Jurnal TIMES , Vol. V No 1 : 17-19 , 2016
ISSN : 2337 - 3601

PENERAPAN SIRKUIT HAMILTON UNTUK MENENTUKAN RUTE TERPENDEK PERJALANAN SALESMAN PT HEALTH WEALTH INTERNATIONAL (HWI)

Edi Wijaya[1] Vera[2]
Sekolah Tinggi Manajemen Informatika Dan Komputer TIME[1,2]
Jl. Merbabu No. 32 AA – BB Medan[1,2]
e-mail : wiwileosummer@gmail.com[1]

Abstrak

Penelitian ini bertujuan untuk menganalisis dan merancang aplikasi rute perjalanan salesman dengan Sirkuit Hamilton pada PT. Health Wealth International yang mencakup pembelian, penjualan, dan rute perjalanan. Metodologi yang digunakan untuk melakukan proses analisis dan perancangan pada penelitian ini adalah metode *Breadth First Search* (BFS). Tools yang digunakan untuk melakukan analisis dan desain adalah *Data Flow Diagram* (DFD). Hasil dari penelitian ini adalah Aplikasi Rute Perjalanan dengan Sirkuit Hamilton pada PT Health Wealth International yang terkomputerisasi yang dapat digunakan untuk menyediakan informasi rute perjalanan yang berguna dalam kegiatan pengiriman barang perusahaan.

Kata Kunci : Sirkuit Hamilton, Sirkuit Euler, Graf, Travelling Salesperson Problem (TSP)

1. Pendahuluan

Dalam kehidupan sehari-hari, sering diperlukan efisiensi dalam pekerjaan terutama dalam hal meningkatkan kinerja dari karyawan sehingga dapat meningkatkan bisnis perusahaan. Pembahasan dalam penelitian ini mengangkat topik mengenai bagaimana seorang *salesperson* atau yang biasa dikenal dengan nama *salesman*, melaksanakan tugasnya dengan mengunjungi semua kota hanya sekali dan mencari jarak terdekat dari satu kota ke kota-kota yang lain. Nama persoalan ini adalah *Travelling Salesperson Problem* (TSP) yang diilhami oleh masalah seorang pedagang yang berkeliling mengunjungi sejumlah kota. Deskripsi persoalannya adalah sebagai berikut: diberikan sejumlah kota dan jarak antar kota. Jalur yang menghubungkan dua buah kota adalah jalur dua arah. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan. Permasalahan yang dihadapi oleh *salesperson* tersebut adalah bagaimana meminimalkan jarak tempuh yang digunakan dalam mengunjungi setiap kota tersebut dengan sebuah kota sebagai tempat keberangkatannya.

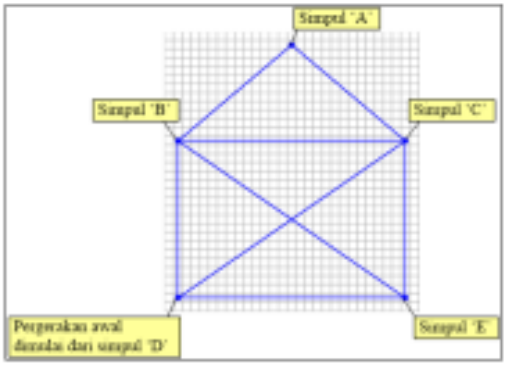
Dalam permasalahan TSP, kota dapat dinyatakan sebagai simpul graf sedangkan sisi menyatakan jalan yang menghubungkan antar dua buah kota. Bobot pada sisi menyatakan jarak antara dua buah kota. Persoalan perjalanan pedagang adalah menentukan sirkuit Hamilton yang memiliki bobot minimum pada sebuah graf terhubung. Lintasan Hamilton adalah lintasan yang melalui tiap simpul di dalam graf tepat satu kali. Bila lintasan itu kembali ke simpul asal membentuk lintasan tertutup (sirkuit), maka lintasan tertutup itu dinamakan sirkuit Hamilton.

2. Metode Penelitian

Metode proses pencarian sirkuit Hamilton dapat dicari dengan menggunakan metode *Breadth First Search* (BFS) dan *Depth First Search* (DFS).

Metode Breadth First Search

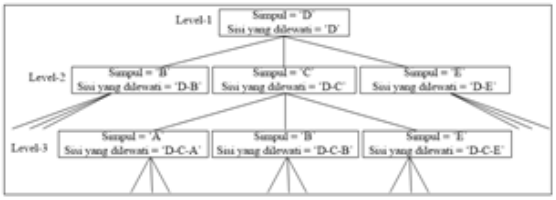
Proses kerja dari sistem akan dimulai dari penentuan data nama dan posisi *node* (keadaan). Proses ini akan dilakukan secara acak oleh komputer. Pencarian dengan metode BFS atau disebut juga pencarian melebar pertama dimulai dari *node* akar (keadaan awal) terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Demikian seterusnya hingga ditemukannya solusi. Setiap *node* dalam pohon pencarian adalah berupa simpul yang sedang ditempati pada keadaan itu dan sisi-sisi yang telah ditelusuri sebelumnya. Agar lebih jelas, perhatikan gambar 1 berikut ini.



Gambar 1 Contoh Lintasan Yang Akan Dicari Solusinya

Sesuai dengan gambar 1, maka hubungan antar simpul yang terdapat dalam lintasan tersebut adalah (A – B), (A – C), (B – C), (B – D), (B – E), (C – E), (C – D) dan (D – E). Prosedur pencarian BFS akan dimulai dari simpul ’D’ sebagai *node* akar atau keadaan awal. Kemudian, dari *node* akar dikembangkan semua *node* anak yang menghubungkan simpul ’D’ dengan simpul yang lain. Dalam hal ini adalah hubungan (B – D), (C –

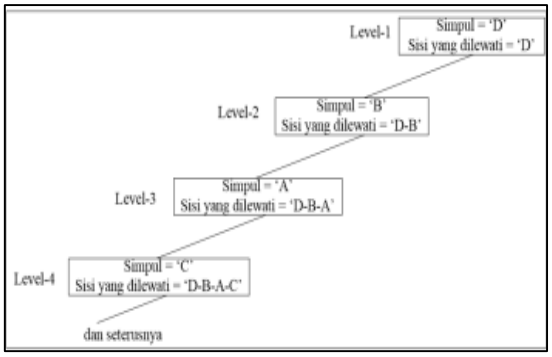
D) dan (D – E), sehingga *node* anak pada level-2 menempati simpul B, C dan E. Demikian seterusnya, hingga semua sisi dalam graf dilewati. Pencarian tidak akan melewati sisi yang telah dilewati sebelumnya. Prosedur pencarian BFS dapat dilihat pada gambar 2 berikut.



Gambar 2 Prosedur Pencarian BFS

Metode Depth First Search

Pencarian dengan metode DFS atau disebut juga pencarian mendalam pertama dimulai dari *node* akar. Pada metode ini, proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan pencarian ke *node-node* yang selevel. Dari *node* akar (level-1), dikembangkan satu *node* anak (level-2), kemudian dari *node* baru pada level-2 dikembangkan lagi *node* anak pada level berikutnya (level-3). Demikian seterusnya hingga ditemukannya solusi. Jika tidak ada langkah yang dapat ditempuh pada *node* yang sedang diperiksa, maka pencarian akan kembali ke *node* induk untuk dikembangkan lagi *node* anak lainnya. Setiap *node* dalam pohon pencarian adalah berupa simpul yang sedang ditempati pada keadaan itu dan sisi-sisi yang telah ditelusuri sebelumnya. Agar lebih jelas, prosedur pencarian DFS untuk menyelesaikan graf dapat dilihat pada gambar 3 berikut.



Gambar 3 Prosedur Pencarian DFS

Tidak seperti pencarian BFS, pencarian DFS tidak akan mengembangkan semua *node* anak yang mungkin dari suatu *node*. Pencarian DFS hanya akan mengembangkan satu *node* anak dan pencarian berlanjut ke *node* anak yang baru terbentuk. Sesuai gambar 4.6, prosedur pencarian DFS akan dimulai dari simpul ’D’ sebagai *node* akar atau keadaan awal. Kemudian, dari *node* akar dikembangkan satu *node* anak yang menghubungkan simpul ’D’ dengan simpul ’B’, kemudian dari *node* baru tersebut dikembangkan lagi node anak yang menghubungkan simpul ’B’ dengan simpul ’A’. Demikian seterusnya, hingga semua sisi dalam graf dilewati. Bila tidak ada sisi yang dapat dilewati lagi dan solusi belum ditemukan, maka pencarian mundur satu langkah ke *node* induk. Pencarian tidak akan melewati sisi yang telah dilewati sebelumnya.

Pada penelitian ini, akan digunakan metode BFS untuk melakukan pencarian sirkuit Hamilton dari sebuah *graph*. Hal ini dikarenakan proses pencarian dengan menggunakan metode DFS akan memerlukan waktu yang lebih lama jika dibandingkan dengan metode BFS. Berikut dijabarkan algoritma yang digunakan untuk mencari sirkuit Hamilton dengan metode BFS:

Algoritma BFS ini digunakan untuk mencari solusi atau penyelesaian dari sirkuit atau lintasan dengan metode BFS. Metode BFS akan memeriksa dan mengembangkan setiap *node* pada setiap level. Pencarian dimulai dari *node* akar (keadaan(1)), kembangkan anak dari *node* tersebut, ke level berikutnya, periksa dan kembangkan *node* anak dari kiri ke kanan, ke level berikutnya, dari kiri ke kanan dan seterusnya hingga ditemukan solusi. Solusi dalam permasalahan ini adalah suatu keadaan dimana setiap sisi di dalam graf telah dilewati satu kali.

Algoritma pencarian BFS adalah sebagai berikut

1. Set N =1, IndeksGoal = 0 dan Selesai = *False*.
2. Selama Selesai = *False* dan bStop = *False*, lakukan prosedur di bawah ini:
   a. Untuk i = 1 sampai jumlah array dari variabel ‘Hubungan’,
      i. A = Split(Hubungan(i), “,”).
      ii. Jika Keadaan(N).Titik = A(0) maka periksa jika sisi belum pernah ditelusuri (InStr(1, Keadaan(n).Sisi, "," & Keadaan(n).Titik & "," & A(1) & ",") = 0 dan InStr(1, Keadaan(n).Sisi, "," & A(1) & "," & Keadaan(n).Titik & ",") = 0), maka:
         1) Set j = UBound(Keadaan) + 1.
         2) ReDim Preserve Keadaan(j).
         3) Set Keadaan(i).Titik = A(1), .Sisi = Keadaan(n).Sisi & A(1) & ",", .Induk = n, .Level = Keadaan(n).Level + 1 dan ReDim .Anak(0).
         4) k = UBound(Keadaan(n).Anak) + 1.
         5) ReDim Preserve Keadaan(n).Anak(k).
         6) Keadaan(n).Anak(k) = j.
         7) Periksa apakah telah mencapai goal State. Set S = Split(Keadaan(j).Sisi, ","). Periksa jika UBound(S) - 2 = UBound(Hubungan) maka set Selesai = True, IndeksGoal = j dan keluar dari perulangan.
      iii. Jika Keadaan(N).Titik = A(1) maka periksa jika sisi belum pernah ditelusuri (InStr(1, Keadaan(n).Sisi, "," & Keadaan(n).Titik & "," & A(0) & ",") = 0

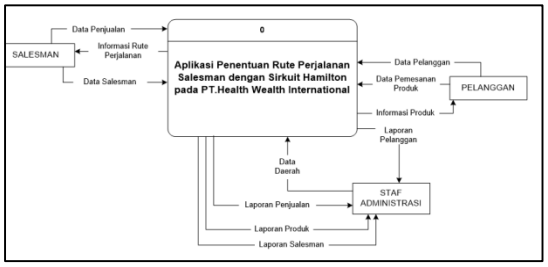
dan InStr(1, Keadaan(n).Sisi, "," & A(0) & "," & Keadaan(n).Titik & ",") = 0), maka:

1) Set j = UBound(Keadaan) + 1.
2) ReDim Preserve Keadaan(j).
3) Set Keadaan(i).Titik = A(0), .Sisi = Keadaan(n).Sisi & A(0) & ",", .Induk = n, .Level = Keadaan(n).Level + 1 dan ReDim .Anak(0).
4) k = UBound(Keadaan(n).Anak) + 1.
5) ReDim Preserve Keadaan(n).Anak(k).
6) Keadaan(n).Anak(k) = j.
7) Periksa apakah telah mencapai goal State. Set S = Split(Keadaan(j).Sisi, ","). Periksa jika UBound(S) - 2 = UBound(Hubungan) maka set Selesai = True, IndeksGoal = j dan keluar dari perulangan.

b. Periksa dan kembangkan keadaan berikutnya, set N = N +1.

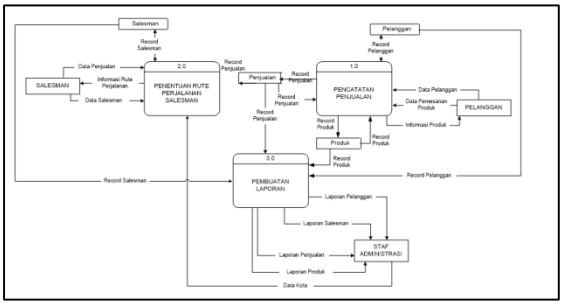
3. Jika bStop = *False* (*user* tidak menekan tombol 'Stop'), maka:
   a. ReDim Temp(0).
   b. Set j = IndeksGoal.
   c. Telusuri dari *node* solusi ke *node* akar. Selamat Keadaan(j).Induk > = 0, maka set i = UBound(temp) + 1, ReDim Preserve Temp(i), temp(i) = j dan j = Keadaan(j).Induk.
   d. Redim Solusi(UBound(temp)).
   e. Untuk i = UBound(temp) sampai 1, step -1 maka set Solusi(i) = temp(UBound(temp) - i + 1).
   f. Munculkan pesan "Solusi berhasil ditemukan !"
4. Jika bStop = True (user menekan tombol 'Stop'), maka munculkan pesan "Pencarian dihentikan !".

3. Perancangan DFD



Gambar 4 Diagram Konteks logika

Dari diagram konteks pada gambar 4 diatas, terlihat bahwa dalam sistem informasi penjualan pada perusahaan terdapat tiga buah entitas yaitu *sales*, pelanggan dan manajer operasional. Dari diagram konteks tersebut, dapat dirincikan lebih lanjut proses-proses yang terlibat dalam sistem seperti terlihat pada DFD Level 0 pada gambar 5 berikut ini:



Gambar 5 DFD Level 0

Dari gambar 5 di atas, terlihat bahwa terdapat tiga proses yang terdapat dalam DFD level 0 sistem informasi operasional pada perusahaan yang sedang berjalan, yaitu Penjualan, Pengaturan Rute Perjalanan Sales dan Pembuatan Laporan.

Agar dapat memperoleh informasi mengenai semua proses transaksi yang terjadi, maka data transaksi tersebut akan diolah dan dibuat menjadi bentuk laporan.

4. Hasil

Hasil yang didapat dari evaluasi ini adalah bahwa pencarian menggunakan metode BFS lebih efektif dan lebih cepat dibanding dengan menggunakan metode DFS. Metode ini akan memeriksan dan mengembangkan setiap node pada setiap levelnya. Dan tentunya dengan adanya rancangan ini akan memberikan iformasi mengenai rute perjalanan salesman (TSP) yang paling efisian dan singkat walaupun ada rute perjalanan lainnya.

5. Daftar Pustaka

[1] Edhy Sutanta,, 2011, Basis Data dalam Tinjauan Konseptual , CV.Andi Offset, Yogyakarta
[2] Erick Kurniawan , 2011, Cepat Mahir Visual Basic 2010, Andi Publisher.
[3] Jubilee Enterprise , 2014, Buku Pintar Database dengan Ms Access, Penerbit Elex Media Komputindo, Jakarta.
[4] Herry Raditya Wibowo , Jubilee Enterprise 2014, Visual Basic Database, Penerbit Elex Media Komputindo, Jakarta.
[5] Junindar, 2008, Panduan Lengkap Menjadi Programmer Membuat Aplikasi Penjualan Menggunakan VB.Net. Cetakan ke-3. Media Kita, Jakarta.
[6] Dr. Eng. Admi Syarif , 2015, Algoritma Genetika; Teori dan Aplikasi , Edisi 2, Penerbit Graha Ilmu.