalan Kuliah merupakan pengaturan penempatan waktu dan ruangan berdasarkan jumlah kuliah dan akademik sejenis, dengan memperhatikan sejumlah aturan yang berhubungan dengan kapasitas dan lokasi dari ruangan yang tersedia, waktu bebas yang diperlukan dan sejumlah aturan lain yang berkaitan dengan toleransi untuk dosen, dan hubungan antar mata kuliah khusus.

Menurut Kamus Besar Bahasa Indonesia, kuliah adalah pelajaran yg diberikan di perguruan tinggi. Terdapat berbagai istilah yang dikenal pada kegiatan kuliah (perkuliahan) di perguruan tinggi, diantaranya mahasiswa, dosen, mata kuliah, SKS, dan semester.

1. Mahasiswa adalah orang yang belajar di perguruan tinggi dan merupakan pelaku kegiatan kuliah di perguruan tinggi yang menerima pelajaran dari dosen.
2. Dosen adalah tenaga pengajar pada perguruan tinggi dan merupakan pelaku kegiatan kuliah di perguruan tinggi yang memberi pelajaran kepada mahasiswa.
3. Mata kuliah adalah satuan pelajaran di tingkat perguruan tinggi yang diajarkan oleh dosen kepada mahasiswa.
4. SKS (Satuan Kredit Semester), adalah satuan yang digunakan dalam Sistem Kredit Semester. Sistem kredit semester adalah suatu sistem penyelenggaraan perkuliahan di perguruan tinngi yang menyatakan beban studi mahasiswa, beban kerja dosen dan beban penyelenggaraan perkuliahan dengan menggunakan Satuan Kredit Semester (SKS). Besarnya satu satuan kredit semester (1 SKS) dalam kegiatan perkuliahan pada berbagai perguruan tinggi terkadang berbeda namun ada juga yang sama. Di Universitas Pendidikan Indonesia, nilai 1 SKS dalam kegiatan perkuliahan adalah setara dengan

alokasi waktu lima puluh (50) menit kegiatan tatap muka terjadwal antara dosen dengan mahasiswa.

5. Semester adalah satuan terkecil untuk lamanya masa program pendidikan. Jenjang pendidikan strata satu (S1) terdiri atas 8 semester. Satu tahun akademik (penyelenggaraan kegiatan perkuliahan dalam satu tahun) dibagi menjadi dua semester, yaitu semester ganjil dan semester genap

2.2 Algoritma *Ant colony*

2.2.1 Algoritma

Pada awalnya kata algorisma adalah istilah yang merujuk kepada aturan-aturan aritmetis untuk menyelesaikan persoalan dengan menggunakan bilangan numerik arab, namun pada abad ke-18 istilah ini telah berkembang menjadi algoritma suatu urutan langkah atau prosedur yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan.Kata algoritma berasal dari latinisasi nama seorang ahli matematika dari Uzbekistan Al Khawārizmi (hidup sekitar abad ke-9), sebagaimana tercantum pada terjemahan karyanya dalam bahasa latin dari abad ke-12 "Algorithmi de numero Indorum". Berikut ini beberapa pengertian menurut para ahli :

a. Abu Ja'far Mohammad Ibn Musa Al Khoarismi "Suatu metode khusus untuk menyelesaikan suatu persoalan."
b. Goodman Hedet Niemi "Urut-urutan terbatas dari operasi-operasi terdefinisi dengan baik, yang masing-masing membutuhkan memory dan waktu yang terbatas untuk menyelesaikan suatu masalah."
c. Donald E. Knuth "Algor itma adalah sekumpulan aturan-aturan berhingga yang memberikan sederetan operasi-operasi untuk menyelesaikan suatu jenis masalah yang khusus.

Knuth (1973) menyatakan 5 komponen utama dalam algoritma yaitu finiteness, definiteness, input, output dan effectiveness. Sehingga dalam merancang sebuah algoritma ada 3 (tiga) komponen yang harus ada yaitu:

a. Komponen masukan (input) terdiri dari pemilihan variable, jenis variable, tipe variable, konstanta dan parameter (dalam fungsi).

b. Komponen keluaran (output) merupakan tujuan dari perancangan algoritma dan program. Permasalahan yang diselesaikan dalam algoritma dan program harus ditampilkan dalam komponen keluaran. Karakteristik keluaran yang baik adalah benar (menjawab) permasalahan dan tampilan yang ramah (Frendly).
c. Komponen proses (processing) merupakan bagian utama dan terpenting dalam merancang sebuah algoritma. Dalam bagian ini terdapat logika masalah, logika algoritma (sintaksis dan semantik), rumusan, metode (rekursi, perbandingan, penggabungan, pengurangan dan lain-lain).

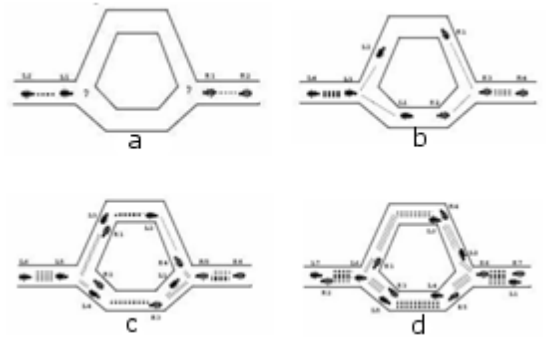
2.2.2 *Ant colony*

Ant colony pertama kali dikenalkan oleh Marco Dorigo pada tesis PhD tahun 1992. Algoritma semut lebih menitik beratkan pada perilaku kebiasaan atau yang sering dilakukan (dilewati) oleh semut. Dengan kemampuan dan keunikan semut, yang memiliki kemampuan secara alami (riil ant) untuk menemukan alur/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* :

1. Pada awalnya, semut berkeliling secara acak.
2. Ketika semut-semut menemukan jalur yang berbeda misalnya sampai pada persimpangan, mereka akan mulai menentukan arah jalan secara acak seperti Gambar 2.1.a.
3. Sebagian semut memilih berjalan ke atas dan sebagian lagi akan memilih berjalan ke bawah seperti Gambar 2.1.b.
4. Ketika menemukan makanan mereka 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 Gambar 2.1.c.

6. *Feromon* yang ditinggalkan oleh semut di jalur yang lebih pendek aromanya akan lebih kuat dibandingkan *feromon* di jalur yang lebih panjang seperti Gambar 2.1.d.
7. Semut-semut lain akan lebih tertarik mengikuti



Gambar 2.1 Cara Kerja Semut

Terdapat tiga gagasan utama yang dapat diambil oleh Algoritma Semut yang berasal dari riil ant:

1. Semut-semut (ants) mempunyai suatu pilihan probabilistic atau kemungkinan untuk memilih suatu lintasan/alur dengan nilai pheromone yang tinggi.
2. Alur yang lebih pendek cenderung untuk suatu tingkat pertumbuhan nilai pheromone yang lebih tinggi.
3. Dengan menggunakan suatu system komunikasi tak langsung melalui phromonea edge.

2.2.3 *Ant colony* Optimazation

Algoritma *Ant colony Optimization* 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 1991. Sesuai dengan nama algoritmanya, ACO di inspirasi 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 mereka, 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, individu lain, kelompok, dan untuk membantu proses reproduksi. Berbeda dengan hormon, *feromon* menyebar ke luar tubuh dan hanya dapat mempengaruhi dan dikenali oleh individu lain yang sejenis, proses peninggalan *feromon* ini dikenal sebagai stigmergy. Semut dapat mencium *feromon* dan ketika mereka memilih jalur mereka, mereka 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 probabilitik 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 mereka.

Algoritma ACO telah banyak digunakan untuk menghasilkan penyelesaian yang mendekati optimal [Bernd Bullnheimer et al., 1997, p.1]. Aplikasi algoritma semut dalam kehidupan sehari-hari mencakup beberapa persoalan sebagai berikut.

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 ditempatkan 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.
d. Vehicle Routing Problem (VRP), pengaturan jalur kendaraan
e. Pewarnaan graf

Koloni semut yang nyata dan artificial terdapat banyak kemiripan [Marco Dorigo et al., 2006, p.3]. Keduanya terbentuk dari sebuah populasi yang terdiri dari individu-individu yang berkerja sama untuk mencapai tujuan. Semut artificial hidup di dunia virtual, karenanya mereka 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, mereka berpindah secara sekuen melewati set batasan dari permasalahan.
b. Update *feromon* (penumpukan dan penguapan *feromon*) tidak dilakukan dengan jalan yang sama pada semut yang nyata dan semut arficial. 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, yaitu:

Langkah 1 :

a. Inisialisasi harga parameter-parameter algoritma.

Parameter-parameter yang diinisialisasikan adalah :

1. Intensitas jejak semut antar titik dan perubahannya (Ƭij).
2. Banyak titik (n) termasuk koordinat (x,y) atau jarak antar titik (dij).
3. titik berangkat dan titik tujuan.
4. Tetapan siklus semut (Q).

5. Tetapan pengendali intensitas jejak semut (α), nilai $\alpha \geq 0$.
6. Tetapan pengendali visibilitas (β), nilai $\beta \geq 0$.
7. Visibilitas antar wilayah = 1/dij (ɳij).
8. Banyak semut (m).
9. Tetapan penguapan jejak semut (ρ) , nilai ρ harus > 0 dan < 1 untuk mencegah jejak pheromone yang tak terhingga.
10. Jumlah siklus maksimum (NCmax) bersifat tetap selama algoritma dijalankan, sedangkan Ʈij akan selalu diperbaharui nilainya pada setiap siklus algoritma mulai dari siklus pertama (NC=1) sampai tercapai jumlah siklus maksimum (NC=NCmax) atau sampai terjadi konvergensi.

b. Inisialisasi titik pertama setiap semut.

Setelah inisialisasi Ʈij dilakukan, kemudian m semut ditempatkan pada titik pertama tertentu secara acak.

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 tabuk (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 tabuk sebagai titik tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua titik satu persatu dikunjungi atau telah menempati tabuk. Jika s menyatakan indeks urutan kunjungan, titik asal dinyatakan sebagai tabuk(s) dan titik-titik lainnya dinyatakan sebagai {N-tabuk},

maka untuk menentukan titik tujuan digunakan persamaan probabilitas titik untuk dikunjungi sebagai berikut :

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\Sigma_{k' e\{Ntabu_k\}}[\tau_{ik'}]^{\alpha}[\eta_{ik'}]^{\beta}}$$ Untuk j {*N-tabu*$_k$} ..... (1)

$P_{ij}^{k} = 0$ Untuk j lainnya ..... (2)

dengan i sebagai indeks titik asal dan j sebagai indeks titik tujuan.

Langkah 4 :

a. Perhitungan panjang rute setiap semut.

Perhitungan panjang rute tertutup (length closed tour) atau Lk setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan ini dilakukan berdasarkan tabuk masing-masing dengan

persamaan berikut :

$Lk = dtabu_k(n), tabu_k(1) + \Sigma_{s=1}^{n=1}\ dtabu_k(s), tabu_k(s-1)$ .....(3)

dengan dij adalah jarak antara titik i ke titik j yang dihitung berdasarkan persamaan :

$$dij = \sqrt{(xi - xj)^2 + (yi - yj)^2}$$.....(4)

b. Pencarian rute terpendek.

Setelah Lk setiap semut dihitung, akan didapat harga minimal panjang rute tertutup setiap siklus atau LminNC dan harga minimal panjang rute tertutup secara keseluruhan adalah atau Lmin.

c. Perhitungan perubahan harga 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. Persamaan perubahan ini adalah :

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$ ..... (5)

Dengan $\Delta\tau_{ij}^{k}$ adalah perubahan harga intensitas jejak kaki semut antar titik setiap semut yang dihitung berdasarkan persamaan:

$\Delta\tau_{ij}^{k} = \frac{Q}{L_k}$, untuk (i,j) kota asal dan titik tujuan dalam *tabu*$_k$ .....(6)

$\Delta\tau_{ij}^{k}$ untuk (i,j) lainnya ……...…….. (7)

Langkah 5 :

a. Perhitungan harga intensitas jejak kaki semut antar titik untuk siklus selanjutnya. Harga intensitas jejak kaki semut antar titik pada semua lintasan antar titik ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewati. 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

$\tau_{ij} = \rho . \tau_{ij} + \Delta \tau_{ij}$ .....(8)

b. Atur ulang harga perubahan intensitas jejak kaki semut antar titik. Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar titik perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6 :

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. Penampilan algoritma semut disesuaikan pada beberapa parameter yaitu: α, β, lintasan penting yang dilalui serta menarik, ρ, persisteansi lintasan, Ʈij(0), tingkat identitas lintasan, m, banyaknya jumlah semut, dan Q. Parameter-parameter tersebut digunakan untuk memperjelas solusi-solusi yang baik serta dengan perhitungan yang minimum.

Algoritma tersebut mengacu pada:

1. {Initialization}

   Initialize Ʈij and ɳij, Ɣ (ij).

2. {Contruction}
   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
   tabuk.
   until ant k has completed its solution.
   end for
3. {Trail update}
   For each ant move (ij) do
   compute Ţij
   update the trail matrix.
   end for
4. {Terminating condition}
   If not(end test)go to step 2

2.3 SQL Server

SQL Server adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur client/server. Istilah client, server, dan client/server dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada level yang sangat umum, sebuah client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (resource) dari komponen sistem lainnya. Sedangkan sebuah server adaah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

Secara garis besar, SQL Server mempunyai 3 (Tiga) jenis Transact SQL yaitu :

a. Data Definition Language (DDL), merupakan bagian dari sistem manajemen database yang dipakai untuk mendefinisikan dan mengatur semua atribut dan properti dari sebuah database

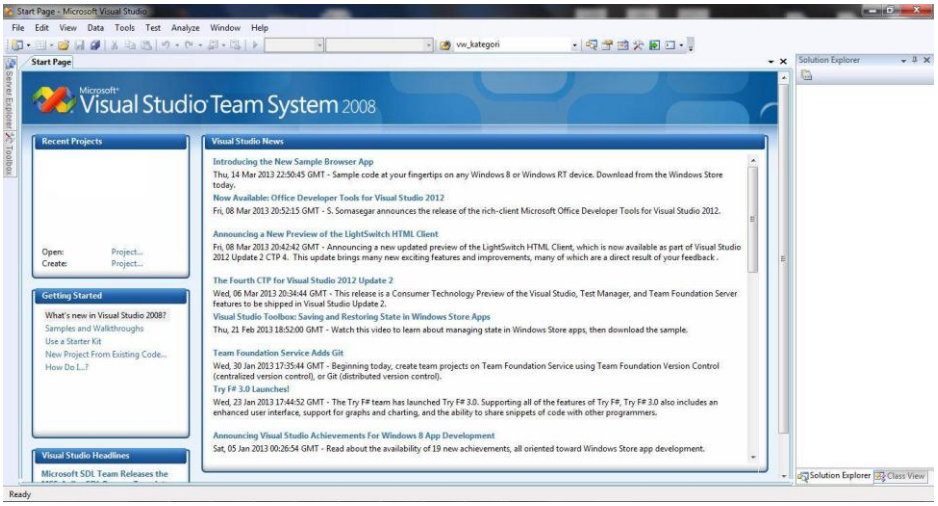
b. Data Manipulation Language (DML), merupakan perintah – perintah yang digunakan untuk menampilkan, menambah, mengubah, dan menghapus data di dalam obyek – obyek yang didefinisikan oleh DDL.
c. Data Control Language (DCL), digunakan untuk mengontrol hak – hak pada obyek – obyek database.

2.4 VB.net

Microsoft **Visual Basic .NET** adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa *BASIC*. Dengan menggunakan alat ini, para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi command-line, VB.NET tampil dengan wajah yang sama sekali berbeda dari VB6 dalam hal kaidah pemrograman, terutama dengan fasilitas strong typing dan code safety. Disamping itu, sifat .NET Framework yang dirancang dengan nuansa OOP juga harus diikuti, sehingga VB.NET dapat dikatakan sebagai full OOP programming.

Tujuan umum dari VB.NET, yaitu :

1. Menciptakan aplikasi distribusi tinggi. Tren dalam aplikasi bisnis lebih mengarah pada model terdistribusi tingkat tinggi. Aplikasi pada generasi yang akan datang akan mempunyai elemen masing-masing yang terdistribusi diantara organisasi yang bermacam-macam.
2. Menyederhanakan pengembangan *software*.
3. Interface pengguna yang lebih baik daripada web.
4. Mendukung bahasa pemrograman yang bervariasi.
5. Platform yang efektif di masa yang akan dating



Gambar 2.2 Tampilan VB.net