ORIGINAL ARTICLE

PERANCANGAN STRUKTUR DATA YANG EFISIEN UNTUK PEMROGRAMAN ANALISIS JARINGAN

Wayan Firdaus Mahmudy, Ani Budi Astuti
Jurusan Matematika, FMIPA, Universitas Brawijaya

ABSTRAK

Pada penelitian ini telah dirancang Tipe Data Abstrak (*Abstract Data Type*/ADT) *graph* yang efisien untuk masalah analisis jaringan. ADT graph ini dirancang untuk menyimpan data matriks berukuran besar sesuai kapasitas memori komputer.

ADT tersebut digunakan untuk menyusun perangkat lunak untuk menyelesaikan persoalan pencarian rute terpendek (*shortest route*/*shortest path*) dan persoalan minimasi jaringan atau rentang pohon minimal (*minimal spanning tree*).

Program yang telah dibuat dengan memanfaatkan struktur data graph mampu memecahkan masalah jaringan (jarak terpendek dan rentang pohon minimal) dengan tepat dan memberikan pemecahan langkah demi langkah.

ABSTRACT

An efficient Graph *Abstract Data Type* (ADT) to solve network analysis problem has been designed. The Graph ADT is designed to manage big size matrix depend on computer‘s memory.

The ADT is used to create software and solving *shortest route* / *shortest path* problems and *minimal spanning tree* problems. This software can give an exact solution of *shortest route* / *shortest path* problems and *minimal spanning tree* problems and it’s steps.

PENDAHULUAN

Analisis jaringan merupakan suatu masalah dalam penelitian operasional yang mencakup persoalan pencarian rute terpendek (*shortest route*/*shortest path*), persoalan minimasi jaringan atau rentang pohon minimal (*minimal spanning tree*) dan persoalan aliran maksimum (*maximal flow*). Pemecahan persoalan dalam analisis jaringan secara manual memerlukan waktu yang lama dan ketelitian perhitungannya tidak terjamin sehingga diperlukan suatu perangkat lunak atau program komputer sebagai alat bantu. Telah tersedia beberapa paket program untuk memecahkan masalah ini tetapi kebanyakan tidak menunjukkan langkah demi langkah penyelesaian persoalan yang diberikan sehingga kurang mendukung untuk proses belajar mengajar. Selain itu juga terdapat keterbatasan banyaknya data yang dapat dimasukkan.

Pembuatan program untuk masalah analisis jaringan memerlukan memori komputer yang cukup besar untuk menampung data masukan dan data untuk pemrosesan. Dengan menggunakan model struktur data standart yang terdapat dalam kompiler bahasa pemrograman (dalam penelitian ini akan digunakan bahasa pemrograman Pascal dengan kompiler Delphi 2.0) maka memori yang dialokasikan kompiler tersebut tidak akan mencukupi. Untungnya kompiler Delphi 2.0 mengijinkan pemrogram untuk mendefinisikan tipe data

buatan (*users defined*) yang biasa disebut tipe data abstrak (*abstract data type*/*ADT*).

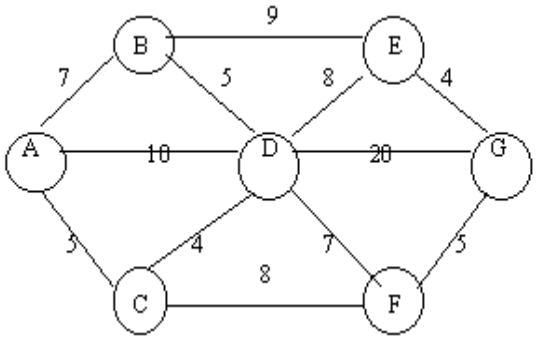
Yang menjadi masalah dalam perancangan suatu ADT adalah bagaimana dengan menggunakan memori komputer seminimum mungkin dihasilkan kecepatan eksekusi program yang maksimum.

Dalam penelitian ini dirancang ADT yang efisien untuk masalah analisis jaringan. ADT yang telah dibuat akan digunakan dalam penyusunan program untuk memecahkan persoalan pencarian rute terpendek (*shortest route*/*shortest path*) dan persoalan minimasi jaringan atau rentang pohon minimal (*minimal spanning tree*).

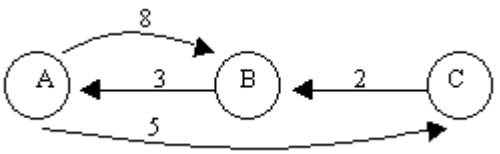
TINJAUAN PUSTAKA

Suatu jaringan terdiri atas suatu set titik-titik yang dihubungkan yang disebut *node*. Node-node tersebut dilambangkan dengan sebuah huruf atau angka. Pada Gambar 1 dan Gambar 2 node-node tersebut dilambangkan dengan huruf. Node tertentu dihubungkan dengan node lain dengan sebuah garis yang disebut busur (*edge*) (Dimyati, 1992).

Suatu busur bisa berarah atau tak berarah. Gambar 1 menunjukkan jaringan dengan busur tak berarah, Gambar 2 menunjukkan jaringan dengan busur berarah. Angka-angka pada busur merupakan suatu nilai yang tergantung permasalahannya.



Gambar 1. Jaringan dengan busur tak berarah



Gambar 2. Jaringan dengan busur berarah

Persoalan Rute Terpendek (*Shortest Route*)

Pada persoalan rute terpendek, yang menjadi masalah adalah mencari rute dengan jarak terpendek dari suatu node ke node yang lainnya dengan angka pada busur merupakan jarak node yang dihubungkan oleh busur tersebut. Pada masalah nyata, node yang ada bisa mewakili suatu kota, sedangkan angka pada busur merupakan jarak antar kota.

Untuk memecahkan persoalan rute terpendek bisa digunakan algoritma Dijkstra atau algoritma Floyd (Dimyati, 1992) (Wirt, 1976). Algoritma Dijkstra digunakan untuk mencari rute terpendek dari satu node sumber ke semua node yang lain. Algoritma Floyd digunakan untuk mencari rute terpendek dari semua node ke semua node yang lain. Dalam penelitian ini digunakan algoritma Floyd. Di bawah ini disajikan algoritma Floyd:

```
procedure Floyd (n:integer
  var A:array[1..n,1..n] of real;
  C:array[1..n,1..n] of real);
var i,j,k:integer;
begin
  for i:=1 to n do
  for j:=1 to n do
  begin
    { salin nilai matriks C ke A }
    A[i,j] = C[i,j];
    { set matriks path }
    Path[i,j]:=0;
  end;

  for k:=1 to n do
  for i:=1 to n do
  for j:=1 to n do
    { periksa apakah didapatkan }
    { jarak yang lebih kecil    }
    { jika melalui k             }
    if (A[i,k]+A[k,j])<A[i,j] then
    begin
      { perbarui matriks A }
      A[i,j] := A[i,k] + A[k,j];
      { catat lintasan/jalur }
      Path[i,j] := k;
    end;
end; { procedure }
```

C merupakan matrik untuk menyimpan jarak antar node secara langsung; ***A*** untuk menyimpan jarak antar node yang terdekat; ***Path*** merupakan matriks global untuk menyimpan jalur/lintasan dengan jarak terdekat; ***n*** menyatakan banyaknya node. Untuk dua node yang tidak ada hubungan/jalur langsung maka pada matrik ***C*** diisi nilai yang tak terhingga; sedangkan pada pemrogramannya bisa digunakan suatu nilai real yang sangat besar.

Inti dari algoritma Floyd di atas adalah untuk mencari jarak terdekat antara dua node (node *i* dan node *j*) adalah dengan mencoba melalui semua node yang lain (node *k*).

Persoalan Rentang Pohon Minimal (*Minimal Spanning Tree*)

Pada persoalan rentang pohon minimal, yang menjadi masalah adalah menghubungkan semua node yang ada dengan jarak minimum. Persoalan ini merupakan variasi persoalan rute terpendek, perbedaannya terletak pada rute yang dicari. Pada rute terpendek, dicari lintasan/rute dari sumber ke tujuan yang memberikan total jarak minimum, sedangkan pada persoalan rentang pohon minimal yang dipersoalkan adalah menentukan busur-busur yang menghubungkan node-node yang ada pada jaringan sehingga diperoleh panjang busur total yang minimum. Pada masalah yang nyata misalkan persoalan pemasangan kabel telepon yang menghubungkan semua kota.

Persoalan rentang pohon minimal ini bisa diselesaikan dengan cara sebagai berikut (Dimyati, 1992):

1. Dipilih secara sembarang salah satu node, kemudian node ini dihubungkan dengan node lain yang terdekat.
2. Ditentukan node lain yang belum terhubung yang terdekat dengan node-node yang sudah terhubung. Node ini kemudian dihubungkan.

Struktur Data Graph

Graph adalah suatu ADT yang digunakan untuk menangani masalah jaringan. Secara sederhana, graph dalam Pascal bisa dituliskan sebagai berikut (Schneider, 1987):

```
const
  MaxNode = 100;

type
  Graph = object
    { data }
    N : integer;
    A : array[1..MaxNode,
        1..MaxNode] of real;
    { metoda manipulasi data }
    procedure ……
    function ……
  end;
```

N digunakan untuk menyimpan banyaknya node, ***A*** merupakan array dua dimensi untuk menyimpan jarak antar node.

Struktur data di atas merupakan struktur data statis. Apabila banyaknya node yang ada semakin bertambah maka ukuran ***A*** harus ditambah, padahal ukuran data termasuk program dalam Pascal dibatasi sampai 64 KB. Untuk memecahkan masalah ini bisa digunakan struktur data dinamis dengan menggunakan pointer (Kadir, 1990).

Dengan menggunakan pointer, struktur data graph di atas bisa dimodifikasi sebagai berikut:

```
const
  MaxNode = 100;
  PtrData = ^RecordData;
  RecordData = record
    Item : integer;
    Next : PtrData;
  end;
type
  Graph = object
    { data }
    N : integer;
    A : PtrData;
    { metoda manipulasi data }
    procedure ……
    function ……
  end;
```

Penggunaan pointer seperti struktur data di atas memerlukan penanganan tersendiri yang menambah kompleksitas program. Karena itu perlu disusun suatu ADT graph dengan menggunakan data dinamis tetapi harus dirancang sedemikian rupa sehingga mudah untuk digunakan dan waktu untuk mengaksesnya cukup cepat (kompleksitas waktunya rendah).

Perancangan Input Program

Perangkat lunak dibuat dengan bahasa pemrograman *Borland Delphi 2.0* yang bekerja pada lingkungan sistem operasi 32 bit. Perangkat lunak menerima masukan data berupa file text dan keluarannya juga berupa file text yang berisi penyelesaian masalah langkah demi langkah.

Misalkan terdapat masalah jaringan seperti pada Gambar 2. Matriks jarak dari jaringan tersebut disimpan dalam file teks (misalnya DATA1.DAT) dengan format sebagai berikut:

```
3
0   8   5
3   0   1000
1000   2   0
```

Baris pertama dari file di atas menyatakan banyaknya node dan baris selanjutnya merupakan nilai busur. Untuk menyatakan node yang tak terhubung langsung bisa digunakan nilai yang jauh lebih besar dari nilai lainnya (untuk contoh di atas digunakan nilai 1000).

Perancangan Obyek TDataMatrixReal

Merupakan inti dari pemecahan masalah dalam penelitian ini. Berisi data dan metoda untuk memanipulasi data matriks berukuran besar. Data disimpan dalam memori dinamis dengan menggunakan

gabungan array dan pointer. Array satu dimensi digunakan untuk menyimpan pointer yang menunjuk langsung ke blok data yang telah dialokasikan. Banyaknya blok yang dialokasikan disesuaikan secara dinamis sesuai kebutuhan. Kerangka obyek ini adalah:

```
const
TDataMatrixMaxElement = 600;
TDataMatrixMaxBlock   = 600;

type
TDataMatrixRealData =
  array[0..TDataMatrixMaxElement-1]
  of real;
TDataMatrixRealPtr =
  ^TDataMatrixRealData;

TDataMatrixReal = object
  { data }
  row, col, block : word;
  Data : array
    [0..TDataMatrixMaxBlock-1]of
    TDataMatrixRealPtr;
  { metoda }
  constructor Init
              (xrow,xcol:word);
  destructor Done;
  procedure SetValueAll (D:Real);
  function  Get (i,j:word) : Real;
  procedure Put (i,j:word; D:Real);
end;
```

Data yang ada pada obyek ini adalah:

- Row : menyimpan banyaknya baris
- Col : menyimpan banyaknya kolom
- Block : menyimpan banyaknya blok yang dibutuhkan
- Data : untuk menyimpan data matriks

Metoda yang ada pada obyek ini adalah:

- Init : untuk memesan memori.
- Done : untuk mengembalikan memori yang telah dipesan.
- Get : mendapatkan nilai elemen matrik pada baris *i*, kolom *j*.
- Put : mengisi nilai elemen matrik pada baris *i*, kolom *j*.

Perancangan Obyek TGraph

Obyek untuk menangani data jaringan. Berisi data matriks jarak antar node dan metoda untuk membaca nilai tersebut dari file. Tugas utama obyek ini adalah meangalokasikan serta mendealokasikan memori untuk menyimpan data jaringan. Kerangka obyek ini adalah:

```
const
GRAPHMAXNODE = 600;

type
TGraph = object
  n : integer;
  A : TDataMatrixReal;
  constructor InitFromFile
              (FileName:string);
  constructor InitCopy
              (var Graph:TGraph);
  destructor Done;
  function  Get (i,j:word) : Real;
  procedure Put (i,j:word; D:Real);
end;
```

Data yang ada pada obyek ini adalah:

- n : menyimpan banyaknya node
- A : data jarak antar node.

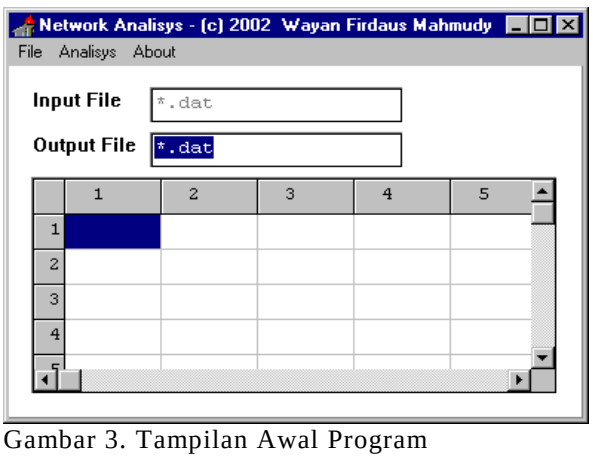
Metoda yang ada pada obyek ini adalah:

- InitFromFile : untuk mengambil data jaringan dari file untuk dimuat ke memori.
- InitCopy :menyalin data jaringan yang ada di memori ke alamat memori yang lain.

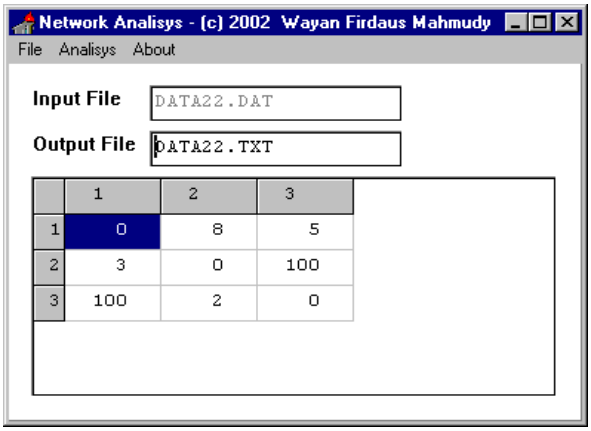
– Get : mendapatkan nilai elemen matrik jarak pada baris *i*, kolom *j*.
– Put : mengisi nilai elemen matrik jarak pada baris *i*, kolom *j*.

HASIL DAN PEMBAHASAN

Tampilan antar muka (*interface*) program dirancang secara visual dengan menggunakan fasilitas yang disediakan oleh Delphi. *Interface* program dirancang sedemikian rupa sehingga program mudah untuk digunakan. Pertama kali program dijalankan akan muncul tampilan seperti pada Gambar 3. Dengan memilih menu File, Open dan memilih file data yang akan diproses dihasilkan tampilan seperti pada Gambar 4. Setelah data dibuka, analisis dilakukan dengan memilih menu Analisys dan hasil analisis akan disimpan ke file.



Gambar 3. Tampilan Awal Program



Gambar 4. Tampilan Program Setelah File Dibuka

Pemecahan Persoalan Rute Terpendek (*Shortest Route*)

Misalkan terdapat masalah jaringan seperti pada Gambar 1. Hasil dari program adalah adalah file text yang berisi:

```
Step 1
    0     7     5    10   999   999   999
    7     0    12     5     9   999   999
    5    12     0     4   999     8   999
   10     5     4     0     8     7    20
  999     9   999     8     0   999     4
  999   999     8     7   999     0     5
  999   999   999    20     4     5     0
...
...
Step 7
    0     7     5     9    16    13    18
    7     0     9     5     9    12    13
    5     9     0     4    12     8    13
    9     5     4     0     8     7    12
   16     9    12     8     0     9     4
   13    12     8     7     9     0     5
   18    13    13    12     4     5     0

MATRIX OF SHORTEST DISTANCE
    0     7     5     9    16    13    18
    7     0     9     5     9    12    13
    5     9     0     4    12     8    13
    9     5     4     0     8     7    12
   16     9    12     8     0     9     4
   13    12     8     7     9     0     5
   18    13    13    12     4     5     0


MATRIX OF PATH
```

```
   0    0    0    3    2    3    6
   0    0    4    0    0    4    5
   0    4    0    0    4    0    6
   3    0    0    0    0    0    5
   2    0    4    0    0    7    0
   3    4    0    0    7    0    0
   6    5    6    5    0    0    0

PATH FROM A NODE TO ANOTHER NODE

From 1 to 2
From 1 to 3
From 1 to 3 to 4
From 1 to 2 to 5
…
From 7 to 6 to 3 to 1
From 7 to 5 to 2
From 7 to 6 to 3
From 7 to 5 to 4
From 7 to 5
From 7 to 6
```

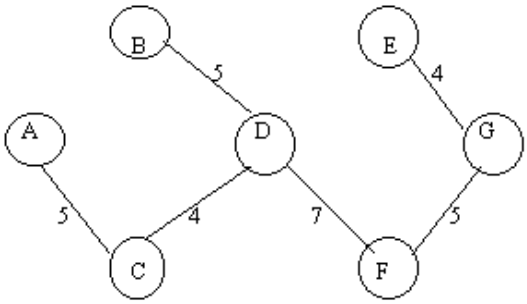
Pada hasil di atas terdapat dua matriks, yang pertama adalah matriks jarak minimum antar node dan matrik jalur terpendeknya.

Pemecahan Persoalan Rentang Pohon Minimal (*Minimal Spanning Tree*)

Untuk masalah jaringan seperti pada Gambar 1, hasil dari analisis untuk mencari rentang pohon minimal adalah:

```
CONNECTED NODE
   1 ->     3  Distance =     5
   3 ->     4  Distance =     4
   4 ->     2  Distance =     5
   4 ->     6  Distance =     7
   6 ->     7  Distance =     5
   7 ->     5  Distance =     4
Total distance =    30
```

Dari hasil diatas bisa digambarkan node yang harus dihubungkan sebagai berikut:



Gambar 5. Hasil Rentang Pohon Minimal

Pengujian Data Berukuran Besar

Pada uji coba data matriks berukuran 600x600 yang membutuhkan ruang memori 600x600x4 byte atau 1.37 Mbyte, program ini membutuhkan waktu hampir 3 menit untuk meyelesaikan masing-masing masalah di atas (ujicoba pada prosessor Pentium233 dan memori 64 Mbyte).

KESIMPULAN

- Struktur data graph yang telah dirancang dan diimplementasikan mampu untuk menyimpan data matriks berukuran besar sesuai kapasitas memori komputer.
- Program yang telah dibuat dengan memanfaatkan struktur data graph mampu memecahkan masalah jaringan (jarak terpendek dan rentang pohon minimal) dengan tepat dan memberikan pemecahan langkah demi langkah.

SARAN

- Perlu dikembangkan struktur data graph yang mampu memanfaatkan penyimpan eksternal (*disk*) sebagai tempat pemrosesan sementara (*temporer*) untuk memecahkan

masalah jaringan yang besar yang datanya tidak cukup disimpan di memori internal.

DAFTAR PUSTAKA

Dimyati, Tjutju Tarliah. (1992). *Operations Research, Model-model Pengambilan Keputusan*. Edisi Kedua. Sinar Baru, Bandung.

Kadir, Abdul. (1990). *Turbo Pascal Untuk IBM PC*. Elex Media Komputindo, Jakarta.

Santoso, Insap. (1992). *Struktur Data Menggunakan Turbo Pascal*. Andi Offset, Yogyakarta.

Schneider, G Michael. (1982). *An Introduction to Programming and Problem Solving with PASCAL*. Second Edition. John Wiley & Sons, Inc.

Schneider, G Michael. (1987). *Advanced Programming and Problem Solving with PASCAL*. Second Edition. John Wiley & Sons, Inc.

Taha, Hamdy A. (1982). *Operations Research, An Introduction*. Macmillan Publishing Co., Inc. New York.

Wirth, Niklaus. (1976). *Algorithms + Data Structures = Programs.* Prentice Hall Int.