MODUL 9

GRAPH

DASAR TEORI

A. Definisi dan Konsep Graph

Graph (Graf) adalah kumpulan dari simpul dan busur yang secara matematis dinyatakan sebagai:

$$G = (V, E)$$

G = Graph

V = Vertex/Node/Simpul/Titik

E = Edge/Busur/Ruas/Garis

Contoh: Suatu graph G(E,V) dengan elemen-elemen sebagai berikut:

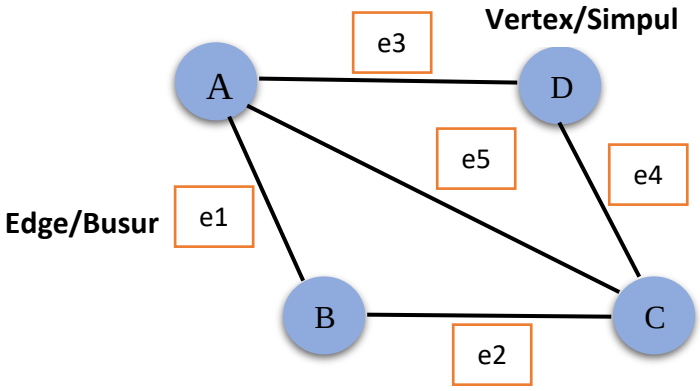
V mengandung 4 simpul: A, B, C, D

E mengandung 5 busur:

e1 = (A,B) e2 = (B,C) e3 = (A,D)

e4 = (C,D) e5 = (A,C)

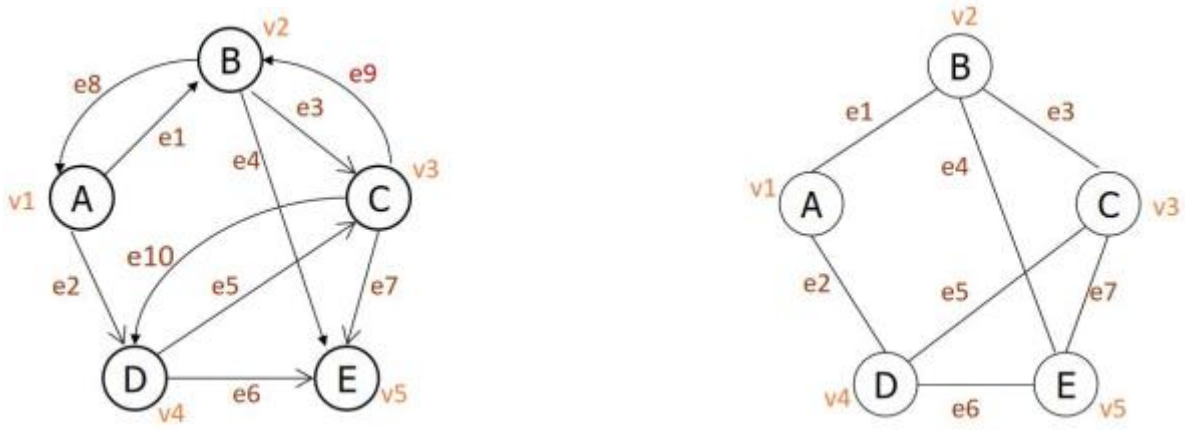
Dapat digambarkan:



Gambar 1 Contoh Graph

B. Jenis-Jenis Graph

Graph Berarah (Directed) dan Tak Berarah (Undirected)



Directed Graph

Undirected Graph

Gambar 2 Directed dan Undirected Graph

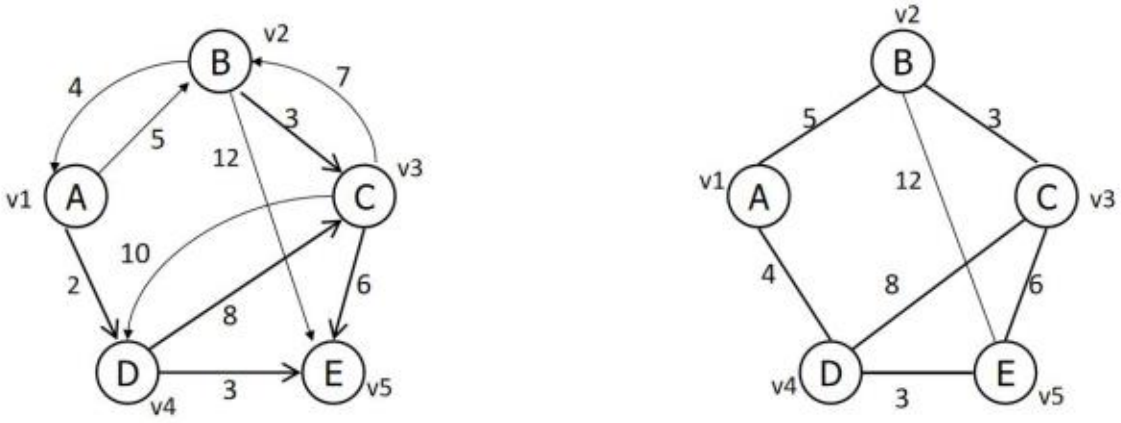
Graph berarah (directed graph):

Urutan simpul mempunyai arti. Misal busur AB adalah e1 sedangkan busur BA adalah e8.

Graph tak berarah (undirected graph):

Urutan simpul dalam sebuah busur tidak diperhatikan. Misal busur e1 dapat disebut busur AB atau BA.

Graph Berbobot (Weighted Graph)

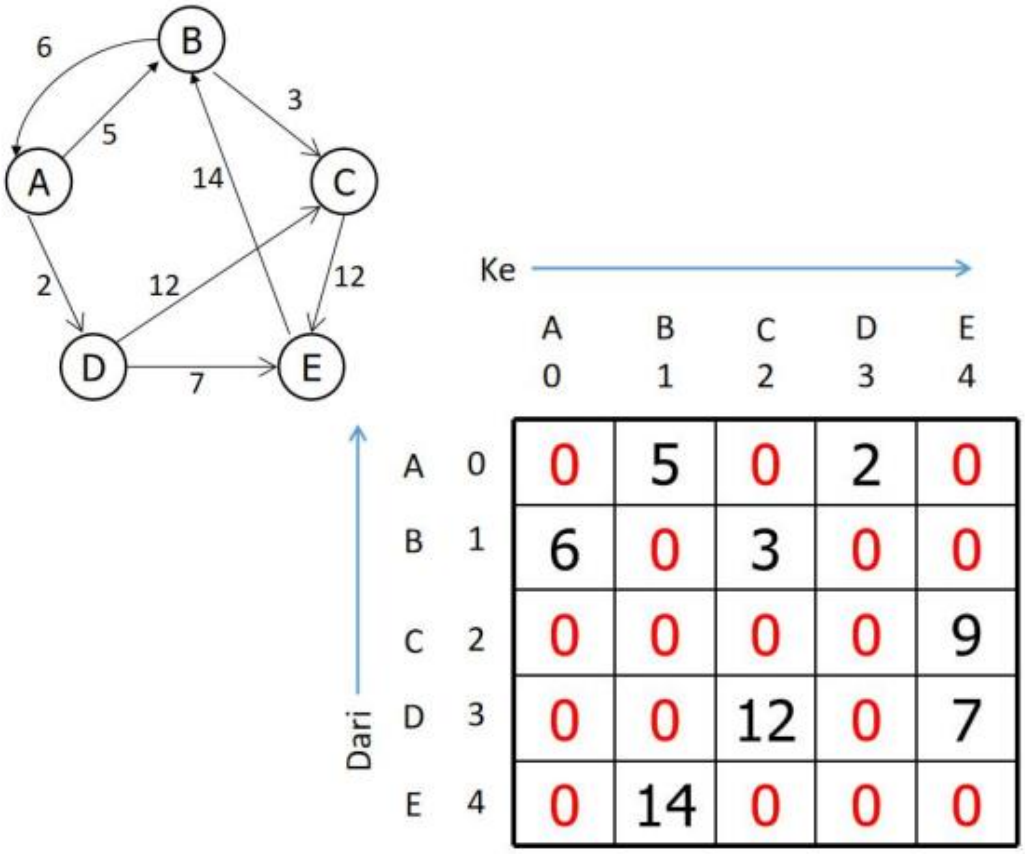


Gambar 3 Graph Berbobot

Jika setiap busur mempunyai nilai yang menyatakan hubungan antara 2 buah simpul, maka busur tersebut dinyatakan memiliki bobot.

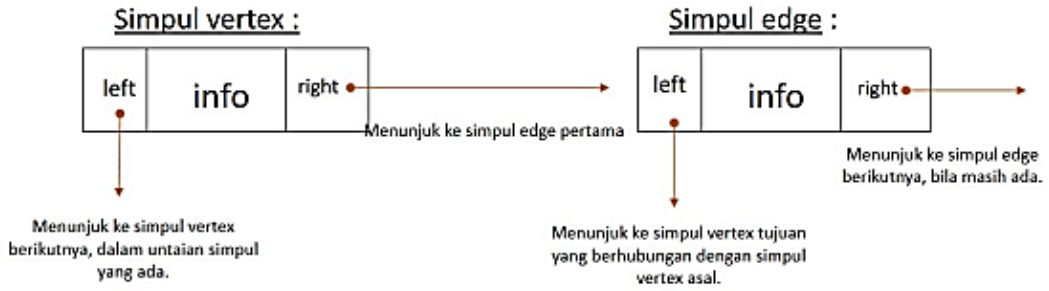
C. Representasi Graph

Representasi dengan Matriks



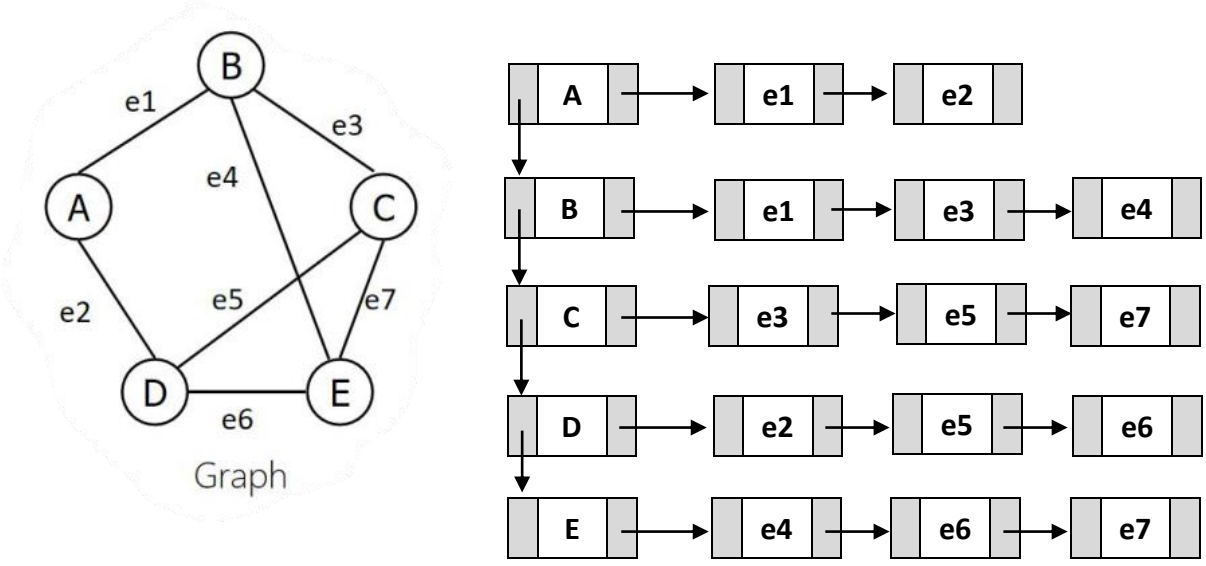
Gambar 4 Representasi Graph dengan Matriks

Representasi dengan Linked List



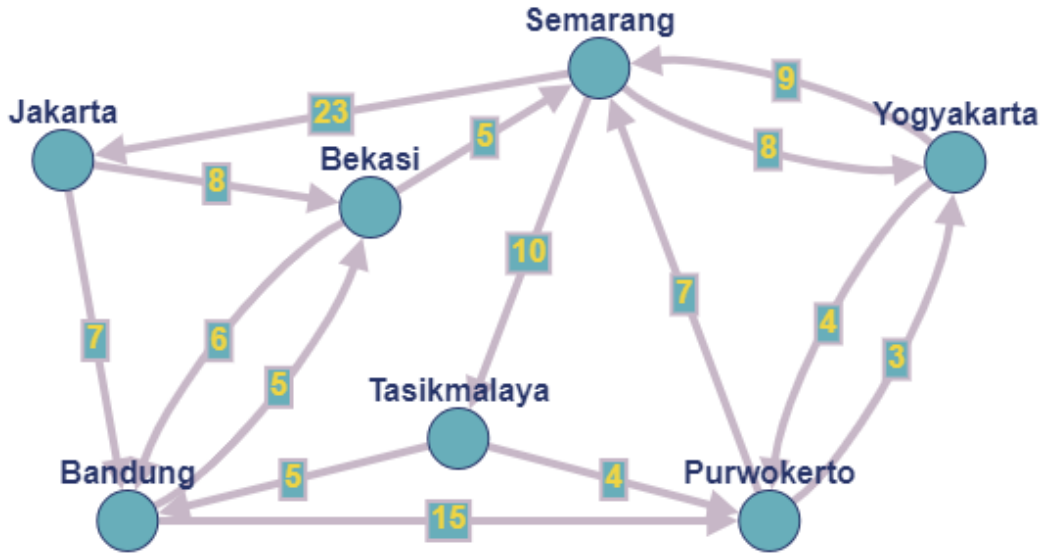
Gambar 5 Representasi Graph dengan Linked List

Yang perlu diperhatikan dalam membuat representasi graph dalam bentuk linked list adalah membedakan antara simpul vertex dengan simpul edge. Simpul vertex menyatakan simpul atau vertex dan simpul edge menyatakan busur (hubungan antar simbol). Struktur keduanya bisa sama bisa juga berbeda tergantung kebutuhan, namun biasanya disamakan. Yang membedakan antara simpul vertex dengan simpul edge nantinya adalah anggapan terhadap simpul tersebut juga fungsinya masing-masing.



Gambar 6 Representasi Graph dengan Linked List

GUIDED



Gambar 7 Ilustrasi Graph

Berdasarkan ilustrasi graph berarah dan berbobot di atas, berikut representasi dengan matriks.

Dari/Ke	Jakarta	Bandung	Bekasi	Tasikmalaya	Semarang	Purwokerto	Yogyakarta
Jakarta	0	**7**	**8**	0	0	0	0
Bandung	0	0	**5**	0	0	**15**	0
Bekasi	0	**6**	0	0	**5**	0	0
Tasikmalaya	0	**5**	0	0	0	**4**	0
Semarang	**23**	0	0	**10**	0	0	**8**
Purwokerto	0	0	0	0	7	0	**3**
Yogyakarta	0	0	0	0	**9**	**4**	0

Program Menampilkan Data Graph

```
#include <iostream>

using namespace std;

///PROGRAM MENAMPILKAN DATA GRAPH

//Deklarasi Node/Simpul
string simpul[7] = {"Jakarta",
                    "Bandung",
                    "Bekasi",
                    "Tasikmalaya",
                    "Semarang",
                    "Purwokerto",
                    "Yogyakarta"};

//Deklarasi Edge/Busur
int busur[7][7] = {
                    {0,7,8,0,0,0,0},
                    {0,0,5,0,0,15,0},
                    {0,6,0,0,5,0,0},
                    {0,5,0,0,0,4,0},
                    {23,0,0,10,0,0,8},
                    {0,0,0,0,7,0,3},
                    {0,0,0,0,9,4,0}};

//Tampil Data Graph
void tampilGraph()
{
    for (int baris = 0; baris < 7; baris++){
        cout << " " << simpul[baris] << ": ";
        for (int kolom = 0; kolom < 7; kolom++){
            if (busur[baris][kolom] != 0){
                cout << " " << simpul[kolom] << "(" << busur[baris][kolom]
<< ")" << " ";
            }
        }
    cout << endl;
    }
}

int main()
{
      tampilGraph();

      return 0;
```

```
}
```

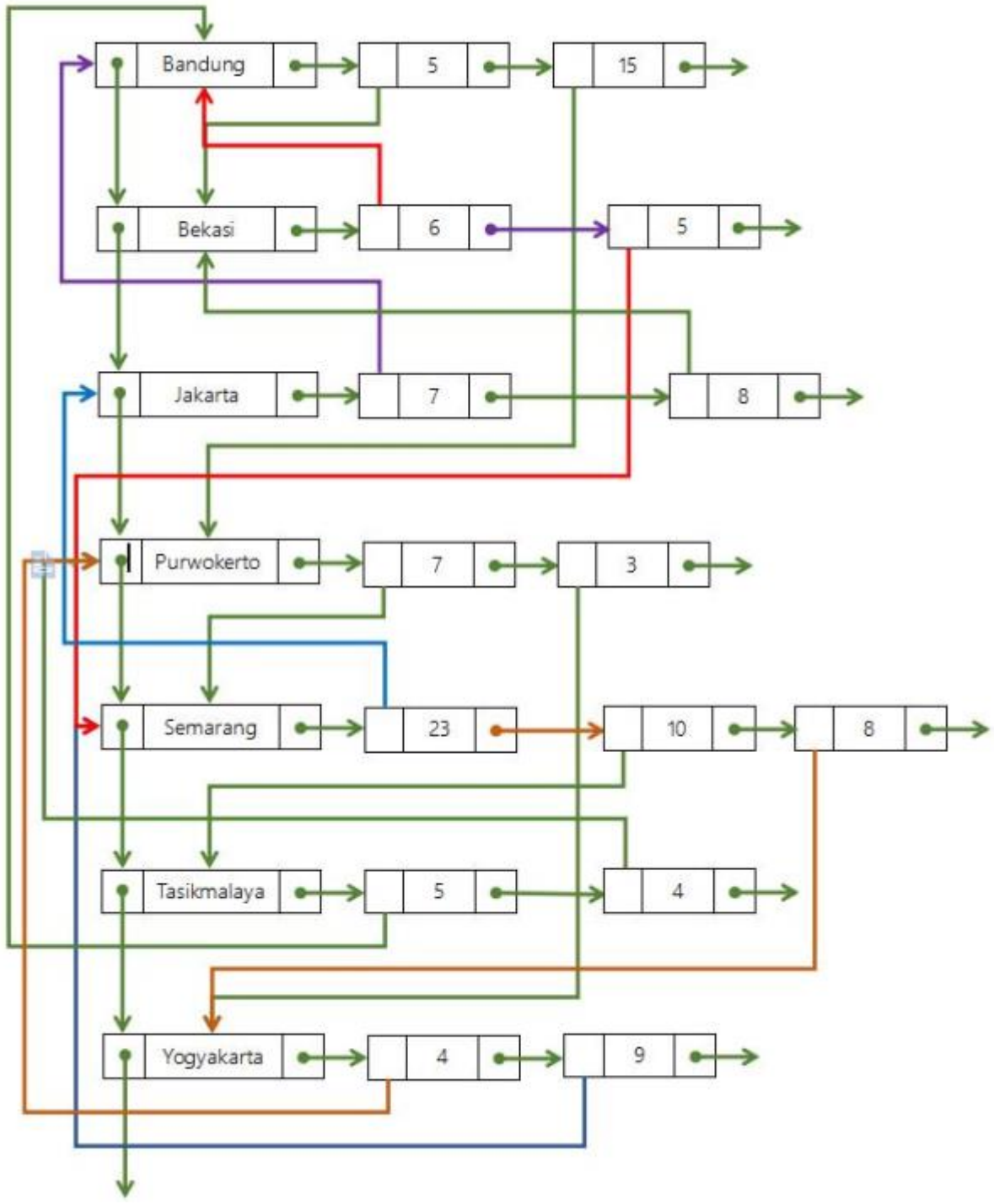
Program Representasi Graph dengan Matriks berdasarkan Input User

```
#include <iostream>

using namespace std;

///PROGRAM REPRESENTASI GRAPH DENGAN MATRIKS BERDASARKAN INPUT USER

//Deklarasi
int jumlahSimpul = 7;
string *dataSimpul; //digunakan untuk membuat array dinamis
int **dataBusur;    //digunakan untuk membuat array dinamis dua dimensi
bool cekMatriks = false;

//Buat Matriks Graph
void buatMatriks()
{
      dataSimpul = new string[jumlahSimpul];   //membuat array string
dinamis

      //Membuat simpul dengan jumlah yang dimasukkan user
      dataBusur = new int*[jumlahSimpul]; //membuat pointer array integer
dinamis
      dataBusur[0]    =    new    int[jumlahSimpul    *    jumlahSimpul];
//menginisialisasi array busur indeks-0 dengan array integer dinamis
berindeks pointer
      for (int i = 1; i < jumlahSimpul; i++){
            dataBusur[i] = dataBusur[i-1] + jumlahSimpul;
      }

      //Membuat matriks 2 dimensi dengan ukuran sesuai masukan user
      //Proses input nama simpul
      cout << " Silahkan masukkan nama simpul" <<endl;     //input nama
simpul sebanyak jumlah simpul

      for (int i = 0; i < jumlahSimpul; i++){
            cout << " Simpul " <<i+1<< " : ";
            cin >> dataSimpul[i];
      }

      //Proses input nilai busur (bobot)
```

```
      cout << " Silahkan masukkan bobot antar simpul" << endl;    //input
bobot busur antar simpul

      for (int baris = 0; baris < jumlahSimpul; baris++){
            for (int kolom = 0; kolom < jumlahSimpul; kolom++){
                  cout  <<  "  "  <<  dataSimpul[baris]  <<  "  -->  "  <<
dataSimpul[kolom] << " : ";  //input bobot sebagai baris dan kolom
                  cin >> dataBusur[baris][kolom];
            }
      }
      cout << endl;

      //Inisialisasi Pengecekan Matriks
      cekMatriks = true;  //atur keberadaan matriks menjadi ada/true
}

//Tampil Matriks Graph
void tampilMatriks()
{
      if (cekMatriks){    //jika matriks ada
          cout << "   ";
            for (int i = 0; i < jumlahSimpul; i++){ //membuat kolom nama
simpul
                  cout << dataSimpul[i] << " ";
            }
            cout << endl;
            for (int baris = 0; baris < jumlahSimpul; baris++){ //membuat
kolom bobot
                cout << " " << dataSimpul[baris] << " ";     //membuat
baris nama simpul
                  for (int kolom = 0; kolom < jumlahSimpul; kolom++){
//membuat baris bobot
                        cout << dataBusur[baris][kolom] << " ";
                  }
                  cout << endl;
            }
      }
      else{   //jika matriks belum ada
      cout << " Tidak ada matriks" << endl;
      }
}

int main()
{
      cout << " Silahkan masukkan jumlah simpul: ";    //input jumlah
simpul
      cin >> jumlahSimpul;
```

```
        buatMatriks();
        tampilMatriks();

        return 0;
}
```

Program Representasi Graph dengan Linked List berdasarkan Input User



```
#include <iostream>
#include <string>

using namespace std;

///PROGRAM REPRESENTASI GRAPH DENGAN LINKED LIST

//Deklarasi Global
int jumlahSimpul = 7;
string *dataSimpul; //digunakan untuk membuat array dinamis
int **dataBusur;    //digunakan untuk membuat array dinamis dua dimensi

//Deklarasi Linked List
struct Graph
{
    string data;
    Graph *kanan;
    Graph *kiri;
};

Graph *simpul, *busur, *awal, *akhir, **alamat, *bantu, *bantu2;

//Inisialisasi
void init()
{
    awal = NULL;
    akhir = NULL;
}

//isEmpty
int isEmpty()
{
    if (awal == NULL && akhir == NULL)
        return 1; //true
    else
        return 0; //false
}

//Buat Matriks
void buatMatriks()
{
    dataSimpul = new string[jumlahSimpul];
    dataBusur = new int*[jumlahSimpul];
    dataBusur[0] = new int[jumlahSimpul * jumlahSimpul]; //inisialisasi
ukuran elemen matriks

    for (int i = 1; i < jumlahSimpul; i++){
```

```
            dataBusur[i] = dataBusur[i-1] + jumlahSimpul;
      }

      //Membuat matriks 2 dimensi dengan ukuran sesuai masukan user
      //Proses input nama simpul
      cout << " Silahkan masukkan nama kota" <<endl;     //input nama
simpul sebanyak jumlah simpul

      for (int i = 0; i < jumlahSimpul; i++){
            cout << " Kota " <<i+1<< " : ";
            cin >> dataSimpul[i];
      }

      //Proses input nilai busur (bobot)
      cout << " Silahkan masukkan bobot antar kota" << endl;    //input
bobot busur antar simpul

      for (int m = 0; m < jumlahSimpul; m++){
            for (int n = 0; n < jumlahSimpul; n++){
                  cout << " " << dataSimpul[m] << " --> " << dataSimpul[n]
<< " : ";  //input bobot sebagai baris dan kolom
                  cin >> dataBusur[m][n];
            }
      }
}

//Buat Simpul Graph
void buatSimpulGraph()
{
    alamat = new Graph*[jumlahSimpul]; //membuat pointer alamat sebanyak
jumlah simpul
    buatMatriks();

    for (int i=0; i<jumlahSimpul; i++){
        if (isEmpty() == 1){
            simpul = new Graph;
            simpul->data = dataSimpul[i];
            simpul->kanan = NULL;
            simpul->kiri = NULL;
            awal = simpul;
            akhir = simpul;
            alamat[i] = awal;
        }
        else{
            simpul = new Graph;
            simpul->data = dataSimpul[i];
            akhir->kiri = simpul;
```

```
            akhir = simpul;
            simpul->kiri = NULL;
            simpul->kanan = NULL;
            alamat[i] = akhir;
        }
    }

    bantu = awal;

    for (int m=0; m<jumlahSimpul; m++){
        bantu2 = bantu;
        for (int n=0; n<jumlahSimpul; n++){
            if (dataBusur[m][n] != 0){
                simpul = new Graph;
                simpul->data = to_string(dataBusur[m][n]);
                bantu2->kanan = simpul;
                simpul->kiri = alamat[n];
                simpul->kanan = NULL;
                bantu2 = simpul;
            }
        }
        bantu = bantu->kiri;
    }
}

//Tampil
void tampilGraph()
{
    if (isEmpty() == 0){
        bantu = awal;
        while (bantu != NULL){
            cout << ends << bantu->data << ": ";
            bantu2 = bantu->kanan;

            while (bantu2 != NULL){
                cout << ends << bantu2->kiri->data << ": " << bantu2-
>data << " ";
                bantu2 = bantu2->kanan;
            }
            cout << endl;
            bantu = bantu->kiri;
        }
    }
    else{
        cout << " Graph masih kosong!" << endl;
    }
}
```

```
int main()
{
    init();

    cout << " Silahkan masukkan jumlah kota: ";
    cin >> jumlahSimpul;

    buatSimpulGraph();
    tampilGraph();

    return 0;
}
```

TUGAS

Modifikasi program Guided III dengan menambahkan fungsi untuk mencari busur terpendek!

~ SELAMAT MENGERJAKAN 😊 ~