

TUGAS 2

Dasar-Dasar Pemrograman 2 CSGE601021 Semester Genap 2016/2017

Batas waktu pengumpulan:

Minggu, 30 April 2017 pukul 23.55 Waktu Scele

Tujuan dari Tugas ini adalah untuk menguji pemahaman Anda terhadap materi bahan kuliah yang telah diajarkan di kelas. Mahasiswa **harus menuliskan sendiri** solusi/kode program dari soal yang diberikan tanpa bantuan orang lain. Belajarlah menjadi mahasiswa yang mematuhi integritas akademik. **Sikap Jujur merupakan sebuah sikap yang dimiliki mahasiswa Fasilkom UI.**

Peringatan: Jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!

TUGAS 2

Teman Anda, Nadiem Makara, baru saja merintis sebuah *start-up company* berbasis teknologi informasi. *Start-up* yang dinamakan OW-JEK tersebut rencananya akan menyasar transportasi umum yang sudah dikenal luas oleh masyarakat, yaitu, ojek. Ide dari Nadiem Makara ialah menyediakan aplikasi untuk para pengemudi ojek sehingga para pengemudi ojek dan penumpang dapat dengan mudah dipertemukan. *Start-up* Nadiem Makara tersebut sudah dilirik oleh banyak investor. Mereka menginginkan sebuah *prototype* sebagai *proof of concept*. Karena Nadiem Makara tidak memiliki latar belakang maupun pengalaman dalam bidang IT, dia meminta bantuan Anda untuk membangun sebuah *prototype* sistem dari aplikasi tersebut untuk dipresentasikan kepada investor.

Requirement yang diberikan adalah sebagai berikut: Anda akan diberikan map yang merepresentasikan sebuah kota mini. Map tersebut pada dasarnya ialah sebuah matriks dua dimensi yang berukuran 50 (*height*) x 100 (*width*). Secara umum, ada dua karakter dalam map, yaitu, karakter kres ('#') yang merepresentasikan bangunan atau rumah dan karakter spasi (' ') yang merepresentasikan jalan. Setiap titik pada map ditandai oleh sebuah koordinat. Representasi lokasi baris pada map dimulai dari A0 s.d. E9 sedangkan representasi lokasi kolom berada pada nilai Q0 s.d. Z9. Koordinat lokasi sebuah titik ialah gabungan dari lokasi baris dan lokasi kolom, contohnya, lokasi titik A5X7 merupakan titik yang berada pada baris A5 dan kolom X7. Anda tidak perlu khawatir mengenai map tersebut, karena, denah dan sebagian fungsinya sudah dikerjakan oleh programmer lain. Anda hanya perlu *me-reuse* class Map yang sudah ada. Denah dari map yang akan digunakan adalah sebagai berikut:

```
Q      R      S      T      U      V      W      X      Y      Z
012345678901234567890123456789012345678901234567890123456789
A0 #####
1 # #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 # #####
8 # #####
9 #####
B0 # #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 #
9 # #####
C0 # #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 # #####
D0 ##
1 ##
2 ##
3 ##
4 ##
5 ##
6 ##
7 ##
8 ##
9 ##
E0 #####
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
```

Skala 1 karakter : 100 meter

Fungsionalitas yang diinginkan dari *prototype* yang akan dibangun ialah melakukan **kalkulasi biaya jasa** dengan input yang diberikan oleh user yaitu titik keberangkatan dan titik tujuan. Tentu saja, sebelum melakukan kalkulasi, *prototype* harus terlebih dahulu menentukan jalur yang digunakan, dimana, sistem harus memastikan jalur tersebut merupakan **jalur tercepat (shortest path)** di antara kemungkinan jalur-jalur. Untuk perhitungan biasa jasa, harga akan disesuaikan dengan tipe dari service. Dalam *prototype* ini, ada tiga tipe, yaitu, OW-JEK Regular, OW-JEK Sporty, dan OW-JEK Exclusive. Perbedaan biaya jasa antara ketiga tipe tersebut ialah sebagai berikut:

<i>type</i>	Regular	Sporty	Exclusive
<i>minYearAllowed</i>	2012	2015	2016
<i>minTopSpeed (kmh)</i>	-	140	-
<i>minCc (cc)</i>	-	-	500
<i>costPerKm</i>	1.000	3.000	5.000
<i>first2KmCost</i>	3.000	-	-
<i>first5KmCost</i>		20.000,00	
<i>fixedCost</i>	-	-	10.000
<i>protectionCost</i>	-	10% (from total)	5% (from total)
<i>promo</i>	40% (6 km pertama)	60% (8 km pertama)	50% (dari biaya total)

Bisa dilihat pada tabel diatas, perhitungan biaya jasa untuk masing-masing tipe berbeda-beda. Pada tipe regular, perhitungan biaya dilakukan dengan melihat harga per kilometer (***costPerKm***) yaitu Rp 1.000,00 per km dimulai dari kilometer ke 2,1 (2 km 100 meter), sedangkan, untuk dua kilometer pertama menggunakan harga Rp. 3.000,00. Pada tipe ini, diberlakukan tarif promo, yaitu, potongan harga 40% hanya untuk 6 km pertama saja. Sedangkan, km selanjutnya, diberlakukan harga normal.

Untuk kelas sporty, harga per kilometer yang diberlakukan ialah Rp. 3.000,00 per km dimulai pada km ke 5,1 (5 km 100 meter). Untuk kilometer ke 1 s.d. 5 diberlakukan harga flat sebesar Rp. 20.000,00. Pada tipe sporty juga terdapat promo yaitu potongan sebesar 60% untuk 8 km pertama. Sedangkan untuk kilometer selanjutnya diberlakukan harga normal. Pada tipe sporty terdapat protection cost, yaitu, sebesar 10% dari nilai total setelah dilakukan pemotongan harga promo.

Untuk tipe exclusive, harga per kilometer yang diberlakukan ialah sebesar Rp 5.000,00 per km dimulai dari kilometer ke 0,1 (100 meter). Pada tipe ini juga terdapat fixed cost yang harus dibayar, yaitu, sebesar Rp. 10.000,00. Terdapat juga potongan harga promo untuk, yaitu, sebesar 50% dari biaya total. Sama seperti tipe sporty, pada tipe exclusive juga terdapat protection cost yaitu sebesar 5% dari biaya total setelah dikurangi pemotongan harga promo.

Setelah mengetahui *requirement* yang diberikan, Anda pun mulai melakukan analisis. Sejenak Anda mengingat kembali materi-materi yang baru saja disampaikan di kelas, yaitu, tentang *inheritance* dan *polymorphism*. Anda mulai melihat kembali tabel diatas, dan menemukan bahwa ada beberapa persamaan dan ada beberapa perbedaan pada masing-masing kelas. Anda pun mulai memikirkan solusi dengan menggunakan konsep *inheritance*, yaitu, membuat sebuah *super class* dengan *field-field* berupa informasi umum yang terdapat pada semua tipe. Sedangkan, untuk *field-field* yang bersifat spesifik, Anda akan definisikan pada setiap subclass yang meng-*extends* super class tersebut. Selain itu, Anda juga mulai berpikir bagaimana untuk mengimplementasi method ***getCost(from, to)***, ***getPromo()***, dll yang berbeda-beda pada masing-masing tipe. Untuk kebutuhan ini, Anda akan menerapkan konsep *polymorphism* yaitu dengan membuat sebuah *interface* yang akan di-*implement* pada masing-masing *subclass*.

Setelah itu, Anda mulai memikirkan solusi untuk masalah *shortest path*, yaitu, menemukan jalan tercepat untuk titik pemberangkatan dan tujuan yang diberikan oleh user. Anda mulai mengingat kembali materi *recursif* yang telah diajarkan di kelas dan mencoba untuk menerapkan konsep tersebut untuk menemukan solusi *shortest path*. Anda pun mulai membuat *draft pseudocode* sederhana untuk problem tersebut, yaitu sebagai berikut:

```

if finish point:
    yes:
        if this path < older path:
            yes:
                store new path;
                stop;
            no:
                stop;
        no:
            if branch
            yes:
                if this path < older path:
                    yes:
                        store new path to this branch;
                        for each branch, recursively, call this function again.
                    no:
                        stop;
            no:
                next;

```

Tentu saja, *draft* tersebut masih hanya gambaran kasar dari algoritma *shortest path*. Kedepannya, Anda mungkin akan melakukan optimasi algoritma tersebut setelah mencari referensi algoritma *shortest path* di internet maupun sumber lainnya. Anda juga mungkin tidak akan menggunakan algoritma diatas dan menggantinya dengan algoritma lain yang Anda rasa lebih baik.

Dalam prototype ini, akan ada tiga perintah yang digunakan, yaitu:

1. show map

Perintah ini akan menampilkan map yang ada. Output dari perintah **show map** adalah sebagai berikut:

```

>> show map

      Q      R      S      T      U      V      W      X      Y      Z
0123456789012345678901234567890123456789012345678901234567890123456789
A0 #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 # #####
B0 # #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 #
9 # #####
C0 # #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 # #####
D0 #
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 #
E0 #####
1 #####

```

```

2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####

```

2. go from [START] to [FINISH] with OW-JEK [TYPE]

Perintah ini merupakan perintah utama dalam *prototype*. [START] merupakan lokasi titik pemberangkatan sedangkan [FINISH] merupakan lokasi titik tujuan. Lokasi [START] dan [FINISH] hanya boleh berada pada jalan (karakter spasi ' ') bukan gedung (karakter kres '#'), jika tidak maka input dianggap tidak valid. Perintah ini akan mencari jalan tercepat (*shortest path*) dari titik [START] ke titik [FINISH] dan menampilkan peta yang sudah disertai dengan *shortest path*. Titik [START] pada peta akan ditandai dengan karakter 'S' sedangkan titik [FINISH] pada peta direpresentasikan oleh karakter 'F'. Shortest path sendiri ditandai dengan kumpulan karakter titik '.' dari mulai titik [START] hingga titik [FINISH]. Selain menampilkan *shortest path*, *prototype* juga melakukan kalkulasi biaya jasa berdasarkan aturan yang telah disampaikan sebelumnya. Kalkulasi biaya disesuaikan dengan [TYPE] dari jasa OW-JEK yang digunakan. Seperti telah disebutkan sebelumnya, ada tiga jenis tipe jasa OW-JEK yaitu: Regular, Sporty, dan Exclusive. Contoh input dan output dari penggunaan fungsi ini adalah sebagai berikut:

```
>> go from D9Q2 to A1Y0 with OW-JEK Regular
```

```

      Q      R      S      T      U      V      W      X      Y      Z
0123456789012345678901234567890123456789012345678901234567890123456789
A0 #####
1 # .....F #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # .....
8 # #####
9 # .....
B0 # #####
1 # .....
2 # .....
3 # .....
4 # .....
5 # .....
6 # .....
7 # .....
8 # .....
9 # .....
C0 # .....
1 # .....
2 # .....
3 # .....
4 # .....
5 # .....
6 # .....
7 # .....
8 # .....
9 # .....
D0 # .....
1 # .....
2 # .....
3 # .....
4 # .....
5 # .....
6 # .....
7 # .....
8 # .....
9 #S .....
E0 #####
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####

```

Terimakasih telah melakukan perjalanan dengan OW-JEK.

```

[Jarak] 11,6 KM
[TipeO] Regular
[2KMPe] Rp 3.000,00 (+)
[KMSel] Rp 9.600,00 (+)
[Promo] Rp 2.400,00 (-)
[TOTAL] Rp 10.200,00

```

>> go from D9Q2 to A1Y0 with OW-JEK Exclusive

```

      Q      R      S      T      U      V      W      X      Y      Z
0123456789012345678901234567890123456789012345678901234567890123456789
A0 #####
1 # .....F #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 # #####
B0 #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # .....
9 # #####
C0 #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 # #####
D0 #####
1 # #####
2 # #####
3 # #####
4 # #####
5 # #####
6 # #####
7 # #####
8 # #####
9 #S #####
E0 #####
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
```

Terimakasih telah melakukan perjalanan dengan OW-JEK.
[Jarak] 11,6 KM
[TipeO] Exclusive
[Fixed] Rp 10.000,00 (+)
[KMSel] Rp 58.000,00 (+)
[Promo] Rp 34.000,00 (-)
[Prtns] Rp 1.700,00 (+)
[TOTAL] Rp 35.700,00

Adapun, contoh output dari input yang tidak valid adalah sebagai berikut:

>> go from D9Q2 to A4W1

input tidak valid: A4W1 bukan jalan

3. exit

Perintah ini digunakan jika user ingin keluar dari sistem.

***NOTE:** Untuk hal-hal lain yang tidak dijelaskan diatas, Anda bebas berasumsi dan berkreasi sebebas mungkin.

Poin Penilaian

1. Polymorphism dan Inheritance (30 %)
 - a. Polymorphism (15 %)
 - b. Inheritance (15 %)
2. Fungsi `go from [START] to [FINISH]` (70%)
 - a. Shortest path (40%)
 - b. Kalkulasi biaya jasa (30%)
2. Error Handling (+5%)
3. Dokumentasi (+5%)