Penerapan Graf Dalam Penentuan Jalur Terpendek Untuk Berkeliling Sekretariat Himpunan

Ichlasul Amal 13511075
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
ichlasul@students.itb.ac.id

***Abstrak*—Pekerjaan berkunjung ke sekretariat himpunan merupakan hal yang biasa dilakukan Divisi Intra Kampus HMIF ITB (Himpunan Mahasiswa Teknik Informatika Institut Teknologi Bandung). Untuk menentukan jalur yang terpendek yang dilalui persoalan ini dapat direpresentasikan ke dalam graf dan mencari sirkuit Hamilton dengan bobot minimum. Persoalan ini serupa dengan travelling salesman problem. Persoalan ini selanjutnya diimplementasikan menggunakan teknik Branch and Bound dalam bahasa C. Dari hasil ini diperoleh jalur terpendek yang harus dilewati untuk mengunjungi seluruh sekretariat himpunan di ITB.**

***Kata Kunci*—sirkuit Hamilton, travelling salesman problem, algoritma Branch and Bound, himpunan ITB**

I. PENDAHULUAN

Salah satu rutinitas harian yang dilakukan Divisi Intra Kampus HMIF ITB adalah berkunjung ke seluruh sekretariat himpunan lain yang ada di ITB. Kegiatan ini biasanya dilakukan untuk membagikan undangan, kuesioner, kartu ucapan, dan sebagainya. Sampai saat ini umumnya pekerjaan ini tidak dapat selesai dalam waktu satu hari. Kegiatan berkunjung umumnya dilakukan tanpa rencana dan hanya asal berangkat saja. Alangkah lebih baik jika kegiatan ini direncanakan terlebih dahulu untuk melalui jalur terpendek sehingga dapat mempersingkat waktu. Akan tetapi, penentuan jalur terpendek ini tidak mungkin secara manual atau coba-coba. Ada 29 himpunan mahasiswa jurusan yang ada di ITB, terlalu banyak untuk dilakukan pendekatan manual. Oleh karena itu, penulis berinisiatif untuk menyelesaikan masalah ini menggunakan konsep graf untuk memperoleh sirkuit terpendek yang harus ditempuh.

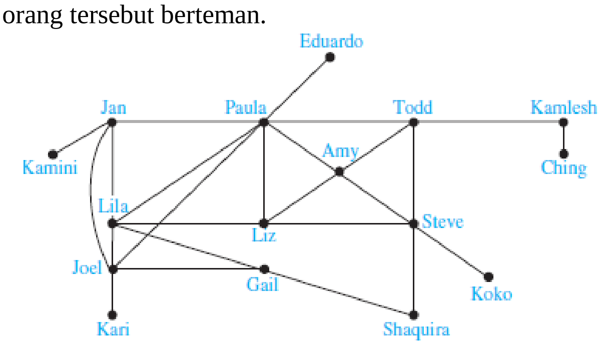
II. TEORI GRAF

A. Definisi Graf

Secara matematis, graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis $G = (V, E)$. V merupakan himpunan tak kosong dari simpul (*nodes* atau *vertices)* dan E adalah himpunan sisi (*edges*). Setiap sisi terhubung dengan pasangan simpul [rinaldi] [rossen].

Dari definisi tersebut dapat diperoleh bahwa graf minimal terdiri dari sebuah simpul dan boleh untuk tidak memiliki sisi. Jika dituliskan $E = (e_1, e_2, \dots, e_n)$ dan $V = (v_1, v_2, \dots, v_n)$ maka sisi dapat dituliskan sebagai $e_i = (v_j, v_k)$.

Secara umum graf dapat divisualisasikan seperti gambar 2-1. Graf tersebut merepresentasikan simpul sebagai akun social network maupun orang secara nyata, sedangkan sisi sebagai hubungan pertemanan yang dimiliki [rossen]. Simpul yang terhubung melalui sisi menunjukkan bahwa orang tersebut berteman.



Gambar 2-1 Contoh graf dalam merepresentasikan hubungan pertemanan

B. Terminologi Graf

Ada beberapa terminologi dasar dalam graf yang diambil dari [rinaldi]. Beberapa terminologi yang digunakan pada makalah ini didefinisikan sebagai berikut.

1. Sisi ganda (*multiple edges*). Sisi $e_i = (v_j, v_k)$ dan $e_{i+1} = (v_k, v_j)$ dinamakan sisi ganda karena kedua sisi ini menghubungkan dua buah simpul yang sama.
2. Gelang atau kalang (*loop)*. Suatu sisi $e_i = (v_j, v_j)$ dinamakan gelang karena sisi ini berawal dan berakhir di simpul yang sama.
3. Graf sederhana (*simple graph*). Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana.
4. Graf tak sederhana (*unsimple graph)*. Kebalikan dari graf sederhana, graf tak sederhana ialah graf yang mengandung gelang atau sisi ganda.
5. Graf tak berarah (*undirected graph)*. Graf yang setiap sisinya tidak memiliki orientasi arah disebut graf tak berarah. Dalam graf tak berarah, urutan simpul yang dihubungkan oleh sisi tidak diperhatikan. Jadi $(v_i, v_j) = (v_j, v_k)$.
6. Graf berarah (*directed graph)*. Graf yang setiap

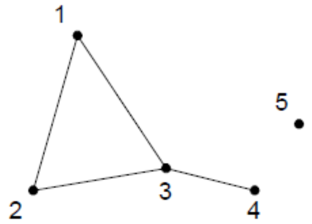
sisinya memiliki orientasi arah disebut graf berarah. Dalam graf berarah, urutan simpul yang dihubungkan oleh sisi diperhatikan. Jadi $(v_i, v_j) \neq (v_j, v_k)$. Simpul dalam graf berarah sering dinamakan dengan busur atau *arc*.

7. Bertetangga (*Adjacent*). Dua buah simpul pada graf tak berarah G dikatakan bertetangga jika keduanya terhubung langsung dengan sebuah sisi. Dengan kata lain v_j bertetangga dengan v_k jika sisi $e_i = (v_j, v_k)$ ada dalam graf G.
8. Bersisian (*Incident*). Untuk sembarang sisi $e_i = (v_j, v_k)$, sisi e_i dikatakan bersisian dengan simpul v_j dan v_k.
9. Derajat (*degree*) suatu simpul pada graf tak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Dalam graf berarah, derajat simpul merupakan penjumlahan dari jumlah busur yang masuk ke simpul dan jumlah busur yang keluar dari simpul. Derajat dari simpul v_j dinotasikan sebagai $d(v_j)$.
10. Lintasan (*Path)* yang panjangnya n dari simpul awal v_1 ke simpul tujuan v_{n+1} di dalam graf G ialah barisan berselang-seling antara simpul dan sisi yang berbentuk $v_1, e_1, v_2, e_2, \dots, v_n, e_n, v_{n+1}$ sedemikian sehingga $e_1 = (v_1, v_2), e_2 = (v_2, v_3), \dots, e_n = (v_n, v_{n+1})$ adalah sisi-sisi dari graf G.
11. Sirkuit (*circuit*) adalah lintasan yang berawal dan berakhir pada simpul yang sama.
12. Terhubung (*connected*). Graf G dikatakan graf terhubung jika untuk setap pasang simpul v_j dan v_k didalam himpunan V terdpak lintasan dari v_j ke v_k. Jika tidak, maka G disebut graf tak terhubung.
13. Graf berbobot (*weighted graph*) adalah graf yang setiap sisinya diberi sebuah harga/bobot.
14. Graf Lengkap (*Complete Graph)* adalah graf sederhana yang setiap simpulnya memiliki sisi ke seluruh simpul lainnya. Graf lengkap dengan n buah simpul dilambangkan dengan K_n. Setiap simpul pada K_n berderajat $n-1$.
15. Graf Lingkaran adalah graf sederhana yang setiap simpulnya berderajat dua. Graf lingkaran dengan n simpul dilambangkan dengan C_n. Simpul terakhir dari graf ini terhubung dengan simpul pertama

C. Representasi Graf

Untuk mengolah graf dalam komputer maka terlebih dahulu graf harus diolah di dalam memori. Salah satu representasi yang paling umum dan digunakan dalam makalah ini adalah matriks ketetanggan (*adjacency matrix).*

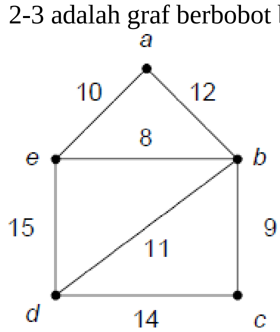
Misalkan $G = (V, E)$ adalah graf dengan n simpul. Matriks ketetanggan G adalah persegi yang berukuran $n \times n$. Bila matriks tersebut dinamakan $A = [a_{ij}]$, maka $a_{ij} = 1$ (*true*) jika simpul i dan j bertetangga, sebaliknya $a_{ij} = 0$ jika simpul i dan j tidak bertetangga. Contoh representasi dapat dilihat pada gambar 2-2.



$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

Gambar 2-2 Contoh graf sederhana (kiri). Representasi dalam matriks ketetanggaan (kanan).

Untuk graf berbobot, a_{ij} menyatakan bobot tiap sisi yang menghubungkan simpul i dengan simpul j. Gambar 2-3 adalah graf berbobot beserta matriks ketetanggaannya.



$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{bmatrix} \infty & 12 & \infty & \infty & 10 \\ 12 & \infty & 9 & 11 & 8 \\ \infty & 9 & \infty & 14 & \infty \\ \infty & 11 & 14 & \infty & 15 \\ 10 & 8 & \infty & 15 & \infty \end{bmatrix}
\end{array}
$$

Gambar 2-3 Contoh graf berbobot (kiri). Representasi dalam matriks ketetanggaan (kanan).

C. Sirkuit Hamilton

Sirkuit Hamilton adalah sirkuit yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul asal yang sekaligus simpul akhir dilalui dua kali.

Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton. Hingga saat ini belum ada cara yang mangkus untuk menentukan sirkuit Hamilton. Untuk menunjukkan bahwa suatu graf tertentu mempunyai sirkuit Hamilton, kita terpaksa menggunakan cara eksplisit atau mengenumerasi kemungkinan sirkuit yang ada.[rinaldi] Beberapa teorema tentang sirkuit Hamilton adalah sebagai berikut.

1. Teorema Dirac. Jika G adalah graf sederhana dengan n buah simpul $(n \geq 3)$ sedemikian sehingga derajat tiap simpul paling sedikit $n/2$ $(d(v) \geq n/2$ untuk setiap simpul v di $G)$, maka G adalah graf Hamilton.
2. Teorema Ore. Jika G adalah graf sederhana dengan n buah simpul $(n \geq 3)$ sedemikian sehingga $d(v) + d(u) \geq n$ untuk setiap simpul tidak bertetangga u dan v, maka G adalah graf Hamilton.
3. Setiap graf lengkap adalah graf Hamilton.
4. Di dalam graf lengkap G dengan n buah simpul $(n \geq 3)$ terdapat sebanyak $(n-1)!/2$ buah sirkuit Hamilton.
5. Di dalam graf lengkap G dengan n buah simpul $(n \geq 3$ dan n ganjil) terdapat sebanyak $(n-1)/2$ buah sirkuit Hamilton yang saling lepas (tidak ada sisi yang bersisian). Jika n genap dan $n \geq 4$ maka di dalam G terdapat $(n-2)/2$ buah sirkuit Hamilton yang saling lepas.

D. Travelling Salesman Problem

Travelling Salesman Problem atau biasa disebut TSP merupakan persoalan yang sangat terkenal dalam teori graf maupun algoritma. Deskripsi persoalannya adalah sebagai berikut. Diberikan sejumlah kota dan jarak antar kota. Tentukan sirkuit terpendek yang harus dilalui pedagang itu untuk berangkat dari kota asal dan pergi menyinggahi setiap kota tepat satu kali kemudian kembali ke kota asal.

Persoalan ini dapat direpresentasikan dalam graf. Setiap kota menyatakan simpul graf sedangkan sisi menyatakan jalan yang menghubungkan tiap kota. Bobot pada sisi menyatakan jarak antara dua buah kota. Persoalan ini tidak lain adalah menentukan sirkuit Hamilton yang memiliki bobot minimum. Meskipun persoalan ini bernama travelling salesman tetapi penerapannya sangat luas dalam bidang sehari-hari maupun bidang sains dan teknik.

Persoalan TSP ini merupakan persoalan yang sulit dipandang dari sisi komputasi. Untuk memecahkan persoalan ini kita harus mengenumerasi sebanyak $(n-1)!\ /\ 2$ sirkuit Hamilton, menghitung panjang rute masing-masing sirkuit, dan memilih sirkuit dengan rute terpendek. Bayangkan untuk $n = 20$ akan terdapat $19!/2$ atau sekitar 6×10^6 buah penyelesaian.

III. Representasi Dalam Graf

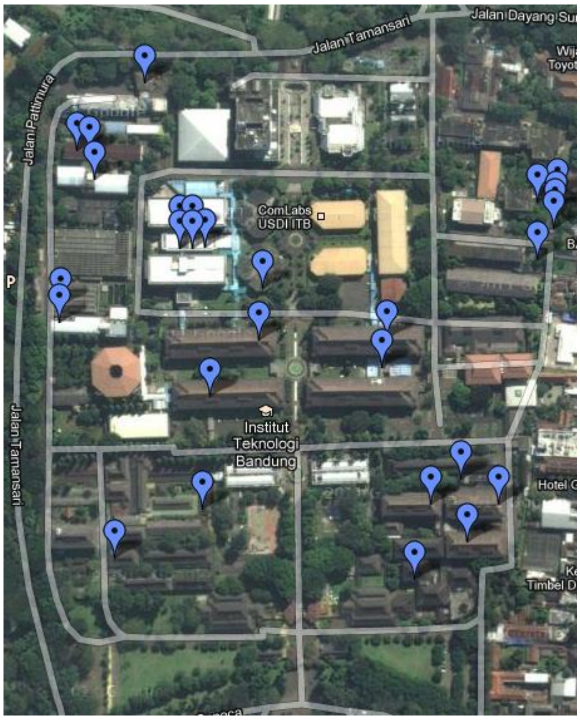
A. Visualisasi Persoalan

Persoalan mengunjungi sekretariat himpunan di ITB masing-masing satu kali dan kembali ke sekretariat semula merupakan persoalan yang sama dengan travelling salesman problem. Dalam hal ini simpul pada graf merepresentasikan sekretariat, sisi merepresentasikan jalan penghubung, dan bobot pada sisi merepresentasikan jarak jalan antar sekretariat. Hal ini dapat divisualisasikan pada gambar 3-1 dengan masing-masing lokasi sekretariat yang berupa simpul ditandai dengan simbol biru biru sedangkan jalan yang menghubungkan tiap sekretariat berupa sisi yang tidak digambarkan karena akan terlalu rumit. Cukup dibayangkan bahwa simpul-simpul tersebut terhubung satu sama lain membentuk graf lengkap.

B. Asumsi Persoalan

Dalam persoalan ini dilakukan asumsi untuk mempermudah pemecahan masalah. Asumsi yang dilakukan sebagai berikut.

1. Pengukuran yang hanya menggunakan kakas Google Maps[3] tanpa mengukur langsung. Hal ini tentu saja untuk mempermudah pengukuran. Dikhawatirkan terdapat perbedaan jarak yang cukup signifikan dibandingkan dengan kondisi nyata.
2. Jarak antar sekretariat hanya diukur berdasarkan jarak terdekat dan tidak diukur berdasarkan jalan umum yang harus dilalui. Dalam hal ini seolah-olah untuk berkunjung ke sekretariat lain kita dapat terbang melalui lintasan terpendek. Padahal dalam kenyataannya jalan untuk menuju tiap sekretariat berkelok-kelok dan bukan garis lurus.



Gambar 3-1 Visualisasi lokasi sekretariat sebagai simpul graf

Tabel 3-1 Penomoran Masing-Masing Sekretariat

No	Nama
1	Himpunan Mahasiswa Teknik Sipil (HMS)
2	Himpunan Mahasiswa Fisika (Himafi)
3	Himpunan Mahasiswa Mesin (HMM)
4	Keluarga Mahasiswa Penerbangan (KMPN)
5	Himpunan Mahasiswa Matematika (Himatika)
6	Keluarga Mahasiswa Teknik Industri (MTI)
7	Himpunan Mahasiswa Astronomi (Himastron)
8	Keluarga Mahasiswa Sekolah Bisnis dan Manajemen (KM-SBM)
9	Himpunan Mahasiswa Elektroteknik (HME)
10	Himpunan Mahasiswa Farmasi (HMF)
11	Himpunan Mahasiswa Informatika (HMIF)
12	Keluarga Mahasiswa Teknik Kelautan (KMKL)
13	Himpunan Mahasiswa Fisika Teknik (HMFT)
14	Mahasiswa Teknik Material (MTM)
15	Himpunan Mahasiswa Teknik Kimia (Himatek)
16	Himpunan Mahasiswa Biologi (Nymphaea)
17	Himpunan Mahasiswa Geofisika dan Meteorologi (HMME)
18	Himpunan Mahasiswa Tambang (HMT)
19	Ikatan Mahasiswa Metalurgi (IMMG)
20	Ikatan Mahasiswa Geodesi (IMG)
21	Himpunan Mahasiswa Teknik Perminyakan (Patra)
22	Himpunan Mahasiswa Teknik Geofisika (Terra)
23	Himpunan Mahasiswa Kimia (Amisca)
24	Himpunan Mahasiswa Oseanografi (HMO)
25	Himpunan Mahasiswa Teknik Lingkungan (HMTL)
26	Himpunan Mahasiswa Planologi (HMP)
27	Ikatan Mahasiswa Gunadharma (IMA-G)
28	Keluarga Mahasiswa Seni Rupa (KMSR)
29	Himpunan Mahasiswa Teknik Geologi (GEA)

3. Mengabaikan ketinggian sekretariat. Di ITB ketinggian tiap bangunan tidak sama, seharusnya faktor ketinggian ini mempengaruhi juga jarak yang harus ditempuh. Contohnya saja sekretariat KMKL berada di lantai 4 gedung Labtek VI sedangkan sekretariat HMFT berada di *basement* TVST. Contoh lainnya adalah tanah di wilayah utara ITB jauh lebih tinggi dibandingkan dengan di wilayah selatan.
4. Mengabaikan jarak sekretariat himpunan yang terlalu jauh. Dalam pengukuran diabaikan jarak himpunan yang terlalu jauh, misalnya KMSR dengan KM-SBM. Selain untuk mempermudah pengukuran hal ini juga termasuk dalam *heuristic* atau seni dalam pemecahan masalah. Dalam hal ini diasumsikan tidak mungkin jalur terpendek yang harus dilalui akan melewati jalur ini.
5. Mengabaikan batasan-batasan lain yang mungkin ada dalam penerapan di lapangan. Misal saja ada sekretariat himpunan yang harus dikunjungi terlebih dahulu dan sebagainya.

C. Pengukuran Jarak

Jarak dari masing-masing sekretariat himpunan dalam meter diukur secara manual menggunakan kakas Google Maps. Hasil yang diperoleh dapat dilihat pada tabel 3-2. Penomoran masing-masing sekretariat sesuai dengan tabel 3-1. Dalam hal ini isian kosong merupakan jarak yang tidak diukur dan diasumsikan terlalu jauh untuk dijadikan kemungkinan jalur terpendek yang dilalui.

III. IMPLEMENTASI

A. Representasi dalam Matriks

Sebelum mengolah di dalam suatu program komputer maka data jarak pada tabel 3-2 harus terlebih dahulu direpresentasikan dalam struktur data yang dimengerti komputer. Representasi yang dipilih dalam makalah ini adalah representasi matriks ketetanggaan seperti pada subbab IIC. Representasi ini dipilih karena algoritma yang menggunakan representasi ini tersedia dan representasi ini yang paling umum digunakan. Selain itu representasi ini mampu merepresentasikan tidak hanya ketetanggaan tiap simpul namun juga bobot dari lintasan antar dua simpul.

Tabel 3-2 Jarak Antara Masing-Masing Sekretariat dalam meter

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1		96	230	255					311	331	179	247	291							366					336	304	333	285	
2	96		233	236					217	240	107	170	220	263	264	249	248			280				248	247	216	252	212	
3	230	233		16	161	139	163	239	309	311	160	189	195	142	151	144	156							133					
4	255	236	16		146	124	147	226	309	310	166	190	192	131	142	135	147							124					
5			161	146		25	13	84			255	238	207	113	122	133	140							124					
6			139	124	25		33	105			232	215	189	94	105	113	122							103					
7			163	147	13	33		91			264	249	219	125	135	145	153							135					
8			239	226	84	105	91		350	332	303	268	223	147	150	163	166				386			160					
9	311	217	309	309				350		26	165	121	137	231	220	212	201	232	222	169	215	209	179	223	128	137	182	202	215
10	331	240	311	310				332	26		175	123	127	220	209	201	193	211	202	188	192	188	162	214	150	160	205	227	196
11	179	107	160	166	255	232	264	303	165	175		69	112	156	154	141	140	378	371	291	360	360	335	142	249	232	279	259	366
12	247	170	189	190	238	215	249	268	121	123	69		50	126	119	107	103	314	308	274	295	299	276	115	232	224	274	268	304
13	291	220	195	192	207	189	219	223	137	127	112	50		94	84	77	67	292	289	302	272	281	262	88	259	259	308	309	284
14		263	142	131	113	94	125	147	231	220	156	126	94		13	20	28	377	375		357	370	355	16					371
15		264	151	142	122	105	135	150	220	209	154	119	84	13		14	17	346	344		326	340	327	18					342
16		249	144	135	133	113	145	163	212	201	141	107	77	20	14		12	347	345		327	349	326	13					342
17		248	156	147	140	122	153	166	201	193	140	103	67	28	17	12		336	333		316	329	315	24					331
18									232	211	378	314	292	377	346	347	336		10	295	20	31	62	361	282	314	338	385	19
19									222	202	371	308	289	375	344	345	333	10		286	19	20	51	358	271	304	325	374	9
20	366	280							169	188	291	274	302					295	286		288	265	237	386	44	65	47	106	264
21								386	215	192	360	295	272	357	326	327	316	20	19	288		30	53	339	272	302	327	373	23
22									209	188	360	299	281	370	340	349	329	31	20	265	30		33	354	252	286	308	356	12
23									179	162	335	276	262	355	327	326	315	62	51	237	53	33		339	221	253	276	325	41
24		248	133	124	124	103	135	160	223	214	142	115	88	16	18	13	24	361	358	386	339	354	339						356
25	336	247							128	150	249	232	259					282	271	44	272	252	221			38	60	104	262
26	304	216							137	160	232	224	259					314	304	65	302	286	253		38		49	73	295
27	333	252							182	205	279	274	308					338	325	47	327	308	276		60	49		60	318
28	285	212							202	227	259	268	309					385	374	106	373	356	325		104	73	60		366
29									215	196	366	304	284	371	342	342	331	19	9	264	23	12	41	356	262	295	318	366	

Secara umum representasi matriks yang digunakan dalam implementasi tepat sama dengan tabel 3-2. Dengan kolom tabel sebagai indeks kolom matriks dan baris tabel sebagai indeks baris matriks. Nilai yang kosong pada tabel 3-1 seharusnya diisi dengan nilai tak hingga namun dalam implementasi ini cukup dipilih angka yang cukup besar. Pada algoritma yang dipilih telah disediakan penanganan khusus untuk jarak yang terlalu jauh/tidak tersambung yaitu dengan menggunakan jarak sama dengan 0.

B. Representasi Fisik

Untuk mempermudah maka data berupa matriks tersebut dituliskan pada suatu file eksternal. Isi dari file eksternal ini cukup sisi segitiga atas dari matriks karena pada dasarnya matriks yang terbentuk simetris berdasarkan diagonal. Selain itu juga dilakukan penggantian nilai jarak yang melebihi 300 dengan nilai 0. Hal ini dilakukan untuk membantu heuristic menjadi lebih baik lagi. Dalam analisa awal, penggunaan metode ini mampu meningkatkan efektifitas algoritma hingga 16,67 %. Contoh 15 baris pertama dan 15 baris terakhir dari file eksternal yang diberi nama input.txt adalah sebagai berikut.

```
96
230
255
0
0
0
0
0
0
179
247
291
0
0
0
. . .
0
0
0
0
0
38
60
104
262
49
73
295
60
0
0
```

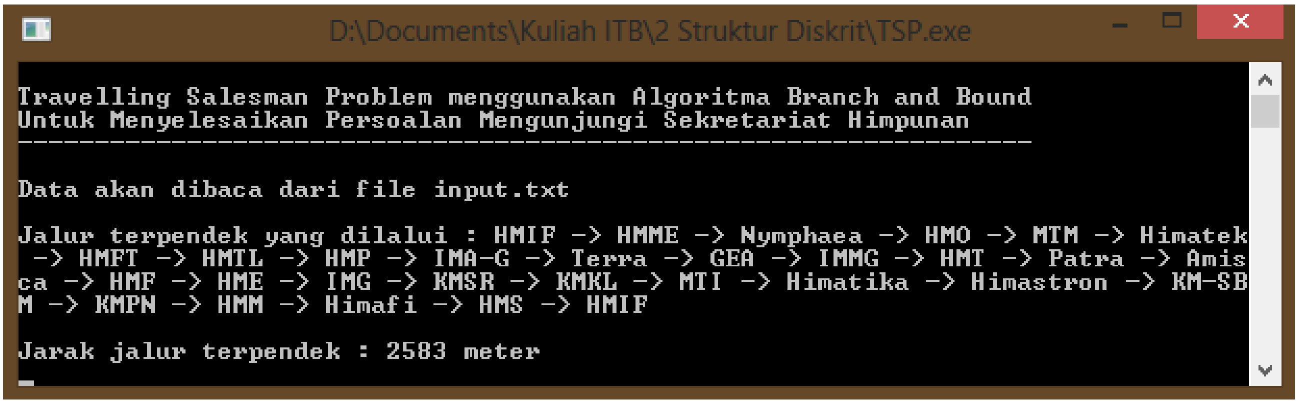
C. Algoritma

Untuk mengimplementasikan hal ini tidak mungkin digunakan cara naif atau *brute force* dengan mengenumerasi seluruh kemungkinan yang ada. Pada subbab IID ditunjukkan bahwa jumlah enumerasi yang dibutuhkan untuk 20 simpul saja mencapai hingga 6×10^6 buah penyelsaian. Pada persoalan berkeliling sekretariat himpunan ini terdiri dari 29 simpul sehingga semakin mustahil untuk dilakukan enumerasi.

Algoritma yang dicoba dipilih pada makalah ini adalah algoritma Branch and Bound. Algoritma Branch and Bound terdiri dari enumerasi seluruh kandidat solusi secara sistematis. Selanjutnya subset dari kandidat dieliminasi secara massal menggunakan batas atas dan bawah yang diestimasi. Algoritma yang digunakan dalam implementasi ini awalnya terdapat dalam [4] namun telah dilakukan modifikasi yang cukup banyak. Implementasi dalam bahasa C adalah sebagai berikut.

```c
#include <stdio.h>

int cost=0;

void get(int a[30][30],int n,int
visited[30]) {
  int i,j,max,src,dest,wt;
  FILE *F;

  for(i=1;i<=n;i++) {
    visited[i]=0;

    for(j=1;j<=n;j++) {
      a[i][j]=99999;
    }
  }

  max=n*(n-1)/2;
  src = 1;
  dest = 2;
  F = fopen("input.txt", "r");

  for (i=1;i<=max;i++) {
    fscanf(F, "%d",&wt);
    a[src][dest]=wt;
    a[dest][src]=wt;
    dest++;
    if (dest > 29) {
      src++;
      dest = src + 1;
    }
  }
}

void mincost(int city,int n,int
a[30][30],int visited[]) {
  int i,ncity;
  visited[city]=1;
  switch (city) {
    case 1 :
      printf("HMS -> ");
      break;
    case 2 :
      printf("Himafi -> ");
      break;
    case 3 :
      printf("HMM -> ");
      break;
    case 4 :
      printf("KMPN -> ");
      break;
```

```
    case 5 :
      printf("Himatika -> ");
      break;
    case 6 :
      printf("MTI -> ");
      break;
    case 7 :
      printf("Himastron -> ");
      break;
    case 8 :
      printf("KM-SBM -> ");
      break;
    case 9 :
      printf("HME -> ");
      break;
    case 10 :
      printf("HMF -> ");
      break;
    case 11 :
      printf("HMIF -> ");
      break;
    case 12 :
      printf("KMKL -> ");
      break;
    case 13 :
      printf("HMFT -> ");
      break;
    case 14 :
      printf("MTM -> ");
      break;
    case 15 :
      printf("Himatek -> ");
      break;
    case 16 :
      printf("Nymphaea -> ");
      break;
    case 17 :
      printf("HMME -> ");
      break;
    case 18 :
      printf("HMT -> ");
      break;
    case 19 :
      printf("IMMG -> ");
      break;
    case 20 :
      printf("IMG -> ");
      break;
    case 21 :
      printf("Patra -> ");
      break;
    case 22 :
      printf("Terra -> ");
      break;
    case 23 :
      printf("Amisca -> ");
      break;
    case 24 :
      printf("HMO -> ");
      break;
    case 25 :
      printf("HMTL -> ");
      break;
    case 26:
      printf("HMP -> ");
      break;
    case 27 :
      printf("IMA-G -> ");
      break;
    case 28 :
      printf("KMSR -> ");
      break;
    case 29 :
      printf("GEA -> ");
      break;
  }
  ncity = least(city,n,a,visited);
  if (ncity==99999) {
    ncity = 11;
    printf("HMIF");
    cost += a[city][ncity];
    return;
  }
  mincost(ncity,n,a,visited);
}

int least(int c,int n,int a[30][30],int
visited[30]) {
  int i,nc=99999,min=99999,kmin;
  for(i=1;i<=n;i++) {
    if((a[c][i]!=0)&&(visited[i]==0)) {
      if(a[c][i]<min) {
        min=a[i][1]+a[c][i];
        kmin=a[c][i];
        nc=i;
      }
    }
  }

  if (min!=99999)
    cost+=kmin;

  return nc;
}

void put() {
  printf("\nJarak jalur
terpendek:%d\n",cost);
  getchar();
  getchar();
}

main() {
  int n,a[30][30],visited[30];

  printf("\nTravelling Salesman
Problem Using Branch and Bound");
  printf("\nUntuk Menyelesaikan
Persoalan Mengunjungi Sekretariat
Himpunan");

printf("\n***************************
*********************");
  n = 29;
  printf("\nData akan dibaca dari file
input.txt");
  get(a,n,visited);
  printf("\nJalur terpendek yang
dilalui : ");
  mincost(11,n,a,visited);
  put();
}
```



Gambar 4-1 Hasil implementasi menggunakan Algoritma Branch and Bound dan bahasa C

IV. HASIL DAN ANALIS

Dari implementasi diperoleh hasil seperti terlihat pada gambar 4-1. Dari hasil ini ditunjukkan bahwa jarak minimal yang dilalui sekitar 2,5 km. Hasil ini masih berada dalam kewajaran sehingga penulis cukup yakin dengan kebenaran implementasi. Selain itu program ini dapat berjalan dalam waktu instan (kurang dari 1 detik) untuk masukan simpul sebanyak 29.

Jadi menurut program jalur terpendek yang harus dilewati untuk mengunjungi seluruh sekretariat himpunan masing-masing sekali dan diawali dari HMIF adalah melalui HMME, Nymphaea, HMO, MTM, Himatek, Patra, IMMG, GEA, Terra, KMSR, IMA-G, HMP, HMTL, Amisca, HMT, IMG, HMFT, KMKL, HMF, HME, KM-SBM, Himatika, Himastron, MTI, KMPN, HMM, Himafi, HMS, kemudian kembali lagi ke HMIF. Visualisasi dari jalur yang harus dilewati dapat dilihat pada gambar 4-2.

Gambar 4-2 Visualisasi jalur yang harus dilalui untuk melewati seluruh sekretariat

Dari beberapa tanggapan responden, hasil dari implementasi ini belum optimal karena masih sering dilakukan penyeberangan dari daerah barat ke timur dan ada juga dari utara ke selatan. Analisa yang mungkin dilakukan adalah karena adanya kesalahan data masukan. Seperti dilihat pada Tabel 3-2 data yang dibutuhkan sangat banyak sehingga sangat rawan untuk terjadinya kesalahan. Data ini juga harus diubah menjadi representasi fisik yang bisa saja terjadi kesalahan kembali. Selain itu saat pengukuran jarak sekretariat dapat saja terjadi kesalahan pengukuran atau kesalahan dalam menuliskan data.

V. KESIMPULAN

Dari makalah ini telah dijelaskan beberapa dasar teori graf yang digunakan untuk merepresentasikan persoalan mencari jalur terpendek untuk berkeliling sekretariat himpunan tepat sekali dalam wujud graf. Selain itu telah dilakukan pengukuran jarak masing-masing sekretariat dan merepresentasikannya dalam wujud matriks dan representasi fisik

Dari hasil implementasi telah berhasil diperoleh jalur terpendek yang harus dilewati untuk mengunjungi seluruh sekretariat himpunan masing-masing sekali dan diawali dari HMIF. Implementasi ini dapat ditingkatkan lebih baik lagi dengan memasukkan berbagai faktor yang diabaikan pada subbab IIIB terutama faktor jalan yang berkelok-kelok dalam menuju suatu himpunan. Selain itu implementasi dapat dikembangkan untuk persoalan lain yang lebih umum dan terdiri dari simpul yang lebih banyak. Faktor lain yang perlu ditingkatkan adalah akurasi atau optimisasi hasil program. Karena menurut tanggapan responden hasil dari program belum menunjukkan jalur yang benar-benar terpendek.

VI. UCAPAN TERIMA KASIH

Terima kasih kepada Farid Firdaus yang telah memberi inspirasi ide masalah dalam makalah ini.

Terima kasih kepada Pandu Kartika Putra dan Faiz Ilham Muhammad yang telah memberikan masukan terhadap hasil implementasi.

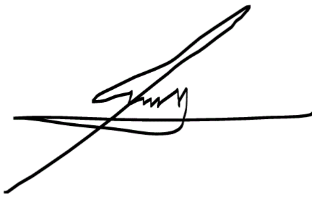
DAFTAR PUSTAKA

[1] K. H. Rosen, "Discrete Mathematics and Its Application", 7th ed, New York: McGraw-Hill, 2012, hal. 641-706
[2] R. Munir, "Matematika Diskrit", Revisi Keempat, Bandung: Informatika, 2010, hal. 356-424
[3] http://maps.google.com/, Diakses 17 Desember 2012. 08:15
[4] http://wiki.answers.com/Q/Could_you_give_the_C_code_for_travelling_salesman_problem_using_branch_and_bound, Diakses 18 Desember 2012. 11:50

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Desember 2012



Ichlasul Amal 13511075