

# **LAPORAN TUGAS BESAR**

## **IF2110/Algoritma dan Struktur Data**

**Mobita: Program *life simulation game* kurir berbahasa C  
berbasis *command-line interface***

Dipersiapkan oleh:

Kelompok 5

13520116 – Mahesa Lizardy

13520138 – Gerald Abraham Sianturi


13520142 – Rania Dwi Fadhilah

13520155 – Jundan Haris

13520167 – Aldwin Hardi Swastia

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|   |   |                        |   |                         |
|---|---|------------------------|---|-------------------------|
|  | <b>Sekolah Teknik<br/>Elektro dan<br/>Informatika ITB</b> | <b>Nomor Dokumen</b>   |   | <b>Halaman</b>          |
|   |   | <i>IF2110-TB-05-03</i> |   | <i>34</i>               |
|   |   | <i>Revisi</i>          | - | <i>18 November 2021</i> |

# Daftar Isi

|   |    |
|---|----|
| 1 Ringkasan.....                              | 3  |
| 2 Penjelasan Tambahan Spesifikasi Tugas ..... | 3  |
| 2.1 Daftar Pesanan.....                       | 3  |
| 3 Struktur Data (ADT).....                    | 3  |
| 3.1 ADT Point .....                           | 3  |
| 3.2 ADT List Statis.....                      | 3  |
| 3.3 ADT List Dinamis.....                     | 4  |
| 3.4 ADT Matriks .....                         | 4  |
| 3.5 ADT Mesin Karakter dan Mesin Kata .....   | 4  |
| 3.6 ADT Queue .....                           | 5  |
| 3.7 ADT Stack (Modified to Tas) .....         | 5  |
| 3.8 ADT Linked List.....                      | 5  |
| 3.9 ADT Player .....                          | 5  |
| 4 Program Utama .....                         | 6  |
| 5 Data Test .....                             | 6  |
| 5.1 Move.....                                 | 6  |
| 5.2 Pick Up.....                              | 7  |
| 5.3 DROP_OFF .....                            | 8  |
| 5.4 MAP .....                                 | 9  |
| 5.5 TO_DO.....                                | 10 |
| 5.6 IN_PROGRESS .....                         | 10 |
| 5.7 BUY .....                                 | 10 |
| 5.8 INVENTORY.....                            | 11 |
| 5.9 HELP.....                                 | 11 |
| 5.10 EXIT .....                               | 11 |
| 6 Test Script.....                            | 11 |
| 7 Pembagian Kerja dalam Kelompok .....        | 13 |
| 8 Lampiran .....                              | 14 |
| 8.1 Deskripsi Tugas Besar 2.....              | 14 |
| 8.2 Notula Rapat.....                         | 30 |
| 8.3 Notula Asistensi .....                    | 31 |
| 8.4 Log Activity Anggota Kelompok.....        | 34 |

# 1 Ringkasan

Pada era digital, industri *game* turut berkembang. Salah satu jenis permainan yang mengalami perkembangan pesat adalah *Life Simulation Game*. Oleh karena itu, sebagai usaha untuk beradaptasi dengan perkembangan tersebut, serta memenuhi beban tugas besar mata kuliah Algoritma dan Struktur Data, didesainlah suatu program permainan sederhana menggunakan bahasa C dengan jenis tersebut dan bertemakan simulasi kurir pengantar *item*. Persoalan yang dihadapi adalah bagaimana mendesain program permainan tersebut dengan fitur utama, yakni melacak pesanan, navigasi peta, serta mengambil dan menurunkan barang menggunakan ADT tertentu secara tepat guna. Pada laporan ini, kami membahas ADT yang digunakan dalam program beserta detailnya, serta menampilkan hasil *running* program. Dengan demikian, dengan digunakannya ADT yang ada, fungsi-fungsi yang ada di program bisa lebih efisien penggunaannya karena dapat diakses secara global.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Daftar Pesanan

Daftar pesanan merupakan ADT “CollOfItems” yang berisi *item-item* yang ada yang telah difilter sehingga hanya *item-item* yang sudah memenuhi kondisi waktu tertentu, yakni waktu pesanan masuknya lebih kecil atau sama dengan waktu yang sedang berjalan, belum pernah di-*pick up* atau di-*drop off*.

## 3 Struktur Data (ADT)

Untuk mempermudah menyelesaikan permasalahan kompleks pada tugas ini, kami membentuk beberapa ADT, baik yang telah dikerjakan melalui praktikum (murni dari praktikum atau dilakukan modifikasi), maupun ADT lain yang baru dibentuk. Berikut beberapa ADT yang digunakan pada program beserta detailnya:

### 3.1 ADT Point

ADT Point digunakan untuk menyimpan posisi bangunan atau menunjukkan dimana suatu bangunan itu berada. ADT Point memiliki tiga atribut, *C* bertipe *char* untuk menyimpan nama bangunan, *X* dan *Y* bertipe *int* untuk menyimpan koordinat dari bangunan tersebut. Dibuat juga makro untuk mengakses ketiga atribut dari ADT Point tersebut. Jika *P* adalah Point, maka Absis(*P*) mengakses *X* dari *P*, Ordinat(*P*) mengakses *Y* dari *P*, dan LOCC(*P*) mengakses *C* dari *P*. ADT Point tepat untuk merepresentasikan posisi bangunan karena kedua hal tersebut memiliki koordinat *X* dan *Y*. Selain itu *C* bisa digunakan untuk menyimpan nama bangunan. Misalkan Headquarter yang direpresentasikan dengan *char* ‘8’ memiliki koordinat (1,1) maka implementasi pada ADT Point-nya adalah Point dengan *C* = ‘8’, *X* = 0, dan *Y* = 0. ADT Point ini disimpan dan diimplementasikan pada “point.h” dan “point.c”.

### 3.2 ADT List Statis

ADT list statis memiliki atribut berupa ‘ElType *contents*[*CAPACITY*]’ sebagai memori tempat penyimpanan elemen. ElType sendiri merupakan tipe elemen list yang berupa *integer*. ‘*CAPACITY*’ merupakan kapasitas penyimpanan dari list ini. Persoalan yang diselesaikan oleh

ADT ini adalah tempat penyimpanan gadget. Gadget terdefinisi menjadi empat ID, dengan ID pertama berupa kain pembungkus waktu, ID kedua untuk senter pembesar, ID ketiga untuk pintu kemana saja, dan ID keempat untuk mesin waktu. Dengan menggunakan list statis, ID dapat disimpan dalam bentuk list. Misalkan seorang pemain memiliki gadget dengan ID 1 sebanyak dua buah, maka list gadget yang dia miliki akan berisi [1,1]. Kapasitas dari list statis ini sendiri disesuaikan dengan kapasitas penyimpanan maksimum gadget, yaitu lima item. ADT ini digunakan pada file bernama “gadget.h” dan “gadget.c”.

### 3.3 ADT List Dinamis

ADT List Dinamis digunakan untuk menyimpan semua bangunan yang ada pada map file konfigurasi. ADT List Dinamis ini memiliki tiga atribut. Atribut yang pertama adalah buffer yang bertipe ElTypeLD, dimana ElTypeLD merupakan sebuah ADT Point. Atribut yang kedua dan yang ketiga bertipe integer, yaitu nEff dan capacity. nEff digunakan untuk menyimpan banyaknya elemen efektif dari List, sedangkan capacity digunakan untuk menyimpan ukuran elemen. ADT List Dinamis ini cocok untuk menyimpan bangunan karena bentuknya yang seperti array satu dimensi. Bangunan yang disimpan berbentuk ADT Point yang menyimpan nama bangunan serta koordinatnya. Selain itu, seperti namanya, List Dinamis bersifat dinamis. Misalkan player saat ini berada di bangunan ‘A’, maka semua bangunan yang bisa dicapai dari ‘A’ dapat disimpan dalam ADT List Dinamis. Ketika player sedang di bangunan ‘B’, tentu bangunan yang bisa dicapai dari ‘B’ berbeda dari saat dia di bangunan ‘A’. karena jumlahnya berbeda juga, sehingga ADT List Dinamis ini cocok untuk menyimpan daftar bangunan. ADT List Dinamis disimpan dan diimplementasikan pada file bernama “listdin.h” dan “listdin.c”.

### 3.4 ADT Matriks

ADT Matriks digunakan untuk menyimpan matriks ketetanggan. Secara umum, ADT Matriks ini adalah array dua dimensi. ADT Matriks memiliki tiga atribut, atribut pertama yaitu ‘contents[row][col]’ yang bertipe ElType. ElType sendiri adalah sebuah integer. Dua atribut lainnya yaitu rowEff serta colEff yang menunjukkan ukuran matriks tersebut. Untuk menyimpan sebuah matriks (matriks adjacency), tentu saja ADT Matriks adalah ADT yang paling tepat. Kita juga membuat modifikasi ADT Matriks dengan mengganti jenis ElType dari integer ke Point. Modifikasi ini dinamakan MatrixMap. MatrixMap digunakan untuk merepresentasikan bangunan ke map. Misalkan Map berukuran 10x15, maka pada koordinat-koordinat tertentu akan diisi dengan Point bangunan. Sisanya akan diisi dengan Point ‘0’. ADT Matriks dan ADT MatrixMap ini disimpan dan diimplementasikan pada “matrix.h”, “matrix.c”, “map.h”, serta “map.c”.

### 3.5 ADT Mesin Karakter dan Mesin Kata

ADT ini digunakan untuk membaca input user maupun file. ADT Mesin Kata dengan menggunakan fungsi-fungsi pada Mesin Karakter akan membaca seluruh input selain BLANK dan NEWLINE. Dalam membaca input user, dibuat sebuah variabel global yaitu tape yang isinya adalah stdin. Kemudian dalam membaca inputan user selanjutnya digunakan fungsi advWord. Dalam membaca input dari file digunakan fungsi bawaan fopen yang dijadikan value dari variabel global filetape. Dalam membaca satu demi satu input file, digunakan fungsi advWordFile.

### 3.6 ADT Queue

ADT ini dibuat sebagai antrian masuknya pesanan pada To-Do-List pemain. Queue ini terurut berdasarkan waktu masuknya pesanan.

### 3.7 ADT Stack (Modified to Tas)

ADT ini dibuat sebagai mekanisme tas yang digunakan untuk menyimpan item yang diambil di suatu lokasi tertentu. Awalnya, kapasitas awal tas tersebut adalah 3 dan dapat berkembang hingga kapasitas maksimal 100. Karena menggunakan ADT Stack maka aturan masuk item adalah LIFO (*Last In First Out*). Item yang dapat di-drop off hanya *item* yang berada pada tumpukan teratas tas. Jika pada saat mengambil *item* di lokasi terdapat beberapa pesanan, urutan *item* yang masuk akan sesuai dengan urutan pesanan masuk. ADT Stack pada tas berisi:

- “item”: yang elemennya berisi item yang merupakan elemen stack itu sendiri,
- “idxTop”: yang merupakan index paling atas pada stack,
- “heavyItem”: yang merupakan jumlah heavy item yang berada di stack,
- “currCapacity”: menyimpan kapasitas stack sekarang,
- “maxCapacity”: menyimpan jumlah maksimal item yang bisa dimuat tas.

ADT tersebut digunakan di beberapa file seperti drop\_off, pick\_off, to\_do\_list dan ability

### 3.8 ADT Linked List

ADT ini dibuat sebagai mekanisme menampilkan daftar pesanan yang harus dikerjakan (*to do list*) dan yang sedang dikerjakan (*in progress*). Untuk mekanisme pertama, yaitu *to do list*, akan ditampilkan kepada *user* beberapa informasi, di antaranya lokasi tempat *pick up item*, lokasi *drop off item*, jenis *item*, dan jika jenis item adalah Perishable Item, akan ditampilkan pula waktu pesanan hangus. Alasan penggunaan ADT ini pada mekanisme *to do list* adalah karena mengingat proses penampilan *to do list* mempertimbangkan waktu pesanan masuk, dimana waktu pesanan masuk terkecil akan menjadi *to do list* teratas, sehingga perlu dilakukan *sorting* dan *insert* (algoritma *sorting* tepat untuk kasus ini. Selanjutnya, untuk *in progress list*, hanya diperlukan prosedur untuk menampilkan isi tas dari *stack*. Karena tidak perlu dilakukan *sorting* (karena mekanisme tas sudah didesain dengan *stack*), digunakan *stack linked list* untuk menampilkan *in progress list*.

### 3.9 ADT Player

ADT *player* adalah struktur data yang dibuat untuk menyimpan data-data dari *player*. Terdapat beberapa atribut dari struktur data ini, antara lain :

- Listpos gadget : menyimpan gadget yang dimiliki oleh *player*. Bentuk penyimpanannya adalah dalam bentuk ID dari gadget, sebagai contoh : [1,1,3,4].
- int money: menyimpan uang dari *player*.
- int time: menyimpan waktu dari *player*.
- Point location: menyimpan posisi *player*.

ADT ini dibuat untuk memudahkan dan memusatkan data dari *player*.

## 4 Program Utama

Program utama dimulai dengan pilihan *Play game* dan *Exit*. Apabila memilih *Play game*, permainan akan dimulai (Mobita). *User* akan bermain untuk menyelesaikan *pick-up* pesanan dan *drop-off* hingga seluruh pesanan telah dikirimkan atau pemain memutuskan untuk keluar dari permainan. Apabila pemain memilih *Exit*, program akan berhenti.

## 5 Data Test

### 5.1 Move

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

ENTER COMMAND : MOVE
Posisi Mobita saat ini : 8 (1,1)
Posisi yang dapat dicapai :
1. C (1,9)
2. E (2,3)
3. F (3,1)
Posisi yang ingin dituju? (0 jika ingin membatalkan)
ENTER COMMAND FOR MOVE : 3
Posisi Mobita saat ini : F (3,1)
Waktu saat ini: 1
ENTER COMMAND : MAP
* * * * *
* 8      C      D  B *
*   E                *
* F          G      H *
*   I                    *
*   K                      *
*   L                        *
* M      N                O *
*   P                      *
* A  Q                    *
* * * * *
```

Player berpindah ke lokasi lain di map sesuai dengan *adjacency*. Saat player berpindah, waktu akan bertambah sebanyak 1 satuan. Apabila player memiliki ability speedboost waktu

baru akan bertambah setelah bergerak sebanyak 2 kali. Sedangkan saat player membawa satu heavy item, waktu akan bertambah sebanyak 2 dan akan bertambah 1 satuan setiap heavy item lain yang diambil.

## 5.2 *Pick Up*

```
Posisi Mobita saat ini : E (2,3)
Waktu saat ini: 1
2. M -> B (Heavy Item)
3. B -> M (Normal Item)

ENTER COMMAND : PICK_UP
Pesanan berupa Normal Item berhasil diambil!

ENTER COMMAND : TO_DO
1. M -> B (Heavy Item)
2. B -> M (Normal Item)
```

Melakukan *pick-up* terhadap pesanan yang lokasi tempat pesanannya sesuai yang ada di *to-do-list* pada elemen pertama. kemudian meletakkannya di dalam tas jika masih terdapat kapasitas penyimpanan pada tas dengan elemen yang terakhir masuk akan berada di elemen teratas(top) pada tas. Jika tas penuh maka akan ditampilkan pesan yang menyatakan bahwa tas sudah penuh dan tidak dapat memasukan item kedalam tas

### 5.3 DROP\_OFF

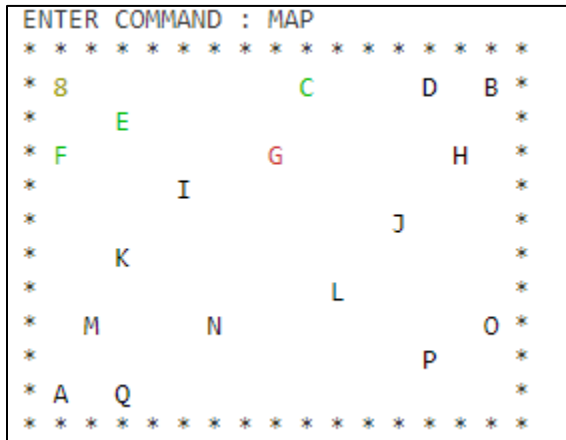
```
ENTER COMMAND : MAP
* * * * *
* 8      C      D  B *
*   E          *
* F      G      H *
*   I          *
*           J      *
*   K          *
*           L      *
* M      N      O *
*           P      *
* A  Q          *
* * * * *

ENTER COMMAND : MOVE
Posisi Mobita saat ini : I (4,5)
Posisi yang dapat dicapai :
1. E (2,3)
2. F (3,1)
3. G (3,8)
4. J (5,12)
5. K (6,3)
6. L (7,10)
Posisi yang ingin dituju? (0 jika ingin membatalkan)
ENTER COMMAND FOR MOVE : 5
Posisi Mobita saat ini : K (6,3)
Waktu saat ini: 4
ENTER COMMAND : MOVE
Posisi Mobita saat ini : K (6,3)
Posisi yang dapat dicapai :
1. F (3,1)
2. I (4,5)
3. L (7,10)
4. M (8,2)
5. N (8,6)
Posisi yang ingin dituju? (0 jika ingin membatalkan)
ENTER COMMAND FOR MOVE : 5
Posisi Mobita saat ini : N (8,6)
Waktu saat ini: 5
ENTER COMMAND : DROP_OFF
Pesanan Normal Item berhasil diantarkan
Uang yang didapatkan: 200
```

Menurunkan pesanan telah di-*pick-up* sebelumnya. Saat berhasil menurunkan pesanan maka player akan diberikan reward berupa uang dan ability tertentu tergantung pada jenis item yang di drop off. Jika tempat drop off pesanan tidak sesuai dengan posisi player dan tidak ada item pada tas (tas kosong) maka akan ditampilkan pesan kesalahan.



## 5.4 MAP



Menampilkan MAP permainan. Map atau peta 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 oranye 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

Data test yang digunakan adalah file konfigurasi yang sama dengan file konfigurasi contoh pada spesifikasi tugas besar. Map di atas adalah map pada waktu T=1 dan belum ada item yang dipickup, sehingga belum ada bangunan yang berwarna biru.



Pada gambar di atas, bangunan biru adalah lokasi dropoff item teratas dari tas, kuning posisi mobita, merah lokasi pickup, hijau lokasi yang dapat dicapai oleh mobita, dan warna hitam adalah bangunan sisanya.

## 5.5 TO\_DO

```
ENTER COMMAND FOR MOVE : 2
Posisi Mobita saat ini : E (2,3)
Waktu saat ini: 1

ENTER COMMAND : TO_DO
1. G -> N (Normal Item)
2. M -> B (Heavy Item)
```

Menampilkan pesanan yang telah masuk ke dalam *to-do-list* berdasarkan waktu.

## 5.6 IN\_PROGRESS

```
ENTER COMMAND FOR MOVE : 2
Posisi Mobita saat ini : E (2,3)
Waktu saat ini: 1
2. M -> B (Heavy Item)
3. B -> M (Normal Item)

ENTER COMMAND : PICK_UP
Pesanan berupa Normal Item berhasil diambil!

ENTER COMMAND : TO_DO
1. M -> B (Heavy Item)
2. B -> M (Normal Item)

ENTER COMMAND : IN_PROGRESS
1. Normal Item (Tujuan: N)
```

Menampilkan tujuan pengiriman item yang telah di-*pick-up* sebelumnya dan berada di tas.

## 5.7 BUY

```
ENTER COMMAND : BUY
Uang yang tersedia : 200 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 FOR BUY: 1
Uang tidak cukup untuk membeli gadget.
```

Melakukan pembelian *gadget* menggunakan uang *player*. Command ini hanya dapat dilakukan saat *player* berada di *Headquarter*. Apabila uang *player* tidak cukup, maka *output* yang akan keluar adalah “Uang tidak cukup untuk membeli gadget”. Namun, apabila uang *player* mencukupi dan jumlah *gadget* pada *inventory* belum penuh (5 item), maka uang *player* akan berkurang seharga *gadget* dan ID *gadget* akan masuk ke dalam listpos.

## 5.8 INVENTORY

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

Menampilkan kepemilikan *gadget player* dan menggunakannya saat *player* memutuskan untuk menggunakannya. Apabila *gadget* tidak tersedia, maka *output* yang akan keluar adalah “Tidak ada *gadget* yang dapat digunakan!”. Namun, apabila *gadget* tersedia, maka kemampuan *gadget* dapat digunakan dan *gadget* yang sudah terpakai tersebut akan terhapus dari listpos *gadget*.

## 5.9 HELP

```
ENTER COMMAND : HELP
Berikut command yang tersedia
1. MOVE      -> Untuk berpindah ke lokasi selanjutnya
2. PICK_UP   -> Untuk mengambil item di lokasi sekarang
3. DROP_OFF  -> Untuk mendrop off item di lokasi sekarang (Khusus item yang baru saja di pick up)
4. MAP       -> Untuk menampilkan peta
5. TO_DO     -> Untuk menampilkan pesanan yang sudah bisa dilakukan pengambilannya (PICK_UP)
6. IN_PROGRESS -> Untuk menampilkan barang yang ada di tas dan siap untuk diantarkan
7. BUY       -> Untuk menampilkan gadget yang dapat kamu beli
8. INVENTORY -> Untuk menampilkan gadget yang telah kamu beli dan siap dipakai
9. HELP      -> Untuk menampilkan list command dan kegunaannya
```

Menampilkan list *command* yang tersedia pada permainan.

## 5.10 EXIT

```
ENTER COMMAND : EXIT
Apakah kamu yakin ingin keluar? (Y/N)
Y
PS C:\Users\geral\Dropbox\My PC gery\Downloads\alstrukdat_mobita>
```

Keluar dari permainan dan kembali ke terminal.

## 6 Test Script

| No. | Fitur yang Dites | Tujuan Testing  | Langkah-Langkah Testing  | Input Data Test | Hasil yang Diharapkan                    | Hasil yang Keluar                                   |
|-----|------------------|---|--|-----------------|--|---|
| 1   | MOVE             | Mengetahui perpindahan player                             | Jalankan perintah move, pilih lokasi yang dituju, cek map perpindahan player         | Data Test 5.1   | Player berpindah ke lokasi yang dituju   | Player berpindah ke lokasi yang dituju              |
| 2   | PICK_UP          | Mengetahui apakah command <i>pick up</i> dapat dijalankan | Jalankan perintah move, cari lokasi tempat pick up pesanan yang terdapat pada to do, | Data Test 5.2   | Player berhasil melakukan <i>pick up</i> | Player berhasil melakukan <i>pick up</i> , dan item |

| No. | Fitur yang Dites | Tujuan Testing   | Langkah-Langkah Testing  | Input Data Test | Hasil yang Diharapkan   | Hasil yang Keluar  |
|-----|------------------|--|--|-----------------|---|--|
|     |                  | sesuai ketentuan   |  |                 |   | disimpan ke dalam tas  |
| 3   | DROP_OFF         | Mengetahui apakah command <i>drop off</i> dapat dijalankan sesuai ketentuan tertentu | Jalankan perintah move, cari lokasi tempat <i>drop off</i> pesanan yang terdapat pada <i>in progress</i> , | Data Test 5.3   | Player berhasil melakukan <i>drop off</i>   | Player berhasil melakukan <i>pick up</i> , dan item dikeluarkan dari tas, serta mendapatkan uang dan ability |
| 4   | MAP              | Mengetahui fungsi MAP berjalan dengan baik atau tidak                                | Masukkan “MAP” pada “ENTER COMMAND”  | Data Test 5.4   | Mencetak map pada layar dengan ketentuan warna tertentu   | Mencetak map pada layar dengan ketentuan warna tertentu  |
| 5   | TO_DO            | Menampilkan <i>item</i> pesanan yang sudah masuk (di-order)                          | Menjalankan perintah <i>TO_DO</i> pada awal dan pertengahan permainan                                      | Data test 5.5   | Mencetak <i>to do list</i> jika ada <i>item</i> yang sudah masuk (di-order)                               | Mencetak <i>to do list item</i>  |
| 6   | IN_PROGRESS      | Menampilkan <i>item</i> yang sedang dalam pengantaran                                | Menjalankan perintah <i>IN_PROGRESS</i> pada awal permainan dan pertengahan pertanian                      | Data test 5.6   | Mencetak string yang menunjukkan tidak ada <i>item</i> yang sedang dibawa atau <i>item-item</i> jika ada. | Mencetak <i>item-item yang sedang diantarkan</i>   |
| 7   | BUY              | Mengetahui apakah command buy dapat berjalan dengan baik atau tidak                  | Memasukkan command “BUY” pada permainan dan membeli suatu gadget   | Data test 5.7   | Menyimpan ID gadget pada data player untuk digunakan di kemudian hari                                     | Menyimpan ID gadget pada data player   |
| 8   | INVENTORY        | Mengetahui apakah command inventory dapat berjalan dengan baik atau tidak            | Memasukkan command “INVENTORY” pada permainan kemudian memilih suatu gadget untuk digunakan.               | Data test 5.8   | Gadget yang telah digunakan terhapus dari list dan gadget dapat diimplementasikan.                        | Gadget terhapus dari list & kemampuan gadget dijalankan.   |
| 9   | HELP             | Menampilkan <i>command-command</i> yang bisa dilakukan pada program permainan        | Memasukan <i>command HELP</i>  | Data test 5.9   | Menampilkan semua <i>command</i> yang dapat dilakukan pada program  | Pesan ditampilkan.   |

| No. | Fitur yang Dites | Tujuan Testing                              | Langkah-Langkah Testing                    | Input Data Test | Hasil yang Diharapkan   | Hasil yang Keluar                           |
|-----|------------------|---|--|-----------------|---|---|
| 10  | EXIT             | Keluar dari program dan kembali ke terminal | Memasukan <i>command EXIT</i> dan <i>Y</i> | Data test 5.10  | Keluar dari program ketika memasukan <i>Y</i> saat validasi atau tetap dalam program jika tidak | Keluar dari program dan kembali ke terminal |

## 7 Pembagian Kerja dalam Kelompok

| No | Nama                    | Tugas   |
|----|-------------------------|---|
| 1  | Rania Dwi Fadhillah     | a) <i>Buy</i> dan <i>inventory</i><br>b) Implementasi efek <i>gadget</i><br>c) Mekanisme uang <i>player</i> |
| 2  | Aldwin Hardi Swastia    | a) <i>Main Program</i><br>b) <i>User interface</i><br>c) Fungsional program                                 |
| 3  | Mahesa Lizardy          | a) <i>Pick up, drop off</i> , dan <i>to do list</i><br>b) Implementasi jenis dan <i>ability</i> item        |
| 4  | Jundan Haris            | a) <i>Move &amp; map</i><br>b) Mekanisme waktu  |
| 5  | Gerald Abraham Sianturi | a) <i>To do list, in progress, help</i><br>b) Mekanisme tas <i>player</i>                                   |

## 8 Lampiran

### 8.1 Deskripsi Tugas Besar 2

# 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 ditinggalkannya 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.

27/10/2021

pg. 1 of 16

## 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.

27/10/2021

pg. 4 of 16

- 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



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

- Karakter 'B' 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: DRDP_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  *
*  H  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 di lokasi 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.

27/10/2021

pg. 10 of 16

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

\*\*\*\*\*

\*B        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 → Normal Item

27/10/2021

pg. 12 of 16

- 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
00010110000000000000
00000000000000010001
00001000100000000000
10001001100000000000
00110000100000000000
10000011010000000000
10000100010100000000
00010100111000000000
001110010010000100
000001110011100000
000000011100100100
000000100100111000
000000000111001010
010000000001001001
000000000001110001
0000000001010000010
000000000000100100
010000000000011000
10
1 G N N
3 G N H
2 B M N
```

27/10/2021

pg. 13 of 16

```

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 -> H (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:

- Membaca informasi konfigurasi dari file eksternal
- Membaca command dari interaksi user terhadap program, baik di dalam game atau di menu start, dan
- (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

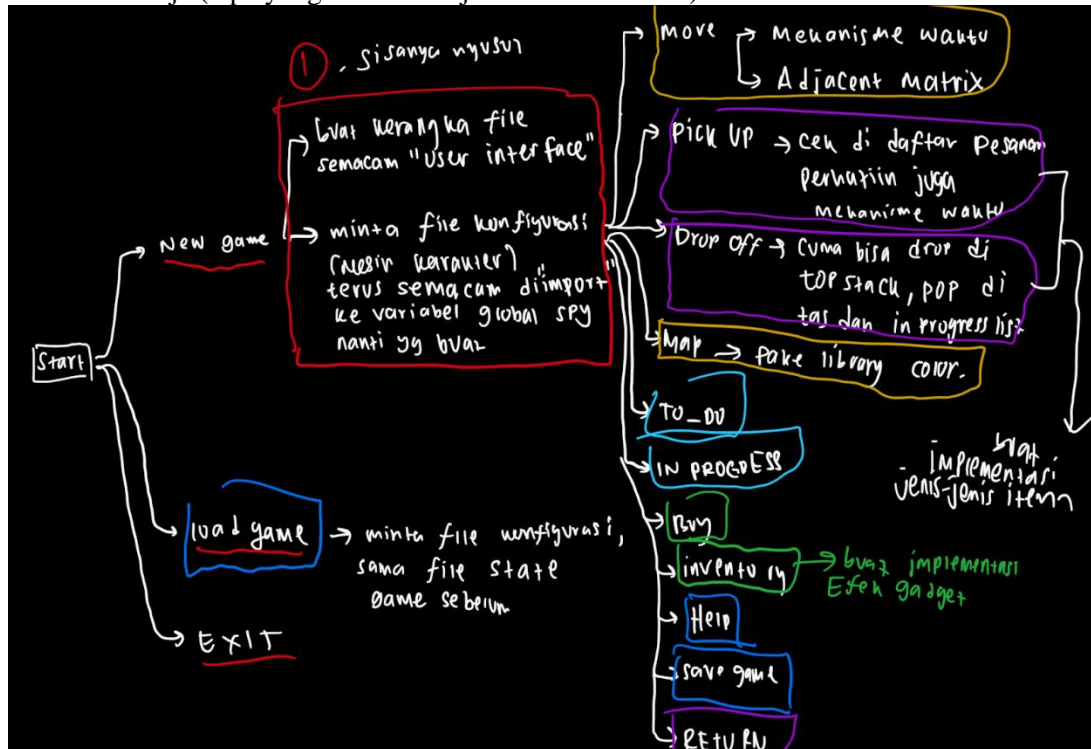
1. Sebagai saran, manfaatkan Makefile untuk mempermudah proses kompilasi dan penjalanan program. Bila sulit dalam menggunakan Makefile, bisa diakali dengan menggunakan shell script/batch file.
2. Usahakan buat struktur program yang serapi mungkin. Jangan buat semuanya pada file yang sama.
3. Manfaatkan 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.

## 8.2 Notula Rapat

Meeting kedua: 28 Oktober 2021

### Yang akan dibahas:

- Pembagian tugas
- Sistem kerja (Apa yang harus dikerjakan terlebih dahulu)



### Pembahasan:

Deadline: 8 November

- Main program [Aldwin] (ini harus dikerjakan duluan(?))
  - User interfacenya di terminal (Desain)
  - Fungsionalnya juga, kalau user input X, keluarnya B, terus konekin ke fungsi mana (Semacam driver), tapi driver
  - Nge"import" file konfigurasi ke variabel global yang nanti bisa diakses sama yang buat "fungsi-fungsi kecil" [MesinKar, jangan pake fopen fopenan]
- Urus MOVE dan MAP [Jundan]
  - Move: Buat **mekanisme waktu**, dan Adjacent Matrix dari Konfigurasi File, dll.
  - MAP: Pake library color, dll.
- Urus PICK UP dan DROP OFF [Mahesa]
  - Pake ADT yang udah ada dari praktikum, nanti paling ganti nama ADTnya aja, dll.
  - Implementasi **jenis dan ability item**
- Urus TO DO dan IN PROGRESS dan HELP [Gery]
  - Kerja sama sama orang yang (urus PICKUP dan DROP OFF), sama (urus MOVE dan MAP), dll.
  - Mekanisme **Tas Player**
- Urus BUY dan INVENTORY [Rania]
  - Buat **implementasi efek gadget (penggunaan gadget)** dll.
  - Mekanisme **uang player**
- Urus (Bonus) SAVE GAME dan (Bonus) LOAD GAME dan (Bonus) RETURN [ ]







### 8.3 Notula Asistensi

**Form Asistensi Tugas Besar  
IF2110/Algoritma dan Struktur Data  
Sem. 1 2021/2022**

No. Kelompok/Kelas : 5/K3  
Nama Kelompok :  
Anggota Kelompok (Nama/NIM) :  
1. Mahesa Lizardy/13520116  
2. Gerald Abraham Sianturi/13520138  
3. Rania Dwi Fadhillah/13520142  
4. Jundan Haris/13520155  
5. Aldwin Hardi Swastia/13520167  
  
Asisten Pembimbing : Gregorius Jovan Kresnadi/13518135







---

# Asistensi I

|  |  |
|--|--|
| <b>Tanggal :</b> 26 Oktober 2021   | <b>Catatan Asistensi:</b><br>1. Daftar pesanan sama to-do-list bedanya apa?<br>Awalnya masuk daftar pesanan, akan masuk to do list, kalau udah ada sesuai waktu<br>2. Pembagian tugas gimana bagusnya kak? bobot tiap tugas?<br>Coba <i>adjust</i> sendiri. Supaya cari pembagian tugas dengan baik. Penilaian nanti dipengaruhi <i>self</i> dan <i>peer assessment</i><br>3. Kalau Array ukurannya statik atau dinamis?<br>Sebenarnya bisa diperhitungkan sendiri, misal kalau inventory pasti 5, jadi pasti statik. Kalau inventory maksimal 100, coba dipikiri sendiri bagusnya gimana.<br>4. Adjacent matrix itu bagaimana?<br>Matriksnya berdasarkan hubungan di map, apakah bisa dari suatu titik ke titik lain. Itu ditentukan dari masukan pas <i>file konfigurasi</i> .<br>5. Ketika mau <i>pick up</i> barang di suatu lokasi, dan di lokasi itu ada banyak, apakah bisa langsung mengambil dengan "Pick up pick up ..."<br>Boleh, tapi harus common pick up lebih dari 1 kali<br>6. Kalau baca <i>file konfigurasi</i> boleh pakai fungsi <i>scanf</i> kah kak?<br>Tidak, harus menggunakan mesin karakter. Beda kasus kalau misal nerima <i>command</i> seperti PICK UP, MAP, MOVE, dsb. Ini baru yang bisa pakai <i>scanf</i><br>7. Kalau ada barang di lokasi yang sama dan waktu pesanan masuknya sama, prioritas berdasarkan apa?<br>Berdasarkan urutan di konfigurasi file.<br><br><b>Kesimpulan:</b><br>Bagi tugas secepatnya. Sarannya coba dibuat <i>flow</i> dari game. Atau dari pembagian tiap kebutuhan-kebutuhan tiap menu. Di mapping.<br><br>Bikin yang untuk show <i>map</i> dan baca <i>file konfigurasi</i> cukup ribet, jadi bisa 1 org aja.<br><br>Prioritasin fitur utama. Kalau bonus bisa nanti, dan juga karena yang bonus kemungkinan akan ada revisi juga. |
| <b>Tempat :</b> Google Meeting   |  |
| <b>Kehadiran Anggota Kelompok:</b><br>No<br>NIM<br>Tanda tangan<br><br>1. 13520116<br><br>2. 13520138<br><br>3. 13520142<br><br>4. 13520155<br><br>5. 13520167<br> |  |
|  | <b>Tanda Tangan Asisten:</b><br><br>Digitally signed by<br>Gregorius Jovan Kresnadi<br>Date: 2021.11.01 21:44:07<br>+07'00'  |



# Asistensi II

|  |  |
|--|--|
| <b>Tanggal :</b> 10 November 2021<br><b>Tempat :</b> Google Meeting  | <b>Catatan Asistensi:</b>  |
| <b>Kehadiran Anggota Kelompok:</b><br>No<br>NIM<br>Tanda tangan<br><br>1. 13520116<br><br>2. 13520138<br><br>3. 13520142<br><br>4. 13520155<br><br>5. 13520167<br> | 1. Perishable item, kalau speed bost ikutin waktunya tetep atau sesuai sebelumnya, tiap bangunan kurang 1?<br>Perishable ikutin yang "time global"<br>2. Perishable item bisa hangus dan langsung ke hapus dari tas, apakah pada algoritma bisa mengakses <i>buffer</i> nya secara langsung?<br>Boleh, tidak masalah.<br>Revisi, sebenarnya bisa pakai yang algoritma <i>tower of hanoi</i> , (sepertinya nyimpen item yang terpaksa ke pop untuk bisa ngakses perishable item ke stack lain, terus item yang terpaksa kepindah itu dibalikin lagi)<br><br><b>Kesimpulan:</b><br>Jangan lupa <i>deadline</i> tanggal 18 jam 21.10, jangan mepet <i>upload</i> di Olympianya. |
|  | <b>Tanda Tangan Asisten:</b><br><br> Digitally signed by<br>Gregorius Jovan<br>Kresnadi<br>Date: 2021.11.17<br>21:43:41 +07'00'  |

#### 8.4 Log Activity Anggota Kelompok

| Waktu dan tanggal          | Aktivitas yang dikerjakan  | Anggota Kelompok yang bekerja |
|----------------------------|--|-------------------------------|
| Minggu, 24/10/2021 (11:00) | Kumpul perdana, membahas spesifikasi dan menyamakan pemahaman.   | Seluruh anggota kelompok      |
| Selasa, 26/10/2021 (20:30) | Asistensi pertama.   |                               |
| Kamis, 28/10/2021 (21:00)  | Kumpul kedua, membahas pembagian tugas.  |                               |
| Selasa, 9/11/2021 (21:00)  | Kumpul ketiga, membahas progres pengerjaan masing-masing.  |                               |
| Rabu, 10/11/2021 (20:30)   | Asistensi kedua.   |                               |
| Senin, 15/11/2021 (21:00)  | Melanjutkan progres melakukan pengerjaan bersama untuk integrasi fungsionalitas-fungsionalitas yang ada. |                               |
| Selasa, 16/11/2021 (20:00) | Melanjutkan progress.  |                               |
| Rabu, 17/11/2021           | Mencoba me-run dan <i>debug</i> program untuk evaluasi program.  |                               |
| Kamis, 18/11/2021          | <i>Finishing</i> program dan <i>submission</i> tugas.  |                               |