APLIKASI GRAF DALAM BISNIS TRAVEL BANDUNG-BOGOR

Achmad Giovani – NIM : 13508073

Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganeca 10 Bandung
e-mail: if18073@students.if.itb.ac.id

ABSTRAK

Makalah ini membahas tentang aplikasi graf dalam bisnis travel dengan rute Bandung-Bogor. Graf terdiri dari simpul dan jalur penghubung antar-simpul. Dengan merepresentasikan jalur yang dilalui sebuah mobil angkutan antar-jemput dengan graf, maka dapat dicari jalur terpendek yang dilalui. Dalam makalah ini akan dijelaskan algoritma untuk mencari rute terpendek. Dimisalkan ada tujuh penumpang, sehingga ada tujuh titik di Bandung dan tujuh titik di Bogor. Setelah menemukan jalur terpendek, kemudian dihitung perkiraan biaya operasional mobil tersebut, dalam hal ini adalah pemakaian bahan bakar. Untuk mencari jalur terpendek, dalam makalah ini dijelaskan dua cara, yaitu cara manual dengan menggambar graf beseta bobotnya, dan dengan cara membuat sebuah program yang dapat menentukan jalur terpendek sekaligus menghitung perkiraan biaya operasional sebuah mobil angkutan antar-jemput.

Kata kunci: Graf, Bobot, Algoritma, Program, Jalur, Terpendek, Biaya, Operasional

1. PENDAHULUAN

Graf merupakan salah satu struktur data yang sangat berguna dalam informatika. Berbagai objek diskrit dapat direpresentasikan dengan menggunakan graf. Objek yang dapat direpresentasikan sebagai graf adalah objek yang dapat digambarkan menjadi beberapa simpul dan beberapa lintasan/sisi/*edge*. Contoh objek diskrit misalnya Jembatan Königsberg dan senyawa kimia.

Selain kedua contoh yang telah disebutkan, graf juga dapat merepresentasikan kota/lokasi yang saling terhubung oleh jalan/lintasan. Dalam makalah ini, permasalahannya dibuat lebih khusus, yaitu merepresentasikan rute perjalanan angkutan antar-jemput (*travel*) Bandung-Bogor sebagai graf. Lokasi merupakan simpul dan jalan merupakan lintasan.

Pokok permasalahan yang dibahas adalah bagaimana cara menentukan rute perjalanan terpendek untuk meminimalisasi biaya operasional kendaraan *travel*, sehingga tarif perjalanan akan murah. Untuk menghitung estimasi biaya operasional kendaraan *travel*, yang pertama dilakukan adalah menentukan jalur terpendek yang akan dilalui, menghitung jarak yang telah ditempuh, kemudian menghitung estimasi biaya berdasarkan konsumsi BBM kendaraan yang digunakan.

2. DEFINISI DAN DESKRIPSI GRAF

2.1 Definisi Graf

Graf terdiri dari himpunan tidak-kosong dari simpul-simpul (*vertices*) dan himpunan sisi/lintasan yang menghubungkan sepasang simpul. Secara matematis dapat ditulis sebagai berikut:

Graf $G = (V,E)$

$V = \{v_1,v_2,..,v_n\}$ = himpunan tidak-kosong dari simpul-simpul (*vertices*)

$E = \{e_1,e_2,\ldots,e_n\}$ =himpunan sisi/lintasan yang menghubungkan sepasang simpul

2.2. Deskripsi Graf

Jenis-jenis graf:

- Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf
  a. Graf sederhana
     Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana.

  b. Graf tak-sederhana
     Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*).

- Berdasarkan orientasi arah pada sisi
  a. Graf tak-berarah
     Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah

  b. Graf berarah
     Graf yang setiap sisinya mempunyai orientasi arah disebut graf berarah

Terminologi graf:

- Ketetanggaan

  Dua buah simpul dikatakan bertetangga bila keduanya terhubung langsung

- Bersisian

  Untuk sembarang sisi $e = (v_j, v_k)$ dikatakan

  e bersisian dengan simpul v_j , atau
  e bersisian dengan simpul v_k

- Simpul Terpencil

  Simpul terpencil ialah simpul yang tidak mempunyai sisi yang bersisian dengannya.

- Graf Kosong

  Graf yang himpunan sisinya merupakan himpunan kosong (N_n).

- Derajat

  Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Notasi: $d(v)$

- Lintasan

  Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n di dalam graf G ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang berbentuk $v_0, e_1, v_1, e_2, v_2, \ldots , v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, ... , $e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G.

- Siklus atau Sirkuit

  Lintasan yang berawal dan berakhir pada simpul yang sama disebut sirkuit atau siklus.

- Terhubung

  Dua buah simpul v_1 dan simpul v_2 disebut terhubung jika terdapat lintasan dari v_1 ke v_2.

  G disebut graf terhubung (*connected graph*) jika untuk setiap pasang simpul v_i dan v_j dalam himpunan V terdapat lintasan dari v_i ke v_j.

  Jika tidak, maka G disebut graf tak-terhubung (*disconnected graph*).

- Upagraf Rentang

  Upagraf $G_1 = (V_1, E_1)$ dari $G = (V, E)$ dikatakan upagraf rentang jika $V_1 = V$ (yaitu G_1 mengandung semua simpul dari G).

- Cut-set

  Cut-set dari graf terhubung G adalah himpunan sisi yang bila dibuang dari G menyebabkan G tidak terhubung. Jadi, *cut-set* selalu menghasilkan dua buah komponen.

- Graf Berbobot

  Graf berbobot adalah graf yang setiap sisinya diberi sebuah harga (bobot). Terminologi inilah yag menjadi

Beberapa graf khusus:

- Graf Lengkap

  Graf lengkap ialah graf sederhana yang setiap simpulnya mempunyai sisi ke semua simpul lainnya. Graf lengkap dengan n buah simpul dilambangkan dengan K_n. Jumlah sisi pada graf lengkap yang terdiri dari n buah simpul adalah $n(n-1)/2$.

- Graf Lingkaran

  Graf lingkaran adalah graf sederhana yang setiap simpulnya berderajat dua. Graf lingkaran dengan n simpul dilambangkan dengan C_n.

- Graf Teratur

  Graf yang setiap simpulnya mempunyai derajat yang sama disebut graf teratur. Apabila derajat setiap simpul adalah r, maka graf tersebut disebut sebagai graf teratur derajat r. Jumlah sisi pada graf teratur adalah $nr/2$.

Lintasan dan Sirkuit Euler:

- Lintasan Euler ialah lintasan yang melalui masing-masing sisi di dalam graf tepat satu kali.

- Sirkuit Euler ialah sirkuit yang melewati masing-masing sisi tepat satu kali..

- Graf yang mempunyai sirkuit Euler disebut graf Euler (*Eulerian graph*). Graf yang mempunyai lintasan Euler dinamakan juga graf semi-Euler (*semi-Eulerian graph*).

Graf tidak berarah memiliki lintasan Euler jika (graf semi-Euler) dan hanya jika terhubung dan memiliki dua buah simpul berderajat ganjil atau tidak ada simpul berderajat ganjil sama sekali.

Graf tidak berarah *G* adalah graf Euler (memiliki sirkuit Euler) jika dan hanya jika setiap simpul berderajat genap.

Graf berarah *G* memiliki sirkuit Euler jika dan hanya jika *G* terhubung dan setiap simpul memiliki derajat-masuk dan derajat-keluar sama.

G memiliki lintasan Euler jika dan hanya jika *G* terhubung dan setiap simpul memiliki derajat-masuk dan derajat-keluar sama kecuali dua simpul, yang pertama memiliki derajat-keluar satu lebih besar derajat-masuk, dan yang kedua memiliki derajat-masuk satu lebih besar dari derajat-keluar.

Lintasan dan Sirkuit Hamilton:

- Lintasan Hamilton ialah lintasan yang melalui tiap simpul di dalam graf tepat satu kali.
- Sirkuit Hamilton ialah sirkuit yang melalui tiap simpul di dalam graf tepat satu kali, kecuali simpul asal (sekaligus simpul akhir) yang dilalui dua kali.
- Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi-Hamilton.

Syarat cukup supaya graf sederhana *G* dengan n (≥ 3) buah simpul adalah graf Hamilton ialah bila derajat tiap simpul paling sedikit $n/2$ (yaitu, $d(v) \geq n/2$ untuk setiap simpul v di G). *(coba nyatakan dalam "jika p maka q")*

Setiap graf lengkap adalah graf Hamilton.

Di dalam graf lengkap *G* dengan n buah simpul ($n \geq 3$), terdapat $(n-1)!/2$ buah sirkuit Hamilton.

Di dalam graf lengkap *G* dengan n buah simpul ($n \geq 3$ dan n ganjil), terdapat $(n-1)/2$ buah sirkuit Hamilton yang saling lepas (tidak ada sisi yang beririsan). Jika n genap dan $n \geq 4$, maka di dalam G terdapat $(n-2)/2$ buah sirkuit Hamilton yang saling lepas.

Beberapa aplikasi graf:

- Lintasan terpendek
- Persoalan pedagang keliling
- Persoalan tukang pos Cina
- Pewarnaan graf

3. STRUKTUR DATA GRAF

Dalam makalah ini jenis graf yang digunakan adalah graf berbobot. Bobot sebuah sisi merepresentasikan jarak atar-wilayah. Sebuah graf memiliki 3 elemen utama, yaitu simpul, sisi/lintasan, dan bobot sisi. Struktur data graf cukup sederhana, hanya saja pemakaiannya terkadang cukup rumit.

Sebuah simpul dapat memiliki atau tidak memiliki simpul tetangga. Jika tidak memiliki simpul tetangga (*neighbour*), simpul tersebut merupakan simpul terpencil. Setiap simpul memiliki jumlah simpul tetangga yang dapat berbeda-beda.

Untuk kasus *travel*, setiap simpul yang melambangkan tempat penumpang naik dan turun (dalam kota) harus semuanya dilewati, karena jika tidak, akan ada penumpang yang tidak terangkut dan/atau tidak sampai tujuan. Karena itu setiap simpul memiliki tipe *boolean: vis* (*visited*/telah dikunjungi). Pada keadaan akhir (*final state*) semua simpul tempat penumpang naik-turun harus pernah dikunjungi paling sedikit satu kali. Sedangkan simpul yang boleh tidak dilewati (misal Jakarta dan Padalarang), *vis=false* jika tidak dilewati, *vis=true* jika dilewati. Pada keadaan akhir, salah satu simpul *vis*-nya harus *true*, tidak boleh kedua-duanya *true* atau kedua-duanya *false*. Definisi tipe-nya adalah sebagai berikut:

```
type node <
            nd : character
                   {nama node}
            nei : array of character
              {daftar node tetangga}
            vis : boolean
              {node-nya sudah
dikunjungi atau belum}
          >
```

Koneksi pada graf direpresentasikan dengan list *edge*. Contoh list-nya adalah:

Tabel 1. Contoh Representasi Hubungan antar-simpul Graf dengan Menggunakan List Edge

Elemen List	Simpul 1	Simpul 2	Bobot
AB	A	B	4.0
BC	B	C	5.5
AC	A	C	3.5

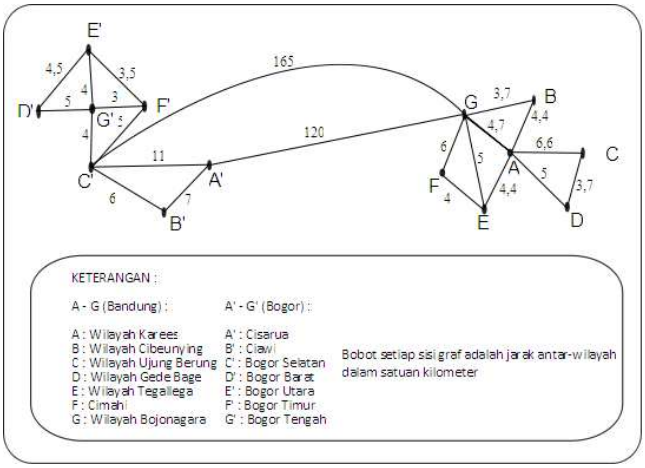
Elemen list adalah *array of character* yang terdiri dari dua karakter. Karakter pertama adalah simpul pertama, karakter kedua adalah simpul kedua. Sedangkan bobot sisi adalah bilangan *real*. Daftar koneksi didefinisikan sebagai larik dari koneksi. Selain larik koneksi, daftar koneksi mengandung jumlah koneksi (*integer*).

Definisi tipe-nya adalah sebagai berikut:

```
type conn <
          c : array of character {jml
elemen maksimum = 2}
          jarak : real
          {jarak antar node}
          >
type listkoneksi <
                  con : array of conn
                  jumkoneksi : integer
                 >
```

4. APLIKASI GRAF DALAM BISNIS TRAVEL BANDUNG-BOGOR

Jalan yang menghubungkan antar-wilayah yang dilalui travel dapat digambarkan sebagai berikut :



Gambar 1. Representasi Graf untuk Rute Perjalanan Bandung-Bogor

Jika gambar di atas kurang jelas terlihat, perbesarannya dapat dilihat pada lampiran.

Ada dua cara yang dapat dilakukan untuk menentukan rute terpendek yang dilalui sebuah kendaraan *travel*. Cara pertama adalah dengan cara manual, yaitu menggambar graf kemudian mencari rute terpendek dengan menghitung jarak yang ditempuh secara manual. Untuk jalur yang tidak ada atau sedikit percabangannya, cara ini mudah dilakukan, tetapi bila banyak simpul yang memiliki simpul tetangga cukup banyak, akan merepotkan dan memakan waktu lama.

Cara kedua adalah menggunakan program yang dapat menentukan dengan jalur terpendek dengan algoritma tertentu. Untuk permasalahan mencari jalur terpendek, deskripsi algoritmanya adalah:

- Tentukan titik awal dan titik akhir
- *Assign* semua titik yang akan dilewati (termasuk titik akhir), kecuali titik awal sebagai "belum dikunjungi". Sedangkan titik awal di-*assign* sebagai "telah dikunjungi"
- Pilih simpul tetangga yang memiliki koneksi dengan titik awal, kemudian simpan nilai bobot/jarak jika telah menemukan jalur menuju simpul "terdekat"
- Proses pemilihan berlangsung sebanyak n-1 kali, dimana n adalah jumlah simpul tetangga.
- Setelah itu, titik awal digeser ke simpul yang "jaraknya" paling dekat. *Assign* titik awal yang baru sebagai "sudah dikunjungi".
- Jarak total yang sebelumnya 0, ditambah dengan nilai jarak hasil pemilihan simpul terdekat
- Pergeseran simpul dan penyimpanan nilai jarak total dilakukan hingga titik awal sama dengan titik akhir.

Untuk dapat membentuk program, diperlukan fungsi-fungsi pendukung, yaitu:

- Fungsi koneksi (untuk menghitung jarak antar-simpul)
- Fungsi min (untuk menentukan nilai minimun dari dua buah bilangan real)
- Fungsi allnodevisited (untuk menentukan apakah sebuah simpul sudah pernah dikunjungi atau belum)
- Fungsi idxlast (menghasilkan indeks terakhir dari larik karakter)
- Fungsi jarakterpendek (fungsi utama, yang menghasilkan jarak terpendek yang ditempuh oleh kendaraan)
- Fungsi hitungbiaya (untuk menghitung biaya operasional berdasarkan jarak yang ditempuh, harga BBM, dan konsumsi BBM per liternya)

Fungsi jarakterpendek hanya perlu digunakan saat akan menentukan rute terpendek di dalam kota, jika digunakan untuk menentukan rute terpendek antarkota, kurang begitu berguna karena dapat dengan mudah dikerjakan secara manual (misalnya sisi GA' dan GC').

Dengan adanya fungsi-fungsi diatas, maka program utama pun dapat dibuat dengan mudah. Dalam program utama, fungsi yang dipanggil secara langsung hanya fungsi hitungbiaya. Tetapi, jika fungsi hitungbiaya dipanggil, fungsi-fungsi lain juga dipanggil secara tidak langsung.

Algoritma beberapa fungsi yang dipakai adalah sebagai berikut:

```
function koneksi (N : node, M :
node) -> real
       kamus lokal
              i : integer
              L : listkoneksi
       ALGORITMA
              L.con[i].c[1]:=N
              L.con[i].c[2]:=M

              ->L.con[i].jarak
```

```
function min (a : real, b : real)
-> real
       kamus lokal

       ALGORITMA
              if (a <= b) then
                     -> a
              else
                     -> b
              endif
```

```
function allnodevisited() ->
boolean
       kamus lokal
              A,B,C,D,E,F,G : node
       ALGORITMA
              if (A.vis and B.vis
and C.vis and D.vis and
                     E.vis and
F.vis and G.vis) then
                     -> true
              else
                     -> false
              endif
```

```
function idxlast (c : array of
character) -> integer
       kamus lokal
              i,j : integer
       ALGORITMA
              i := 1
              j := i + 1
              while (c[j] <> Nil)
                     i := i + 1
                     j := j + 1
              endwhile
              -> i
```

Algoritma program dan fungsi yang lain dapat dilihat pada lampiran.

Sebagai contoh, ada tujuh penumpang dari Bandung yang akan dijemput mobil *travel*. Misalkan kantor *travel* ada di A (Bandung) dan di E' (Bogor). Maka A adalah titik awal dan E' adalah titik akhir. Gunakan fungsi jarakterpendek, rute terpendek saat di Bandung adalah A-C-D-A-B-G-F-E-G (jarak tempuh = 6,6 + 3,7 + 5 + 4,4 + 3,7 + 6 + 4 + 5 = 38,4 km). setelah itu ada dua pilihan untuk sampai di Bogor, dari simpul G ke C' atau A'? Pilih A' karena lebih dekat (jarak tempuh = 38,4 + 120 = 158,4 km). Setelah sampai di A' tentunya harus lewat B' karena B' dan C' harus dikunjungi. Jalur terpendek adalah A'-B'-C' (jarak tempuh = 158,4 + 7 + 6 = 171,4 km). Di C' kembali gunakan fungsi jarakterpendek, hasilnya adalah rute C'-G'-F'-E'-G'-D'-E' (jarak total = 171,4 + 4 + 4 + 3,5 + 3 + 5 + 4,5 = 194,4 km).

Misalkan konsumsi BBM rata-rata mobil adalah 10 km/liter dan BBM yang dipakai adalah premium, biaya operasional (hanya pemakaian BBM) adalah (194,4*4500) div 10 = 87.480 rupiah. Berarti setiap penumpang membayar sebanyak 12.500 untuk pemakaian BBM, sisanya biaya lain-lain termasuk sewa mobil dan biaya perawatan. Tarif travel di indonesia bervariasi, tergantung jauh dekatnya tujuan. Kalau travel Bandung-Bogor yang biasa saya pakai, biayanya 65.000 rupiah jika lewat Puncak atau 85.000 jika lewat Cipularang. Berarti, biaya di luar pemakaian BBM adalah antara 52.500 sampai dengan 72.500.

5. KESIMPULAN

Graf dapat merepresentasikan banyak persoalan, salah satunya adalah menentukan rute terpendek perjalanan dalam kota dan antar-kota, dalam hal ini adalah perjalanan mobil *travel*. Yang pertama dilakukan adalah menggambar graf yang sesuai, kemudian buat daftar koneksi antar-simpul, kemudian jalankan program.

Program dibuat menggunakan struktur data graf untuk merepresentasikan koneksi antar-simpul. Program ini efektif digunakan pada kondisi dimana banyak percabangan antara titik awal dan titik akhir, misalnya di dalam kota. Jika tidak banyak percabangan, dapat dipakai cara manual. Jika kedua kondisi ada (banyak percabangan dan sedikit percabangan), sebaiknya program dipakai agar proses pemilihan jalur dapat dilakukan dengan cepat dan mudah.

DAFTAR REFERENSI

[1] Munir, Rinaldi. (2003). Matematika Diskrit. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Liem, Inggriani. (2008). Diktat Struktur Data. Departemen Teknik Informatika, Institut Teknologi Bandung.

[3] http://maps.google.com/ (tanggal akses 7 Desember 2009 pukul 14.30)

LAMPIRAN

Definisi tipe dan fungsi-fungsi dasar:

```
type node <
                    nd : character
                    nei : array of character   {daftar node tetangga}
                    vis : boolean                          {node-nya sudah

dikunjungi atau belum}
             >
type conn <
                    c : array of character {jml elemen maksimum = 2}
                    jarak : real           {jarak antar node}
             >
type listkoneksi<
                    con : array of conn
                    jumkoneksi : integer
             >
```

```
function koneksi (N : node, M : node) -> real
       kamus lokal

              i : integer
              L : listkoneksi
       ALGORITMA
              L.con[i].c[1]:=N
              L.con[i].c[2]:=M

              ->L.con[i].jarak
```

```
function min (a : real, b : real) -> real
       kamus lokal

       ALGORITMA
              if (a <= b) then
                     -> a
              else
                     -> b
              endif
```

```
function allnodevisited() -> boolean
       kamus lokal
              A,B,C,D,E,F,G : node
       ALGORITMA
              if (A.vis and B.vis and C.vis and D.vis and
                     E.vis and F.vis and G.vis) then
                     -> true
              else
                     -> false
              endif
```

```
function idxlast (c : array of character) -> integer
       kamus lokal
              i,j : integer
       ALGORITMA
              i := 1
              j := i + 1
              while (c[j] <> Nil) do
                     i := i + 1
                     j := j + 1
              endwhile
              -> i
```

Dua fungsi utama yang dipakai:

```
function jarakterpendek (nodeawal : node, nodeakhir : node) -> real
       kamus lokal
              i : integer
              temp : real
              jalur : real
              initnode : node
              node D,E,F,G : node
              nextnode: node
              L : listkoneksi
              n : integer
       ALGORITMA
              i := 1 {inisialisasi}
              temp := 0
              nodeawal := C
              initnode := nodeawal       {initial node (node awal)}
              C.vis := true
              D.vis := false
              E.vis := false
              F.vis := false
              G.vis := false
              n := idxlast(C.nei)

              jalur = min(koneksi(c,g),koneksi(c,f))

              if (jalur = L.con[i].jarak) then
                     nextnode.nd := L.con[i].c[2]
              endif

              while (initnode <> nodeakhir) do
                     for i=1 to n-1 do
                            jalur := min(koneksi(initnode,N1.nei[i]),
                                   koneksi(initnode,N1.nei[i+1]))
                     endfor
                     initnode := nextnode
                     initnode.vis := true
                     temp := temp + jalur
              endwhile

              if (allnodevisited()) then
                     -> temp
              endif
```

```
function hitungbiaya (nAw : node, nAkh : node, KB : real, hargabbm : integer) ->

integer
       kamus lokal

       ALGORITMA
              -> (jarakterpendek(nAw,nAKh)*hargabbm) div KB
              {KB : konsumsi bbm kendaraan (km/liter)}
```

Program utama:

```
PROGRAM HitungBiayaOpr
 {program menentukan jalur/rute terpendek dari suatu graf yang
  merepresentasikan jalur antar-wilayah serta jaraknya dalam
  satuan kilometer}
```

```
KAMUS
  hslbiaya : integer {hasil akhir program ini}
  kBBM : integer     {konsumsi BBM dalam km/liter}
  hBBM : integer     {harga BBM dalam rupiah}
  C,E : node       {titik awal dan akhir}

  function koneksi (N : node, M : node) -> real

  function min (a : real, b : real) -> real

  function allnodevisited() -> boolean

  function idxlast (c : array of character) -> integer

  function jarakterpendek (nodeawal : node, nodeakhir : node) -> real

  function hitungbiaya (nAw : node, nAkh : node, KB : real, hargabbm : integer) ->

integer
```

```
ALGORITMA
  hslbiaya := hitungbiaya (C,E,kBBM,hBBM)
```

Gambar Representasi Graf Rute Perjalanan Travel Bandung-Bogor: