

Spesifikasi Tugas Besar

IF2110 Algoritma dan Struktur Data

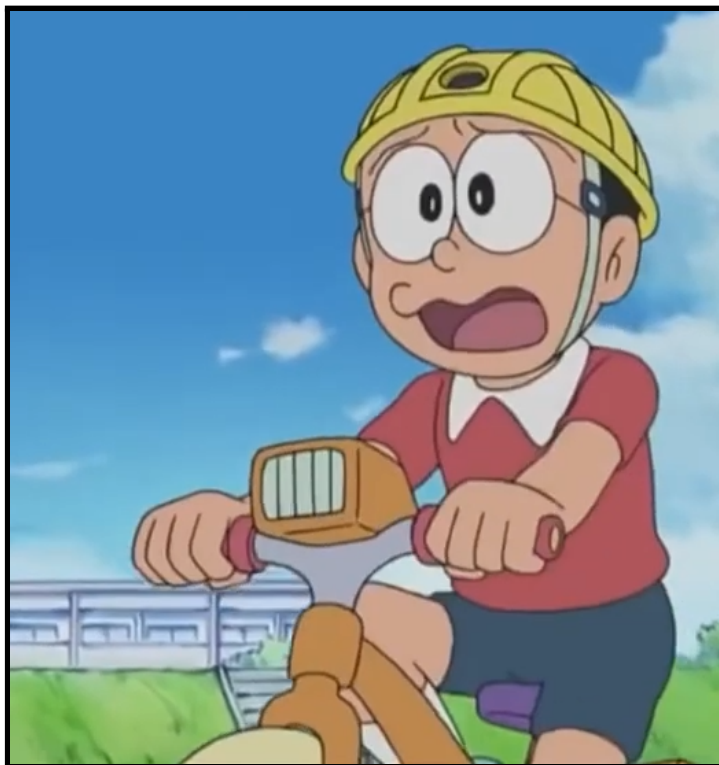
Mobilita

Revisi

~~ver. 21 Oktober 2021~~

ver. 27 Oktober 2021

Latar Belakang



“Mobita menjadi seorang kurir demi mencari nafkah untuk keluarganya”

Dengan adanya pandemi COVID-19, usaha orang tua Mobita mengalami penurunan pendapatan. Setelah diringkangkannya PPKM, Mobita ingin membantu orang tuanya mendapat penghasilan menjadi seorang kurir. Tugas kalian adalah membantu Mobita membuatkan aplikasi yang akan membantunya melacak pesanan, navigasi peta, mengambil dan menurunkan barang.

Spesifikasi Umum

Buatlah sebuah permainan berbasis **CLI** (command-line interface) tentang pengantaran barang. Permainan ini dibuat dalam **bahasa C** dengan menggunakan struktur data yang sudah kalian pelajari di mata kuliah ini. Kalian boleh menggunakan struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini.

Game Mechanics

1. Main Menu

- a. Pada awal permainan, pemain dapat memilih untuk bermain game atau keluar dari permainan.
- b. Command-command pada Main Menu:
 - i. **NEW GAME**
Meminta input nama file konfigurasi level, lalu memulai permainan. Jika nama file invalid, maka akan diminta ulang hingga benar.
 - ii. **EXIT**
Menutup program.
 - iii. **LOAD GAME (Bonus)**
Selain mengakses file konfigurasi, program mengakses file eksternal yang menyimpan state game sebelumnya. File konfigurasi dapat menyimpan state dari game yang telah disave. Setelah file dibaca, maka game akan dilanjutkan dari state tersebut.
- c. Setelah melakukan command **NEW GAME** (atau **LOAD GAME**), program akan melakukan load pada **folder file eksternal** untuk mendapatkan layout Map dan daftar urutan pesanan.

2. Mekanisme Waktu

Dalam permainan ini, waktu berjalan ketika Mobita berpindah dari satu tempat ke tempat lain. Pada dasarnya, setiap perpindahan menambah 1 unit waktu. Mekanisme waktu ini akan menentukan kapan pesanan masuk ke dalam To Do List Mobita dan kapan pesanan tertentu hangus. Waktu harus ditunjukkan di tampilan game setiap sebelum menerima command utama.

3. Daftar Pesanan

Mobita akan mendapatkan daftar pesanan yang harus dikerjakan. Setiap item dalam daftar pesanan akan berisi:

- a. Waktu pesanan masuk
- b. Lokasi tempat pick up item (pick up point)
- c. Lokasi tempat drop off item (drop off point)

- d. Jenis item
- e. Waktu pesanan hangus bila jenisnya Perishable item

Daftar pesanan akan di-input menggunakan file konfigurasi. Pesanan-pesanan akan masuk ke dalam To Do List Mobita sesuai dengan atribut Waktu Pesanan Masuk jika Waktu Pesanan Masuk pesanan lebih besar atau sama dengan waktu sekarang.

4. To Do List

Pesanan yang masuk dari daftar pesanan akan masuk ke dalam list ini. List pesanan dapat dikerjakan tanpa berurutan. List ini dapat ditampilkan pada In-Game Menu dengan atribut sebagai berikut:

- a. Lokasi tempat pick up item (Pick up point)
- b. Lokasi tempat drop off item (Drop off point)
- c. Jenis item
- d. Waktu pesanan hangus bila jenisnya Perishable item

5. In Progress List

Item yang diambil di pick up point akan masuk ke dalam list ini. Jika item memiliki efek, maka efek akan aktif. Jika item di-drop off pada lokasi yang sesuai, maka item dihapuskan dari list dan reward dari item tersebut akan langsung aktif.

6. Tas

Tas digunakan untuk menyimpan item yang diambil di lokasi. Awalnya, Mobita hanya dapat membawa **3 item** sekaligus. Aturan masuk item adalah **LIFO** (Last In First Out). Item yang dapat di-drop off adalah hanya item di tumpukan teratas tas. Jika pada saat mengambil item di lokasi terdapat beberapa pesanan, urutan item yang masuk akan sesuai dengan urutan pesanan masuk. Kapasitas tas maksimum setelah diperbesar oleh ability dan gadget adalah 100 item.

7. Jenis Item

Setiap jenis item memiliki karakteristik sebagai berikut:

a. Normal Item

- Item ini tidak memiliki efek samping ketika di-pick up atau reward ability tambahan ketika di-drop off.
- Item ini bernilai 200 Yen.

b. Heavy Item

- Saat mengantar item ini, Mobita bergerak lebih lambat.
- Efek: Setiap pindah ke 1 lokasi maka waktu akan bertambah 1 unit ($1+1 = 2$ unit). Efek ini dapat menumpuk.
- Reward: Mobita akan mendapatkan ability **Speed Boost** jika berhasil mengantarkan item ini

- Item ini bernilai 400 Yen.
- c. Perishable Item**
- Item ini harus diantarkan dengan cepat karena dapat hangus.
 - Efek: Item harus diantarkan ke lokasi drop off dalam waktu yang ditentukan. Jika waktu habis, maka item akan dikeluarkan secara otomatis dari tas, dan pesanan akan dihapus secara otomatis dari In Progress List.
 - Reward: Mobita akan mendapatkan ability **Increase Capacity** jika berhasil mengantarkan item ini.
 - Item ini bernilai 400 Yen.
- d. (Bonus) VIP Item**
- Jangan salah, Zhisuka juga menggunakan layanan Mobita! Karena Mobita adalah ~~simp~~ teman yang baik, pesanan yang Zhisuka buat dijadikan pesanan VIP yang harus dikerjakan lebih dulu.
 - Efek: Efek ini aktif saat pesanan masuk ke To Do List. Pesanan dengan item ini harus segera dilayani (pick up dan drop off), sehingga Mobita tidak bisa melakukan pick up ~~dan drop off~~ untuk item lain.
 - Reward: Mobita akan mendapatkan ability **Return To Sender** jika berhasil mengantarkan item ini.
 - Item ini bernilai 600 Yen.

8. Ability

Ability dibawah ini langsung teraktivasi kala pengantaran suatu item yang memberikan ability dari item tersebut.

- a. Speed Boost**
- Karena berhasil mengantarkan barang berat, maka Mobita mendapatkan semangat dan dapat bergerak lebih cepat.
 - Efek: Waktu akan bertambah 1 unit setiap Mobita berpindah 2 lokasi. Efek ini berlangsung hingga Mobita telah melewati 10 lokasi (berlaku selama 5 unit waktu).
 - Ability ini akan segera hilang bila Mobita melakukan pick up Heavy Item atau mendapatkan ability ketika sedang membawa Heavy Item lain.
- b. Increase capacity (permanent)**
- Karena bungkus Perishable Item dapat digunakan lagi dan diberikan kepada Mobita, maka bungkus ini digantung di luar tas dan dapat digunakan untuk membawa item lebih.
 - Kapasitas tas bertambah 1.
- c. (Bonus) Return to sender**
- Jika ability ini digunakan dengan command RETURN, maka item di atas tas dikembalikan ke lokasi pick up
 - Efek: Pesanan akan dikembalikan ke To Do List di urutan terakhir.

- Efek: Time limit yang terdapat pada pesanan dengan Perishable Item akan di-reset sebelum dikembalikan.
- Ability ini tidak dapat diaktifkan untuk pesanan VIP Item.

9. Gadget

Dengan uang yang didapat Mobita, maka Mobita dapat membeli beberapa gadget dari doraemonangis. Setiap gadget yang terbeli hanya dapat digunakan **tepat sekali** dan kemudian akan hangus. Gadget yang tersedia adalah sebagai berikut:

a. Kain Pembungkus Waktu

Harga: 800 Yen

Setiap kain pembungkus waktu dapat dipakai sekali untuk perishable item teratas pada tas agar kembali ke durasi semula.

b. Senter Pembesar

Harga: 1200 Yen

Senter pembesar dapat digunakan untuk meningkatkan kapasitas tas sebesar dua kali lipat, namun tidak melebihi batas maksimum kapasitas tas.

c. Pintu Kemana Saja

Harga: 1500 Yen

Pintu Kemana Saja dapat digunakan sekali untuk berpindah ke lokasi yang diinginkan tanpa menambahkan unit waktu.

d. Mesin Waktu

Harga: 3000 Yen

Mesin waktu dapat digunakan untuk mengurangi waktu sebanyak 50 unit. (jika waktu kurang dari 50 unit, maka waktu menjadi 0 unit). **Pesanan yang masuk dalam jangka waktu 50 unit tersebut tidak akan hilang/kembali ke To Do List, dan tidak akan muncul kembali ke To Do List ketika waktu masuknya dilampaui lagi.**

e. (Bonus) Senter Pengecil

Harga: 800 Yen

Senter pengecil dapat digunakan untuk menghilangkan efek dari satu heavy item jika terdapat pada tumpukan teratas tas. Efek dari senter pengecil ini akan berlangsung sampai melakukan **drop off / return** pertama kali setelah penggunaan gadget ini.

10. Inventory Gadget

Setiap gadget yang terbeli dapat dimasukkan ke dalam inventory dan setiap gadget yang hangus otomatis terbuang dari inventory. Inventory memiliki kapasitas sebesar 5. Jika inventory penuh, maka Mobita tidak dapat membeli gadget.

11. Peta

Untuk membantu Mobita, terdapat peta dengan ukuran **N x M**. Ukuran peta minimal 10 x 10 dan maksimal 20 x 30 (ditentukan oleh konfigurasi permainan). Titik (i, j) merupakan sebuah petak pada baris ke-i dan kolom ke-j.

Berikut adalah ilustrasi peta 10 x 15 yang ditampilkan kepada pemain

```
*****
*8      C  D B*
*  E          *
*F      G    H*
*  I          *
*          J    *
*  K          *
*          L    *
*  M  N          O*
*          P    *
*A Q          *
*****
```

Petak diberi boundary karakter *. Karakter pada petak melambangkan lokasi yang ada di petak tersebut. Berikut adalah legenda peta:

- Karakter '8' menunjukkan Headquarters tempat Mobita memulai / mengakhiri tugasnya, dan membeli gadget.
- Karakter berwarna kuning menunjukkan posisi Mobita.
- Karakter berwarna merah menunjukkan lokasi pick up pesanan.
- Karakter berwarna biru menunjukkan lokasi drop off item teratas pada tas.
- Karakter berwarna hijau menunjukkan lokasi yang dapat dicapai (destinasi).
- Karakter berwarna hitam adalah lokasi tanpa pick up / drop off / mobita.

Prioritas pewarnaan karakter adalah sebagai berikut:

- Mobita > Drop off > Pick up > Destination > Neutral

12. Lokasi dan Command

Di setiap lokasi, Mobita dapat menginput command-command berikut (setiap contoh command ini dapat diubah sesuai kebutuhan, tidak perlu sama dengan contoh):

a. MOVE

Command ini digunakan untuk menampilkan pilihan lokasi yang dapat disinggahi Mobita. Lokasi yang dapat disinggahi akan berdasarkan adjacency matrix pada file konfigurasi. Setelah dipilih, Mobita akan berpindah ke lokasi tersebut dan waktu akan bertambah 1 unit.

```
Waktu: 10
ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. D (3,1)
2. C (3,5)
3. F (3,15)
4. I (5,12)
5. K (6,3)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND: 5

Mobita sekarang berada di titik K (6,3)!
Waktu: 11
ENTER COMMAND:
```

b. PICK_UP

Command ini digunakan untuk mengambil item jika ada pesanan yang harus diambil pada lokasi. Jika tidak ada pesanan pada lokasi, maka tampilkan pesan bahwa tidak ada pesanan (pesan dibebaskan). Jika ada beberapa pesanan di lokasi pick up, maka item yang diambil adalah pesanan yang masuk duluan.

Contoh skenario 1:

```
ENTER COMMAND: PICK_UP
Pesanan berupa Normal Item berhasil diambil!
Tujuan Pesanan: H
```

Contoh skenario 2:

```
ENTER COMMAND: PICK_UP
Pesanan tidak ditemukan!
```

c. DROP_OFF

Command ini digunakan untuk mengantarkan item ke lokasi jika item di tumpukan teratas tas sesuai dengan pesanan. Jika tidak ada pesanan yang perlu diantarkan ke lokasi, maka tampilkan pesan bahwa tidak ada pesanan (pesan dibebaskan).

Contoh skenario 1:

```
ENTER COMMAND: DROP_OFF
Pesanan Normal Item berhasil diantarkan
Uang yang didapatkan: 200 Yen
```

Contoh skenario 2:

```
ENTER COMMAND: DROP_OFF
Tidak dapat pesanan yang dapat diantarkan!
```

d. MAP

Command ini digunakan untuk memunculkan peta. Untuk keterangan lebih lanjut, lihat bagian **Peta**.

```
ENTER COMMAND: MAP
*****
*8      C  D B*
*  E          *
*F      G    H *
*  I          *
*          J    *
*  K          *
*          L    *
*  M  N          O*
*          P    *
*A Q          *
*****
```

e. TO_DO

Command ini digunakan untuk menampilkan pesanan yang masuk ke To Do List. Pesanan ditampilkan secara berurutan sesuai dengan pesanan yang masuk duluan.

```
ENTER COMMAND: TO_DO
Pesanan pada To Do List:
1. G -> F (Normal Item)
2. D -> H (Normal Item)
```

f. IN_PROGRESS

Command ini digunakan untuk menampilkan pesanan yang sedang dikerjakan. Pesanan ditampilkan secara berurutan sesuai dengan pesanan yang terakhir masuk.

```
ENTER COMMAND: IN_PROGRESS
Pesanan yang sedang diantarkan:
1. Normal Item (Tujuan: H)
```

g. BUY

Command ini hanya dapat dipanggil di Headquarters. Command ini digunakan untuk menampilkan gadget yang dapat dibeli lalu membelinya. Setelah gadget ditampilkan maka pemain dapat melakukan input untuk membeli gadget yang diinginkan.

Contoh skenario 1:

```
ENTER COMMAND: BUY
```



```
Uang Anda sekarang: 1000 Yen
Gadget yang tersedia:
1. Kain Pembungkus Waktu (800 Yen)
2. Senter Pembesar (1200 Yen)
3. Pintu Kemana Saja (1500 Yen)
4. Mesin Waktu (3000 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)

ENTER COMMAND: 1
Kain Pembungkus Waktu berhasil dibeli!
Uang Anda sekarang: 200 Yen
```

Contoh skenario 2:

```
ENTER COMMAND: BUY
Uang Anda sekarang: 1000 Yen
Gadget yang tersedia:
1. Kain Pembungkus Waktu (800 Yen)
2. Senter Pembesar (1200 Yen)
3. Pintu Kemana Saja (1500 Yen)
4. Mesin Waktu (3000 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)

ENTER COMMAND: 4
Uang tidak cukup untuk membeli gadget!
```

h. INVENTORY

Command ini digunakan untuk menampilkan list isi inventory. Setelah gadget ditampilkan, pemain dapat melakukan input untuk menggunakan gadget yang diinginkan. Apabila tidak ada gadget yang dapat digunakan, tampilkan pesan bahwa tidak ada gadget yang dapat digunakan (pesan dibebaskan).

Contoh skenario 1:

```
ENTER COMMAND: INVENTORY
1. Kain Pembungkus Waktu
2. -
3. -
4. Senter Pembesar
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 1
Kain Pembungkus Waktu berhasil digunakan!
```

Contoh skenario 2:

```
ENTER COMMAND: INVENTORY
1. -
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)
```

```
ENTER COMMAND: 1
Tidak ada Gadget yang dapat digunakan!
```

i. **HELP**

Command ini digunakan untuk menampilkan seluruh command yang dapat dimasukkan beserta deskripsinya. Penjelasan deskripsi dibebaskan dengan syarat deskripsi masih mendeskripsikan command sesuai spek.

```
ENTER COMMAND: HELP
1. MOVE -> Untuk berpindah ke lokasi selanjutnya
2. PICK_UP -> Untuk mengambil item dilokasi sekarang
...
8. INVENTORY -> Untuk melihat gadget yang dimiliki dan menggunakannya
9. HELP -> Untuk mengeluarkan list command dan kegunaannya
```

j. **(Bonus) SAVE_GAME**

Command ini digunakan untuk melakukan save state dari permainan yang sedang dijalankan. Hasil dari save ini disimpan pada suatu file yang kemudian dapat dilakukan load game di awal permainan dimulai.

```
ENTER COMMAND: SAVE_GAME
Masukkan nama save file permainan: 21-10-2021
Save file berhasil terbuat!
```

k. **(Bonus) RETURN**

Command ini digunakan untuk mengembalikan item di tumpukan teratas pada tas kembali ke lokasi pick up jika Mobita memiliki ability Return To Sender. Untuk keterangan lebih lanjut, lihat bagian Ability > Return To Sender.

Contoh skenario 1:

```
ENTER COMMAND: RETURN
Normal Item berhasil dikembalikan ke Pick Up Point D
```

Contoh skenario 2:

```
ENTER COMMAND: RETURN
Mobita tidak memiliki Ability Return To Sender!
```

13. Game Flow

a. **Start Game**

Mobita akan memulai harinya di lokasi Headquarters, mendapat jumlah pesanan yang harus diselesaikan, serta mulai mendapat pesanan yang masuk ke To Do List.

Tujuan dari game ini adalah menyelesaikan semua pesanan dengan waktu secepat mungkin

b. End Game

Jika semua pesanan selesai diantar (dan semua pesanan dengan Perishable Item sudah expire / diantar) Mobita harus kembali ke Headquarters. Kemudian akan ada pesan yang menampilkan game sudah selesai, berapa jumlah item yang berhasil diantar, dan berapa lama waktu yang dilampaui.

14. (Bonus) Save/Load Game

Dalam melakukan *save game*, perlu disimpan *state-state* yang dibutuhkan ke dalam suatu file, apa yang dituliskan dibebaskan, namun tentunya harus dapat dibaca saat *load game*. Saat melakukan *load game*, permainan kembali seperti keadaan *state* yang telah disimpan *file* yang di-load tersebut. File yang disimpan berupa file konfigurasi awal yang di-append dengan state-state yang disimpan.

Contoh Tampilan Game

Berikut adalah contoh state dari permainan yang akan digunakan untuk beberapa command yang penting. Misalkan Mobita berada di titik I, Pick Up Point di titik M, dan Drop Point di titik H. Perhatikan bahwa Anda tidak perlu mengimplementasikan output sama persis dengan contoh.

Mobita berada di posisi **I (4,5)**

Waktu: 3

Uang yang dimiliki: 0 Yen

ENTER COMMAND: MAP

8 C D B

* E *

*F G H *

* I *

* J *

* K *

* L *

* M N O*

* P *

*A Q *

Mobita berada di posisi **I (4,5)**

Waktu: 3

Uang yang dimiliki: 0 Yen

Jumlah pesanan yang harus dikerjakan: 12

ENTER COMMAND:

Contoh Konfigurasi Permainan

Konfigurasi file yang akan diinput pada awal game akan berisi sebagai berikut:

- Ukuran peta ($10 \leq N \leq 20$; $10 \leq M \leq 30$)
- Koordinat Headquarters
- Jumlah lokasi ($4 \leq L \leq 26$)
- Karakter lokasi dan koordinat lokasi
- Matriks hubungan lokasi (*adjacency matrix*), di mana elemen pertama menunjukkan adjacency Headquarters dengan lokasi lain
- Jumlah pesanan ($5 \leq P \leq 30$)
- Waktu pesanan masuk, bangunan lokasi pick up, bangunan lokasi drop off, dan jenis item sebagai berikut:
 - $N \rightarrow$ Normal Item

- H → Heavy Item
- P → Perishable Item
 - Untuk setiap *perishable item* akan ada tambahan input waktu item, menyatakan kapan *perishable item* tersebut akan hangus
- V → VIP Item

Berikut adalah contoh konfigurasi file:

```

10 15
1 1
17
A 10 1
B 1 15
C 1 9
D 1 13
E 2 3
F 3 1
G 3 8
H 3 14
I 4 5
J 5 12
K 6 3
L 7 10
M 8 2
N 8 6
O 8 15
P 9 13
Q 10 3
0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0
0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0
0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1
0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
10
1 G N N
3 G N H
2 B M N

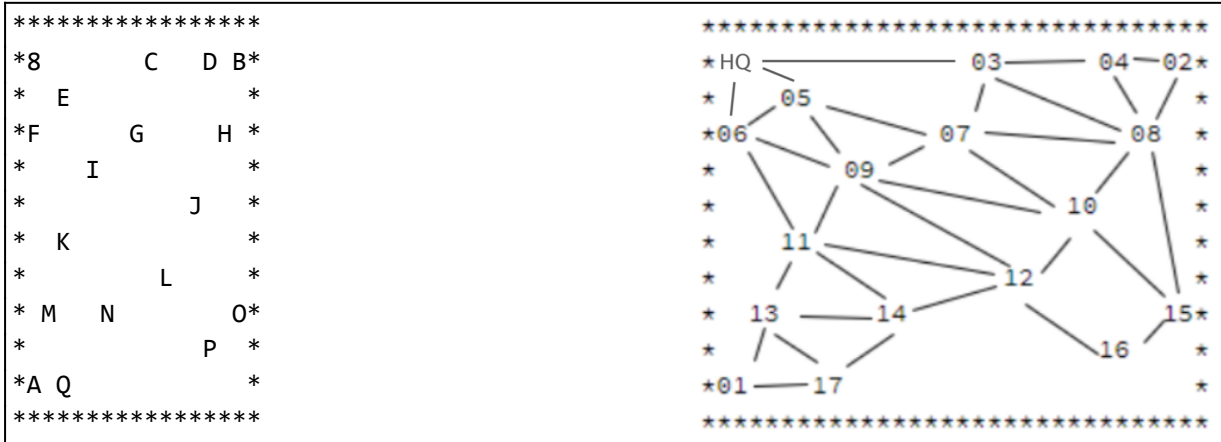
```

```

1 M B H
5 C D N
3 F E N
10 P Q P 20
5 G N P 10
8 O P V
8 Q L V

```

File konfigurasi di atas bersesuaian dengan tampilan peta berikut (gambar jalur hanya ilustrasi hubungan):



Daftar pesanan yang akan dihasilkan dari file konfigurasi di atas dan dibaca dari To Do List pada waktu 10 adalah sebagai berikut:

1. G -> N (Normal Item)
2. M -> B (Heavy Item)
3. B -> M (Normal Item)
4. G -> N (Heavy Item)
5. F -> E (Normal Item)
6. C -> D (Normal Item)
7. G -> N (Perishable Item, sisa waktu 10)
8. O -> P (VIP Item)
9. Q -> L (VIP Item)
10. P -> Q (Perishable Item, sisa waktu 20)

Penggunaan Modul Print Berwarna

Karena tampilan pada peta diharuskan menggunakan warna berbeda, asisten akan menyediakan sebuah modul untuk melakukan print berwarna pada terminal. Modul ini dapat digunakan pada Windows Terminal, Terminal VSCode, Windows Powershell dan terminal UNIX. Silahkan unduh **pcolor.c** dan **pcolor.h** pada Olympia, serta lihatlah contoh pada pcolor.h untuk penggunaannya. Anda tentu dapat memodifikasi modul tersebut jika diperlukan.

Daftar ADT yang Digunakan

Anda diwajibkan menggunakan ADT di bawah ini. Selain itu, Anda dapat pula menggunakan ADT lain, namun cantumkan analisis alasan kenapa menggunakan ADT tersebut pada laporan.

1. ADT Point

ADT ini digunakan untuk menunjukkan posisi bangunan.

2. ADT List Statis

ADT ini digunakan untuk menyimpan gadget yang sudah dibeli.

3. ADT List Dinamis

ADT ini digunakan untuk menyimpan daftar bangunan yang tersedia pada peta.

4. ADT Matriks

ADT ini digunakan sebagai representasi tampilan peta dan untuk merepresentasikan hubungan jalan antar bangunan (menggunakan adjacency matriks).

5. ADT Mesin Karakter dan Mesin Kata

ADT ini digunakan untuk:

- a. Membaca informasi konfigurasi dari file eksternal
- b. Membaca command dari interaksi user terhadap program, baik di dalam game atau di menu start, dan
- c. (Bonus) Membaca dan menuliskan *save file* ke suatu file eksternal

6. ADT Queue

ADT ini digunakan untuk urutan pesanan yang akan masuk.

7. ADT Stack

ADT ini digunakan sebagai tas/ransel untuk menyimpan item yang di pick up.

8. ADT Linked List

ADT ini digunakan untuk menyimpan daftar pesanan yang harus dikerjakan dan sedang dikerjakan.

9. ADT Lain

ADT ini dibuat dan digunakan untuk abstraksi beberapa hal lain. Mahasiswa dipersilakan untuk mendefinisikan sendiri.

Catatan Tambahan

1. Sebagai saran, manfaatkan Makefile untuk mempermudah proses kompilasi dan penjalanan program. Bila sulit dalam menggunakan Makefile, bisa diakali dengan menggunakan shell script/batch file.
2. Usahakan buat struktur program yang serapi mungkin. Jangan buat semuanya pada file yang sama.
3. Manfaatkan ADT yang sudah kalian buat dalam praktikum semaksimal mungkin.
4. Perhatikan bahwa nilai untuk bonus akan lebih kecil dibandingkan dengan fitur utama. Silakan prioritaskan fitur-fitur yang lebih penting terlebih dahulu.