PENGAPLIKASIAN GRAF DALAM KEHIDUPAN SEHARI-HARI

Nurio Juliandatu Masido – NIM : 13505083

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15083@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang cara pengaplikasian graf dalam kehidupan sehari-hari dan juga memberikan beberapa contoh pengaplikasian serta memberi gambaran betapa bergunanya teori graf dalam kehidupan sehari-hari.

Di dalam buku Diktat Kuliah Matematika Diskrit yang disusun oleh Bapak Rinaldi Munir, M.T. disebutkan bahwa Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Ada banyak sekali contoh penggunaan graf di dalam kehidupan contohnya saja dalam pembuatan peta, dimana satu kota dihubungkan dengan kota lain apabila terdapat jalan atau sarana transportasi yang menghubungkan kedua kota tersebut. Selain itu juga graf dapat kita temukan dalam visualisasi silsilah keluarga yang menggunakan pohon keturunan. Pohon merupakan salah satu contoh graf khusus.

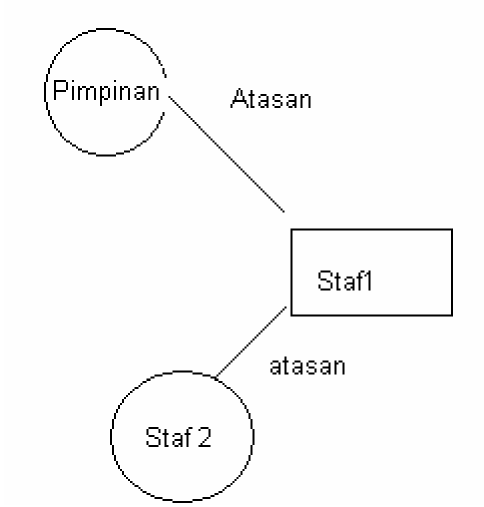
Adapun dalam makalah ini sangat ditekankan pengaplikasian graf dalam kehidupan sehari-hari sehingga terdapat banyak contoh pengaplikasian graf yang menarik untuk diaplikasikan dalam kehidupan sehari-hari.

***Kata kunci**: graf, aplikasi graf, pohon, algoritma, simpul, sisi, optimasi, keputusan, pengambilan, diskrit, matematika, vertex, node, edge, kehidupan, aras, lintasan, path, euler, sirkuit, bobot, Djikstra.*

1. Pendahuluan

Graf didefinisikan sebagai pasangan himpunan (V,E), yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (verticers atau node) = {v1, v2, v3,...} dan E himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul {e1, e2, e3, ...} atau dapat di tulis dengan notasi G = {V,E}.

Dengan definisi demikian graf dapat digunakan berbagai objek diskrit, terutama graf sering digunakan untuk memodelkan berbagai persoalan untuk memudahkan penyelesaiannya. Misalnya seseorang ingin menggambarkan diagram hubungan relasi kerja seorang pimpinan dengan staf-stafnya, maka sang pimpinan dapat dijadikan suatu objek diskrit (simpul/vertex), demikian juga staf-stafnya, dan akan terdapat sisi-sisi (edges) yang menghubungkan satu dan lainnya untuk menggambarkan hubungan (relationship) antara objek-objek (simpul) tadi. Seperti terlihat pada gambar 1, dimana seorang Pimpinan membawahi Staf1, dan Staf2 dibawahi Staf 1. Dari sini dapat dilihat kekuatan graf dalam mendeskripsikan objek-objek diskrit sehingga graf sampai saat ini banyak digunakan.



Gambar 1 Relasi dengan graf

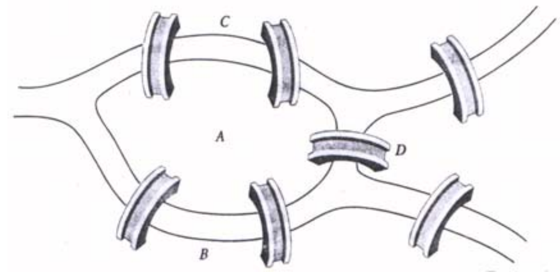
Dengan kekuatannya ini graf merupakan salah satu cabang penting dalam matematika yang terus dikembangkan terutama dalam ilmu komputer dimana dengan graf dapat merepresentasikan banyak sekali model persoalan.

2. Dasar-dasar Pengaplikasian Graf

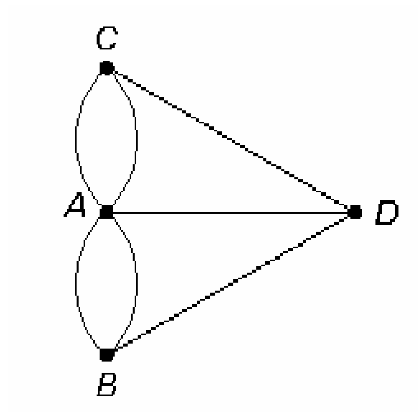
Graf digunakan dalam kehidupan sehari-hari terutama untuk mendeskripsikan model persoalan dan menggambarkannya secara konkret dan jelas. Selain itu graf juga dipergunakan untuk mempermudah menyelesaikan berbagai macam persoalan-persoalan yang sulit diselesaikan dengan perhitungan dan pertimbangan biasa. Hal ini disebabkan sifat-sifat dan teori-teori yang telah dipelajari pada graf.

2.1 Sejarah Graf

Menurut catatan sejarah, graf pertama kali digunakan dalam menyelesaikan permasalahan jembatan Königsberg (1736).



Gambar 2.a Jembatan Königsberg



Gambat 2.b Graf yang merepresentasikan jembatan Königsberg

Masalah jembatan Königsberg ini adalah : mungkinkah melalui ketujuh buah jembatan itu masing-masing tepar satu kali, dan kembali lagi ke tempat semula? Kemudian tahun 1736 seorang matematikawan Swiss, L.Euler, adalah orang pertama yang berhasil menemukan jawaban maslah itu dengan memodelkan masalah ini ke dalam graf. Daratan (titik-titik yang dihubungkan oleh jembatan) dinyatakan sebagai titik (noktah) yang disebut **simpul** (*vertex*) dan jembatan dinyatakan sebgai garis-garis yang disebut **sisi** (edge). Setiap titik diberi label huruf A, B, C, dan D. Graf yang dibuat Euler seperti tampak pada gambar 2.b.

Euler mengungkapkan bahwa tidak mungkin seseorang berjalan melewati tepat satu kali masing-masing jembatan dan kembali lagi ke tempat semula karena pada graf model jembatan Königsberg itu tidak semua simpul berderajat genap (derajat sebuah simpul adalah jumlah sisi yang bersisian dengan simpul yang bersangkutan).

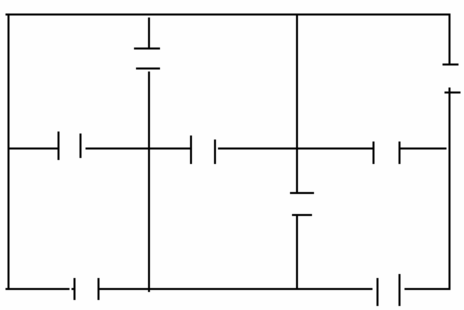
Apabila sebuah graf dapat dilalui setiap sisinya masing-masing satu kali dan kembali lagi ke tempat semula, maka graf tersebut dikatakan memiliki sirkuit Euler.

Sampai saat ini graf telah berkembang penggunaannya menjadi jauh lebih luas.

2.2 Bagaimana Membangun Graf Berdasarkan Permasalahan Yang dihadapi

Selanjutnya akan dibahas tentang cara-cara menggunakan graf dalam berbagai persoalan dan permasalahan untuk mempermudah dan memperjelas permasalahan yang sedang dihadapi.

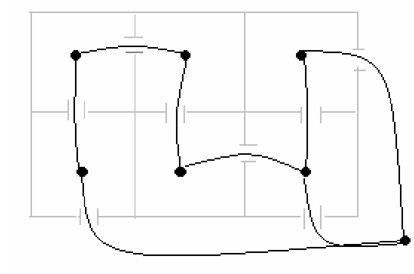
Inti dari cara pengaplikasian graf ini adalah bagaimana kita bisa membaca permasalahan, kemudian mendefinisikan apa yang akan menjadi objek diskrit yang kemudian akan menjadi simpul-simpul dari graf yang akan kita bangun untuk menggambarkan permasalahan yang kita hadapi tadi, apabila telah kita dapatkan simpul-simpul maka akan mudah bagi kita untuk membangun graf dengan memberi sisi pada simpul-simpul yang saling berhubungan.



Gambar 2.c Persoalan Rumah

Misalkan gambar di atas adalah denah lantai dasar sebuah gedung. Apakah dimungkinkan berjalan melalui setiap pintu di lantai itu hanya satu kali saja jika kita boleh mulai memasuki pintu yang mana saja?

Kita tidak akan menyelesaikan permasalahan tersebut di bahasan ini, tapi hanya untuk membangun graf yang memodelkan permasalahan ini. Maka pertama-tama kita harus melihat dahulu apakah permasalahan ini dapat dimodelkan menjadi graf, ternyata setiap ruangan termasuk bagian luar rumah dapat dijadikan simpul, sehingga setiap pintu akan menjadi sisi-sisi yang menghubungkan setiap simpul dalam graf yang akan kita bentuk, maka graf yang akan terbentuk adalah seperti terlihat dalam gambar 2.d berikut ini.



Gambar 2.d Graf dari Persoalan Rumah

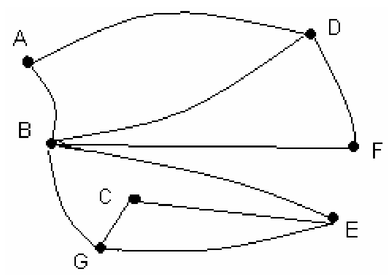
Contoh yang lain misalnya ada 6 jenis zat kimia yang perlu disimpan di gudang. Beberapa pasang dari zat tersebut tidak dapat disimpan di tempat (ruangan) yang sama, karena campuran gasnya bersifat eksplosif (mudah meledak). Untuk zat-zat semacam itu perlu dibangun ruang-ruang yang terpisah yang dilengkapi ventilasi dan penyedot udara keluar yang berlainan. Jika lebih banyak ruangan yang digunakan maka akan lebih banyak biaya yang diperlukan untuk penyimpanan. Karena itu perlu diketahui jumlah minimum ruangan yang diperlukan untuk menyimpan zat-zat kimia tersebut. Berikut daftar zat-zat yang tidak dapat di simpan dalam satu ruangan.

Zat Kimia	Tidak Dapat disimpan bersama
A	B,D
B	A,D,E,F,G
C	E,G
D	A,F,B
E	B,C,G
F	B,D
G	B,C,E

Tabel 1 Distribusi Reaktif Zat-Zat Kimia

Disini kita tentu saja hanya mencoba menggambarkan graf yang menyatakan persoalan ini. Dapat kita lihat dengan jelas bahwa setiap zat kimia A,B,C,D,E,F. dan G dengan sendirinya menjadi simpul-simpul graf. Kemudian akan kita gambarkan sisi-sisi yang menggambarkan hubungan eksplosif, yang artinya saling menyebabkan campuran yang bersifat eksplosif.

Dengan begitu hasil graf akan tampak seperti gambar 2.e di bawah ini.



Gambar 2.e Graf Zat Kimia

Pada graf tersebut tampaklah bahwa apabila ada sisi antara dua buah simpul maka zat-zat kimia yang diwakili oleh simpul-simpul tadi tidak dapat disimpan dalam satu ruangan yang sama.

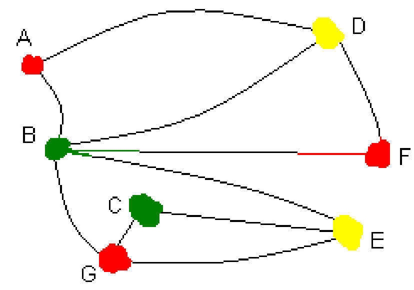
2.3 Penyelesaian Masalah Dengan Graf

Bagaimana menyelesaikan sebuah masalah dengan menggunakan graf. Setelah menggambarkan dan memvisualisasikan permasalahan dengan graf maka untuk menyelesaikan permasalahan yang dihadapi kita juga dapat menggunakan berbagai teori dan sifat-sifat graf. Tentu untuk menyelesaikannya kita harus mempunyai pandangan bagaimana persoalan yang dihadapi ini secara keseluruhan. Setelah itu kita dapat mencari penyelesaiannya dengan menggunakan sifat-sifat dan teori-teori mengenai graf.

Untuk menyelesaikan persoalan rumah pada sub-bahasan 2.2 jelas bahwa akan dipergunakan teorema mengenai sirkuit dan lintasan euler, yaitu sebuah graf memiliki sirkuit euler apabila semua simpul yang berada pada graf tersebut berderajat ganjil, dan akan memiliki lintasan euler apabila terdapat tepat dua simpul yang berderajat ganjil.

Pada graf persoalan rumah gambar 2.d dapat dilihat bahwa graf tersebut memiliki enam simpul yang terdiri dari 4 simpul berderajat dua (genap) dan 2 simpul berderajat 3 ganjil, sehingga kita bisa mengambil kesimpulan bahwa dimungkinkan untuk berjalan melalui setiap pintu pada rumah tersebut masing-masing hanya satu kali, karena graf rumah tersebut memiliki lintasan euler karena memiliki tepat dua simpul yang berderajat ganjil.

Untuk permasalahan zat kimia maka kita tinggal menentukan bilangan kromatik dari graf tersebut. Yaitu dengan pewarnaan simpul. Ternyata graf tersebut dapat diwarnai dengan hanya tiga warna, seperti terlihat pada gambar 2.f.

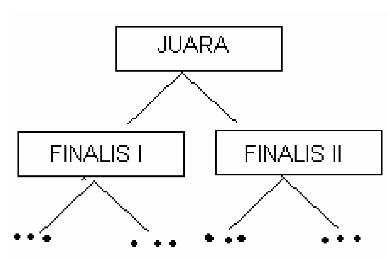


Gambar 2.f Pewarnaan Graf

Artinya cukup mengalokasikan biaya untuk membuat 3 buah ruangan untuk tempat penyimpanan zat-zat kimia tadi.

Misalkan pada suatu perlombaan terdapat 5000 orang peserta yang akan bertanding. Apabila hanya terdapat satu pemenang dan menggunakan sistem gugur, berapa buah pertandingan yang harus diadakan?

Untuk persoalan ini kita harus menentukan dahulu graf apa yang terbentuk, karena sang juara hanya satu orang dan ia pasti berasal dari pertandingan antara 2 orang, yang masing-masingnya berasal dari pertandingan antara 2 orang juga, artinya yang terbentuk adalah sebuah graf berbentuk pohon biner seperti yang terlihat pada gambar 2.g.



Gambar 2.g Pohon Biner pada Kejuaraan

Yang artinya jumlah pertandingan yang harus dilakukan adalah jumlah semua simpul yang memiliki 2 anak simpul pada pohon biner yang memiliki 5000 buah dauh (simpul yang ridak memiliki anak).

Untuk itu kita perlu membangun pohon biner tersebut terlebih dahulu, yaitu pertama-tama membuat pohon biner sampai terdapat jumlah daun yang hampir mendekati 5000. Maka pohon biner tersebut akan memiliki jumlah simpul pada tiap arasnya sebagai berikut :

- Aras 0 (Akar) = 2^0 = 1
- Aras 1 = 2^1 = 2
- Aras 2 = 2^2 = 4
- Aras 3 = 2^3 = 8
- Aras 4 = 2^4 = 16
- Aras 5 = 2^5 = 32
- Aras 6 = 2^6 = 64
- Aras 7 = 2^7 = 128
- Aras 8 = 2^8 = 256
- Aras 9 = 2^9 = 512
- Aras 10 = 2^{10} = 1024
- Aras 11 = 2^{11} = 2048
- Aras 12 = 2^{12} = 4096

Kita berhenti dahulu disini karena jumlah daun telah mendekati 5000, sampai disini kita telah memiliki $2^0 + 2^1 + .. + 2^{10} + 2^{11} = 4095$ pertandingan, dan telah ada 4096 buah daun. Untuk menjadikan jumlah daun menjadi 5000, maka akan ditambahkan sebanyak (5000 - 4096) pertandingan lagi, sehingga pada aras ke-12 terdapat 904 buah simpul yang merupakan hasil pertandingan dari 1808 peserta yang berasal dari aras ke-13.

Jumlah pertandingan yang harus dilakukan dengan menghitung jumlah semua simpul dikurangi dengan 5000 (jumlah daun) yang menyatakan semua peserta yang mengikuti kejuaraan.

Jumlah pertandingan yang harus diadakan agar terdapat seorang juara dan menggunakan sistem gugur adalah 4999 yang di dapat dengan hasil perhitungan simpul (1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 + 1024 + 2048 + 4096 + 1808 - 5000) atau dengan hasil perhitungan pertandingan 4095 + 904 pertandingan.

3. Contoh-contoh Aplikasi Graf

Sub-bahasan ini membahas tentang beberapa aplikasi graf penting dalam kehidupan sehari-hari.

3.1 Mencari Lintasan Terpendek (Shortest Path)

Persoalan ini sangat penting dalam pengoptimasian keputusan yang akan di ambil. Graf yang digunakan untuk permasalahan ini adalah graf berbobot (***weighted graph***). Terpendek disini bukan berarti bahwa kita hanya menghitung jalan terpendek yang bisa ditempuh untuk menuju satu titik dari satu titik lainnya, bisa juga terpendek ini berarti menghitung biaya minimum yang bisa dikeluarkan untuk melakukan sesuatu. Terpendek disini berarti menghitung bobot total minimum yang bisa dicapai dalam mencapai suatu tujuan dari satu titik tolak tertentu.

Sebuah graf G = (V, E) disebut sebuah graf berbobot (weight graph), apabila terdapat sebuah fungsi bobot bernilai real W pada himpunan E,

$$W : E \rightarrow R,$$

nilai W(e) disebut bobot untuk sisi e, $\forall$ e $\in$ E. Graf berbobot tersebut dinyatakan pula sebagai G = (V, E, W).

Dengan begitu akan sangat mudah merepresentasikan sebuah graf dengan menggunakan sebuah matriks yang berdimensi (simpul+1)x(simpul+1). Dengan elemen matriks berupa bobot sisi yang menghubungkan dua buah sisi. Sisi-sisi yang sama akan diisi 0, dan apabila tidak terdapat hubungan langsung antara dua buah sisi, maka akan diisi Infinity (Tak Hingga ( )).

Misal pada sebuah graf G yang terdiri dari 6 simpul A, B, C, D, E, F, dengan hubungan sekaligus bobot masing-masing

- A -> B(60), D(20), F(10)
- B -> A(60), C(25), E(35)
- C -> B(25), D(21), F(43)
- D -> A(20), C(21)
- E -> B(35)
- F -> A(10), C(43)

Maka matriks yang merepresentasikan graf G dapat berbentuk seperti di bawah ini :

Simpul	A	B	C	D	E	F
A	0	60	∞	20	∞	10
B		0	25	∞	35	∞
C			0	21	∞	43
D				0	∞	∞
E					0	∞
F						0

Tabel 2 Cara Pendeskripsian Graf dengan Matriks

Contoh-contoh penerapan pencarian lintasan terpendek misalnya :

1. Pencarian lintasan yang dilalui dari sebuah kota ke kota yang lain sehingga lintasan tersebut bernilai minimum dari semua kemungkinan lintasan yang bisa dilalui
2. Menentukan jalur komunikasi terpendek antara dua buah terminal komputer untuk menghemat waktu pengiriman pesan dan biaya komunikasi.

Ada beberapa macam persoalan lintasasn terpendek, antara lain :

a. Lintasan terpendek antara dua buah simpul tertentu
b. Lintasan terpendek antara semua simpul.
c. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.

Untuk menyelesaikan persoalan mencari lintasan terpendek ini diantaranya dapat digunakan algoritma **Dijkstra** (ditemukan Edsger Wybe Dijkstra) yang dapat digunakan pada graf berarah maupun graf tak berarah. Algoritma ini terdiri dari langkah-langkah dan menggunakan prinsip gr*eedy* dimana pada setiap langkah kita memilih sisi minimum dan memasukkan ke dalam himpunan solusi.

Ide dasar dari algoritma tersebut adalah sebagai berikut:

Misalkan dalam sebuah graf berbobot G = (V, E, W) ingin dicari suatu lintasan terpendek P(a, z) dari simpul $a \in V$ ke simpul $z \in V$. Untuk itu, dicari dahulu lintasan terpendek dari a ke suatu simpul lain b di V, P(a, b), kemudian dicari lagi lintasan terpendek P(a, c) dari a ke suatu simpul lain lagi c di V, dan seterusnya. Lalu berhenti apabila lintasan P(a, z) telah diperoleh. Caranya adalah sebagai berikut:

1) Selama proses berlangsung ditentukan dua himpunan $S \subseteq V$ dan $B \subseteq V$, dengan

   $S \cup B = V$ dan $S \cap B = \phi$, dan

   S = himpunan simpul-simpul yang sudah diketahui lintasan terpendeknya dari a

   B = himpunan simpul-simpul yang belum diketahui lintasan terpendeknya dari a

2) Apabila ada hubungan dari satu simpul ke simpul lainnya maka pada bobot sisi yang bersisian dengan kedua simpul tersebut diberi nilai sesuai dengan bobot (jarak) antara kedua simpul tersebut, jika tidak ada hubungan maka akan diberi nilai tak hingga (Infinity/∞).

3) Misalkan B adalah suatu subset dari V dan $a \notin B$, dan

   S = V - B. Lintasan terpendek dari a ke salah satu simpul di B dapat ditentukan sebagai berikut:

   Untuk setiap simpul $t \in B$, diberi label L(t) yang menyatakan panjang suatu lintasan terpendek di antara semua lintasan dari a ke t yang tidak melibatkan simpul lain di B, (boleh melibatkan simpul lain di S), L(t) disebut label untuk t terhadap S.

4) Ambil simpul t_1 di B yang memiliki label terkecil di antara semua label-label simpul di B, maka $L(t_1)$ adalah panjang lintasan terpendek dari a ke t_1

Bukti:

   Misalkan $L(t_1)$ bukan panjang lintasan terpendek dari a ke t_1, maka ada suatu lintasan $P(a, t_1)$ dari a ke t_1 yang panjangnya kurang dari $L(t_1)$. Lintasan $P(a,t_1)$ ini pasti melibatkan satu atau lebih simpul di $B_1 = B - \{t_1\}$. Andaikan u adalah simpul pertama di B_1 yang ditemui dalam lintasan $P(a,t_1)$ itu. Hal ini berarti bahwa

   $L(u) < L(t_1)$

   Kontradiksi dengan $L(t_1)$ merupakan label terkecil di antara label-label simpul di B.

5) Cara efisien untuk menentukan label baru dari label sebelumnya adalah sebagai berikut: Misalkan semua simpul di B telah memiliki label terhadap suatu S, dan x adalah simpul di B dengan label terkecil. Bentuk

   $S' = S \cup \{x\}$ dan $B' = B - \{x\}$,

   maka label L'(t) untuk simpul t di B' terhadap S' adalah

   $L'(t) = \min [L(t), L(x) + W(x, t)]$

   karena, lintasan terpendek dari a ke t tanpa melibatkan simpul lain di B' dapat merupakan

- lintasan yang tidak melibatkan x dan simpul lain di B', maka L'(t) bisa = L(t),
- lintasan yang terdiri dari suatu lintasan dari a ke x yang tidak melibatkan simpul di B' kemudian diteruskan oleh sisi dari x ke t, maka L'(t) bisa = L(x) + W(x, t).

Adapun Algoritma Dijkstra ini dapat pula dinyatakan dengan pseudo-code dan menggunakan matriks untuk representasi sebuah graf adalah sebagai berikut :

```
procedure Dijkstra (input
n:matriks, input a:integer
{simpul awal})
{Mencari lintasan terpendek dari
simpul awal a ke semua simpul
yang lainnya. Masukan matriks
ketetanggaan (m) dari graf
berbobot G dan simpul awalnya a;
Keluaran sebuah larik S yang
merupakan larik lintasan
terpendek dari a ke semua simpul
lainnya.
}
Deklarasi
      S : larik integer [1..n]
      D : larik integer[1..n]
      I  : integer
Algoritma
      {Langkah 0 - Inisialisasi}
      for  i from 1 to n
            do
            S_i ← 0
            D_i ← m_ai
      endfor

      {Langkah-1}

      Sa  ←  1  {karena simpul a
            adalah sompul asal
            lintasan terpendek,
            jadi simpul a sudah
            pasti terpilih dalam
            lintasan terpendek}
      Da ← ∞      {Tidak ada
            lintasan terpendek
            dari simpul a ke
            simpul a sendiri}

      {Langkah-2, 3, … n-1}

for i from 2 to (n-1 )
     do
      cari j sedemikian sehingga
         S_j = 0 dan Dj = min (D_1,
         D_2,  ... D_n)
      S_j  ← 1 {Simpul j sudah
         terpilih di dalam
         lintasan terpendek}
      perbarui D_i, untuk i =
         1,2,3,4,.. n
endfor
```

3.2 Pengambilan Keputusan dengan Menggunakan Pohon Keputusan

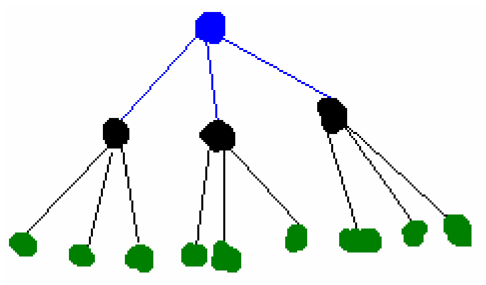
Dengan menggunakan pohon keputusan kita akan lebih mendapat banyak gambaran sebelum mengambil keputusan sehingga kita dapat mengoptimalkan keputusan yang kita ambil.

Menurut **sir Francis Bacon** dalam mengambil keputusan perlu memperhatikan langkah-langkah berikut :

1. Merumusakan/mendefinisikan masalah;
2. Pengumpulan informasi yang relevan;
3. Mencari alternatif tindakan;
4. Analisis alternatif;
5. Memilih alternatif terbaik;
6. Melaksanakan keputusan dan evaluasi hasil.

Maka penggunaan pohon keputusan dapat diaplikasikan pada langkah-langkah 1-4 dimana salah satu cara mewujudkan pohon keputusannya adalah dengan menggambarkan permasalahan yang kita hadapi sebagai pohon berakar yang akarnya adalah hal yang kita tuju (tujuan pemecahan masalah), kemudian anak dari akar ini adalah semua hal yang memungkinkan tujuan ini tercapai, demikian seterusnya sampai ke bawah, sehingga bagian paling bawah pada kondisi kita pada saat sekarang ini dan apa-apa yang dapat kita lakukan untuk mencapai langkah-langkah berikutnya.

Salah satu contoh mungkin akan tampak seperti pada gambar 3.1 dimana simpul berwarna biru ( ) adalah merupakan tujuan yang harus kita capai, sehingga simpul-simpul berwarna hitam ( ) adalah langkah-langkah yang mungkin untuk dilakukan agar tujuan kita tadi tercapai, kemudian di bawah masing-masing simpul hitam terdapat simpul berwarna hijau yang merupakan semua langkah yang harus ditempuh agar langkah berikutnya yang merupakan simpul berwarna hitam di atasnya tercapai.



Gambar 3.1 Salah Satu Contoh Pohon Keputusan

Dengan adanya pohon keputusan ini maka keputusan yang akan kita ambil pastinya akan lebih baik karena kita dapat melihat segala kemungkinan yang akan terjadi apabila kita merumuskan permasalahannya dengan baik.

3.3 Perancangan Navigasi Web dengan Graf

Dengan menggunakan graf, proses merancang struktur navigasi pada sebuah website akan terasa lebih mudah. Pertama-tama kita akan membuat halaman-halaman utama yang akan dituju langsung (direct) dari halaman index (Halaman Depan), kemudian kita harus membuat sebuah link untuk kembali ke halaman utama dari setiap halaman tersebut, demikian juga dari halaman-halaman bagian-bagian tadi, dari setiap sub-bagiannya juga harus ada link ke halaman utama dan juga ke halaman bagian asalnya tadi.

Fungsi graf di sini adalah untuk mempermudah proses perancangan website secara keseluruhan.

3.4. Persoalan Perjalanan Pedagang (TSP)

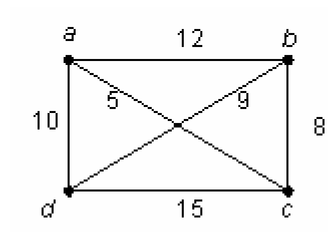
Diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui oleh seorang pedagang bila pedagang itu berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali dan kembali lagi ke kota asal keberangkatan.

==> menentukan sirkuit Hamilton yang memiliki bobot minimum.

Aplikasi TSP:

1. Pak Pos mengambil surat di kotak pos yang tersebar pada *n* buah lokasi di berbagai sudut kota.
2. Lengan robot mengencangkan *n* buah mur pada beberapa buah peralatan mesin dalam sebuah jalur perakitan.
3. Produksi *n* komoditi berbeda dalam sebuah siklus.

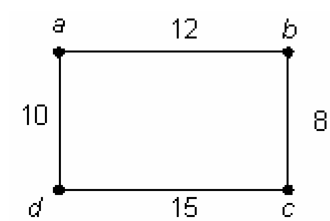
Jumlah sirkuit Hamilton di dalam graf lengkap dengan n simpul: $(n - 1)!/2$.



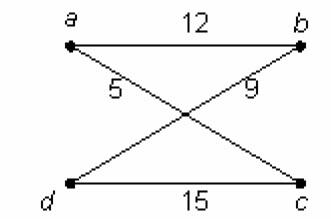
Gambar 3.2 Sirkuit Hamilton pada sebuah graf

Graf di atas memiliki $(4 – 1)!/2 = 3$ sirkuit Hamilton, yaitu:

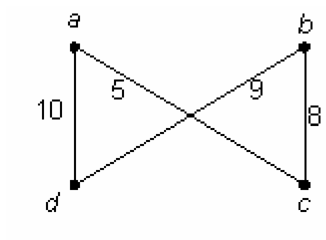
$I_1 = (a, b, c, d, a)$ atau (a, d, c, b, a) ==> panjang $= 10 + 12 + 8 + 15 = 45$
$I_2 = (a, c, d, b, a)$ atau (a, b, d, c, a) ==> panjang $= 12 + 5 + 9 + 15 = 41$
$I_3 = (a, c, b, d, a)$ atau (a, d, b, c, a) ==> panjang $= 10 + 5 + 9 + 8 = 32$



(a)



(b)



(c)

Gambar 3.2 (a, b, dan c) Sirkuit Hamilton pada sebuah graf

Jadi, sirkuit Hamilton terpendek adalah $I_3 = (a, c, b, d, a)$ atau (a, d, b, c, a) dengan panjang sirkuit $= 10 + 5 + 9 + 8 = 32$.

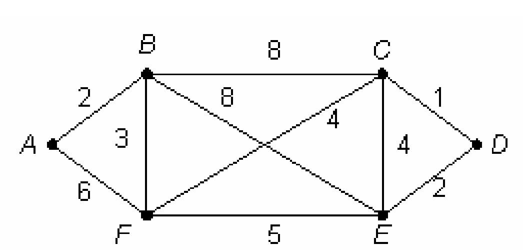
Jika jumlah simpul $n = 20$ akan terdapat $(19!)/2$ sirkuit Hamilton atau sekitar 6×10^{16} penyelesaian.

3.5. Persoalan Tukang Pos Cina

Dikemukakan oleh Mei Gan (berasal dari Cina) pada tahun 1962.

Masalahnya adalah sebagai berikut: seorang tukang pos akan mengantar surat ke alamat-alamat sepanjang jalan di suatu daerah. Bagaimana ia merencanakan rute perjalanannya supaya ia melewati setiap jalan tepat sekali dan kembali lagi ke tempat awal keberangkatan.

Cara mengatasi permasalahan ini adalah dengan menentukan sirkuit Euler di dalam graf.



Gambar 3.3 Sirkuit Euler di dalam Graf

3.6 Mengaplikasikan Graf dalam Kehidupan

Ada banyak hal dalam kehidupan ini dimana kita dapat mengaplikasikan teori graf. Aplikasi teori Graf bisa kita saksikan bila Anda mengendarai mobil atau menumpangi angkot Bandung. Kita punya banyak alternatif jalan untuk sampai ke tujuan. Namun ada jalan yang satu arah, ada titik-titik kemacetan, ada jam-jam sibuk. Dengan pemahaman tentang teori Graf, para ahli transportasi mengatur agar lampu lalu lintas menyala dengan warna dan saat yang tepat.

Selain itu, pengetahuan tentang teori Graf juga diaplikasikan dalam pengaturan jalur penerbangan agar tidak ada pesawat yang bersimpangan, pengaturan jalur telekomunikasi ketika ada pelanggan yang melakukan panggilan telefon, pengaturan jalur pelayaran, serta segala sistem yang berkaitan dengan objek diskrit dan hubungannya dengan objek yang lain.

4. Kesimpulan

Kesimpulan yang dapat dimbil dari studi dan implementasi graf ini adalah :

1. Dengan penggunaan graf dalam menyelesaikan permasalahan kita akan mendapat kemudahan dalam menyelesaikannya. Hal ini disebabkan karena sifat-sifat dan teori-teori mengenai graf yang memudahkan penyelesaian.
2. Untuk menyelesaikan persoalan dengan menggunakan graf mula-mula kita harus bisa memodelkan permasalahan dengan menggunakan graf, kemudian baru kita dapat mencari solusi dengan membaca persoalan dan menyelesaikan dengan sifat-sifat dan teori-teori mengenai graf.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2004). Diktat Kuliah IF2151 Matematika Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] *UI*. (2004). Web Universitas Indonesia. *http://www.cs.ui.ac.id/WebKuliah/MD1/DGRAF_05_8_10_18.DOC*. Tanggal akses: Selasa, 2 Januari 2007 pukul 19:17.

[3] *Soenhadji, Iman*. (2004). Teori Pengambilan Keputusan. Web Perpustakaan Universitas Gunadarma. *http://library.gunadarma.ac.id/*. Tanggal akses: Selasa, 2 Januari 2007 pukul 19:50.