

# **LAPORAN TUGAS BESAR**

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


### **MOBILITA**

Dipersiapkan oleh:

Kelompok 11 / K-03

Nelsen Putra	13520130
Fachry Dennis Heraldî	13520139
Muhammad Gerald Akbar Giffera	13520143
Rizky Ramadhana P. K.	13520151
David Karel Halomoan	13520154

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung  
Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>IF2110-TB-11-03</i>		<i>29</i>
		<i>Revisi</i>	<i>1</i>	<i>18 November 2021</i>

# Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas .....	3
2.1    Spesifikasi Jenis Item: VIP Item .....	3
2.2    Spesifikasi Ability: Return to Sender.....	4
2.3    Spesifikasi Gadget: Senter Pengecil.....	4
2.4    Spesifikasi Save/Load Game.....	4
3 Struktur Data (ADT) .....	4
3.1    ADT Point .....	4
3.2    ADT List Statis.....	5
3.3    ADT List Dinamis .....	5
3.4    ADT Matriks .....	5
3.5    ADT Mesin Karakter.....	6
3.6    ADT Mesin Kata .....	6
3.7    ADT Queue .....	6
3.8    ADT PrioQueue.....	6
3.9    ADT Stack.....	7
3.10    ADT Node .....	7
3.11    ADT Linked List .....	7
3.12    ADT Barang .....	8
3.13    ADT Bangunan.....	8
4 Program Utama .....	8
5 Algoritma-Algoritma Menarik.....	10
5.1    SAVE_GAME.....	10
5.2    getCommand().....	11
6 Data Test .....	11
6.1    Data Test 1 .....	11
6.2    Data Test 2 .....	12
6.3    Data Test 3 .....	12
6.4    Data Test 4 .....	13
6.5    Data Test 5 .....	13
6.6    Data Test 6 .....	14
6.7    Data Test 7 .....	14
6.8    Data Test 8 .....	15
6.9    Data Test 9 .....	15
6.10    Data Test 10.....	16
6.11    Data Test 11.....	17
6.12    Data Test 12.....	18
6.13    Data Test 13.....	18
6.14    Data Test 14.....	19
6.15    Data Test 15.....	19
6.16    Data Test 16.....	20
6.17    Data Test 17.....	20

6.18	Data Test 18.....	20
7	Test Script .....	21
8	Pembagian Kerja dalam Kelompok .....	23
9	Lampiran .....	24
9.1	Deskripsi Tugas Besar 2.....	24
9.2	Notulen Rapat.....	25
9.3	Log Activity Anggota Kelompok.....	26

## 1 Ringkasan

Mobilita merupakan permainan berbasis CLI (command-line interface) tentang pengantaran barang yang mensimulasikan player menjadi seorang kurir dengan melacak pesanan, menavigasi peta, mengambil, dan menurunkan barang. Premis dari permainan ini adalah tokoh utama dari permainan ini, Mobita, menjadi seorang kurir demi mencari nafkah untuk keluarganya karena usaha orang tua Mobita mengalami penurunan akibat pandemi COVID-19. Bermula dari permasalahan tersebut, pemain diminta untuk membuatkan suatu aplikasi yang dapat membantunya dalam menjalankan profesinya sebagai kurir yang ditulis dalam bahasa C dengan menggunakan struktur data yang telah dipelajari di mata kuliah IF2110 Algoritma dan Struktur Data.

Adapun permainan simulasi ini memiliki mekanisme yang diimplementasi dalam 13 bagian utama antara lain, main menu, mekanisme waktu, daftar pesanan, to do list, in progress list, tas, jenis item, ability, gadget, inventory gadget, peta, lokasi & command, dan game flow. Tujuan dari permainan ini adalah menyelesaikan semua pesanan secepat mungkin dengan memanfaatkan fitur-fitur yang disediakan oleh permainan.

Laporan mencakup deskripsi umum persoalan game, penjelasan tambahan spesifikasi tugas, penjelasan struktur data yang digunakan, penjelasan program utama, data-data dan script yang digunakan untuk menguji keberjalanan game, dan pembagian tugas serta notulensi rapat dan lampiran lainnya.

Hasil dari tugas besar ini adalah kelompok kami mampu memahami dan mengimplementasikan pengetahuan mengenai struktur data yang telah dipelajari di mata kuliah IF2110 Algoritma dan Struktur Data. Kami mempelajari juga bagaimana cara menuangkan pemahaman tersebut dengan menuliskan program secara prosedural menggunakan bahasa C. Dengan mengerjakan tugas besar ini juga kami berhasil membuat permainan sederhana dan belajar banyak hal dengan mengeksplorasi hal-hal baru dalam pengerjaan tugas besar ini.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Spesifikasi Jenis Item: VIP Item

VIP Item merupakan jenis item khusus yang bernilai 600 Yen. Pesanan dengan jenis item VIP akan dikerjakan terlebih dahulu dibandingkan dengan item jenis lain. Oleh karena itu, jenis item VIP ini akan memiliki efek atau pengaruh terhadap pesanan masuk ke dalam

STEI- ITB	IF2110-TB-11-03	Halaman 3 dari 29 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.		

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. Apabila Mobita telah selesai atau berhasil mengantarkan item ini, ia akan mendapatkan ability Return to Sender yang spesifikasinya dapat dilihat pada penjelasan di bawah ini.

## **2.2 Spesifikasi Ability: Return to Sender**

Ability Return to Sender merupakan ability yang dapat diaktifkan dengan menjalankan command RETURN. Jika ability ini digunakan, maka item di atas tas akan dikembalikan ke lokasi pick up. Akibatnya, pesanan akan dikembalikan ke To Do List dan diletakkan di urutan terakhir. Selain itu, time limit yang terdapat pada pesanan dengan jenis Perishable Item akan di-reset sebelum dikembalikan. Ability ini tidak dapat diaktifkan untuk pesanan VIP Item.

## **2.3 Spesifikasi Gadget: Senter Pengecil**

Senter pengecil merupakan gadget dengan harga 800 Yen yang dapat digunakan untuk menghilangkan efek dari satu heavy item jika terdapat pada tumpukan teratas pada tas. Ketika senter pengecil akan digunakan, akan dicek terlebih dahulu apakah tumpukan teratas pada tas merupakan heavy item atau tidak. Pada implementasinya, akan ada variabel yang menampung nilai boolean keberadaan senter pengecil. Apabila berhasil diaktifkan (heavy item berada pada tumpukan teratas) variabel keberadaan senter pengecil bernilai true. Selama variabel ini bernilai true maka ketika melakukan perpindahan, penambahan nilai waktu akibat heavy item pada tumpukan teratas akan dibatalkan. Efek dari senter pengecil akan berakhir ketika melakukan drop off/return pertama kali dengan mengembalikan variabel keberadaan senter pengecil bernilai false.

## **2.4 Spesifikasi Save/Load Game**

Untuk melakukan save game, dapat dijalankan command SAVE\_GAME pada terminal. Dalam melakukan save game, akan disimpan state-state yang dibutuhkan ke dalam suatu file yang nantinya dapat dibaca saat melakukan load game. Load game sendiri dapat dilakukan dengan menjalankan command LOAD GAME pada menu awal game di terminal. Saat melakukan load game, permainan akan kembali pada keadaan state yang telah disimpan dalam file yang di-load tersebut. File yang disimpan berupa file konfigurasi awal yang di-append dengan state-state yang disimpan.

# **3 Struktur Data (ADT)**

## **3.1 ADT Point**

- Sketsa struktur data:
  - a. Absis : menyatakan sumbu-x pada koordinat kartesius
  - b. Ordinat : menyatakan sumbu-y pada koordinat kartesius
- Persoalan yang diselesaikan: digunakan untuk mengimplementasikan ADT lain yaitu bangunan dan barang, yang memudahkan pengerjaan fungsi-fungsi di dalam program yang berhubungan dengan pergerakan dan map.

- Alasan pemilihan: memudahkan pengerjaan fungsi lain karena bisa membandingkan antar point secara langsung dan juga membuat source code lebih readable.
- Diimplementasi sebagai ADT Point dengan nama file header "point.h" dan file body "point.c"

### 3.2 ADT List Statis

- Sketsa struktur data:
  - a. Contents[LISTCAPACITY] : memori tempat penyimpanan elemen integer (ElType)
  - b. nEff : menyatakan kolom yang terisi di sebuah list
- Persoalan yang diselesaikan: command buy dan inventory
- Alasan pemilihan: ADT List Statis digunakan sebagai inventory untuk menyimpan gadget yang dibeli dari command buy. Pada spesifikasi kapasitas inventory telah dibatasi menjadi maksimal lima buah gadget. Hal ini menunjukkan bahwa panjang dari list yang akan deklarasi terdefinisi sehingga menggunakan ADT List Statis menjadi inventory adalah pilihan yang tepat.
- Direpresentasikan sebagai ADT List dengan header "list.h" dan file body "list.c"

### 3.3 ADT List Dinamis

- Sketsa struktur data:
  - a. Buffer : memori tempat penyimpanan elemen, dalam program ini ADT list dinamis disesuaikan sehingga menyimpan ADT bangunan.
  - b. nEff : banyaknya elemen efektif
  - c. capacity : kapasitas maksimum list dinamis
- Persoalan yang diselesaikan: Untuk menampung informasi bangunan yang berisi lokasi dan urutan antrian bangunan yang harus dituju sesuai dengan pesanan
- Alasan pemilihan: Karena pesanan yang masuk tidak dapat dipastikan berapa jumlahnya dan dapat bertambah sewaktu-waktu seiring bertambahnya waktu, maka digunakan list dinamis agar alokasi memori dapat disesuaikan dengan penambahan masuknya pesanan.
- Direpresentasikan sebagai ADT ListDin dengan header "listdin.h" dan file body "listdin.c"

### 3.4 ADT Matriks

- Sketsa struktur data:
  - a. Contents[row][col] : Elemen dalam matriks yang bertipe Eltype
  - b. rowEff : jumlah baris matriks yang terisi elemen
  - c. colEff : jumlah kolom matriks yang terisi
- Persoalan yang diselesaikan: pembacaan map dan adjacency matrix
- Alasan pemilihan: karena map dan adjaceny matrix direpresentasikan sebagai sebuah matriks, sehingga akan lebih mudah untuk mengaksesnya untuk fungsi-fungsi lain di dalam program.

- Diimplementasikan sebagai ADT Matrix dengan header "matrix.h" dan file body "matrix.c"

### 3.5 ADT Mesin Karakter

- Sketsa struktur data:
  - a. currentChar : menyatakan karakter yang sedang dibaca
  - b. eot : menyatakan state mesin karakter, jika true maka mesin karakter sudah mencapai akhir dari tape
  - c. tape : file yang dibaca oleh mesin karakter
- Persoalan yang diselesaikan: penerimaan input command dari user
- Alasan pemilihan: diperlukan sebagai dasar dari ADT Mesin Kata
- Direpresentasikan sebagai ADT Mesin Karakter dengan header "charmachine.h" dan file body "charmachine.c"

### 3.6 ADT Mesin Kata

- Sketsa struktur data:
  - a. endWord: menyatakan state akhir dari sebuah kata
  - b. currentWord: berisi kata yang sedang diakses oleh mesin kata
  - c. contents: berisi karakter-karakter yang membentuk kata
  - d. length: berisi nilai panjang kata.
- Persoalan yang diselesaikan: input command dari user.
- Alasan pemilihan: sebagai metode untuk membaca string/input command yang dimasukkan oleh pengguna dalam program.
- Direpresentasikan sebagai ADT Mesin Kata dengan header "wordmachine.h" dan file body "wordmachine.c"

### 3.7 ADT Queue

- Sketsa struktur data:
  - a. Buffer : memori tempat penyimpanan elemen dengan prinsip FIFO (first in first out), dalam program ini queue disesuaikan untuk menyimpan ADT barang untuk memudahkan pengerjaan program.
  - b. idxHead : menyatakan index dari elemen pertama pada queue
  - c. idxTail : menyatakan index dari elemen terakhir pada queue
- Persoalan yang diselesaikan: pick\_up, drop\_off
- Alasan pemilihan: Karena sistem pesanan yang sesuai dengan prinsip FIFO sehingga ADT Queue tepat untuk diimplementasikan untuk membuat list pesanan yang didapat dari hasil config, akan tetapi karena ..... maka akan lebih tepat jika merepresentasikan pesanan kedalam bentuk ADT PrioQueue.
- Direpresentasikan sebagai ADT Queue dengan header "queue.h" dan file body "queue.c"

### 3.8 ADT PrioQueue

- Sketsa struktur data:

STEI- ITB	IF2110-TB-11-03	Halaman 6 dari 29 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.		

- a. Buffer : memori tempat penyimpanan elemen dengan prinsip FIFO (first in first out), dalam program ini queue disesuaikan untuk menyimpan ADT barang untuk memudahkan pengerjaan program.
- b. idxHead : menyatakan index dari elemen pertama pada queue
- c. idxTail: menyatakan index dari elemen terakhir pada queue
- Persoalan yang diselesaikan: pick\_up, drop\_off
- Alasan pemilihan: Karena sistem pesanan yang sesuai dengan prinsip FIFO sehingga ADT PrioQueue tepat untuk diimplementasikan untuk membuat list pesanan yang didapat dari hasil config.
- Direpresentasikan sebagai ADT PrioQueue dengan header "prioqueue.h" dan file body "prioqueue.c"

### 3.9 ADT Stack

- Sketsa struktur data:
  - a. Buffer : memori tempat penyimpanan elemen dengan prinsip LIFO (Last in First out), dalam program ini stack disesuaikan untuk menyimpan ADT barang untuk memudahkan pengerjaan program.
  - b. idxTop: menyatakan index dari elemen paling 'atas' pada stack
  - c. Neff : menyatakan jumlah elemen pada stack
- Persoalan yang diselesaikan: bag
- Alasan pemilihan: karena sesuai spesifikasi tas juga menggunakan sistem LIFO sama seperti stack.
- Direpresentasikan sebagai ADT Stack dengan header "stack.h" dan file body "stack.c"

### 3.10 ADT Node

- Sketsa struktur data:
  - a. Info: memori tempat penyimpanan elemen Barang
  - b. Next: menyatakan alamat memori setelahnya
- Persoalan yang diselesaikan: to\_do, in\_progress
- Alasan pemilihan: karena diperluka untuk ADT ListLinked
- Direpresentasikan sebagai ADT Node dengan header "node.h" dan "node.c"

### 3.11 ADT Linked List

- Sketsa struktur data:
  - a. First: menyatakan elemen pertama list yang berisi Node
- Persoalan yang diselesaikan: in\_progress, to\_do
- Alasan pemilihan: karena pada antrian pesanan, penggunaan queue kurang tepat untuk menghasilkan todo list karena pada queue hanya bisa mengakses HEAD, sehingga digunakan linkedList untuk menjalankan fungsi tersebut
- Direpresentasikan sebagai ADT ListLinked dengan header "list\_linked.h" dan file body "list\_linked.c"

### 3.12 ADT Barang

- Sketsa struktur data:
  - a. Waktu\_pickup : menyatakan waktu pickup barang
  - b. Waktu\_order : menyatakan waktu pemesanan barang
  - c. Lokasi\_pickup : menyatakan bangunan tempat mengambil barang
  - d. Lokasi\_dropoff : menyatakan bangunan tempat mengantarkan barang
  - e. Jenis\_barang : menyatakan jenis barang (heavy/normal/perishable)
  - f. Waktu\_expired : menyatakan waktu expired jika bertipe perishable
- Persoalan yang diselesaikan : pick\_up, drop\_off, map, buy
- Alasan pemilihan: karena banyaknya variabel yang harus diperhitungkan pada setiap barang selama keberjalanan program sehingga akan lebih mudah jika dibuat ADT khusus yang menyimpan semua data tersebut.
- Direpresentasikan sebagai ADT barang dengan header "barang.h"

### 3.13 ADT Bangunan

- Sketsa struktur data:
  - a. Name : menyatakakan nama bangunan (1 karakter)
  - b. Location : menyatakan lokasi bangunan, bertipe Point(x,y)
  - c. Antrean : tempat pesanan pada setiap gedung disimpan, jika lebih dari 1 maka akan diakses dengan sistem FIFO.
- Persoalan yang diselesaikan: map, move, pick\_up, drop\_off
- Alasan pemilihan: karena sebuah bangunan memiliki beberapa variabel maka akan lebih mudah jika dibuat kedalam satu ADT.
- Direpresentasikan sebagai ADT bangunan dengan header "bangunan.h"

## 4 Program Utama

Program utama dari tugas besar ini adalah file "mainDriver.c". File ini merupakan file main driver dari seluruh file header ADT maupun fitur yang diimplementasikan. Mula-mula, file main driver akan membaca file "config.h" yang berisi fungsi untuk melakukan parsing file konfigurasi permainan dan file "initconfig.h" yang berisi preprocessor file header ADT dan fitur yang diimplementasi serta deklarasi variabel-variabel yang akan digunakan secara global pada permainan. Selanjutnya pada file mainDriver akan dilakukan deklarasi ulang variabel-variabel global yang sudah dideklarasikan pada file "initconfig.h". Setelah melakukan deklarasi kamus-kamus umum, program masuk ke algoritma utama yang merupakan looping.

Tampilan pertama yang muncul pada layar adalah Menu Utama. Ada tiga aksi utama yang dapat diakses oleh pemain pada awalnya, yaitu "NEW GAME" untuk memulai permainan, "LOAD GAME" untuk melanjutkan permainan yang telah dijalankan sebelumnya, dan "EXIT" untuk terminasi program. Perintah "NEW GAME" dan "LOAD GAME" keduanya akan meminta file konfigurasi game yang sesuai, file konfigurasi dipastikan valid sehingga program bisa langsung melakukan parsing untuk memuat data-data yang terkandung di dalam file konfigurasi tersebut. Setelah parsing berhasil, program akan masuk kedalam looping state untuk permainan agar dapat dimainkan.



Looping akan berlangsung hingga syarat looping menjadi false yang diwakili oleh sebuah flag, flag tersebut adalah sebuah variabel boolean yang menunjukkan suatu game dinyatakan selesai. Selama program melakukan looping, program akan menampilkan 5 buah informasi dasar yang berguna untuk player selama memainkan permainan ini. Informasi tersebut antara lain "Posisi", "Uang", "Waktu", "Pesanan", dan "Tas". "Posisi" menunjukkan posisi pemain sekarang. "Uang" menunjukkan uang yang dimiliki pemain untuk membeli gadget. "Waktu" menunjukkan waktu yang telah berlalu seiring dilakukannya perpindahan antar lokasi. "Pesanan" menunjukkan banyaknya pesanan yang masuk dan harus dikerjakan seiring bertambahnya waktu. "Tas" menunjukkan banyaknya item yang berada di dalam tas dan harus diantarkan.

Selanjutnya pemain akan diminta untuk memasukkan inputan, ditandai dengan tulisan "ENTER COMMAND: ". Program akan memanggil prosedur yang bersesuaian dengan command yang diinputkan oleh pemain. Pertama, pemain dapat menginput command "MAP". Layar akan menampilkan peta sederhana yang berisi huruf-huruf kapital berwarna. Huruf yang berwarna kuning menunjukkan posisi pemain sekarang. Pemain akan memulai dari Headquarter yang memiliki simbol '8'. Headquarter merupakan lokasi memulai dan mengakhiri permainan serta melakukan transaksi pembelian gadget.

Pemain dapat berpindah dengan memberikan input command "MOVE". Layar akan menampilkan lokasi mana saja yang dapat dituju oleh pemain menggunakan prinsip graf sederhana. Pemain hanya dapat berpindah pada lokasi yang bertetangga saja. Ketika melakukan command "MOVE" waktu akan bertambah 1 kecuali jika membawa Heavy Item maka waktu akan bertambah menjadi 2. Seiring bertambahnya waktu, maka akan ada pesanan yang masuk dalam waktu yang bersesuaian. Pesanan yang masuk dan harus dikerjakan dapat dilihat dengan memasukkan command "TO\_DO". Layar akan menampilkan informasi pesanan berupa lokasi asal, tujuan pesanan, dan jenis barang pesanan.

Ketika pemain berada pada lokasi yang sesuai, misal pada lokasi asal pesanan, maka item pesanan dapat diambil dengan memasukkan command "PICK\_UP". Item tersebut akan masuk ke dalam tas dengan sistem LIFO. Jika pemain membawa Heavy Item maka setiap kali berpindah, waktu akan bertambah sesuai dengan banyak Heavy Item yang dibawa. Jika pemain membawa Perishable Item, item tersebut harus segera diantarkan karena memiliki masa berakhir dan dapat hangus sewaktu-waktu. Adapun untuk jenis VIP Item, item tersebut menjadi prioritas utama. Diimplementasi dengan Priority Queue, ketika pesanan berjenis VIP Item masuk, maka pemain hanya dapat mengambil VIP Item saja. Selanjutnya, setelah pesanan diambil, kita dapat mengecek apa-apa saja pesanan yang sedang dikerjakan dengan memasukkan command "IN\_PROGRESS". Dengan begitu kita dapat menentukan pesanan untuk diantarkan ke lokasi yang sesuai.

Apabila pemain sudah berpindah ke lokasi yang sesuai, command "DROP\_OFF" dapat dijalankan untuk mengantarkan pesanan dengan syarat lokasi tujuan item teratas tas sesuai dengan pesanan. Adapun untuk item-item spesial yang berhasil diantarkan, pemain akan menerima reward berupa ability. Apabila berhasil mengantarkan pesanan Heavy Item, maka pemain akan mendapatkan reward ability speed boost, yaitu kemampuan untuk bergerak lebih cepat. Waktu akan bertambah 1 unit setiap pemain berpindah 2 lokasi dan efeknya berlangsung hingga pemain melewati 10 lokasi atau ketika pemain mengambil pesanan heavy item lain. Apabila pemain berhasil mengantarkan pesanan Perishable Item, pemain akan menerima reward ability increase capacity. Tas akan bertambah kapasitasnya secara permanen sebesar 1. Apabila pemain berhasil mengantarkan VIP Item, pemain akan menerima reward Return to sender yang dapat mengembalikan item di tumpukan teratas tas ke lokasi pick up. Reward ability secara otomatis

aktif kecuali untuk ability Return to Sender yang diaktifkan ketika pemain memasukkan command "RETURN". Pemain juga akan menerima uang untuk setiap item yang diantarkan. Besaran uang yang diterima disesuaikan dengan jenis item.

Untuk membantu pemain, adapun gadget-gadget yang dapat memudahkan pekerjaan di permainan ini. Pemain dapat membeli gadget di Headquarter. Ketika pemain sudah di Headquarter, pemain dapat memasukkan command "BUY". Layar akan menampilkan list gadget yang dapat dibeli beserta harganya. Ada lima gadget yang dapat dibeli dengan harga yang beragam. Setiap gadget yang berhasil dibeli akan masuk ke dalam inventory, maksimal banyaknya gadget yang dapat dibeli dan disimpan di dalam inventory adalah lima. Untuk menggunakan gadget, pemain dapat memasukkan command "INVENTORY". Layar akan menampilkan apa-apa saja gadget yang telah dibeli. Pemain dapat menggunakan gadget asalkan syarat untuk menggunakan gadget terpenuhi. Gadget yang pertama dengan harga 800 yen adalah Kain Pembungkus Waktu. Pemain dapat menggunakannya sekali untuk perishable item teratas pada tas agar kembali ke durasi semula. Gadget yang kedua dengan harga 1200 yen adalah Senter Pembesar. Senter Pembesar digunakan sekali untuk meningkatkan kapasitas tas sebesar dua kali lipat, namun tidak melebihi batas maksimum kapasitas tas. Gadget yang ketiga dengan harga 1500 yen adalah Pintu Kemana Saja. Pintu Kemana Saja digunakan sekali untuk berpindah tanpa menambah unit waktu. Gadget yang keempat dengan harga 3000 yen adalah Mesin Waktu. Mesin Waktu digunakan sekali untuk mengurangi waktu sebanyak 50 unit, jika waktu kurang dari 50 unit, maka waktu menjadi 0 unit tanpa mengubah pesanan masuk. Gadget yang kelima dengan harga 800 yen adalah Senter Pengecil. Senter Pengecil digunakan sekali untuk menghilangkan efek dari satu heavy item jika terdapat pada tumpukan teratas tas.

Ketika berada ditengah permainan, adapun command "HELP" yang dapat menampilkan apa-apa saja command yang dapat dimasukkan oleh pemain beserta keterangan fungsi dari command tersebut. Adapun command "SAVE" dapat dijalankan juga jika pemain ingin menyimpan state permainan yang sedang berjalan. Hasil dari save akan disimpan pada suatu file yang kemudian dapat dilakukan load game di awal saat program ini dijalankan. Pemain juga dapat melakukan "EXIT" untuk melakukan terminasi permainan tanpa melakukan penyimpanan.

Dengan memanfaatkan command-command di atas, pemain diharapkan dapat menyelesaikan permainan. Permainan dianggap selesai jika semua pesanan selesai diantar dan tidak ada lagi pesanan yang perlu dikerjakan. Ketika mencapai tahap ini, pemain kembali ke Headquarter dan akan mendapatkan pesan yang menampilkan permainan sudah selesai, informasi jumlah item yang berhasil diantar, dan informasi lama waktu yang dilampaui.

## 5 Algoritma-Algoritma Menarik

### 5.1 SAVE\_GAME

Algoritma SAVE\_GAME dibuat dengan melakukan modifikasi pada charmachine, charmachine diberi dua state baru yaitu saveMode (menandakan bahwa charmachine akan emnulis ke file, bukan membaca) dan withFile (menandakan bahwa charmachine akan berinteraksi antara dengan file dari luar atau stdin). Kedua state ini dinisiasi pada fungsi start(). Fungsi adv() juga diubah untuk dapat menerima karakter yang akan ditulis ke dalam file. Ditambahkan juga fungsi close() untuk melakukan penutupan charmachine setelah pembacaan file eksternal.

## 5.2 `getCommand()`

Algoritma `getCommand()` dibuat untuk memfasilitasi input pengguna menggunakan `charmachine`, `charmachine` diberi state baru yaitu `withFile` (menandakan bahwa `charmachine` akan berinteraksi antara dengan file dari luar atau `stdin`). State ini dinisiasi pada fungsi `start()`. Digunakan global variable `command` untuk menyimpan `command` sehingga fungsi ini dapat diakses dari mana saja.

## 6 Data Test

```
6 6
2 3
4
A 1 3
B 2 4
C 4 1
D 5 5
0 0 1 1 1
0 0 1 1 1
1 1 0 1 1
1 1 1 0 1
1 1 1 1 0
4
1 A B H
2 B C N
3 D A P
4 D A H
```

Di atas adalah konfigurasi untuk mengetes alur jalannya permainan. Konfigurasi tersebut memiliki ukuran map 6 x 6 dengan empat buah bangunan. Terdapat empat buah pesanan dengan jenis yang berbeda-beda. Untuk seluruh test pada bagian ini akan menggunakan konfigurasi di atas.

### 6.1 Data Test 1

```
rizky@rizky-VirtualBox:~/IF2110_TB_11_03$ . script.sh
rizky@rizky-VirtualBox:~/IF2110_TB_11_03$ ./bin
MENU UTAMA
Tentukan aksi yang mau dilakukan!
- NEW GAME
- LOAD GAME
- EXIT
Masukkan pilihan aksi: █
```

*Proses Compile dan Startup Program*

Untuk meng-*compile* program dapat digunakan script yang telah disediakan dalam file script.sh. Proses kompilasi tersebut menghasilkan file bernama bin.exe yang dapat dijalankan dengan comman ‘./bin’. Dapat dilihat bahwa program lulus *compile* dan bagian menu utama dapat ditampilkan

## 6.2 Data Test 2

```
MENU UTAMA
Tentukan aksi yang mau dilakukan!
- NEW GAME
- LOAD GAME
- EXIT
Masukkan pilihan aksi: NEW GAME
Masukkan nama file: filetidakada.txt

Nama file tidak valid! (Tidak ada di dalam folder penyimpanan).
Masukkan ulang nama file!
Masukkan nama file: datatest.txt

Loading game...
Game Started

Posisi: 8 (2.00,3.00) | Uang: 0
Waktu: 0 | Pesanan: 0 | Tas: 0
ENTER COMMAND:
```

Perintah NEW GAME

Perintah NEW GAME akan memulai permainan baru dan meminta nama file konfigurasi sampai valid. Dapat dilihat bahwa validasi nama file bekerja dengan baik.

## 6.3 Data Test 3

```
ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. B (2.00,4.00)
2. C (4.00,1.00)
3. D (5.00,5.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: d
Input tidak valid !
ENTER COMMAND: =
Input tidak valid !
ENTER COMMAND: 5
Input tidak valid !
ENTER COMMAND: 1
Mobita sekarang berada di titik B (2.00,4.00)!
```

Perintah MOVE

Perintah MOVE mula-mula akan mencetak posisi yang dapat dicapai berdasar matriks adjacency. Dapat dilihat bahwa pembacaan matriks adjacency berhasil karena A tidak dapat diraih dari HQ. Input pilihan lokasi juga akan divalidasi sampai valid seperti terlihat pada gambar.

## 6.4 Data Test 4

```
Posisi: 8 (2.00,3.00) | Uang: 0
Waktu: 0 | Pesanan: 0 | Tas: 0
ENTER COMMAND: TO_DO

Tidak ada pesanan yang harus diantar

Posisi: B (2.00,4.00) | Uang: 0
Waktu: 1 | Pesanan: 1 | Tas: 0
ENTER COMMAND: TO_DO

1. A -> B (Heavy Item)
```

Perintah TO\_DO

Perintah TO\_DO akan menampilkan pesanan masuk yang belum di PICK\_UP. Dapat dilihat bahwa perintah TO\_DO berhasil menampilkan pesanan masuk sesuai file konfigurasi.

## 6.5 Data Test 5

```
Posisi: B (2.00,4.00) | Uang: 0
Waktu: 1 | Pesanan: 1 | Tas: 0
ENTER COMMAND: PICK_UP

Pesanan tidak ditemukan!

Posisi: A (1.00,3.00) | Uang: 0
Waktu: 2 | Pesanan: 2 | Tas: 0
ENTER COMMAND: PICK_UP

Pesanan berupa Heavy Item berhasil diambil!
Tujuan Pesanan: B
```

Perintah PICK\_UP

Perintah PICK\_UP akan mengambil pesanan yang masuk bila lokasi sesuai. Bila tidak ada pesanan di lokasi tersebut, program akan menampilkan pesan peringatan seperti pada gambar di atas. Bila berhasil, program akan memasukkan pesanan ke tas dan menghapus pesanan dari TO\_DO sesuai pada gambar di bawah ini

```
Posisi: A (1.00,3.00) | Uang: 0
Waktu: 2 | Pesanan: 1 | Tas: 1
ENTER COMMAND: TO_DO

1. B -> C (Normal Item)
```

Pesanan yang Di-pickup Dihapus dari TO\_DO

## 6.6 Data Test 6

```
Posisi: A (1.00,3.00) | Uang: 0
Waktu: 2 | Pesanan: 1 | Tas: 1
ENTER COMMAND: MOVE

Posisi yang dapat dicapai:
1. B (2.00,4.00)
2. C (4.00,1.00)
3. D (5.00,5.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: 1
Mobita sekarang berada di titik B (2.00,4.00)!

Posisi: B (2.00,4.00) | Uang: 0
Waktu: 4 | Pesanan: 3 | Tas: 1
ENTER COMMAND: █
```

Mobita Bergerak Lebih Lambat Apabila Membawa Barang *Heavy*

Bagian ini bermaksud mengetes pertambahan waktu yang lebih banyak apabila Mobita membawa barang berkategori *Heavy*. Hal ini dapat bekerja dengan baik sesuai gambar di atas.

## 6.7 Data Test 7

```
Posisi: A (1.00,3.00) | Uang: 0
Waktu: 6 | Pesanan: 3 | Tas: 1
ENTER COMMAND: DROP_OFF

Tidak ada pesanan yang dapat diantarkan!

Posisi: B (2.00,4.00) | Uang: 0
Waktu: 8 | Pesanan: 3 | Tas: 1
ENTER COMMAND: DROP_OFF

Pesanan Heavy Item berhasil diantarkan
Reward ability Speed Boost berhasil didapatkan dan diaktifkan
Waktu hanya bertambah 1 unit ketika kamu berpindah 2 lokasi
Uang yang didapatkan: 400 Yen
```

Perintah DROP\_OFF

Perintah DROP\_OFF hanya akan bekerja jika ada pesanan dalam tas dan lokasi saat ini sesuai dengan lokasi tujuan dari pesanan paling atas di tas. Selain itu, program akan memberikan pesan peringatan seperti pada gambar

## 6.8 Data Test 8

```
Posisi: B (2.00,4.00) | Uang: 400
Waktu: 8 | Pesanan: 3 | Tas: 0
ENTER COMMAND: MOVE

Posisi yang dapat dicapai:
1. 8 (2.00,3.00)
2. A (1.00,3.00)
3. C (4.00,1.00)
4. D (5.00,5.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: 3
Mobita sekarang berada di titik C (4.00,1.00)!

Posisi: C (4.00,1.00) | Uang: 400
Waktu: 8 | Pesanan: 3 | Tas: 0
ENTER COMMAND: MOVE

Posisi yang dapat dicapai:
1. 8 (2.00,3.00)
2. A (1.00,3.00)
3. B (2.00,4.00)
4. D (5.00,5.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: 4
Mobita sekarang berada di titik D (5.00,5.00)!

Posisi: D (5.00,5.00) | Uang: 400
Waktu: 9 | Pesanan: 3 | Tas: 0
ENTER COMMAND: 
```

Ability *Speed Boost* aktif sesaat setelah berhasil mengantarkan barang *Heavy*. Setelah berhasil mengantarkan pesanan barang *Heavy*, Mobita bisa bergerak lebih cepat. Waktu hanya bertambah satu unit setelah bergerak dua kali seperti pada gambar.

## 6.9 Data Test 9

```
Posisi: D (5.00,5.00) | Uang: 400
Waktu: 9 | Pesanan: 3 | Tas: 0
ENTER COMMAND: BUY

Command BUY hanya dapat dipanggil di Headquarters!
ENTER COMMAND: BUY
Uang Anda sekarang: 0 Yen
```

```

ENTER COMMAND: BUY
Uang Anda sekarang: 400 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: 

```

Perintah BUY

Perintah BUY akan menampilkan saldo sementara dan gadget apa saja yang dapat dibeli. Gambar tengah adalah hasil eksekusi sebelum menyelesaikan pesanan barang *Heavy* (Data test 7), sedangkan gambar bawah adalah hasil eksekusi setelah menyelesaikan pesanan tersebut. Bila uang tidak mencukupi, akan diberi pesan peringatan seperti pada gambar.

### 6.10 Data Test 10

```

Posisi: C (4.00,1.00) | Uang: 400
Waktu: 8 | Pesanan: 2 | Tas: 1
ENTER COMMAND: MAP

* * * * *
*      A      *
*      8 B    *
*             *
* C           *
*             *
*             D *
*             *
* * * * *

```

Perintah MAP

Perintah MAP akan menunjukkan seluruh lokasi yang ada dan warna yang bersesuaian. Seperti yang ditunjukkan pada gambar, Mobita sedang berada di C dan membawa pesanan paling atas yang memiliki tujuan A. Di D dan B terdapat pesanan yang belum diambil. Sedangkan HQ adalah lokasi yang dapat dicapai dari C. Sebetulnya lokasi A, B, dan D juga dapat diraih dari C namun pewarnaan lokasi DROP\_OFF dan PICK\_UP memiliki prioritas yang lebih tinggi.



## 6.11 Data Test 11

```
Posisi: C (4.00,1.00) | Uang: 800
Waktu: 10 | Pesanan: 1 | Tas: 1
ENTER COMMAND: MOVE

Posisi yang dapat dicapai:
1. B (2.00,3.00)
2. A (1.00,3.00)
3. B (2.00,4.00)
4. D (5.00,5.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: 4
Mobita sekarang berada di titik D (5.00,5.00)!

Posisi: D (5.00,5.00) | Uang: 800
Waktu: 10 | Pesanan: 1 | Tas: 1
ENTER COMMAND: PICK_UP

Pesanan berupa Heavy Item berhasil diambil!
Tujuan Pesanan: A

Posisi: D (5.00,5.00) | Uang: 800
Waktu: 10 | Pesanan: 0 | Tas: 2
ENTER COMMAND: MOVE

Posisi yang dapat dicapai:
1. B (2.00,3.00)
2. A (1.00,3.00)
3. B (2.00,4.00)
4. C (4.00,1.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: 2
Mobita sekarang berada di titik A (1.00,3.00)!

Posisi: A (1.00,3.00) | Uang: 800
Waktu: 12 | Pesanan: 0 | Tas: 2
ENTER COMMAND: █
```

*Ability Speed Boost Akan Segera Hilang Apabila Barang Heavy Di-pickup*

Seperti yang dijelaskan pada spesifikasi, *ability speed boost* akan segera hilang setelah pemain melakukan PICK\_UP ke barang *heavy*. Perilaku tersebut tecerminkan pada gambar di atas.

## 6.12 Data Test 12

```
Posisi: C (4.00,1.00) | Uang: 1400
Waktu: 12 | Pesanan: 0 | Tas: 0
ENTER COMMAND: MOVE

Posisi yang dapat dicapai:
1. 8 (2.00,3.00)
2. A (1.00,3.00)
3. B (2.00,4.00)
4. D (5.00,5.00)
Posisi yang dipilih? (ketik 0 jika ingin kembali)
ENTER COMMAND: 1
Mobita sekarang berada di titik 8 (2.00,3.00)!

Selamat, Anda berhasil mengantarkan semua barang!
Total Barang yang Anda antar      : 4
Waktu yang Anda butuhkan          : 13
```

Permainan Akan Selesai Bila Semua Pesanan Telah Diantarkan atau Habis Waktunya dan Pemain Kembali ke HQ

Permainan akan terus berjalan bila masih ada pesanan yang belum masuk, masih berada di TO\_DO, ataupun masih berada di tas. Permainan akan selesai bila hal-hal tersebut sudah tidak terpenuhi. Bila pemain kembali ke HQ, maka statistik permainan terakhir akan ditampilkan seperti pada gambar di atas.

## 6.13 Data Test 13

```
Posisi: 8 (2.00,3.00) | Uang: 0
Waktu: 0 | Pesanan: 0 | Tas: 0
ENTER COMMAND: EXIT
```

Perintah EXIT

Perintah EXIT akan langsung menghentikan permainan seperti gambar di atas.

### 6.14 Data Test 14

```
Posisi: 8 (2.00,3.00) | Uang: 400
Waktu: 6 | Pesanan: 3 | Tas: 0
ENTER COMMAND: INVENTORY

1. Mesin Waktu
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)
ENTER COMMAND: 1
Mesin Waktu berhasil digunakan!

Posisi: 8 (2.00,3.00) | Uang: 400
Waktu: 0 | Pesanan: 3 | Tas: 0
ENTER COMMAND: 
```

Perintah INVENTORY dan *Gadget* Mesin Waktu

Seperti dapat dilihat pada gambar di atas, perintah INVENTORY akan menunjukkan semua *gadget* yang dimiliki. Pada tes ini, dicoba *gadget* mesin waktu. Mesin waktu akan mengurangi waktu sebanyak lima puluh unit. Pada tes ini, waktu akan menjadi nol (tidak mungkin negatif).

### 6.15 Data Test 15

```
Posisi: A (1.00,3.00) | Uang: 0
Waktu: 2 | Pesanan: 1 | Tas: 1
ENTER COMMAND: SAVE_GAME

Masukkan nama file: #

Nama file tidak valid! Nama file harus valid dengan ekstensi '.txt'.
Masukkan nama file: test.pdf

Nama file tidak valid! Nama file harus valid dengan ekstensi '.txt'.
Masukkan nama file: coba.txt

File tersimpan!
```

Perintah SAVE\_GAME

Setelah memasukkan perintah SAVE\_GAME, pemain akan diminta memasukkan nama file sampai valid. Bila belum valid, program akan memberi pesan peringatan dan tidak menyimpan *progress* permainan.

## 6.16 Data Test 16

```
MENU UTAMA
Tentukan aksi yang mau dilakukan!
- NEW GAME
- LOAD GAME
- EXIT
Masukkan pilihan aksi: LOAD GAME
Masukkan nama file: coba.txt

Loading game...
Game Started

Posisi: A (1.00,3.00) | Uang: 0
Waktu: 2 | Pesanan: 1 | Tas: 1
```

Perintah LOAD GAME

Pada perintah LOAD GAME, proses permainan dilanjutkan sama persis seperti saat permainan di SAVE seperti yang terlihat pada gambar.

## 6.17 Data Test 17

```
Posisi: 8 (2.00,3.00) | Uang: 3000
Waktu: 0 | Pesanan: 0 | Tas: 0
ENTER COMMAND: HELP

1. MOVE
Command ini digunakan untuk berpindah ke lokasi selanjutnya dari lokasi pemain berada. Pemain hanya bisa pindah ke lokasi yang berdekatan berdasarkan matriks adjacency. Setelah berpindah lokasi, waktu akan bertambah 1 unit (dalam kondisi normal).

2. PICK_UP
Command ini digunakan untuk mengambil item jika ada pesanan pada lokasi pemain berada. Pemain tidak bisa mengambil item jika tidak ada pesanan pada lokasi pemain berada. Jika ada beberapa pesanan di lokasi pemain berada, maka item yang diambil adalah pesanan yang masuk duluan.
```

Perintah HELP (tidak ditampilkan seluruhnya)

Perintah HELP akan menunjukkan bantuan perintah apa saja yang ada seperti pada gambar di atas. Setelah perintah HELP dijalankan, permainan akan berlanjut.

## 6.18 Data Test 18

```
Posisi: D (5.00,5.00) | Uang: 3000
Waktu: 6 | Pesanan: 1 | Tas: 3
ENTER COMMAND: IN_PROGRESS

1. Perishable Item (Tujuan: A)
2. Normal Item (Tujuan: C)
3. Heavy Item (Tujuan: B)
```

Perintah IN\_PROGRESS

Sesuai gambar di atas, perintah IN\_PROGRESS akan menunjukkan isi tas diurutkan dari yang paling terakhir dimasukkan.

## 7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur Menjalankan Program Permainan	Memeriksa apakah program dapat dicompile dan apakah menu utama berhasil ditampilkan.	Meng-compile source code dengan cara menjalankan script.sh di Linux/run.bat di Windows	Data test 1	Program berhasil dicompile tanpa masalah, dan juga menu utama berhasil ditampilkan.	Program berhasil dicompile, dan menu utama berhasil ditampilkan.
2	Fitur New Game	Memeriksa apakah proses konfigurasi dari file .txt berhasil.	Menjalankan program, lalu memilih NEW GAME, dan input nama file konfigurasi yang valid.	Data test 2	File konfigurasi berhasil dibaca, dan program lanjut ke dalam game dengan waktu = 0.	File konfigurasi berhasil dibaca dan muncul waktu = 0 serta prompt input command.
3	Fitur Load Game	Memeriksa apakah keadaan permainan dikembalikan sama seperti sebelum SAVE	Menjalankan program, lalu menginput LOAD_GAME	Data test 16	Posisi, jumlah pesanan, dan waktu akan kembali seperti sebelum SAVE	Sesuai yang diharapkan
4	Fitur Mekanisme Waktu	Memeriksa apakah skema pertambahan waktu sesuai.	Menginput command MOVE di berbagai kondisi	Data test 3, 6, 8, dan 11	-Bertambah 1 saat tidak membawa barang <i>Heavy</i> dan <i>speed boost</i> tidak aktif - Bertambah lebih banyak ketika membawa barang <i>Heavy</i> - Bertambah lebih sedikit ketika <i>speed boost</i> aktif	Sesuai yang diharapkan
5	Fitur Move	Memeriksa apakah pemain dapat berpindah tempat sesuai spesifikasi, dan ability yang diperoleh.	Menginput command MOVE, pilih lokasi yang diinginkan	Data test 3, 6, dan 8	Muncul list lokasi yang bisa dicapai, pesan berhasil, dan juga perubahan waktu sesuai ability.	Lokasi yang dapat dicapai muncul, lokasi pemain berpindah, dan perubahan waktu juga sesuai ability yang ada.
6	Fitur Pick Up	Memeriksa jika barang berhasil di	Menginput command	Data test 5	Muncul pesan "pesanan berupa	Pesan yang diharapkan

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		pickup, dan apakah masih ada dalam todo list.	PICK_UP, lalu jika ada pesanan dan berhasil diambil maka diperiksa kembali todo list, dengan command TO_DO		... item berhasil diambil”, dan juga sudah dihapus dari to_do list.	muncul, dan juga sudah dihapus dari to_do list.
7	Fitur Drop Off	Memeriksa jika ada barang yang bisa di dropoff, atau sebuah barang berhasil di dropoff oleh pemain, dan menambahkan uang pemain.	Menginput command DROP_OFF.	Data test 7	Muncul pesan “Pesanan .. Item berhasil diantarkan”, penambahan uang pemain, dan juga barang di pop dari stack tas.	Pesan yang diharapkan muncul, uang pemain bertambah, dan barang berhasil di pop dari tas.
8	Fitur Map	Memeriksa apakah map berhasil ditampilkan, warna dan posisi sesuai hasil konfigurasi dan keadaan game pada saat tertentu.	Input command MAP.	Data test 10	Map ditampilkan dengan ukuran, posisi bangunan, dan warna yang sesuai	Map berhasil ditampilkan dengan ukuran, posisi bangunan dan warna yang sesuai.
9	Fitur To Do	Memeriksa apakah todo list berhasil ditampilkan sesuai dengan pesanan pada waktu tertentu.	Input command TO_DO	Data test 4	Todo list berhasil ditampilkan dengan deskripsi jenis item dan lokasi pickup & dropoff	Todo list berhasil ditampilkan sesuai spesifikasi.
10	Fitur In Progress	Memeriksa apakah list pesanan yang sedang dikerjakan berhasil ditampilkan secara berurutan sesuai dengan pesanan yang terakhir masuk	Input command IN_PROGRESS	Data test 17	In Progress list ditampilkan secara berurutan dengan deskripsi tujuan/lokasi dropoff	In Progress list berhasil ditampilkan secara berurutan dengan deskripsi tujuan/lokasi dropoff
11	Fitur Buy	Memeriksa jika list Gadget berhasil ditampilkan, dan apakah bisa membeli gadget.	Input command BUY, pilih gadget yang ingin dibeli.	Data test 9	List gadget berhasil ditampilkan, dan berhasil membeli jika uang pemain cukup.	Sesuai yang diharapkan
12	Fitur Inventory	Memeriksa apakah list isi inventory berhasil ditampilkan serta pemain dapat melakukan input	Input command INVENTORY	Data test 14	List isi inventory sebanyak 5 slot ditampilkan, kemudian pemain melakukan input untuk	List isi inventory berhasil ditampilkan, pemain dapat melakukan

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		untuk menggunakan gadget yang diinginkan			menggunakan gadget yang diinginkan	input gadget yang ingin digunakan
13	Fitur Help	Memeriksa apakah daftar seluruh command yang dapat dijalankan pada program dapat ditampilkan beserta dengan deskripsi fungsinya	Input command HELP	Data test 18	Daftar command beserta deskripsi fungsinya ditampilkan	Daftar command beserta deskripsi fungsinya berhasil ditampilkan
14	Fitur Save Game	Memeriksa apakah permainan disimpan ke file yang benar	Input command SAVE_GAME	Data test 15	File <i>state</i> permainan akan disimpan dalam folder <b>SAVED_GAMES</b>	Sesuai yang diharapkan
15	Fitur Exit	Memeriksa apakah permainan berhenti sesaat setelah menerima perintah	Input command EXIT baik sebelum atau setelah memulai permainan	Data test 13	Permainan akan berhenti	Sesuai yang diharapkan

## 8 Pembagian Kerja dalam Kelompok

No.	Nama	NIM	Tugas
1.	Nelsen Putra	13520130	IN_PROGRESS, INVENTORY
2.	Fachry Dennis Heraldi	13520139	DROP_OFF, BUY
3.	Muhammad Gerald Akbar Giffera	13520143	TO_DO, MAP
4.	Rizky Ramadhana P. K.	13520151	MOVE, PICK_UP
5.	David Karel Halomoan	13520154	ADT, pembacaan config dan command, SAVE, LOAD, RETURN

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar 2




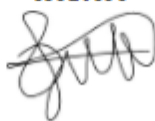


Dengan adanya pandemi COVID-19, usaha orang tua Mobita mengalami penurunan pendapatan. Setelah diringankannya PPKM, Mobita ingin membantu orang tuanya mendapat penghasilan menjadi seorang kurir. Tugas kalian adalah membantu Mobita membuatkan aplikasi yang akan membantunya melacak pesanan, navigasi peta,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

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.


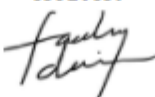






## 9.2 Notulen Rapat

### Asistensi I

Tanggal : 27 Oktober 2021	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- Karakter dianggap string saja karena belum tentu berurutan</li> <li>- Peta punya fungsi untuk menampilkan posisi saja, tetapi untuk perpindahannya bisa disesuaikan</li> <li>- Setiap karakter harusnya tidak ada yang sama</li> <li>- Tidak perlu meng-<i>handle error</i> untuk <i>file config</i> karena <i>file config</i> dapat dipastikan valid</li> <li>- Urutan baris di <i>adjacency matrix</i> sesuai dengan urutan lokasi di file konfigurasi</li> <li>- Kalau mau ngerjain yang bonus <i>Load Game</i>, semua <i>state</i> ditulis di file supaya bisa di-load. Filenya nanti berisi tulis ulang <i>config</i> + tulis <i>state-state</i></li> <li>- Bedanya ADT Queue dan <i>linked list</i> adalah <i>Queue</i> hanya bisa diakses elemen paling depan, sedangkan <i>linked list</i> bisa diakses semua elemennya</li> </ul>
Tempat : Zoom	
<b>Kehadiran Anggota Kelompok:</b> No NIM Tanda tangan  1 13520130  2 13520139  3 13520143  4 13520151  5 13520154 	
	<b>Tanda Tangan Asisten:</b> 

### Asistensi II

Tanggal : 8 November 2021	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- Penambahan poin untuk setiap <i>move</i> adalah 1 poin</li> <li>- Kalau punya <i>extension</i> buat nge-run program, sebisa mungkin untuk tidak <i>include file .c</i> nya, tetap gunakan <i>file .h</i> untuk deliverables nanti</li> <li>- <i>Recommended</i> untuk membuat ADT baru dalam mengimplementasikan <i>ability</i> yang sedang aktif (dengan catatan: apabila waktunya memungkinkan)</li> <li>- Progress sejauh ini baru sekitar 50%</li> <li>- Pertanyaan lebih lanjut dapat disampaikan ke asisten via <i>Line</i></li> </ul>
Tempat : Zoom	
<b>Kehadiran Anggota Kelompok:</b> No NIM Tanda tangan  1 13520130  2 13520139  3 13520143  4 13520151  5 13520154 	
	<b>Tanda Tangan Asisten:</b> 

### 9.3 Log Activity Anggota Kelompok

No.	Nama	NIM	Log Aktivitas
STEI- ITB		IF2110-TB-11-03	Halaman 26 dari 29 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.	Nelsen Putra	13520130	<p>27 Oktober 2021: Menghadiri Asistensi 1 Tugas Besar IF2110</p> <p>29 Oktober 2021: Membuat command IN_PROGRESS</p> <p>8 November 2021: Menghadiri Asistensi 2 Tugas Besar IF2110, Membuat command INVENTORY</p> <p>15 November 2021: Finishing command IN_PROGRESS dan INVENTORY</p> <p>16 November 2021: Memindahkan ADT ke masing-masing folder, membuat driver ADT, menambahkan template README</p> <p>17 November 2021: Menulis laporan</p> <p>18 November 2021: Melengkapi README, merapikan laporan</p>
2.	Fachry Dennis Herald	13520139	<p>23 Oktober 2021: Membuat repositori github</p> <p>27 Oktober 2021: Menghadiri Asistensi 1 Tugas Besar IF2110, melakukan beberapa perubahan pada ADT list, listdin, queue, dan stack, membuat command buy</p> <p>5 November 2021: Membuat command drop_off, melakukan pembaruan dan penyesuaian pada ADT Stack</p> <p>8 November 2021: Menghadiri Asistensi 2 Tugas Besar IF2110</p> <p>9 November 2021: Memperbarui command DROP_OFF dan variabel global pada initconfig.c</p> <p>15 November 2021: Membuat command run.bat untuk memudahkan proses kompilasi program pada sistem operasi Windows</p> <p>15 November 2021: Menginisiasi laporan, membuat ringkasan, merapikan bagian yang diperlukan</p> <p>16 November 2021: Memperbaiki bug pada command buy</p> <p>17 November 2021: Perbaikan pada direktori include, perombakan pada ADT List, perbaikan pada command Inventory, Mengimplementasi fungsi pada tiap gadget</p> <p>18 November 2021: Membuat implementasi bonus gadget senter pengecil, perbaikan pada ADT, merapikan file main, cleansing file, menulis readme</p>
3.	Muhammad Gerald Akbar Gifferra	13520143	<p>27 Oktober 2021: Menghadiri Asistensi 1 Tugas Besar IF2110, menambahkan pcolor pada folder ADT</p> <p>28 Oktober 2021: Membuat ADT Barang</p>

			<p>29 Oktober 2021: Membuat kerangka dari program MAP</p> <p>30 Oktober 2021: Mengupdate fungsi MAP dan TO_DO</p> <p>8 November 2021: Menghadiri Asistensi 2 Tugas Besar IF2110</p> <p>9 November 2021: Membuat perubahan pada ADT linkedList untuk menyesuaikan program, dan mengupdate fungsi TO_DO sehingga mengakses linkedList.</p> <p>17 November 2021: Menulis laporan</p>
4.	Rizky Ramadhana P. K.	13520151	<p>27 Oktober 2021: Menghadiri Asistensi 1 Tugas Besar IF2110</p> <p>28 Oktober 2021: Membuat ADT Bangunan dan command MOVE</p> <p>29 Oktober 2021: Menambahkan command PICK_UP</p> <p>8 November 2021: Menghadiri Asistensi 2 Tugas Besar IF2110, menyempurnakan prosedur penambahan waktu sesuai kriteria, ability speed boost</p> <p>10 November 2021: Perbaikan mekanisme include untuk proses compile</p> <p>15 November 2021: Menambahkan pesanan ke bangunan setiap pertambahan waktu</p> <p>16 November 2021: Menghapus pesanan dari TO_DO apabila sudah dipickup</p> <p>17 November 2021: Membuat data test untuk laporan</p> <p>18 November 2021: Mengganti gambar pada data test dan membuat driver untuk list_linked</p>
5.	David Karel Halomoan	13520154	<p>23 Oktober 2021: Mencoba membuat program config untuk membaca file konfigurasi dan mulai membuat struktur program</p> <p>24 Oktober 2021: Menyelesaikan struktur program, menghadiri meet kelompok pertama, dan push ADT dan struktur program ke github</p> <p>27 Oktober 2021: Menghadiri Asistensi 1 Tugas Besar IF2110 dan menyelesaikan COMMAND_PARSER</p> <p>8 November 2021: Menghadiri Asistensi 2 Tugas Besar IF2110</p> <p>15 November 2021: Memperbaiki bug pada map</p> <p>16 November 2021: Menambahkan fitur input nama file, command HELP, dan fungsi print string berwarna</p> <p>17 November 2021: Memulai pengerjaan command SAVE_GAME</p>

			18 November 2021: Menambahkan implementasi barang VIP, ability RETURN, command SAVE_GAME, dan LOAD GAME dan menulis laporan
--	--	--	---