

LAPORAN TUGAS BESAR
IF2110 Algoritma dan Struktur Data

MOBILITA

Dipersiapkan oleh:

Kelompok 04 – K03

Maria Khelli – 13520115


Mohamad Daffa Argakoesoemah -13520118

Bryan Amirul Husna - 135201146

Frederik Imanuel Louis - 13520163

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

| | | | | |
|-------------------------------------------------------------------------------------|-------------------------------------------------------|------------------------|----------|-------------------------|
|  | Sekolah Teknik Elektro dan Informatika ITB | Nomor Dokumen | | Halaman |
| | | <i>IF2110-TB-04-03</i> | | <i>32</i> |
| | | <i>Revisi</i> | <i>1</i> | <i>18 November 2021</i> |

Daftar Isi

| | |
|-----------------------------------------------|----|
| 1 Ringkasan..... | 4 |
| 2 Penjelasan Tambahan Spesifikasi Tugas | 4 |
| 2.1 Save Game | 4 |
| 2.2 Load Game..... | 4 |
| 2.3 VIP Item..... | 4 |
| 2.4 Return to Sender | 4 |
| 3 Struktur Data (ADT) | 5 |
| 3.1 ADT Point..... | 5 |
| 3.2 ADT List Statis | 5 |
| 3.3 ADT List Dinamis..... | 5 |
| 3.4 ADT Linked List..... | 6 |
| 3.4.1 ADT To Do | 6 |
| 3.4.2 ADT In Progress | 6 |
| 3.5 ADT Matriks | 6 |
| 3.5.1 ADT Map | 7 |
| 3.5.2 ADT Adjacency Matrix | 7 |
| 3.6 ADT Mesin Karakter | 7 |
| 3.7 ADT Mesin Kata..... | 8 |
| 3.8 ADT Queue | 8 |
| 3.9 ADT Stack | 8 |
| 3.10 ADT Tambahan | 8 |
| 3.10.1 ADT Lokasi | 9 |
| 3.10.2 ADT Pesanan | 9 |
| 3.10.3 ADT Mesin Konfigurasi | 9 |
| 3.10.4 ADT Mobita..... | 9 |
| 4 Program Utama | 12 |
| 5 Data Test | 12 |
| 5.1 Data Test 1: Program Utama (main.c) | 12 |
| 5.2 Data Test 2: Move..... | 13 |
| 5.3 Data Test 3: Pick Up..... | 14 |
| 5.4 Data Test 4: Drop Off | 14 |
| 5.5 Data Test 5: Map..... | 15 |
| 5.6 Data Test 6: To Do..... | 16 |
| 5.7 Data Test 7: In Progress | 16 |
| 5.8 Data Test 8: Buy | 17 |
| 5.9 Data Test 9: Inventory | 18 |
| 5.10 Data Test 10: Help | 18 |
| 5.11 Data Test 11: Save | 19 |
| 5.12 Data Test 12: Load..... | 21 |
| 5.13 Data Test 13: Balance | 23 |
| 5.14 Data Test 14: Return | 23 |
| 6 Test Script | 24 |
| 7 Pembagian Kerja dalam Kelompok | 28 |

| | |
|-----------------------------------------|----|
| 8 Lampiran | 28 |
| 8.1 Deskripsi Tugas Besar 2 | 28 |
| 8.2 Notulensi Rapat (Asistensi) | 29 |
| 8.3 Log Activity Anggota Kelompok | 31 |

1 Ringkasan

Program Mobita adalah sebuah permainan berbasis *Command Line Interface* (CLI) yang menyimulasikan pengantaran barang. Diceritakan bahwa karena pandemi COVID-19, Mobita ingin membantu orang tuanya dengan menjadi kurir pengantar barang. Tujuan permainan ini adalah mengantarkan pesanan yang dibawa Mobita kepada masing-masing penerima pesanan dengan waktu sesingkat mungkin.

Berdasarkan latar belakang di atas, kami membuat program permainan tersebut dengan beberapa bagian, antara lain menu, mekanisme waktu, daftar pesanan, *to do list*, *in progress list*, tas, jenis item, *ability*, *gadget*, *inventory gadget*, peta, perintah yang bisa dijalankan di setiap lokasi pada peta, dan kemampuan program untuk menyimpan *state* permainan serta melakukan *load* kembali *state* tersebut. Program ini dibuat menggunakan bahasa C dengan memanfaatkan konsep *Abstract Data Type* (ADT).

Secara garis besar, laporan ini terdiri dari ringkasan, penjelasan tambahan spesifikasi tugas, struktur ADT, program utama, *data test*, *test script*, pembagian kerja dalam kelompok, dan lampiran.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Save Game

Save game adalah fitur menyimpan *state* permainan untuk dilanjutkan pada lain waktu. Save game diimplementasikan dalam command save, dan akan membuat “save file” yang memuat *state* game. Save file tersebut disimpan pada folder *saved_files* dengan ekstensi “.mob.”.

2.2 Load Game

Load game mengambil save file yang dibuat oleh command save dan mengubah *state* permainan sesuai data yang disimpan pada save file.

2.3 VIP Item

VIP item adalah salah satu jenis pesanan yang dapat diantarkan. Jika ada VIP yang masuk, pengguna tidak bisa pick up pesanan yang bukan VIP, tetapi masih bisa drop off pesanan (misal tas penuh tentu perlu drop off terlebih dahulu). Pesanan bertipe ini memiliki harga yang tinggi, yaitu 600 Yen, dan memberikan *ability* return to sender jika berhasil diantarkan.

2.4 Return to Sender

Ability Return to Sender adalah *ability* yang mengirimkan pesanan terakhir yang dipickup (top of tas), dan mengembalikannya (instan) ke lokasi pickup. Ability Return to Sender didapatkan apabila berhasil mengantarkan VIP Item. Ability ini bersifat non-stackable (maksimal satu di “inventory” *ability*).

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------------|
| STEI- ITB | IF2110-TB-04-03 | Halaman 4 dari 32 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

3 Struktur Data (ADT)

3.1 ADT Point

- **Sketsa struktur data:** struktur Point terdiri atas nilai x dan y yang bertipe integer. Primitifnya terdiri atas konstruktor yang membentuk suatu point.
- **Persoalan yang diselesaikan:** mencatat koordinat bangunan pada map.
- **Alasan pemilihan:** Koordinat tiap bangunan pada map terdiri atas dua elemen x dan y yang lebih mudah dinyatakan dalam bentuk struct point dengan komponen absis dan ordinat.
- Diimplementasikan sebagai ADT Point dengan **nama file header "point.h"**.

3.2 ADT List Statis

- **Sketsa struktur data:** struktur Gadget List berbentuk List of Struct Statis dimana masing-masing elemen menyimpan nama gadget (name), harga gadget (price), dan banyak gadget yang dimiliki (owned).
- **Persoalan yang diselesaikan:** mencatat inventory gadget Mobita.
- **Alasan pemilihan:** setiap gadget memiliki nama, harga, masing-masing dapat dibeli lebih dari satu kali. Dengan itu, lebih mudah apabila semua informasi mengenai gadget disimpan dalam bentuk list statis of info gadget karena banyak jenis gadget tidak pernah berubah (5 jenis).
- Diimplementasikan sebagai ADT GadgetList dengan **nama file header "gadgetlist.h"**.

3.3 ADT List Dinamis

- **Sketsa struktur data:** menyimpan pointer elemen pertama dari ADT lokasi, nilai efektif, dan kapasitas dari list.

Di dalam ADT ini, terdapat fungsi primitif untuk membuat, menambahkan, dan menghapus elemen lokasi. Selain itu, ditambahkan fungsi getter untuk mengambil daftar bangunan yang dapat diakses sesuai adjacency matrix.

- **Persoalan yang diselesaikan:** mencatat daftar bangunan yang diinisialisasi saat pembacaan file konfigurasi secara dinamis. Jadi, list dialokasikan sesuai dengan banyaknya bangunan.
- **Alasan pemilihan:** alokasi array (atau list) yang dinamis dapat menghemat memori penyimpanan. Jumlah bangunan maksimal adalah 600 lokasi karena ukuran maksimal map adalah 20×30 . Jika dibuat statis, kita harus selalu mengalokasikan list berisi 600 elemen lokasi. Padahal, jumlah bangunan biasanya masih dalam kisaran puluhan.

- Diimplementasikan sebagai ADT DynamicList dengan **nama file header** “dynamiclist.h”.

3.4 ADT Linked List

Terdapat dua ADT yang mewarisi karakteristik ADT Linked List, yaitu to do dan in progress list.

3.4.1 ADT To Do

- **Sketsa struktur data:** ADT ini terdiri dari Address FIRST yang merupakan elemen pertama dari pesanan. Dalam ADT ini, terdapat fungsi untuk membuat to do list baru, menambah dan menghapus pesanan dari to do list, dan prosedur untuk menampilkan to do list.
- **Persoalan yang diselesaikan:** penyimpanan daftar pesanan dalam to do list
- **Alasan pemilihan:** Dalam permainan ini, to do list akan sering berubah seiring dengan Mobita mengantarkan pesannya. Maka dari itu, dibutuhkan struktur data yang cukup efisien secara umum dalam waktu yang dibutuhkan untuk melakukan operasinya-operasinya.
- **Diimplementasikan** sebagai ADT To Do dengan **nama file header** “linked_list_to_do.h”.

3.4.2 ADT In Progress

- **Sketsa struktur data:** ADT ini terdiri dari Address FIRST yang merupakan elemen pertama dari pesanan. Dalam ADT ini, terdapat fungsi untuk membuat in progress baru, menambah dan menghapus pesanan dari in progress, dan prosedur untuk menampilkan in progress.
- **Persoalan yang diselesaikan:** penyimpanan daftar pesanan dalam in progress
- **Alasan pemilihan:** Dalam permainan ini, in progress akan sering berubah seiring dengan Mobita mengambil dan mengantarkan pesannya. Maka dari itu, dibutuhkan struktur data yang cukup efisien secara umum dalam waktu yang dibutuhkan untuk melakukan operasinya-operasinya.
- **Diimplementasikan** sebagai ADT In Progress dengan **nama file header** “linked_list_in_progress.h”.

3.5 ADT Matriks

Terdapat dua ADT yang mewarisi karakteristik ADT Matriks, yaitu map dan adjacency matrix. Kedua ADT ini dipisah sesuai fungsionalitasnya masing-masing.

3.5.1 ADT Map

- **Sketsa struktur data:** menyimpan matriks (array of array) dari lokasi, indeks efektif baris, dan indeks efektif kolom. Matriks bersifat statis dan berukuran $(\text{baris} + 2) \times (\text{kolom} + 2)$ dengan 4 kolom / baris tambahan sebagai batas map.
- **Persoalan yang diselesaikan:** merepresentasikan peta dengan posisi player dan warna bangunan yang dapat berubah.
- **Alasan pemilihan:** peta memiliki karakteristik dua dimensi sehingga paling tepat diimplementasikan sebagai matriks.
- **Diimplementasikan** sebagai ADT Map dengan **nama file header “map.h”**.

3.5.2 ADT Adjacency Matrix

- **Sketsa struktur data:** menyimpan matriks (array of array) dari tipe data integer dan ukuran efektif dari matriks. Matriks ini selalu merupakan matriks persegi sehingga hanya perlu menyimpan orde matriks saja.
- **Persoalan yang diselesaikan:** merepresentasikan matriks ketetanggaan yang mencatat bangunan mana saja yang dapat diakses dari bangunan X dengan X sesuai urutan masukan file konfigurasi, misalnya HQ selalu dimasukkan pertama, maka baris pertama pasti merepresentasikan bangunan yang dapat diakses dari lokasi HQ.
- **Alasan pemilihan:** sesuai namanya, matriks ketetanggaan paling tepat diimplementasikan dalam ADT Matriks.
- **Diimplementasikan** sebagai ADT AdjMatrix dengan **nama file header “adjmat.h”**.

3.6 ADT Mesin Karakter

- **Sketsa struktur data:** menyimpan variabel dengan scope extern, yaitu currentChar, dan eof, sehingga dapat digunakan dalam file lain. Selain itu, terdapat juga start, adv, dan end untuk membaca karakter lewat input pengguna. Mesin karakter juga dilengkapi dengan fungsi startReadFile, advReadFile, dan endReadFile untuk menangani kasus baca input lewat file eksternal.
- **Persoalan yang diselesaikan:** membaca karakter dari file atau masukan pengguna dan mensimulasikan mesin karakter dunia nyata.
- **Alasan pemilihan:** membaca input per karakter dengan abstraksi mesin karakter.
- **Diimplementasikan** sebagai ADT charmachine dengan **nama file header “charmachine.h”**.

3.7 ADT Mesin Kata

- **Sketsa struktur data:** merupakan ADT ekstensi dari ADT mesin karakter. ADT ini menyimpan variabel scope extern `endWord` dan `currentWord`. Terdapat juga struktur data `Word` yang menyimpan array of character (berfungsi sebagai string) dan panjang dari string tersebut. Maksimal karakter per kata adalah 50.
- **Persoalan yang diselesaikan:** membaca kata dari file atau masukan pengguna.
- **Alasan pemilihan:** membaca input per kata dengan abstraksi mesin kata.
- Diimplementasikan sebagai ADT `wordmachine` dengan **nama file header** “`wordmachine.h`”.

3.8 ADT Queue

- **Sketsa struktur data:** merupakan struktur data *queue* yang diimplementasikan dengan array tipe data Pesanan dengan kapasitas maksimal array 100. ADT ini terdiri atas `idxHead` yang menyimpan indeks head dan `idxTail` yang menyimpan indeks tail.
- **Persoalan yang diselesaikan:** Daftar antrean Pesanan yang belum masuk waktu.
- **Alasan pemilihan:** Pesanan yang memiliki waktu terkecil akan berada paling depan sehingga cocok untuk mengambil Pesanan satu per satu yang sudah masuk waktu.
- Diimplementasikan sebagai ADT `QueuePesanan` dengan **nama file header** “`queuepesanan.h`”.

3.9 ADT Stack

- **Sketsa struktur data:** merupakan struktur data *stack* yang diimplementasikan dengan array dari tipe data Pesanan. Jumlah maksimal Pesanan yang dapat disimpan adalah 100 Pesanan. ADT ini terdiri `IDX_TOP` yang merupakan indeks dari elemen teratas *stack* dan `TOP` yang merupakan elemen teratas dari *stack*.
- **Persoalan yang diselesaikan:** tas Mobita yang digunakan untuk menyimpan item yang diambil di lokasi.
- **Alasan pemilihan:** aturan masuk item dalam tas Mobita menggunakan aturan LIFO (*Last In First Out*) sehingga *stack* sangat cocok digunakan untuk menyelesaikan masalah ini.
- Diimplementasikan sebagai ADT Stack dengan **nama file header** “`stack.h`”.

3.10 ADT Tambahan

Bagian ini berisi struktur data abstrak tambahan yang dibuat untuk memudahkan pemrograman sehingga modularitas bahasa C dapat dimanfaatkan sebaik mungkin.

3.10.1 ADT Lokasi

- **Sketsa struktur data:** menyimpan tipe data point, karakter sebagai nama bangunan, dan tipe data warna melalui definisi yang didefinisikan dari enum bahasa C.
- **Persoalan yang diselesaikan:** representasi lokasi bangunan dan lokasi player (mobita) dalam game.
- **Alasan pemilihan:** jalannya program sangat bergantung dengan informasi lokasi, maka dibuatlah sebuah ADT khusus sehingga dapat dipakai berulang-ulang.
- **Diimplementasikan** sebagai ADT Location dengan **nama file header** “location.h”.

3.10.2 ADT Pesanan

- **Sketsa struktur data:** merupakan struktur untuk menyimpan data pesanan. Struktur ini terdiri atas Location pickup, Location dropOff, waktuIn, price, timeout, timeoutInitial, dan JenisItem. Untuk memudahkan klasifikasi dan operasi, jenis item pesanan dibuat enum tersendiri yaitu JenisItemEnum.
- **Persoalan yang diselesaikan:** penyimpanan data pesanan secara terpadu
- **Alasan pemilihan:** pesanan memiliki banyak informasi sehingga perlu dibuat struktur data tersendiri
- **Diimplementasikan** sebagai ADT Pesanan dengan **nama file header** “pesanan.h”.

3.10.3 ADT Mesin Konfigurasi

- **Sketsa struktur data:** merupakan turunan dari mesin karakter, fungsionalitas utamanya untuk membaca file konfigurasi dari user. Terdapat fungsi-fungsi tambahan misalnya mengonversikan karakter ke angka dan membaca integer yang memiliki digit lebih dari 1.
- **Persoalan yang diselesaikan:** membaca file konfigurasi, terutama untuk membaca lokasi.
- **Alasan pemilihan:** memberikan abstraksi pada program sehingga program utama menjadi lebih bersih.
- **Diimplementasikan** sebagai ADT configmachine dengan **nama file header** “configmachine.h”.

3.10.4 ADT Mobita

- Sketsa struktur data:
 - a. **Properti struktur data** Mobita terdiri atas:

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------------------------|
| STEI- ITB | IF2110-TB-04-03 | Halaman 9 dari 32 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

1. balance: nilai yang merepresentasikan uang Mobita (tipe integer)
 2. tasCapacity: nilai yang merepresentasikan kapasitas tas mobita (tipe integer)
 3. curLoc: menyimpan lokasi Mobita (ADT Location)
 4. toDoList: menyimpan pesanan yang masuk ke aplikasi Mobilita dan belum dipickup oleh Mobita (ADT List)
 5. inProgressList: menyimpan pesanan yang sudah di-pick up oleh Mobita dan belum di-drop off (ADT List)
 6. tas: menyimpan isi tas, dengan top tas merupakan pesanan terakhir yang masuk (ADT Stack)
 7. inventory: menyimpan inventory gadget yang dibeli Mobita (ADT GadgetList)
 8. buildingList: menyimpan daftar bangunan yang ada di peta (ADT DynamicList)
 9. adjMatrix: menyimpan adjacency list bangunan di peta (ADT AdjMatrix)
 10. map: menyimpan grid peta (ADT Map)
 11. speedBoostAbility: menyimpan sisa waktu aktif Speed Boost (tipe integer)
 12. returnToSenderAbility: menyimpan status ability Return to Sender (tipe boolean)
- b. Selain data diatas, Mobita juga menggunakan **variabel global** berikut:
1. globalTime: menyimpan waktu permainan (tipe integer)
 2. daftarPesanan: menyimpan daftar pesanan yang dibaca dari file konfigurasi atau savefile (tipe QueuePesanan)
 3. delivered: menyimpan banyak pesanan yang berhasil diantarkan untuk keperluan skor permainan (tipe integer)
- c. **Fungsi-fungsi** yang disediakan ADT Mobita antara lain:
1. CreateMobita: membentuk Mobita
 2. CommandMove: memindahkan Mobita sesuai peta
 3. CommandPickup: mengambil Pesanan pada lokasi Mobita
 4. CommandDropOff: mengantarkan pesanan yang sudah di-pickup pada lokasi Mobita
 5. CommandMap: menampilkan peta
 6. CommandToDo: menampilkan To Do List
 7. CommandInProgress: menampilkan In Progress List
 8. CommandBuy: mengakses Shop di HQ
 9. CommandInventory: mengecek gadget yang sudah dibeli dan menggunakannya jika diinginkan
 10. CommandHelp: menampilkan Help menu
 11. CommandSave: membuat savefile

12. CommandLoad: mengubah game state sesuai savefile
 13. CommandBalance: menampilkan uang Mobita
 14. CommandReturn: menggunakan ability Return To Sender
- d. Adapun beberapa **fungsi internal yang hanya digunakan oleh ADT Mobita**, yaitu:
1. UseKainWaktu: menggunakan gadget kain waktu (dipanggil melalui fungsi CommandInventory)
 2. UseSenterPembesar: menggunakan gadget senter pembesar (dipanggil melalui fungsi CommandInventory)
 3. UsePintuKemanaSaja: menggunakan pintu kemana saja (dipanggil melalui fungsi CommandInventory)
 4. UseMesinWaktu: menggunakan mesin waktu (dipanggil melalui fungsi CommandInventory)
 5. UseSenterPengecil: menggunakan senter pengecil (dipanggil melalui fungsi CommandInventory)
 6. getInputCommand: mengambil input dari user dengan pesan tertentu (digunakan berbagai fungsi)
 7. updateLocationColor: mengupdate warna lokasi untuk keperluan display peta (digunakan berbagai fungsi)
- **Persoalan yang diselesaikan:** ADT Mobita menjalankan command-command permainan dan menyimpan state game dengan menggunakan ADT-ADT lain yang telah dibuat.
 - **Alasan pemilihan:** Penggabungan ADT-ADT yang dibuat untuk melaksanakan user command akan lebih rapi jika dilakukan dengan ADT yang modular. Proses save dan load pun akan lebih mudah jika state game jelas dinyatakan dalam suatu ADT.
 - Diimplementasikan sebagai ADT Mobita dengan **nama file header "mobita.h"**.

4 Program Utama

Program utama dimulai dengan menginclude semua file header ADT yang telah dibuat. Program utama memuat fungsi main dan beberapa fungsi tambahan. Fungsi tambahan tersebut antara lain:

- `printCmd`: digunakan untuk menampilkan berbagai informasi penting kepada pemain, seperti waktu, lokasi, dan daftar command
- `getCmd`: digunakan untuk meminta command dari user sampai command valid.
- `UpdatePesanan`: digunakan untuk mengupdate to do list dari daftar pesanan yang dimuat pada config file atau save file
- `notDone`: mengecek apakah permainan sudah berakhir (tidak ada lagi pesanan yang dapat diantarkan)

Program main dimulai dengan membuat objek Mobita dan menampilkan menu title screen (new game, load game, atau exit). Kemudian, program memproses file konfigurasi atau save file dan memulai permainan.

Permainan yang berada pada while loop yang hanya akan keluar jika permainan selesai (dicek melalui `notDone`) atau jika user memasukkan command `exit`. Pada tiap iterasi, program akan mengupdate to do list (dengan fungsi `UpdatePesanan`), menampilkan informasi game state pada user (dengan fungsi `printCmd`), dan kemudian meminta input dari user (dengan `getCmd`). Kemudian, program akan memanggil command yang sesuai dari ADT Mobita. Jika program keluar dari while loop, maka program akan menampilkan end message apabila permainan sudah selesai, atau thanks message apabila pemain keluar dari permainan sebelum permainan berakhir.

5 Data Test

5.1 Data Test 1: Program Utama (main.c)

Fitur ini dites untuk memastikan bahwa program utama sudah berjalan. Cara kompilasi program adalah dengan memanggil `make` pada command prompt. Pada Windows dengan MinGW, dapat memanggil `MinGW32-make` atau dengan menginstall Windows Subsystem for Linux, menginstall gcc dan make untuk memanggil `make`. Setelah dipanggil, akan muncul file bertipe `.exe` yaitu executable file untuk menjalankan program utama. Jika dipanggil, tampilannya akan seperti berikut.

```
PS D:\Programming\Programming Portfolio\Github\Tugas Besar\IF2110-Tubes-Alstrukdat> ./main
Selamat datang di permainan Mobilita!
1. New Game
2. Load Game
3. Exit
```

Gambar 5.1 Program utama berhasil

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|----------------------------|
| STEI- ITB | IF2110-TB-04-03 | Halaman 12 dari 32 halaman |
| Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB. | | |

5.2 Data Test 2: Move

Fitur yang dites adalah command move yang digunakan untuk bergerak ke bangunan lain. Hasil yang diharapkan adalah ditampilkan lokasi yang bisa diakses dari lokasi saat ini, kemudian pengguna memilih lokasi yang dituju dan Mobita akan berpindah ke lokasi tujuan. Pesanan baru diupdate tiap dilakukan command move.

```
Enter Command: 1
Posisi yang dapat dicapai:
1. C (1, 9)
2. E (2, 3)
3. F (3, 1)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
Enter Command: 2
Mobita sekarang berada di titik E (2, 3)

Anda mendapatkan 2 pesanan baru!
```

Gambar 5.2 Move berhasil

Waktu bertambah sesuai dengan pesanan yang ada di tas: jika ada Heavy waktu bertambah satu ditambah jumlah item Heavy, jika ada ability speed boost waktu bertambah tiap dua move, dan jika tidak ada keduanya waktu bertambah satu. Akan ada pesan kesalahan jika pengguna memilih lokasi yang tidak ada di daftar dan diulang. Pengguna dapat juga membatalkan perpindahan.

```
Enter Command: 1
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 dipilih? (ketik 0 jika ingin kembali)
Enter Command: 1
Mobita sekarang berada di titik F (3, 1)

Anda mendapatkan 1 pesanan baru!
Waktu sekarang adalah 11
Lokasi: F (3, 1)
```

```
Pesanan yang sedang diantarkan:
1. Heavy Item (Tujuan: B)

Waktu sekarang adalah 9
Lokasi: K (6, 3)
```

Gambar 5.3 Tiap Heavy item menambah 1 unit waktu untuk perpindahan

```
Pesanan HEAVY Item berhasil diantarkan
Uang yang didapatkan: 400 Yen

Waktu sekarang adalah 15
Lokasi: B (1, 15)
```

```
Enter Command: 1
Posisi yang dapat dicapai:
1. D (1, 15)
2. H (3, 14)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
Enter Command: 1
Mobita sekarang berada di titik D (1, 15)

Waktu sekarang adalah 15
Lokasi: D (1, 15)
```

```
Enter Command: 1
Posisi yang dapat dicapai:
1. B (1, 15)
2. C (1, 9)
3. H (3, 14)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
Enter Command: 2
Mobita sekarang berada di titik C (1, 9)

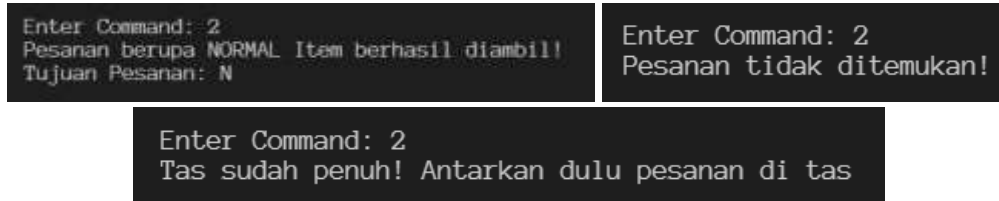
Waktu sekarang adalah 16
Lokasi: C (1, 9)
```

Gambar 5.4 Jika ada ability speed boost dan tidak ada Heavy, waktu bertambah satu tiap dua perpindahan

Maka, tes ini sesuai dengan spesifikasi.

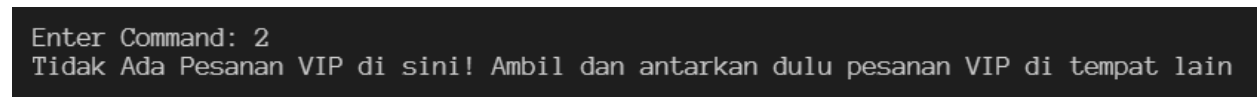
5.3 Data Test 3: Pick Up

Fitur yang dites adalah command pick up yang digunakan untuk menambahkan pesanan ke tas. Hasil yang diharapkan adalah pesanan dipindahkan dari to do ke tas dan in progress list. Jika tas penuh atau tidak ada pesanan di lokasi, ditampilkan peringatan dan tidak jadi diambil.



Gambar 5.5 Pick up berhasil, dan peringatan jika tidak ada pesanan di lokasi atau tas penuh

Jika ada pesanan VIP di tempat lain dan yang diambil saat ini bukan pesanan VIP, maka pengambilan gagal dan ditampilkan peringatan.

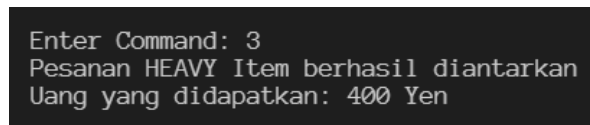


Gambar 5.6 Ada pesanan VIP di tempat lain

Maka, tes ini sesuai dengan spesifikasi.

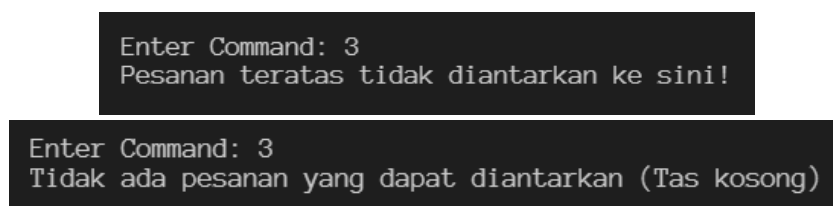
5.4 Data Test 4: Drop Off

Fitur yang dites adalah command drop off yang digunakan untuk mengantarkan pesanan ke tujuan. Hasil yang diharapkan adalah pesanan dihapus dari tas dan in progress list serta saldo Mobita bertambah dan mendapat ability tergantung jenis item. Maka, tes ini sesuai dengan spesifikasi.



Gambar 5.7 Pesanan berhasil diantarkan

Jika top tas tidak diantarkan ke tempat saat ini atau tas kosong, ditampilkan peringatan.

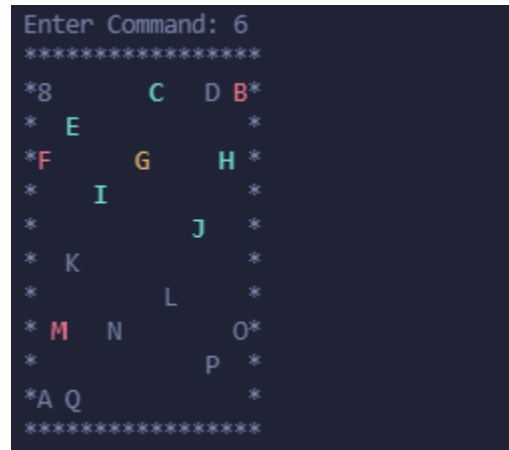


Gambar 5.8 Pesanan gagal diantarkan karena tidak cocok atau tas kosong

Maka, tes ini sesuai dengan spesifikasi.

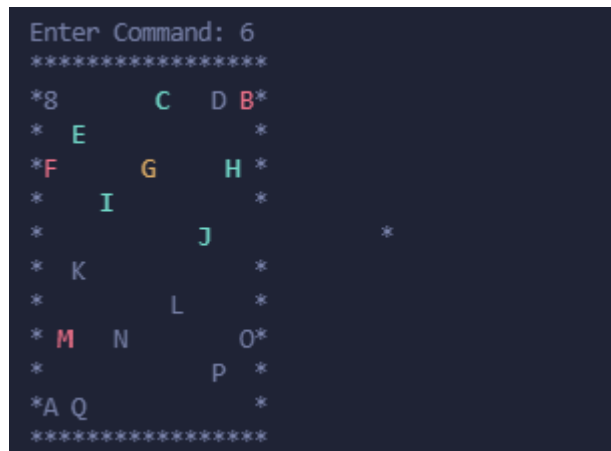
5.5 Data Test 5: Map

Fitur yang dites adalah command map yang digunakan untuk menampilkan peta. Hasil yang diharapkan adalah peta muncul ke layar dengan warna yang sesuai.



Gambar 5.9 Map berhasil

Hanya saja, mungkin terkadang ada masalah pada console sehingga walaupun program (kode) tidak berubah sama sekali, tetap terjadi output yang kurang rapi.



Gambar 5.10 Map gagal (masalah pada console)

5.6 Data Test 6: To Do

Fitur yang dites adalah command to do. Hasil yang diharapkan adalah list pesanan yang harus dikerjakan muncul ke layar.

```
Enter Command: 4
Pesanan pada To Do List:
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)
```

Gambar 5.11 To do tanpa VIP berhasil

Jika ada item VIP, antarmuka pengguna akan terlihat seperti gambar berikut.

```
Enter Command: 4
Pesanan pada To Do List:
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)
```

Gambar 5.12 To do dengan VIP berhasil

Test ini sesuai dengan spesifikasi.

5.7 Data Test 7: In Progress

Fitur yang dites adalah command in progress list. Hasil yang diharapkan adalah akan muncul daftar-daftar pesanan yang sedang diantarkan. Saat tidak ada pesanan, antarmuka pengguna akan tampak seperti gambar berikut.

```
Enter Command: 5
Tidak ada pesanan yang sedang diantarkan!
```

Gambar 5.13 In progress tanpa pesanan berhasil

Sementara itu, jika ada pesanan tampilannya adalah sebagai berikut.


```

Enter Command: 5
Pesanan yang sedang diantarkan:
1. Heavy Item (Tujuan: N)
2. Normal Item (Tujuan: E)
3. Normal Item (Tujuan: N)

```

Gambar 5.14 In progress tanpa pesanan

Test ini sesuai dengan spesifikasi.

5.8 Data Test 8: Buy

Fitur yang dites adalah command buy. Hasil yang diharapkan adalah tampilan *gadget* yang dapat dibeli serta membelinya jika *balance* cukup. Jika *balance* cukup dan item berhasil dibeli, antarmuka pengguna akan tampak seperti gambar berikut.

```

Enter Command: 7
Uang anda sekarang: 800 yen
Mobita HQ Shop:
1. Kain Pembungkus Waktu (800 yen)
2. Senter Pembesar (1200 yen)
3. Pintu Kemana Saja (1500 yen)
4. Mesin Waktu (3000 yen)
5. Senter Pengecil (800 yen)
Gadget mana yang ingin dibeli? (ketik 0 untuk kembali)

Enter Command: 1
Transaksi berhasil. Uang anda sekarang adalah 0 yen

```

Gambar 5.15 Pembelian *gadget* di *shop* berhasil

Sementara itu, jika *balance* tidak mencukupi, tampilan akan seperti gambar berikut.

```

Enter Command: 7
Uang anda sekarang: 0 yen
Mobita HQ Shop:
1. Kain Pembungkus Waktu (800 yen)
2. Senter Pembesar (1200 yen)
3. Pintu Kemana Saja (1500 yen)
4. Mesin Waktu (3000 yen)
5. Senter Pengecil (800 yen)
Gadget mana yang ingin dibeli? (ketik 0 untuk kembali)

Enter Command: 2
Uang anda tidak mencukupi

```

Gambar 5.16 Pembelian *gadget* di *shop* gagal karena *balance* tidak mencukupi

Berdasarkan hasil tes di atas, tes ini sesuai dengan spesifikasi.

5.9 Data Test 9: Inventory

Fitur yang dites adalah command inventory. Hasil yang diharapkan adalah tampilan daftar *gadget* yang dapat dipakai dan *gadget* dapat digunakan jika ada. Jika terdapat *gadget* dan *gadget* digunakan, tampilan antarmuka pengguna akan tampak seperti gambar berikut.

```
Enter Command: 8
1. Kain Pembungkus Waktu (x1)
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 untuk kembali)

Enter Command: 1
Kain Waktu berhasil digunakan!
Karena senang deadline delivery diperpanjang, anda melempar kain tersebut dan melupakannya di pinggir jalan
```

Gambar 5.17 *Gadget* di dalam *inventory* berhasil digunakan

Sementara itu, jika tidak terdapat *gadget* di dalam *inventory*, tampilan akan seperti gambar berikut.

```
Enter Command: 8
Inventory Kosong
Silahkan membeli gadget di HQ!
```

Gambar 5.18 *Gadget* di dalam *inventory* kosong

Berdasarkan hasil tes di atas, tes ini sesuai dengan spesifikasi.

5.10 Data Test 10: Help

Fitur yang dites adalah command help. Hasil yang diharapkan adalah tampilan daftar command yang dapat dijalankan beserta penjelasannya. Jika command help dijalankan, tampilan antarmuka pengguna akan tampak seperti gambar berikut.

```
Enter Command: 11
Command List
1. Move -> Bergerak, bergerak, ayo selesaikan tugas! Command ini akan menampilkan lokasi yang bisa dicapai.
2. Pick Up -> Ambil barang pesananmu, tetapi jangan salah ambil ya!
3. Drop Off -> Antar barang pesananmu, tetapi jangan salah antar ya!
4. To Do List -> Daftar yang harus kamu kerjakan, makanya ayo cepat !! >:(
5. In Progress List -> Daftar yang sedang kamu kerjakan sekarang.
6. Map -> Mau liat kamu di mana? Nggak bisa pakai Google Maps. Ayo lihat di sini saja.
7. Buy -> Beli barang AJAIB, tetapi cuma bisa dibeli di HQ!
8. Inventory -> Barang yang kamu beli dan masih tersedia.
9. Balance -> Uangmu sekarang. Ayo kerja lebih giat!!
10. Return -> Balik ke checkpoint *hanya berlaku pada item tertentu*.
11. Help -> Uh..
12. Save -> Cape main? Sini-sini.
13. Exit
```

Gambar 5.19 Tampilan command help

Berdasarkan hasil tes di atas, tes ini sesuai dengan spesifikasi.

5.11 Data Test 11: Save

Fitur yang dites adalah command save. Hasil yang diharapkan adalah muncul output file pada folder saved_files dengan ekstensi .mob. Setelah memilih command save, antarmuka pengguna akan tampak sebagai berikut.

```
Waktu sekarang adalah 4
Lokasi: M (8, 2)
Command List
1. Move
2. Pick Up
3. Drop Off
4. To Do List
5. In Progress List
6. Map
7. Shop
8. Inventory
9. Balance
10. Return
11. Help
12. Save
13. Exit

Enter Command: 12
Masukkan nama file: slot4
```

Gambar 5.20 UI command save

Selain itu, file .mob yang dihasilkan adalah sebagai berikut.

```

C:\Users\Erik\IF2110-Tubes-Alstrukdat\saved_files>more slot4.mob
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 1 0 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 1 1 1 0 0 1 0 0 1 0 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

```

Gambar 5.21 Hasil savefile halaman pertama

```

5
5 C D N
5 G N P 10
8 O P V
8 Q L V
10 P Q P 20
4
0 0
0
3
M 8 2
4
3 8 G 8 6 N 1 200 -1 -1 N
1 15 B 8 2 M 2 200 -1 -1 N
3 8 G 8 6 N 3 400 -1 -1 H
3 1 F 2 3 E 3 200 -1 -1 N
1
8 2 M 1 15 B 1 400 -1 -1 H
0 0 0 0 0

```

Gambar 5.22 Hasil savefile halaman kedua

Test ini sesuai dengan spesifikasi.

5.12 Data Test 12: Load

Fitur yang dites adalah command load. Hasil yang diharapkan adalah state game berlanjut dari save file yang sudah dibuat pada test save. Berikut merupakan beberapa data gamestate permainan asli.

```

Waktu sekarang adalah 4
Lokasi: M (8, 2)

```

Gambar 5.23 Data waktu dan lokasi gamestate asli

```

Enter Command: 4
Pesanan pada To Do List:
1. G -> N (Normal Item)
2. B -> M (Normal Item)
3. G -> N (Heavy Item)
4. F -> E (Normal Item)

```

Gambar 5.24 Data to do list gamestate asli

```

Enter Command: 5
Pesanan yang sedang diantarkan:
1. Heavy Item (Tujuan: B)

```

Gambar 5.25 Data in progress list gamestate asli

```

Enter Command: 8
Inventory Kosong
Gadget mana yang ingin digunakan? (ketik 0 untuk kembali)

```

Gambar 5.26 Data inventory gamestate asli

```

Enter Command: 9
Uang anda sekarang adalah: 0

```

Gambar 5.27 Data balance gamestate asli

Kemudian, berikut adalah antarmuka pengguna saat melakukan load, dan beberapa data gamestate permainan setelah load.

```

Selamat datang di permainan Mobilita!
1. New Game
2. Load Game
3. Exit

Enter Command: 2
Masukkan nama file: slot4

Waktu sekarang adalah 4
Lokasi: M (8, 2)
Command List
1. Move
2. Pick Up
3. Drop Off
4. To Do List
5. In Progress List
6. Map
7. Shop
8. Inventory
9. Balance
10. Return
11. Help
12. Save
13. Exit

Enter Command: 

```

Gambar 5.28 UI saat melakukan load

```

Waktu sekarang adalah 4
Lokasi: M (8, 2)

```

Gambar 5.29 Data waktu dan lokasi gamestate load

```

Enter Command: 4
Pesanan pada To Do List:
1. G -> N (Normal Item)
2. B -> M (Normal Item)
3. G -> N (Heavy Item)
4. F -> E (Normal Item)

```

Gambar 5.30 Data to do list gamestate load

```
Enter Command: 5
Pesanan yang sedang diantarkan:
1. Heavy Item (Tujuan: B)
```

Gambar 5.31 Data in progress list gamestate load

```
Enter Command: 8
Inventory Kosong
Gadget mana yang ingin digunakan? (ketik 0 untuk kembali)
```

Gambar 5.32 Data inventory gamestate load

```
Enter Command: 9
Uang anda sekarang adalah: 0
```

Gambar 5.33 Data balance gamestate load

Terlihat bahwa data gamestate asli dan load sama. Test ini sesuai dengan spesifikasi.

5.13 Data Test 13: Balance

Fitur yang dites adalah command balance. Hasil yang diharapkan adalah jumlah uang Mobita tampil ke layar. Berikut merupakan antarmuka pengguna saat balance dipanggil.

```
Enter Command: 9
Uang anda sekarang adalah: 400
```

Gambar 5.34 UI pemanggilan balance

Test ini sesuai dengan spesifikasi.

5.14 Data Test 14: Return

Fitur yang dites adalah command return. Hasil yang diharapkan adalah ability return berhasil digunakan jika memang dimiliki. Berikut merupakan antarmuka pengguna saat ability return belum dimiliki.

```
Enter Command: 10
Anda tidak memiliki ability return!
```

Gambar 5.35 UI pemanggilan ability jika ability belum dimiliki

Berikut merupakan antarmuka pengguna jika ability return dimiliki tetapi tas kosong.

```
Enter Command: 10
Tas anda kosong!
```

Gambar 5.36 UI pemanggilan ability jika tas kosong

Berikut merupakan antarmuka pengguna jika ability return berhasil.

Enter Command: 10
Pesanan berhasil dikembalikan!

Gambar 5.37 UI pemanggilan ability jika berhasil

Test ini sesuai dengan spesifikasi.

6 Test Script

| No. | Fitur yang Dites | Tujuan Testing | Langkah-Langkah Testing | Input Data Test | Hasil yang Diharapkan | Hasil yang Keluar |
|-----|------------------|-----------------------------------------------------------------------------|--------------------------------------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------|-------------------------|
| 1 | Program utama | Memastikan permainan dapat dikompilasi dan dijalankan. | Melakukan kompilasi pada file dan menjalankan main.exe. | Data Test 1 | Tidak terjadi error dan program meminta input user. | Sesuai yang diharapkan. |
| 2 | Move | Memastikan pengguna dapat berpindah tempat dan waktu bertambah dengan benar | Membuat kondisi yang memenuhi input data test, menjalankan command | Data Test 2 | Waktu bertambah sesuai jumlah heavy item, waktu bertambah satu tiap dua move, waktu bertambah satu | Sesuai yang diharapkan. |
| 3 | Pick Up | Memastikan pengguna dapat mengambil pesanan | Membuat kondisi yang memenuhi input data test, menjalankan command | Data Test 3 | Pesanan dihapus dari todo list masuk tas dan in progress, pesan peringatan, pesan peringatan, pesan peringatan | Sesuai yang diharapkan. |
| 4 | Drop Off | Memastikan pengguna dapat menurunkan atau | Membuat kondisi yang memenuhi input data test, menjalankan command | Data Test 4 | Pesanan dihapus dari in progress list dan tas serta uang mobita bertambah, | Sesuai yang diharapkan. |

| No. | Fitur yang Dites | Tujuan Testing | Langkah-Langkah Testing | Input Data Test | Hasil yang Diharapkan | Hasil yang Keluar |
|-----|------------------|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| | | mengantarkan pesanan dan mendapatkan imbalan | | | pesan peringatan, pesan peringatan | |
| 5 | Map | Memastikan Map berhasil berjalan sesuai spesifikasi. | Memasukkan command “6” setelah program main sudah dijalankan. | Data Test 5 | Map dapat muncul ke layar dengan warna lokasi yang sesuai dan berubah ketika user melakukan move. | Sesuai yang diharapkan. |
| 6 | To Do | Memastikan To Do berhasil dijalankan sesuai spesifikasi. | Memasukkan command “4” setelah program main sudah dijalankan. | Data Test 6 | Daftar pesanan muncul ke layar dan berubah ketika pemain mengambil tugas. | Sesuai yang diharapkan. |
| 7 | In Progress | Memastikan In Progress berjalan sesuai spesifikasi. | Memasukkan command “5” setelah program main sudah dijalankan. | Data Test 7 | Daftar pesanan muncul ke layar dan berubah ketika pemain mengambil tugas. | Sesuai yang diharapkan. |
| 8 | Buy | Memastikan Buy berjalan sesuai spesifikasi | Memasukkan command “7” lalu memilih <i>gadget</i> yang ingin dibeli dengan memasukkan command sesuai urutan tampil <i>gadget</i> . Langkah ini dilakukan setelah program main sudah dijalankan. | Data Test 8 | Daftar <i>gadget</i> yang dapat dibeli muncul ke layer dan <i>gadget</i> dapat dibeli jika <i>balance</i> mencukupi, begitu sebaliknya | Sesuai yang diharapkan |

| No. | Fitur yang Dites | Tujuan Testing | Langkah-Langkah Testing | Input Data Test | Hasil yang Diharapkan | Hasil yang Keluar |
|-----|------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| 9 | Inventory | Memastikan Inventory berjalan sesuai spesifikasi | Memasukkan command “8” lalu memilih <i>gadget</i> yang ingin digunakan dengan memasukkan command sesuai urutan tampil <i>gadget</i> . Langkah ini dilakukan setelah program main sudah dijalankan. | Data Test 9 | Daftar <i>gadget</i> yang sudah dibeli muncul ke layar dan bisa digunakan (jika ada) | Sesuai yang diharapkan |
| 10 | Help | Memastikan Help dapat tampil sesuai spesifikasi | Memasukkan command “11” setelah menjalankan program main | Data Test 10 | Daftar command yang dapat dijalankan beserta penjelasannya tampil | Sesuai yang diharapkan |
| 11 | Save | Memastikan savefile sukses dibuat | Menjalankan permainan sampai suatu state sembarang dan memanggil command save | Data Test 11 | Seluruh savefile sukses dibuat sesuai format yang telah ditentukan | Seluruh savefile sukses dibuat sesuai format yang telah ditentukan |
| 12 | Load | Memastikan savefile sukses dikonversikan menjadi game state | Memanggil load pada berbagai savefile yang sudah dibuat | Data Test 12 | Seluruh savefile sukses dibaca dan gamestate sama dengan gamestate awal | Seluruh savefile sukses dibaca dan gamestate sama dengan gamestate awal |
| 13 | Balance | Memastikan uang pemain ditampilkan dengan benar | Memanggil balance dari berbagai game state | Data Test 13 | Nilai uang ditampilkan dengan benar | Nilai uang ditampilkan dengan benar |

| No. | Fitur yang Dites | Tujuan Testing | Langkah-Langkah Testing | Input Data Test | Hasil yang Diharapkan | Hasil yang Keluar |
|-----|------------------|---------------------------------------------------------|-------------------------------------------|-----------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| 14 | Return | Memastikan ability return memiliki efek yang diharapkan | Memanggil return dari berbagai game state | Data Test 14 | Item di tas dihapus, pesanan di inprogress list dihapus, dan pesanan dimasukkan di todo list | Item di tas dihapus, pesanan di inprogress list dihapus, dan pesanan dimasukkan di todo list |

7 Pembagian Kerja dalam Kelompok

| No. | Nama | Tugas |
|-----|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Maria Khelli | ADT Mesin Karakter, ADT Adjacency Matrix, ADT Map, ADT Lokasi, ADT List Dinamis, ADT Mesin Konfigurasi, Command Map, Command New Game, Command Help |
| 2. | Mohamad Daffa Argakoesoemah | ADT Stack, ADT To Do, ADT In Progress, ADT Mesin Kata, Command To Do, Command In Progress |
| 3. | Bryan Amirul Husna | ADT Pesanan, ADT Item, ADT Queue, Command Move, Command Pick Up, Command Drop Off, VIP Item (bonus) |
| 4. | Frederik Imanuel Louis | ADT Mobita, ADT Point, ADT List Statis, Command Buy, Command Inventory, Command Save Game, Command Load Game, Command Return to Sender (+Senter, bonus), Main Program |






8 Lampiran

8.1 Deskripsi Tugas Besar 2






Dengan adanya pandemi COVID-19, usaha orang tua Mobita mengalami penurunan pendapatan. Setelah diringkankannya 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.

8.2 Notulensi Rapat (Asistensi)

ASISTENSI 1

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tanggal : 27 Okt 2021 | Catatan Asistensi: Waktu mulai dari nol/satu? Jawab: Nol aja Baca command pake apa? Jawab: Pake mesin kata/ mesin char Kalo adj matrix boleh pake scanf? Jawab: Gaboleh. Graf dijamin terhubung/valid? Jawab: Iya. Adj matrix dijamin simetris. Speed boost ability → 2 move 1 time? maksudnya gimana? Jawab: tiap 2 move → buang 1 time, bukan kebalik 1 time → 2 move. Anggap aja ga ada pecahan (?) Mesin waktu → mengurangi poin waktu? Bukan seperti “undo”. Ingat tujuan game: menyelesaikan pesanan dengan secepatnya. Load/save file boleh diubah konfigurasinya? Jawab: Terserah kalian Makefile harus generate object file? Jawab: Terserah kalian. Ga bakal terlalu dinilai. Kalo heavy item ilang, apakah speed boost balik lagi? Jawab: ilang, kaya temen tubes. Return to sender, bisa return di sembarang tempat apa ke pick up? Jawab: sembarang tempat, kayak teleport. Command bisa pake angka ga? Jawab: command terserah, sesuai kebutuhan aja, kalo mau aman dipake aja mesin katanya |
| Tempat : GMeet | |
| Kehadiran Anggota Kelompok: | |
|  13520115 Maria Khelli  13520146 Bryan Amirul Husna  13520163 Frederik Imanuel Louis  13520118 Mohamad Daffa Argakoesoemah | |
| | Tanda Tangan Asisten:  1358012 Muhammad Hasan |

ASISTENSI 2

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tanggal : 4 November 2021 | Catatan Asistensi: Apa beda data test sama test script? Jawab: data test lebih ke input, kalo test script lebih ke tata caranya. Bab 3 maksud sketsa adt apa? Jawab: penjelasan singkat soal adt, nggak harus spesifik, tapi yang penting tau sebagian besarnya. |
| Tempat : GMeet | |
| Kehadiran Anggota Kelompok:  13520115 Maria Khelli  13520146 Bryan Amirul Husna  13520163 Frederik Imanuel Louis  13520118 Mohamad Daffa Argakoesoemah | |
| | Tanda Tangan Asisten:  13518012 Muhammad Hasan |

8.3 Log Activity Anggota Kelompok

Nama : Maria Khelli – 13520115

23/10/2021 : Membuat repository
24/10/2021 : Membuat ADT Lokasi, ADT Mesin Karakter, ADT Mesin Konfigurasi
26/10/2021 : Membuat ADT List Dinamis, ADT Map, dan ADT Adjacency Matrix
02/11/2021 : Melengkapi ADT List Dinamis dan mengerjakan Command New Game
08/11/2021 : Menyesuaikan Command New Game driver ke ADT Mobita dan main
13/11/2021 : Melengkapi ADT di laporan
14/11/2021 : Melengkapi test script dan notulensi rapat di laporan
15/11/2021 : Melengkapi main program di laporan

Nama : Mohamad Daffa Argakoesoemah – 13520118

24/10/2021 : Membuat ADT Stack dan Mesin Kata
26/10/2021 : Membuat ADT Linked List To Do dan Linked List In Progress, melengkapi primitif ADT Pesanan
27/10/2021 : Memperbaiki bug pada ADT Stack
30/10/2021 : Menambah fungsi pada ADT Stack
06/11/2021 : Memperbaiki dan menambah fungsi pada To Do List, menambah command To Do dan In Progress pada main
15/11/2021 : Membuat ringkasan dan melengkapi ADT di laporan
16/11/2021 : Melakukan testing dan menambah data test di laporan

Nama : Bryan Amirul Husna - 13520146

24/10/2021 : Membuat ADT Pesanan dan QueuePesanan
26/10/2021 : Melengkapi ADT Pesanan
27/10/2021 : Melengkapi ADT Pesanan dan QueuePesanan
06/11/2021 : Membuat fitur Command Move dan PickUp
07/11/2021 : Membuat fitur Command DropOff dan update warna map
11/11/2021 : Membuat fitur pick up VIP
15/11/2021 : Testing dan debugging
16/11/2021 : Melengkapi ADT dan data test di laporan

Nama : Frederik Imanuel Louis – 13520163

25/10/2021 : Membuat ADT Point dan definisi ADT Mobita

28/10/2021 : Membuat ADT List Statis (gadgetlist)

29/10/2021 : Membuat fitur Command Inventory dan Buy

04/11/2021 : Membuat kerangka dasar fitur save dan load

05/11/2021 : Mengubah beberapa definisi berbagai ADT (confilcts)

06/11/2021 : Menambah fungsionalitas gadget

07/11/2021 : Membuat main file

08/11/2021 : Menyelesaikan fitur save dan load

11/11/2021 : Membuat fitur command return (return to sender)

15/11/2021 : Melengkapi laporan