

LAPORAN TUGAS BESAR

IF2110 Algoritma dan Struktur Data

Mobilita

Dipersiapkan oleh:

Kelompok 11

Bariza Haqi 13520018

Maharani Ayu Putri Irawan 13520019


Damianus Clairvoyance Diva P 13520035

Adiyansa Prasetya Wicaksana 13520044

Farrel Farandieka Fibriyanto 13520054

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-11-01</i>		<i>50</i>
		<i>Revisi</i>	<i>1</i>	<i>18/11/2021</i>

Daftar Isi

1 Ringkasan.....	4
2 Penjelasan Tambahan Spesifikasi Tugas	5
2.1 Save Game.....	5
2.2 Load Game	5
2.3 VIP Item dan Ability Return To Sender.....	5
2.4 Gadget Senter Pengecil	5
3 Struktur Data (ADT).....	6
3.1 ADT Point	6
3.2 ADT List Statis.....	6
3.3 ADT List Dinamis.....	6
3.4 ADT Matrix.....	6
3.5 ADT Mesin Karakter.....	7
3.6 ADT Mesin Kata	7
3.7 ADT PColor	7
3.8 ADT PrioQueue.....	8
3.9 ADT Stack.....	8
3.10 ADT Linked List	8
3.11 ADT Map.....	8
4 Program Utama	10
5 Data Test	12
5.1 Data Test 1	12
5.2 Data Test 2	12
5.3 Data Test 3	13
5.4 Data Test 4	13
5.5 Data Test 5	14
5.6 Data Test 6	14
5.7 Data Test 7	15
5.8 Data Test 8	15
5.9 Data Test 9	16
5.10 Data Test 10.....	16
5.11 Data test 11	17
5.12 Data Test 12.....	18
5.13 Data Test 13.....	18
5.14 Data Test 14.....	19
5.15 Data Test 15.....	20
5.16 Data Test 16.....	20
5.17 Data Test 17.....	21
5.18 Data Test 18.....	21
5.19 Data Test 19.....	22
5.20 Data Test 20.....	23
5.21 Data Test 21.....	24
5.22 Data Test 22.....	25

5.23	Data Test 23.....	26
5.24	Data Test 24.....	26
5.25	Data Test 25.....	28
5.26	Data test 26.....	29
5.27	Data Test 27.....	30
5.28	Data Test 28.....	30
6	Test Script.....	32
7	Pembagian Kerja dalam Kelompok	39
8	Lampiran.....	40
8.1	Deskripsi Tugas Besar 2.....	40
8.2	Notulen Rapat.....	40
8.3	Log Activity Anggota Kelompok.....	48

1 Ringkasan

Mobilita merupakan permainan berbasis *Command Line Interface* yang objektifnya untuk mengantarkan pesanan yang di-load dari sebuah file jenis txt dalam waktu sesingkat mungkin. Awal mula game pemain akan ditempatkan pada sebuah *headquarter* dan akan mendapatkan daftar pesanan yang harus diantarkan. Pesanan merupakan item yang harus di-pick up pada suatu bangunan dan harus di-drop off pada bangunan lain sesuai dengan daftar pesanan. Item akan masuk ke daftar pesanan sesuai dengan waktu yang telah didefinisikan pada game dan item juga memiliki beberapa tipe yaitu, *normal*, *heavy*, *perishable*, dan *VIP*. Setiap jenis item memiliki efeknya masing-masing, *normal* tidak memiliki efek spesial apapun, *heavy item* akan membuat waktu yang dilalui pemain bertambah sebanyak 1 unit (memperlambat pemain), *perishable item* memiliki batas waktu untuk diantarkan (jika batas waktu habis maka item akan hangus secara otomatis), dan *VIP item* yang harus segera dilayani karena pemain tidak akan bisa pick up pesanan lain ketika mendapatkan item bertipe *VIP*. Pemain juga akan diberikan *ability* khusus jika mengantarkan item dengan efek tertentu, *ability* yang dimaksud adalah *Speed Boost*, *Increase Capacity*, dan *Return to sender*. Selain *ability* yang didapatkan dari mengantarkan item ber-efek khusus, pemain juga bisa membeli *gadget* yang akan memberikan efek tertentu, *gadget* yang tersedia pada Mobilita adalah Kain Pembungkus Waktu, Senter Pembesar, Pintu Kemana Saja, Mesin Waktu, dan Senter Pengecil.

Berdasarkan deskripsi game tersebut, ada 6 modul yang dibuat oleh kelompok kami, yaitu Gadget yang berisi jenis *gadget* dan *ability* yang ditimbulkan dari *gadget* tersebut; Items berisi jenis *item*, modifikasi dari list untuk menyimpan item, dan tas yang digunakan untuk menyimpan item; List beserta variasinya; Load yang berisi modifikasi dari mesin kata untuk membaca file juga *comand* dan menyimpan *state* dari game; Map yang menampilkan bangunan dan legenda dari bangunan tersebut, serta mekanik dari gerakan pemain; dan Main berisi struktur dari Mobilita.

Laporan mencakup deskripsi umum persoalan game, penjelasan tambahan spesifikasi tugas, enjelasan 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.

Program dibuat menggunakan bahasa C dengan memanfaatkan ADT yang sudah dipelajari pada mata kuliah IF2110 – Algoritma dan Struktur Data. Game ini dimulai dengan menampilkan konfigurasi mana yang diinginkan oleh pemain, lalu input file .txt yang diinginkan. Setelah dilakukan pembacaan konfigurasi oleh program, pemain memulai gamenya di *headquarters* lalu bisa melakukan berbagai command, salah satunya, *MOVE*, *TO_DO*, *IN_PROGRESS*, dan *MAP*. Permainan akan terus berlangsung selama masih ada pesanan yang belum diantarkan, atau sampai pemain melakukan save pada game.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Save Game

Command Save Game berfungsi untuk menyimpan *state* terakhir dari permainan yang sedang dijalankan. Hasil dari *Save Game* ini dapat di-*Load* pada awal game untuk melanjutkan dari *state* terakhir game yang di-save. Game dapat disimpan pada file .txt yang namanya sesuai dengan yang diinginkan oleh pemain. File akan di-*rewrite* jika nama file sudah ada pada direktori, dan file akan dibuat secara otomatis jika nama file yang dimasukkan belum ada.

2.2 Load Game

Command Load Game berfungsi untuk melanjutkan kembali permainan yang *state*-nya telah disimpan pada sebuah file teks yang dipilih oleh pemain. Setelah dilakukan *load game*, pemain dapat melanjutkan permainan dari *state* yang telah disimpan dan dibaca.

2.3 VIP Item dan Ability Return To Sender

Item dengan jenis VIP harus menjadi prioritas dalam pesanan. Ketika VIP Item masuk ke dalam *To Do List*, Pesanan ini harus segera di-*pick up*, pemain tidak bisa melakukan *pick up* item lain ketika terdapat VIP item, tetapi masih bisa melakukan *drop off* terhadap item yang sudah ada di dalam *in progress*. Sebagai konsekuensi dari adanya item VIP, jika item ini berhasil diantarkan maka pemain akan mendapat 600 Yen dan ability *Return To Sender*. Ability ini digunakan dengan *command RETURN*. Efek dari *Return to Sender* adalah item di atas tas akan dikembalikan ke lokasi *pick up*, pesanan akan dikembalikan ke *To Do List* pada urutan terakhir, dan efek time limit yang terdapat pada pesanan dengan *perishable item* akan di-reset sebelum dikembalikan.

2.4 Gadget Senter Pengecil

Senter Pengecil merupakan *gadget* yang dapat dibeli pada *headquarters* seharga 800 Yen. Efek dari *gadget* ini adalah menghilangkan efek dari satu heavy item jika terdapat pada tumpukan teratas tas. Efek ini akan berlangsung sampai melakukan *drop off* atau *return* pertama kali setelah penggunaan gadget ini.

3 Struktur Data (ADT)

Berikut merupakan struktur data (ADT) yang digunakan dalam tugas besar ini, beserta penggunaan dan penjelasannya.

3.1 ADT Point

- Sketsa struktur data: struktur POINT terdiri atas nilai X dan Y yang bertipe *float*. *Prototype*-nya terdiri atas konstruktor pembentuk *point* untuk membentuk suatu *point* dan fungsi untuk menuliskan *point* ke layar.
- Persoalan yang diselesaikan: menentukan posisi setiap bangunan pada *map*.
- Alasan pemilihan: karena bangunan yang dibaca pada konfig dalam bentuk (X, Y) sehingga diperlukan ADT yang bisa merepresentasikan hal ini, dan juga POINT selaras dengan indeks baris dan kolom pada ADT Map.
- Implementasi: ADT Point dengan nama *file header* "point.h".

3.2 ADT List Statis

- Sketsa struktur data: struktur List Statis yang merupakan array of integer berisi elemen integer yang bernilai positif. Memiliki kapasitas sebesar 5, dengan indeks tak terdefinisi -1 dan -999 sebagai elemen tak terdefinisi untuk menunjukkan tidak ada elemen dalam indeks tersebut.
- Persoalan yang diselesaikan: Sistem inventory
- Alasan pemilihan: Inventory memiliki kapasitas tetap yaitu 5 sehingga list statis merupakan pilihan yang tepat
- Implementasi: ADT List Statis diimplementasikan dengan file header "listpos.h".

3.3 ADT List Dinamis

- Sketsa struktur data: terdiri dari buffer berisikan memori sebanyak *ETypeListDin* dikali dengan jumlah elemen efektif, jumlah elemen efektif, dan ukuran elemen. Elemen *ETypeListDin* berisi karakter building dan elemen point building tersebut.
- Persoalan yang diselesaikan: Menyimpan lokasi tiap gedung di map untuk ditampilkan dan fungsi-fungsi lainnya
- Alasan pemilihan: ADT List dinamis cocok untuk menyimpan lokasi gedung karena lokasi gedung yang jumlahnya berbeda-beda untuk tiap konfigurasi dan tidak akan berubah jumlahnya tengah permainan.
- Implementasi: ADT List Dinamis diimplementasikan dalam *file header* "listdin.h"

3.4 ADT Matrix

- Sketsa struktur data: struktur Matrix terdiri atas *ETypeMatrix* yang berupa integer, *rowEff* yang merupakan indeks baris dari Matrix, dan *colEff* yang merupakan indeks kolom dari Matrix. *Prototype* yang didefinisikan terdiri atas konstruktor pembentuk Matrix

kosong, selektor, baca/tulis Matrix, dan operasi lain seperti untuk menghitung jumlah elemen pada Matrix.

- Persoalan yang diselesaikan: representasi *Adjacent Matrix* sebagai hubungan antar bangunan, sebagai basis untuk menentukan legenda pada ADT Map.
- Alasan pemilihan: *Adjacent Matrix* merupakan Matrix sehingga perlu representasi data Matrix untuk membacanya.
- Implementasi: Diimplementasikan sebagai ADT Matrix dengan nama *file header* "matrix.h"

3.5 ADT Mesin Karakter

- Sketsa struktur data: struktur Mesin Karakter terdiri atas sebuah karakter *currentChar*.
- Persoalan yang diselesaikan: ADT ini digunakan untuk membaca karakter demi karakter melalui *console* dan file.
- Alasan pemilihan: Untuk melakukan pembacaan pada file maupun *console*, diperlukan suatu fungsi *standard input* yang difasilitasi oleh ADT Mesin Karakter dalam ekstensinya dengan ADT Mesin Kata.
- Implementasi: Diimplementasikan sebagai ADT *charmachine* dengan nama *file header* "charmachine.h"

3.6 ADT Mesin Kata

- Sketsa struktur data: struktur Mesin Kata terdiri atas sebuah *array of character contents* yang jumlah elemen maksimalnya sebanyak *CAPACITY_WORD* yang dideklarasikan bernilai 50 dan sebuah integer yang merepresentasikan panjang karakter yang disimpan pada *array of character contents*.
- Persoalan yang diselesaikan: Melakukan pembacaan kata/angka sebagai susunan karakter-karakter yang diproses pembacaannya oleh ADT Mesin Karakter.
- Alasan pemilihan: Dalam melakukan pembacaan masukan, ADT Mesin Karakter tidak dapat berjalan sendiri karena hanya memproses karakter demi karakter. Oleh karena itu, diperlukan ADT Mesin Kata untuk menggabungkan karakter-karakter yang ada.
- Implementasi: Diimplementasikan sebagai ADT *wordmachine* dengan nama *file header* "wordmachine.h"

3.7 ADT PColor

- Sketsa struktur data: terdapat struktur data dengan berbagai jenis warna yaitu *NORMAL*, *RED*, *GREEN*, *YELLOW*, *BLUE*, *MAGENTA*, dan *CYAN*. *Prototype* pada ADT ini meliputi prosedur untuk menampilkan karakter berwarna tertentu.
- Persoalan yang diselesaikan: tampilan bangunan pada peta berwarna berdasarkan lokasi dari pemain.
- Alasan pemilihan: Map memerlukan representasi legenda sehingga diperlukan pewarnaan pada bangunan tertentu.
- Implementasi: Diimplementasikan sebagai ADT *PColor* dengan nama *file header* "pcolor.h"

3.8 ADT PrioQueue

- Sketsa struktur data: Dalam implementasi kami, suatu Priority Queue memiliki array berisi pgEls dengan panjang CAPACITY (100), idxHead, dan idxTail yang masing-masing menjelaskan indeks array ke berapa yang merupakan awal dan akhir dari Priority Queue. pgEls akan berisi data-data item pesanan.
- Persoalan yang diselesaikan: Mengurutkan pesanan item berdasarkan waktu
- Alasan pemilihan: ADT ini akan digunakan untuk mengurutkan data pesanan item dari file config.txt yang belum terurut berdasarkan waktu. Karena dalam file config.txt daftar pesanan belum tentu terurut berdasarkan waktu, PrioQueue yang bisa kita atur 'prioritas' nya merupakan waktu muncul item untuk kemudian dipindahkan ke to do list.
- Implementasi: Implementasi ADT ini terdapat di *file header* "prioqueue.h"

3.9 ADT Stack

- Sketsa struktur data: Dalam implementasi kami, ADT Stack akan memiliki array berisi stackEl dengan panjang CAPACITY (100). stackEl memiliki isi yang sama seperti pgEls.
- Persoalan yang diselesaikan: Sistem Bag
- Alasan pemilihan: ADT Stack cocok dengan persoalan 'tas' dalam program karena dibutuhkan untuk LIFO, dengan item yang bisa di-drop off merupakan item teratas tas, atau bisa dibilang *top of stack*.
- Implementasi: ADT ini diimplementasikan dalam *file header* "stack.h"

3.10 ADT Linked List

- Sketsa struktur data: Linked List berisi node-node yang berisi Elements, yaitu info node tersebut, dan next, yaitu alamat di mana node berikutnya berada. Elements akan memiliki isi yang sama seperti pgEls.
- Persoalan yang diselesaikan: Menentukan ADT untuk To Do List dan In Progress
- Alasan pemilihan: Linked List akan dipakai dalam mencatat To Do List dan In Progress karena sifatnya yang dinamis dan untuk In Progress diperlukan untuk mencetak isi tas dari *bottom of stack* sampai *top of stack* dengan menambahkan item baru di awal Linked List.
- Implementasi: ADT ini diimplementasikan dalam *file header* "list_linked.h"

3.11 ADT Map

- Sketsa struktur data: struktur MAP terdiri atas EITypeMap yang merupakan char, rowEffMap yang merupakan indeks dari baris, dan colEffMap yang merupakan indeks dari kolom. Prototype yang didefinisikan terdiri atas konstruktor pembentuk matriks kosong, konstruktor untuk menyalin MAP, baca/tulis MAP, fungsi untuk menampilkan MAP yang tidak berwarna dan berwarna, serta fungsi-fungsi lain untuk pengecekan lokasi.
- Persoalan yang diselesaikan: Representasi dari peta yang diperlukan pada permainan, menampilkan warna sebagai legenda dari peta.

- Alasan pemilihan: untuk memudahkan sinkronisasi antara *Adjacent Matrix* ke dalam bentuk Matriks
- Implementasi: ADT Matriks yang dimodifikasi elemennya sebagai char dengan nama *file header "map.h"*.

4 Program Utama

Program utama dengan nama file *main.c* akan meng-include semua *file header* dari ADT. Program utama akan melakukan *looping* yang pertama untuk menampilkan main menu berupa judul game serta input untuk melakukan *start game* atau *load game*. *Looping* ini akan selesai jika user melakukan mengetikkan *command EXIT* pada game-nya. Semua pembacaan input dari user dilakukan dengan mesin kata, beserta pembacaan file juga dilakukan dengan modifikasi dari mesin kata. Setelah pemain melakukan input *start* ataupun *load*, program akan memulai permainan dan akan menampilkan waktu yang telah dilewati oleh pemain beserta *command* permainan. *Command* yang tersedia dalam permainan Mobilita adalah *MOVE*, *MAP*, *TO_DO*, *IN_PROGRESS*, *PICK_UP*, *DROP_OFF*, *BUY*, *INVENTORY*, *BAG*, *ITEM*, *GADGET*, *ACTIVE_ABILITY*, *RETURN*, *HELP*, dan *EXIT*. Jika *command* yang di-input oleh pemain tidak termasuk ke dalam salah satu *command* yang terdapat dalam permainan, maka program akan mengeluarkan error dan meminta *command* sampai *command* yang di-input termasuk ke dalam salah satu *command* yang valid.

Permainan akan dimulai dengan pemain berada di *headquarters*, waktu permainan bernilai nol, uang yang dimiliki pemain bernilai nol, dan kapaistas *bag* bernilai 3. Namun, jika pemain melakukan *load game* berarti *state* yang dimiliki oleh pemain berdasarkan konfigurasi yang terakhir di-save oleh pemain tersebut. Setelah itu, pemain bisa menggunakan *command* apapun asalkan valid. Objektif dari permainan ini adalah untuk menyelesaikan semua pesanan dalam waktu se-singkat mungkin. Dengan tujuan dari game ini, pemain bisa menentukan sendiri, jalur yang akan memberikan hasil tercepat, dan efek beserta *gadget* yang bisa memberikan keuntungan terhadap permainan. Berdasarkan objektif ini, permainan program utama dibuat dengan *switch case*, berdasarkan *input command* yang dilakukan oleh pemain akan memanggil fungsi yang selaras dengan *command* tersebut.

Command MOVE merupakan salah fungsi yang fundamental terhadap permainan ini karena Mobilita merupakan permainan yang antar-pesanan, *MOVE* membuat pemain berpindah dari suatu bangunan ke bangunan yang lain. *Command MAP* menampilkan peta dan juga legenda dari peta tersebut, dari peta ini dapat terlihat bangunan mana yang harus dikunjungi untuk mengambil ataupun mengantarkan pesanan. *Command TO_DO* dan *IN_PROGRESS* menampilkan pesanan, *TO_DO* menampilkan pesanan yang harus dikerjakan dan *IN_PROGRESS* menampilkan pesanan yang sudah diambil dan sedang dikerjakan. *Command DROP_OFF* melakukan pengantaran pesanan jika pesanan yang seharusnya diantarkan sesuai dengan bangunan yang sekarang ditempati oleh pemain. *Command BUY* berfungsi untuk membeli *gadget*, ini hanya dapat bisa dilakukan jika pemain berada di *headquarters*. *Command INVENTORY* menampilkan *gadget* yang dimiliki oleh pemain, serta bisa dilakukan pemakaian pada *gadget* dengan *command* tambahan angka dari *gadget* yang dimiliki. *Command BAG*, *ITEM*, *GADGET*, dan *ACTIVE_ABILITY* menampilkan informasi yang lengkap terhadap *command* yang bersangkutan, *BAG* menampilkan item yang berada pada *bag*; *ITEM* menampilkan jenis item dan deskripsinya; *GADGET* menampilkan jenis gadget, harga, beserta efeknya; *ACTIVE_ABILITY* menampilkan *ability* yang sedang aktif (*speed boost*). *Command RETURN* menggunakan *ability Return To Sender* yang dapat dipakai setelah pengantaran item VIP. *Command HELP* menampilkan semua *command* yang tersedia beserta deskripsi dari tiap *command*-nya. Terakhir, *Command EXIT*, keluar dari permainan dan meminta pemain akan

diberi pilihan untuk melakukan save dan memasukkan nama file .txt sebagai tempat untuk menyimpan *state* terakhir dari permainan.

5 Data Test

5.1 Data Test 1

Tes ini dilakukan untuk mengetes apakah program sudah dapat di-compile dan dijalankan. Pertama, lakukan kompilasi dengan menjalankan command `gcc -o main main.c` pada terminal di dalam folder. Kedua, jalankan `main` dalam terminal yang sama.

```

      _ _ _ _ _
     | | | | |
    _|_|_|_|_
   | | | | |
  _|_|_|_|_
 | | | | |
_|_|_|_|_|

COMMAND yang tersedia:
NEW_GAME, LOAD_GAME, dan EXIT

ENTER COMMAND: _
```

Gambar 5.1 Tampilan Judul dan Main Menu

5.2 Data Test 2

Tes ini dilakukan untuk mengetes ketika di dalam program, kita bisa memulai game baru dengan menggunakan command `NEW_GAME`. Kemudian, pengguna diperbolehkan memulai game baru menggunakan suatu file konfigurasi. File konfigurasi akan diminta dari user pada tahap ini.

```

COMMAND yang tersedia:
NEW_GAME, LOAD_GAME, dan EXIT

ENTER COMMAND: NEW_GAME
Masukkan nama file: contoh.txt

Waktu: 0

ENTER COMMAND:
```

Gambar 5.2 Tampilan ketika file konfigurasi berhasil dimuat ke dalam program.

```

COMMAND yang tersedia:
NEW_GAME, LOAD_GAME, dan EXIT

ENTER COMMAND: NEW_GAME
Masukkan nama file: nonexistentfile.txt
File nonexistentfile.txt tidak tersedia
Masukkan nama file: contoh.txt

Waktu: 0

ENTER COMMAND: _
```

Gambar 5.3 Tampilan ketika file konfigurasi awalnya tidak tersedia, kemudian tersedia.

5.3 Data Test 3

Dengan menggunakan Data Test 2, akan mengecek apakah command MOVE bekerja dalam program. Pertama, masukkan command `MOVE`, kemudian gerak ke suatu tempat yang diinginkan (dalam hal ini akan tempat random).

```
Waktu: 0

ENTER COMMAND: MOVE
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 (integer): 2

Mobita sekarang berada di titik E (2,3)!

Waktu: 1

ENTER COMMAND:
```

Gambar 5.4 Tampilan program ketika dimasukkan command MOVE dan posisi move random.

```
Waktu: 1

ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. 8 (1,1)
2. F (3,1)
3. G (3,8)
4. I (4,5)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer): 5
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer): 3

Mobita sekarang berada di titik G (3,8)!

Waktu: 2

ENTER COMMAND: 
```

Gambar 5.5 Tampilan program ketika dimasukkan command MOVE dan posisi move yang diluar scope gerak.

5.4 Data Test 4

Dengan menggunakan Data Test 2, kita akan mengetes apakah To Do List berfungsi dengan baik pada program dengan memasukkan command `TO_DO`.

STEI- ITB	IF2110-TB-11-01	Halaman 13 dari 50 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.		

```

Waktu: 2

ENTER COMMAND: TO_DO
Pesanan pada To Do List:
1. G -> N (Normal Item)
2. M -> B (Heavy Item)
3. B -> M (Normal Item)

Waktu: 2

ENTER COMMAND: _

```

Gambar 5.6 Tampilan program ketika dimasukkan command TO_DO

5.5 Data Test 5

Dengan menggunakan Data Test 2, akan mengetes apakah tampilan In Progress berfungsi dengan baik melalui command `IN_PROGRESS`.

```

ENTER COMMAND: IN_PROGRESS
Pesanan yang sedang diantarkan:

Waktu: 2

```

Gambar 5.7 Tampilan program ketika dimasukkan command IN_PROGRESS

```

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

Waktu: 5

ENTER COMMAND: _

```

Gambar 5.8 Tampilan program ketika dimasukkan command IN_PROGRESS dan sedang mengantarkan item

5.6 Data Test 6

Data test ini digunakan untuk mengetes apakah player dapat mengambil barang untuk diantar dengan command `PICK_UP`.

```

Waktu: 1

ENTER COMMAND: PICK_UP
Pesanan tidak ditemukan!

```

Gambar 5.9 Tampilan program ketika dimasukkan command PICK_UP dan tidak ada barang yang bisa di pick up di lokasi player

```

ENTER COMMAND: PICK_UP
Pesanan berupa Normal Item berhasil diambil!

Waktu: 2

```

Gambar 5.10 Tampilan program ketika dimasukkan command PICK_UP dan ada barang di lokasi.

5.7 Data Test 7

Dengan menggunakan Data Test 2, akan mengetes fungsi drop off item yang diantar dalam program dengan mengetik command **DROP_OFF**.

```
ENTER COMMAND: DROP_OFF
Tidak ada pesanan yang dapat diantarkan.
Waktu: 7
```

Gambar 5.11 Tampilan program ketika dimasukkan command DROP_OFF dan tidak ada item yang bisa diantar (salah lokasi atau bag kosong)

```
Waktu: 8

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

Waktu: 8
```

Gambar 5.12 Tampilan program ketika dimasukkan command DROP_OFF dan terdapat item yang bisa diantar

5.8 Data Test 8

Data test ini digunakan untuk menguji fungsi mengakses shop gadget dengan mengetik command **BUY**.

```
ENTER COMMAND: BUY
Uang Anda sekarang: 0
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 1
```

Gambar 5.12 Tampilan program ketika dimasukkan command BUY

```
ENTER COMMAND: BUY
Gadget hanya bisa dibeli saat berada di Headquarters
Waktu: 3
```

Gambar 5.13 Tampilan program ketika dimasukkan command BUY dan player sedang tidak di HQ.

5.9 Data Test 9

Data test ini digunakan untuk menguji fungsi inventory dengan mengetik command **INVENTORY**.

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

Gambar 5.14 Tampilan program ketika dimasukkan command INVENTORY

5.10 Data Test 10

Data test ini digunakan untuk menguji fungsi help dengan mengetik command **HELP**


```
ENTER COMMAND: HELP

/$$ /$$ /$$$$$$$ /$$ /$$$$$
| $$ | $$ |$$_____/ |$$ |$$_ $$
| $$ | $$ |$$ |$$ |$$ \ $$
| $$$$$$| $$$$ |$$ |$$$$$/
| $$_ $$ |$$_/ |$$ |$$_/
| $$ | $$ |$$ |$$ |$$
| $$ | $$ |$$$$$$$| $$$$$$| $$
|_/ |_/|_____/|_____/|_/

COMMANDS

1. New Game
   Ketik NEW_GAME untuk memulai permainan baru.
2. Load Game
   Ketik LOAD_GAME untuk melanjutkan permainan yang telah tersimpan.
3. Exit
   Ketik EXIT untuk keluar dari permainan.
4. Waktu
   Ketik TIME untuk menunjukkan waktu permainan saat ini.
6. To Do List
   Ketik TO_DO untuk menunjukkan To Do List.
7. In Progress List
   Ketik IN_PROGRESS untuk menunjukkan In Progress List.
8. Kapasitas Tas
   Ketik BAG untuk menunjukkan kapasitas tas saat ini.
9. Jenis Item
   Ketik ITEM untuk menunjukkan jenis item yang tersedia.
10. Ability Aktif
   Ketik ACTIVE_ABILITY untuk melihat ability yang sedang aktif.
11. Gadget
   Ketik GADGET untuk menunjukkan daftar gadget yang tersedia.
12. Daftar Inventory
   Ketik INVENTORY untuk menunjukkan daftar inventory yang tersedia.
13. Map
   Ketik MAP untuk menunjukkan peta permainan saat ini.
14. Move
   Ketik MOVE untuk menunjukkan lokasi-lokasi yang dapat disinggahi.
   Lalu, silahkan memilih angka yang hendak dituju.
15. Pick Up
   Ketik PICK_UP untuk mengambil item jika ada pesanan yang harus diambil pada lokasi.
16. Drop Off
   Ketik DROP_OFF untuk mengantarkan item ke lokasi jika item di tumpukan teratas tas sesuai dengan
   pesanan
17. Buy
   Ketik BUY pada posisi Headquarters untuk menampilkan gadget yang dapat dibeli lalu membelinya.
   Setelah muncul daftar gadget, silahkan masukkan angka gadget yang hendak dibeli.

Waktu: 2

ENTER COMMAND:
```

Gambar 5.15 Tampilan program ketika dimasukkan command HELP

5.11 Data test 11

Data test ini digunakan untuk mengetes apakah ability Return To Sender berfungsi dengan lancar. Ability dinyalakan dengan menjalkankan command `RETURN` dalam program.

```
ENTER COMMAND: RETURN
Normal item berhasil dikembalikan ke Pick Up Point G
```

Gambar 5.15 Tampilan program ketika dimasukkan command RETURN.

```
ENTER COMMAND: RETURN
Tidak ada item di bag.
```

Gambar 5.16 Tampilan program ketika dimasukkan command RETURN dan tas sedang kosong.

5.12 Data Test 12

Data test ini digunakan untuk mengetes apakah player dapat meminta cetakan peta game pada program dengan menggunakan command **MAP**.

```
Waktu: 1
ENTER COMMAND: MAP
* * * * *
* 8      C      D      B *
*      E              *
* F      I      G      H *
*              J          *
*      K          L      *
*      M      N          O *
* A      Q          P      *
* * * * *
```

Gambar 5.17 Tampilan program ketika dimasukkan command MAP dan player di titik C

```
Waktu: 0
ENTER COMMAND: MAP
* * * * *
* 8      C      D      B *
*      E              *
* F      I      G      H *
*              J          *
*      K          L      *
*      M      N          O *
* A      Q          P      *
* * * * *
```

Gambar 5.18 Tampilan program ketika dimasukkan command MAP dan player di HQ.

5.13 Data Test 13

Data test ini dipakai untuk mengetes apakah user bisa membeli barang di HQ dengan menggunakan command **BUY**.

STEI- ITB	IF2110-TB-11-01	Halaman 18 dari 50 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.		

```

ENTER COMMAND: BUY
Uang Anda sekarang: 0
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 1
Uang tidak cukup untuk membeli gadget. Nguli lagi yuk :)

```

Gambar 5.19 Tampilan program ketika user membeli item tetapi kurang uang.

```

ENTER COMMAND: BUY
Uang Anda sekarang: 99999
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 1
Kain Pembungkus Waktu berhasil dibeli.

```

Gambar 5.20 Tampilan program ketika user berhasil membeli item.

```

ENTER COMMAND: BUY
Uang Anda sekarang: 95999
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 1
Kapasitas inventory Anda penuh!

```

Gambar 5.21 Tampilan program ketika inventory user penuh dan membeli item.

5.14 Data Test 14

Data test ini menggunakan Data Test 2. Data test ini digunakan untuk mengetes apakah item VIP yang muncul pada To Do List dan In Progress memberi efek VIP pada pengguna

```

ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. 8 (1,1)
2. F (3,1)
3. G (3,8)
4. I (4,5)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer): 3

Mobita sekarang berada di titik G (3,8)!

Waktu: 14

ENTER COMMAND: TO_DO
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)
10. P -> Q (Perishable Item, sisa waktu 20)

Waktu: 14

ENTER COMMAND: PICK_UP
Pesanan tidak ditemukan!

```

Gambar 5.22 Tampilan program ketika Item normal tidak bisa dipungut karena ada item VIP dalam To Do List

5.15 Data Test 15

Data Test ini dijalankan untuk mengetes apakah kita bisa men-save jalannya permainan melalui fitur Save.

```

ENTER COMMAND: EXIT

Apakah game ingin disave (Y atau N)?Y
Masukkan nama file penyimpanan: haloo.txt
Disimpan dalam file haloo.txt

```

Gambar 5.23 Tampilan program ketika men-save game dengan file save haloo.txt

5.16 Data Test 16

STEI- ITB	IF2110-TB-11-01	Halaman 20 dari 50 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.		

Tes ini dilakukan setelah menjalankan program dengan menggunakan command `LOAD_GAME`. Digunakan untuk melanjutkan game dengan file save yang telah disimpan.

```
COMMAND yang tersedia:
NEW_GAME, LOAD_GAME, dan EXIT

ENTER COMMAND: LOAD_GAME
Masukkan nama file: haloo.txt

Waktu: 3

ENTER COMMAND: █
```

Gambar 5.23 Tampilan program ketika meload game dengan file save haloo.txt

5.17 Data Test 17

Tes ini berjalan setelah menggunakan command `EXIT`. Saat user memilih untuk save game maka akan diminta untuk memasukkan nama file penyimpanan yang kemudian disimpan untuk melanjutkan game setelah melakukan load game.

```
ENTER COMMAND: EXIT

Apakah game ingin disave (Y atau N)?Y
Masukkan nama file penyimpanan: save1
Disimpan dalam file save1
```

Gambar 5.23 Tampilan program Ketika melakukan save game dengan nama file save1

5.18 Data Test 18

Data test ini dilakukan untuk mengetes apakah user dapat membeli gadget senter pembesar dan memakainya.

```

ENTER COMMAND: BUY
Uang Anda sekarang: 99199
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 2
Senter Pembesar berhasil dibeli.

Waktu: 0

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

```

Gambar 5.24 Tampilan program ketika membeli dan menggunakan Senter Pembesar

5.19 Data Test 19

Data test ini dipakai untuk mengecek apakah user dapat membeli dan memakai gadget Kain Pembungkus Waktu menggunakan Data Test 2.

```

ENTER COMMAND: BUY
Uang Anda sekarang: 99999
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 1
Kain Pembungkus Waktu berhasil dibeli.

```

Gambar 5.25 Tampilan program ketika pengguna membeli Kain Pembungkus Waktu

```

ENTER COMMAND: IN_PROGRESS
Pesanan yang sedang diantarkan:
1. Perishable Item (Tujuan: N, Waktu tersisa: 4)
2. Heavy Item (Tujuan: N)
3. Normal Item (Tujuan: N)

Waktu: 12

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

Waktu: 12

ENTER COMMAND: IN_PROGRESS
Pesanan yang sedang diantarkan:
1. Perishable Item (Tujuan: N, Waktu tersisa: 10)
2. Heavy Item (Tujuan: N)
3. Normal Item (Tujuan: N)

Waktu: 12

```

Gambar 5.26 Tampilan program ketika memakai gadget Kain Pembungkus Waktu.

5.20 Data Test 20

Data test ini dipakai untuk mengetes apakah pengguna dapat membeli gadget Pintu Kemana Saja dan memakainya.

```

ENTER COMMAND: BUY
Uang Anda sekarang: 99199
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 3
Pintu Kemana Saja berhasil dibeli.

```

Gambar 5.27a Tampilan program ketika membeli Pintu Kemana Saja

```

ENTER COMMAND: INVENTORY
1. Pintu Kemana Saja
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)
1
Posisi yang dapat dicapai:
1. 8
2. A
3. B
4. C
5. D
6. E
7. F
8. G
9. H
10. I
11. J
12. K
13. L
14. M
15. N
16. O
17. P
18. Q

Posisi yang dipilih?

ENTER COMMAND (integer): 18

Mobita sekarang berada di titik Q (10,3)!
Pintu Kemana Saja berhasil digunakan!

Waktu: 0

```

Gambar 5.27b tampilan program ketika memakai gadget Pintu Kemana Saja

5.21 Data Test 21

Data test ini dipakai untuk mengetes apakah pengguna dapat memakai gadget Mesin Waktu

```

Waktu: 0

ENTER COMMAND: BUY 4
Uang Anda sekarang: 97699
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: Mesin Waktu berhasil dibeli.

```

Gambar 5.28 Tampilan program ketika user membeli Mesin Waktu

STEI- ITB	IF2110-TB-11-01	Halaman 24 dari 50 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.		


```

Waktu: 53

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

Waktu: 3

```

Gambar 5.29a Tampilan program ketika user menggunakan Mesin Waktu

```

Waktu: 32

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

Waktu: 0

```

Gambar 5.29b Tampilan program ketika user menggunakan Mesin Waktu

5.22 Data Test 22

Data test ini digunakan untuk mengetes apakah pengguna dapat membeli dan menggunakan gadget Senter Pengecil.

```

ENTER COMMAND: BUY
Uang Anda sekarang: 99999
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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin kau beli? (ketik 0 jika ingin kembali)
Enter Command: 5
Senter Pengecil berhasil dibeli.

```

Gambar 5.30 Tampilan program ketika user membeli Senter Pengecil

```

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

```

Gambar 5.31 Tampilan program ketika user menggunakan Senter Pengecil

5.23 Data Test 23

Data test ini digunakan untuk mengetes apakah waktu perish item perishable berfungsi dengan baik.

```

ENTER COMMAND: IN_PROGRESS
Pesanan yang sedang diantarkan:
1. Perishable Item (Tujuan: N, Waktu tersisa: 2)
2. Heavy Item (Tujuan: N)
3. Normal Item (Tujuan: N)

Waktu: 14

ENTER COMMAND: MOVE 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 (integer):
Mobita sekarang berada di titik C (1,9)!

Ada perishable item yang sudah melewati time limitnya!

Waktu: 16

ENTER COMMAND: _

```

Gambar 5.32 Tampilan ketika ada item perishable yang melewati masa hidup ketika bergerak

5.24 Data Test 24

Data test ini menggunakan Data Test 2. Data Test ini dipakai untuk mengecek apakah pengguna dapat mengambil dan menurunkan item normal, heavy, dan perishable.

```

ENTER COMMAND (integer):
Mobita sekarang berada di titik G (3,8)!

Waktu: 6

ENTER COMMAND: PICK_UP
Pesanan berupa Normal Item berhasil diambil!

Waktu: 6

ENTER COMMAND: PICK_UP
Pesanan berupa Heavy Item berhasil diambil!

Waktu: 6

ENTER COMMAND: PICK_UP
Pesanan berupa Perishable Item berhasil diambil!

Waktu: 6

```

Gambar 5.33 Tampilan ketika user mengambil item Normal, Heavy, dan Perishable

```

ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. C (1,9)
2. E (2,3)
3. H (3,14)
4. I (4,5)
5. J (5,12)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer): 4

Mobita sekarang berada di titik I (4,5)!

Waktu: 8

ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. E (2,3)
2. F (3,1)
3. G (3,8)
4. J (5,12)
5. K (6,3)
6. L (7,10)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer): 4

Mobita sekarang berada di titik J (5,12)!

Waktu: 10

```

Gambar 5.34 Tampilan ketika user bergerak membawa heavy item

```

Waktu: 14

ENTER COMMAND: DROP_OFF
Pesanan Perishable Item berhasil diantarkan.
Uang yang didapatkan: 400 Yen.

Waktu: 14

ENTER COMMAND: DROP_OFF
Pesanan Heavy Item berhasil diantarkan.
Uang yang didapatkan: 400 Yen.

Waktu: 14

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

Waktu: 14

```

Gambar 5.35 Tampilan ketika user menurunkan item Normal, Heavy, dan Perishable.

5.25 Data Test 25

Data test ini dipakai untuk mengetes apakah item VIP dapat diambil dan diantarkan pengguna, serta apakah efeknya berlaku.

```

Mobita sekarang berada di titik M (8,2)!

Waktu: 15

ENTER COMMAND: TO_DO
Pesanan pada To Do List:
1. M -> B (Heavy Item)
2. B -> M (Normal Item)
3. F -> E (Normal Item)
4. C -> D (Normal Item)
5. O -> P (VIP Item)
6. P -> Q (Perishable Item, sisa waktu 20)

Waktu: 15

ENTER COMMAND: PICK_UP
Pesanan tidak ditemukan!

```

Gambar 5.36 Tampilan ketika pengguna ingin mengambil item lain ketika ada efek VIP

```

Mobita sekarang berada di titik 0 (8,15)!
Waktu: 17

ENTER COMMAND: TO_DO
Pesanan pada To Do List:
1. M -> B (Heavy Item)
2. B -> M (Normal Item)
3. F -> E (Normal Item)
4. C -> D (Normal Item)
5. O -> P (VIP Item)
6. P -> Q (Perishable Item, sisa waktu 20)

Waktu: 17

ENTER COMMAND: PICK_UP
Pesanan berupa VIP Item berhasil diambil!
Waktu: 17

```

Gambar 5.37 Tampilan ketika item VIP diambil oleh pengguna

```

ENTER COMMAND: DROP_OFF
Pesanan VIP Item berhasil diantarkan.
Uang yang didapatkan: 600 Yen.
Mendapatkan ability Return To Sender.

Waktu: 17

```

Gambar 5.38 Tampilan ketika item VIP berhasil diantar oleh pengguna

5.26 Data test 26

Data test ini digunakan untuk mengetes apakah item Heavy memberikan efek Speed Boost ketika sudah didrop off.

```

ENTER COMMAND: DROP_OFF
Pesanan Heavy Item berhasil diantarkan.
Uang yang didapatkan: 400 Yen.

Waktu: 31

ENTER COMMAND: ACTIVE_ABILITY
Ability Speed Boost aktif (0 lokasi)

Waktu: 31

ENTER COMMAND: MOVE
Posisi yang dapat dicapai:
1. D (1,13)
2. H (3,14)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer): 1

Mobita sekarang berada di titik D (1,13)!

Waktu: 31

ENTER COMMAND: ACTIVE_ABILITY
Ability Speed Boost aktif (1 lokasi)

```

Gambar 5.39 Tampilan ketika item Heavy diantarkan dan pengguna mendapatkan Speed Boost

5.27 Data Test 27

Data test ini menguji apakah program bisa mengetes apakah pengguna sudah menyelesaikan permainan dengan kriteria tidak ada item lagi yang perlu diantarkan.

```

Mobita sekarang berada di titik 8 (1,1)!

Waktu: 43

ENTER COMMAND: IN_PROGRESS
Selamat anda telah menyelesaikan permainan mobita dalam 43 satuan waktu!

Apakah game ingin disave (Y atau N)?N

Terima kasih telah bermain ^_^)b

```

Gambar 5.40 Tampilan ketika permainan sudah selesai

5.28 Data Test 28

Data test ini menguji apakah gadget Senter Pengecil dapat berfungsi dengan baik.

```

ENTER COMMAND: PICK_UP
Pesanan berupa Heavy Item berhasil diambil!

Waktu: 6

ENTER COMMAND: MOVE 1
Posisi yang dapat dicapai:
1. C (1,9)
2. E (2,3)
3. H (3,14)
4. I (4,5)
5. J (5,12)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer):
Mobita sekarang berada di titik C (1,9)!

Waktu: 8

ENTER COMMAND: INVENTORY
1. Senter Pengecil
2. Pintu Kemana Saja
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)
1
Senter Pengecil berhasil digunakan!

Waktu: 8

ENTER COMMAND: MOVE 1
Posisi yang dapat dicapai:
1. 8 (1,1)
2. D (1,13)
3. G (3,8)
4. H (3,14)
Posisi yang dipilih? (ketik 0 jika ingin kembali)

ENTER COMMAND (integer):
Mobita sekarang berada di titik 8 (1,1)!

```

Gambar 5.41 Tampilan program perubahan waktu ketika bergerak sebelum dan sesudah memakai gadget Senter Pengecil.

6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
1	START GAME	Memeriksa apakah game dapat dijalankan.	Compile main.c di terminal atau run file .exe.	Data Test 1	Menampilkan pesan selamat datang dan pilihan NEW GAME, LOAD GAME, atau EXIT.	Sesuai dengan ekspektasi.
2	NEW GAME	Memeriksa apakah game dapat dimulai dengan kondisi start (dari 0).	<i>NEW_GAME</i>	Data Test 2	Menerima nama file konfigurasi start sampai valid, menampilkan jumlah pesanan, lalu bisa menerima COMMAND.	Sesuai dengan ekspektasi.
3	LOAD GAME	Memeriksa apakah game dapat dimulai dengan kondisi hasil save sebelumnya (ada progress).	<i>LOAD_GAME</i>	Data Test 16	Menerima nama file konfigurasi hasil save sampai valid, lalu menerima COMMAND.	Sesuai dengan ekspektasi.
4	Pengingat Waktu	Memeriksa apakah waktu dapat ditampilkan setiap sebelum COMMAND.	Setelah selesai COMMAND atau sebelum mulai COMMAND.	Hampir semua data test (ada beberapa yang terlewat screenshot)	Menampilkan waktu.	Sesuai dengan ekspektasi.
5	Move player	Memeriksa apakah Mobita dapat berpindah	<i>MOVE</i>	Data Test 3	Menampilkan pilihan lokasi yang dapat dituju	Sesuai dengan ekspektasi.

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
		lokasi dan bertambah waktunya.			berdasarkan adjacency matrix, menerima pilihan lokasi sampai valid, menambah waktu, dan memindahkan lokasi Mobita.	
6	Print map	Memeriksa apakah peta dapat ditampilkan sesuai warna.	<i>MAP</i>	Data Test 12	Menampilkan peta sesuai warna status lokasi.	Sesuai dengan ekspektasi.
7	To do list	Memeriksa apakah dapat menampilkan to-do-list sesuai waktu.	<i>TO_DO</i>	Data Test 4	Menampilkan to do list sesuai urutan waktu munculnya	Sesuai dengan ekspektasi.
8	Pick up item	Memeriksa apakah dapat mem-pick-up item di lokasi pick-up.	<i>PICK_UP</i>	Data Test 6	<p>Apabila ada pick-up di lokasi, informasi item masuk ke dalam in progress list dan muncul pesan berhasil pick-up.</p> <p>Apabila ada lebih dari 1 item, pick-up item yang lebih dulu muncul.</p> <p>Apabila tidak ada pick-up di lokasi atau ada</p>	Sesuai dengan ekspektasi.

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
					VIP item di to do list, muncul pesan gagal pick-up.	
9	Drop off item	Memeriksa apakah dapat men-drop-off item di lokasi drop-off.	<i>DROP_OFF</i>	Data Test 7	<p>Apabila ada di lokasi drop-off barang teratas, informasi item terbangun dari in progress list, muncul pesan berhasil, serta mendapatkan ability dan uang sesuai item.</p> <p>Apabila tidak di lokasi drop-off barang teratas, muncul pesan gagal drop-off.</p>	Sesuai dengan ekspektasi.
10	In Progress List	Memeriksa apakah dapat menampilkan in-progress list sesuai urutan pick-up terakhir.	<i>IN_PROGRESS</i>	Data Test 5	Menampilkan in progress list (yang sedang dikerjakan) sesuai urutan item yang terakhir di-pick-up.	Sesuai dengan ekspektasi.
11	Buy gadget	Memeriksa apakah pengguna dapat membeli gadget.	<i>BUY</i>	Data Test 8 dan 13	<p>Menampilkan list gadget yang dapat dibeli.</p> <p>Apabila tidak di headquarter atau uang tidak cukup atau inventory penuh, muncul</p>	Sesuai dengan ekspektasi.

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
					pesan gagal. Apabila uang cukup dan inventory tidak penuh, muncul pesan berhasil dan mengurangi uang.	
12	Inventory Player	Memeriksa apakah dapat menampilkan gadget yang dimiliki, menggunakan gadget.	<i>INVENTORY</i>	Data Test 9	Menampilkan list gadget yang dimiliki user, pilihan memilih untuk menggunakan gadget. Apabila tidak memiliki tetapi ingin menggunakan gadget, muncul pesan gagal.	Sesuai dengan ekspektasi.
13	Print help	Memeriksa apakah dapat menampilkan bantuan.	<i>HELP</i>	Data Test 10	Menampilkan list bantuan COMMAND beserta fungsinya.	Sesuai dengan ekspektasi.
14	Ability Return to Sender	Memeriksa apakah ability Return To Sender (kembalikan item teratas ke lokasi pick-up) berfungsi.	<i>RETURN</i>	Data Test 11	Bila bukan VIP, item teratas dari bag hilang dari bag dan in-progress-list, lalu item muncul kembali di to-do-list urutan terakhir. Bila perishable	Sesuai dengan ekspektasi.

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
					item, waktu di- reset sebelum dikembalikan.	
15	Efek Normal Item	Memeriksa apakah pemain mendapatkan efek item.	<i>PICK_UP</i> <i>MOVE</i> <i>DROP_OFF</i>	Data Test 24	Pemain mendapatkan 200 Yen setiap drop-off normal item.	Sesuai dengan ekspektasi.
16	Efek Heavy Item	Memeriksa apakah pemain mendapatkan efek item.	<i>PICK_UP</i> <i>MOVE</i> <i>DROP_OFF</i>	Data Test 24 dan 26	Pemain bergerak lebih lambat 1 unit tiap heavy item (kumulatif). Pemain mendapatkan 400 Yen dan ability Speed Boost setiap drop-off heavy item.	Sesuai dengan ekspektasi.
17	Efek Perishable Item	Memeriksa apakah pemain mendapatkan efek item.	<i>PICK_UP</i> <i>MOVE</i> <i>DROP_OFF</i>	Data Test 19, 23, dan 24	Count-down ditampilkan dalam in- progress-list. Bila waktu habis, item hilang dari tas dan in- progress-list. Pemain mendapatkan 400 Yen dan ability Increase Capacity.	Sesuai dengan ekspektasi.
18	Efek VIP	Memeriksa apakah	<i>PICK_UP</i> <i>MOVE</i>	Data Test	Bila ada VIP item dalam to-	Sesuai dengan

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
	Item	pemain mendapatkan efek item.	<i>DROP_OFF</i>	25	do-list, tidak bisa pick-up item lain. Pemain mendapatkan 600 Yen dan ability Return to Sender.	ekspektasi.
19	Ability Speed Boost	Memeriksa apakah pemain mendapatkan efek ability.	Otomatis setelah drop-off heavy item.	Data Test 26	Bila sedang tidak ada heavy item dalam tas dan tidak pick-up heavy item lagi, waktu bertambah 1 unit setiap berpindah 2 lokasi, berlaku selama 5 unit waktu atau 10 lokasi.	Sesuai dengan ekspektasi.
20	Ability Increase Capacity	Memeriksa apakah pemain mendapatkan efek ability.	Otomatis setelah drop-off perishable item.	Data Test 24	Bila tidak penuh, kapasitas tas bertambah 1.	Sesuai dengan ekspektasi.
21	Gadget Kain Pembungkus Waktu	Memeriksa apakah pemain mendapatkan efek gadget.	Bila gadget sudah terbeli, <i>INVENTORY</i> , <nomor gadget>	Data Test 19	Perishable item teratas dari tas kembali ke durasi semula.	Sesuai dengan ekspektasi.
22	Gadget Senter Pembesar	Memeriksa apakah pemain mendapatkan efek gadget.	Bila gadget sudah terbeli, <i>INVENTORY</i> , <nomor gadget>	Data Test 18	Kapasitas tas meningkat dua kali lipat, tetapi tidak lebih dari kapasitas maksimal.	Sesuai dengan ekspektasi.

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
23	Gadget Pintu Ke Mana Saja	Memeriksa apakah pemain mendapatkan efek gadget.	Bila gadget sudah terbeli, <i>INVENTORY</i> , <nomor gadget>	Data Test 20	Pemain berpindah lokasi tanpa menambah waktu.	Sesuai dengan ekspektasi.
24	Gadget Mesin Waktu	Memeriksa apakah pemain mendapatkan efek gadget.	Bila gadget sudah terbeli, <i>INVENTORY</i> , <nomor gadget>	Data Test 21	Waktu berkurang 50 unit, tetapi tidak kurang dari 0.	Sesuai dengan ekspektasi.
25	Gadget Senter Pengecil	Memeriksa apakah pemain mendapatkan efek gadget.	Bila gadget sudah terbeli, <i>INVENTORY</i> , <nomor gadget>	Data Test 22	Bila heavy item berada teratas dalam tas, efeknya hilang sampai drop-off/return.	Sesuai dengan ekspektasi.
26	EXIT	Memeriksa apakah pemain dapat keluar dari game.	<i>EXIT</i>	Data Test 17	Menampilkan pilihan SAVE GAME atau tidak.	Sesuai dengan ekspektasi.
27	SAVE GAME	Memeriksa apakah progress game dapat disimpan dalam suatu file konfigurasi.	SAVE	Data Test 15	Menerima nama file konfigurasi save (bisa file baru atau lama), lalu menyimpan progress game.	Sesuai dengan ekspektasi.
28	END GAME	Memeriksa apakah game dapat selesai.	Semua item telah diantar atau hangus dan Mobita di HQ. Kemudian menjalankan command	Data test 27	Menampilkan pesan game selesai, jumlah item yang berhasil diantar, dan waktu yang dilampaui.	Sesuai dengan ekspektasi.

No.	Fitur yang Dites	Tujuan Testing	Langkah - Langkah Testing (huruf miring menandai masukan)	Input Data Test (huruf miring menandai masukan)	Hasil yang Diharapkan	Hasil yang Keluar
			apapun			

7 Pembagian Kerja dalam Kelompok

No	Nama/NIM	Tugas
1	Bariza Haqi/13520018	ADT, laporan
2	Maharani Ayu Putri Irawan/13520019	ADT Point, ADT Mesin Karakter, ADT Mesin Kata, ADT List Dinamik, ADT Matriks, ADT Node, ADT Linked List, Start Game, Load Game, Save Game, Help, Ability Speed Boost, laporan.
3	Damianus Clairvoyance Diva P./13520035	Colored map, money, time, bag capacity, move, dropoff perishable, heavy item, gadget senter pembesar, gadget pintu ke mana saja, laporan.
4	Adiyansa Prasetya Wicaksana/13520044	ADT Map, Kode program utama, laporan
5	Farrel Farandieka Fibriyanto/13520054	ADT Stack, ADT PrioQueue, Implementasi tas, To Do List dan In Progress serta efek dan fungsi yang mengikuti seperti drop off, dst, implementasi Kain Pembungkus Waktu dan Return To Sender.

8 Lampiran

8.1 Deskripsi Tugas Besar 2

Dengan adanya pandemi COVID-19, usaha orang tua Mobita mengalami penurunan pendapatan. Setelah ditinggalkannya PPKM, Mobita ingin membantu orang tuanya mendapat penghasilan menjadi seorang kurir. Untuk membantu Mobita, dibuatlah aplikasi yang membantu melacak pesanan, navigasi peta, serta mengambil dan menurunkan barang.

Game Mobilita merupakan game *strategy* dan *time-management* yang menyimulasikan petualangan Mobita sebagai kurir. Game ini dimainkan dengan memberikan COMMAND pada Command Line Interface. Game ini dibuat dalam bahasa C. Pada game ini, terdapat sebanyak 18 *commands* yang dapat digunakan pemain dalam memainkan permainan Mobilita ini. Program dibuat dengan mengimplementasikan algoritma dan struktur data yang telah dipelajari dalam mata kuliah IF2110 Algoritma dan Struktur Data.

8.2 Notulen Rapat

STEI- ITB	IF2110-TB-11-01	Halaman 40 dari 50 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.		

Catatan Tubes Alstrukdat

Kelompok 11 K01 2021/2022

Link Jamboard

https://jamboard.google.com/d/1jiiHCSqNHxHCmOYizpQSDfGvYFnlU59TTj_ddDR7IA4/viewer?f=1

Jadwal Meet

No.	Hari	Tanggal	Jam (WIB)
1.	Sabtu	23/10/2021	16.00
2.	Jumat	29/10/2021	
3.	Rabu	10/11/2021	10.00
4.	Jumat	12/11/2021	10.00
5.	Senin	15/11/2021	21.00
6.	Selasa	16/11/2021	21.00
7.	Rabu	17/11/2021	21.30

MEET 1 (23/10/2021)

Functions:

1. Start_game → input file, read, konfigurasi Map
2. Load_game
3. Count_time
4. Count_money
5. Count_capacity
6. ADT Gadget → adaptasi list static
 - a. Buy_gadget
 - b. Remove_gadget
7. Move
8. Pick_up
9. Drop_off
10. Remove_perishable
11. Help
12. Return
13. Inventory

14. Save_game

Data Structures / Variables:

1. To Do List
2. In Progress List
3. Map
4. Inventory (ADT Gadget)
5. Capacity
6. Time
7. Money

ADT

- ADT Mobita
 - ADT Gadget
 - List
 - ADT Tas
 - ADT Stack
 - Capacity
 - Time
 - Money
 - ADT ToDoList
 - List Static [CAPACITY = JumlahPesanan]
 - ADT InProgress
 - ADT Queue [CAPACITY = JumlahPesanan]
 - ADT Bangunan
 - ADT List Dinamis
 - ADT Point
 - ADT Map (ADT Matriks)
 - Input ADT Mesin Karakter & Kata

Pembagian Tugas

1. Start_game → 2 orang (Anca, Rani)
2. Load_game (Adaptasi start) (nanti)
3. Save_game (nanti)
4. Command, Menu (Main nya lah) (nanti)
5. Move, Waktu, Money, Capacity, Perishable Items, Count_money, Count_capacity, Count_time, Konfigurasi → 2 orang (Clair, Bariza)
6. Pick_up, Drop_off, capacity nambah, ToDoList mindah InProgress → 1 orang (Farrel)

Catatan buat meet 2: **Find out gimana buat InEffect**

Target: Sebelum Jumat, 29 Oct 2021, 9.30 a.m.

Meet Asistensi: Jumat 9.30 a.m.

Agenda Asistensi: Tanya klarifikasi halaman 12, tanya kalo ada yang gabisa/dibingungin dari kerjaan pembagian tugas Meet 1

Username Github:

STEI- ITB	IF2110-TB-11-01	Halaman 42 dari 50 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. Farrel → Noxira / 13520054@std.stei.itb.ac.id
2. Anca → apwic/adiyansawicaksana@gmail.com
3. Clair → dclairvoyance / damianus.clairvoyance@gmail.com
4. Bariza → barizahaqi / haqibariza@gmail.com


Log Activity!

Tempat dan Dokumentasi Meet 1

<https://meet.google.com/ceu-dvcw-ctr>

Invitation: IF2110_Tubes_Koordinasi1 @ Sat Oct 23, 2021 4pm - 5pm (WIB) (13520019@std.stei.itb.ac.id)

[Inbox x](#)

 **13520035@std.stei.itb.ac.id** via [google.com](#)
to me, 13520054, 13520044, 13520018

4:03 PM (2 hours ago) ☆ ↶ ⋮

Oct 23
Sat

IF2110_Tubes_Koordinasi1
[View on Google Calendar](#)

When Sat Oct 23, 2021 4pm - 5pm (WIB)

Who 13520054@std.stei.itb.ac.id, 13520044@std.stei.itb.ac.id, 13520018@std.stei.itb.ac.id...

[More options](#)

Agenda
Sat Oct 23, 2021

No earlier events

4pm IF2110_Tubes_Koordinasi1

No later events

You have been invited to the following event.

IF2110 Tubes Koordinasi1

When Sat Oct 23, 2021 4pm - 5pm Western Indonesia Time - Jakarta [more details »](#)

Calendar 13520019@std.stei.itb.ac.id

Who

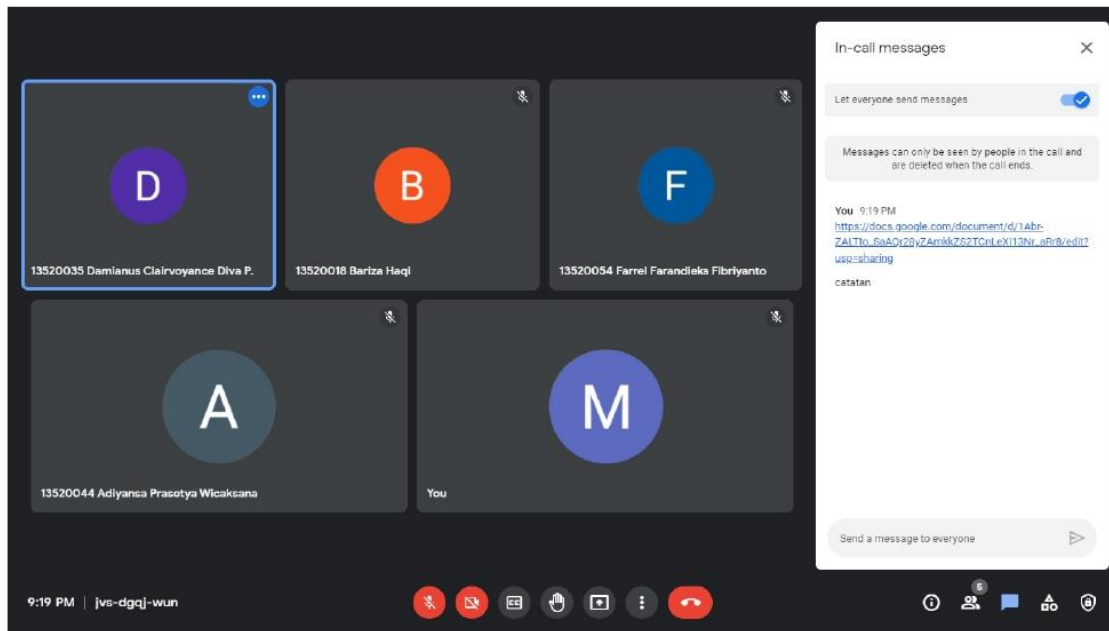
- 13520035@std.stei.itb.ac.id - organizer
- 13520054@std.stei.itb.ac.id
- 13520044@std.stei.itb.ac.id
- 13520019@std.stei.itb.ac.id
- 13520018@std.stei.itb.ac.id

Going (13520019@std.stei.itb.ac.id)? [more options »](#)

Invitation from [Google Calendar](#)

You are receiving this email at the account 13520019@std.stei.itb.ac.id because you are subscribed for invitations on calendar 13520019@std.stei.itb.ac.id

MEET 2 (29/10/2021)



Progress Report

- Start_game (baca file dan baca lewat console sudah selesai)
- Konfigurasi map selesai
- Fungsi count otw, sisanya selesai (current loc pake char)
- Pickup dropoff sudah, todolist sama in progress sudah → orderItems, inEffect VIP selesai
- Capacity nambah gtw bisa set integer aja sbg max atau (?), func mindah todolist ke inprogress udah

Pembagian Tugas

No	Tugas	PJ
1.	Lanjut gadget n inventory	Iza
2.	Main	Anca
3.	Remove perishable	Farrel
4.	Count	Clair
5.	Load save help	Rani



MEET 3 (10/11/2021)

Urutan Load Game:

Time

Bag Capacity

Money

Inventory

Todolist

Inprogresslist

Orderedorders

Current Loc

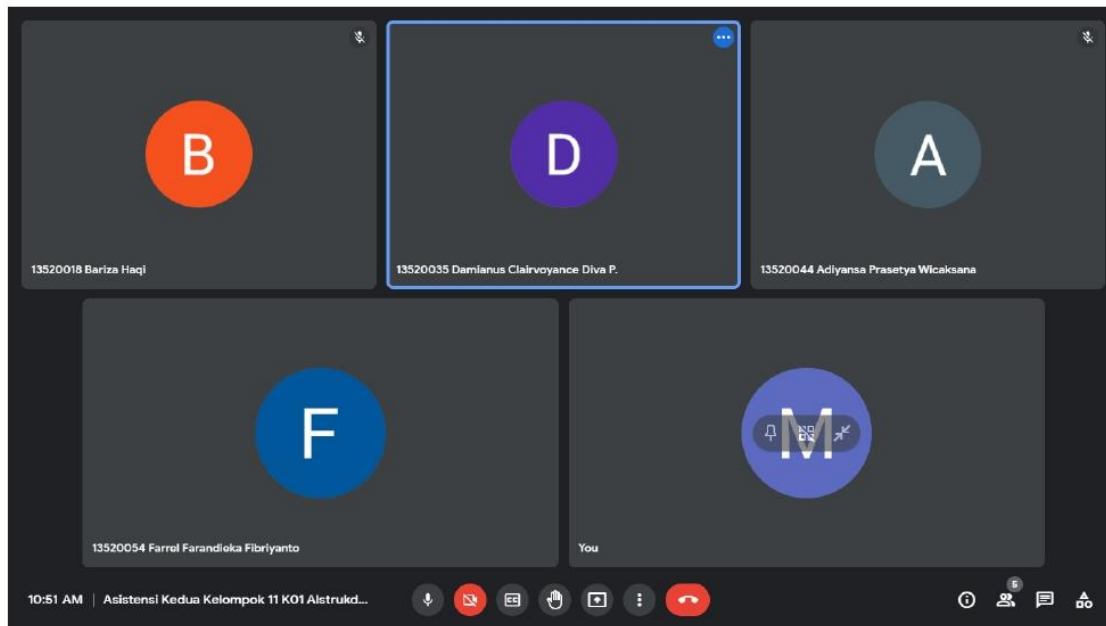
Current Ability

Remaining task:

- Gabungin (Anca)
- Count : ability (3), gadget effect (5)
- Ability 1: Speed boost (Rani)
- Ability 2: Permanent Increase Capacity
- Ability 3: Return (Farrel)
- Gadget 1: Kain Pembungkus Waktu
- Gadget 2: Senter Pembesar (Clair)
- Gadget 3: Pintu Kemana Saja
- Gadget 4: Mesin Waktu (Ija)
- Gadget 5: Senter Pengecil

Dekomposisi count:

1. Ability
2. Gadget Effect



<https://meet.google.com/cfi-ibyx-fxw>

MEET 4 (12/11/2021)

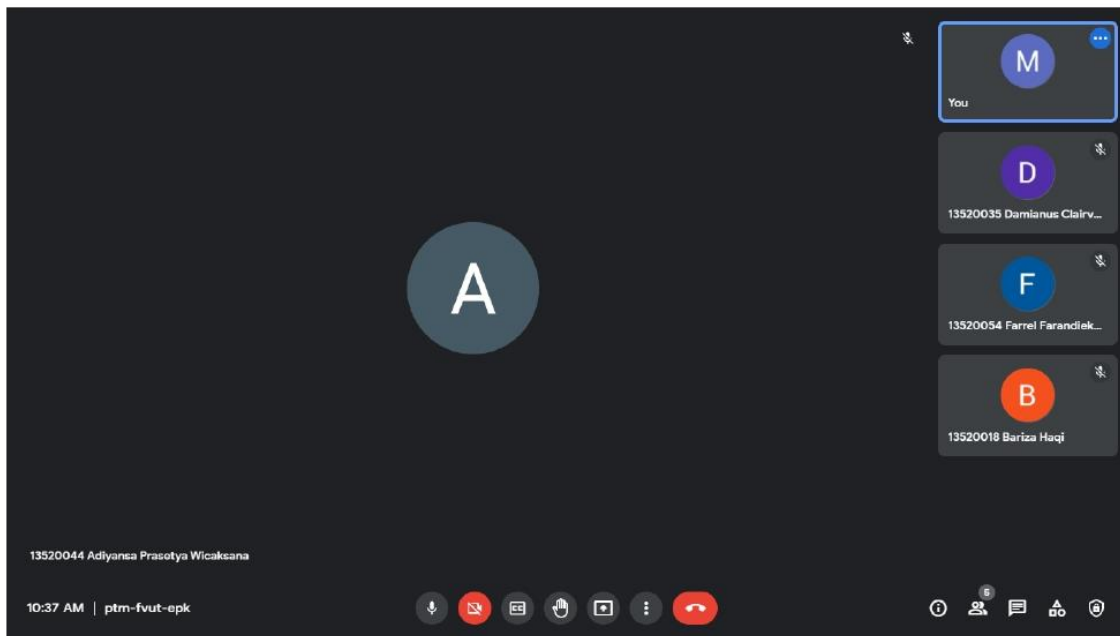
To Do:

- Rani : 2 gadget
- Clair : 2 gadget/ability
- Anca : Game Flow
- Farrel : Return to sender sama waktu
- Ija : laporan

Link word : [Laporan Tubes IF2110 Mobilita.docx](#)

Tempat dan Dokumentasi:

<https://meet.google.com/ptm-fvut-epk>



MEET 5 (15/11/2021)

TO_DO → Selasa 16/11/2021 DONE!

1. Ija → Buy cuma di Headquarters
2. Ija → Gadget masukan selain angka (input selain angka jangan infinite loop → error handling)
3. Rani → Pcolor ganti
4. Anca, Farrel → Commands yang belum: time, bag, item, active_ability, gadget, return to sender
5. Rani → Waktu nya speed boost (setelah ngangkat → Rani)
6. Anca → Strcmp ganti!
7. Rani → Ada yg blm ganti memset sama memcpy
8. Farrel → Todolist
9. (TEST BESOK!) Load game

Pembagian :

1. Anca : Program utama
2. Rani : Notulensi, ADT"
3. Farrel : orderitems
4. Clair : laporan
5. Ija :

MEET 6 (16/11/2021)

Tidak ada catatan, debug kode.

MEET 7 (17/11/2021)

Debug kode

Urutan Save/Load File:
2 bilangan N, M
X, Y (Headquarter)
nLoc
Building sebanyak nLoc
(nLoc+1)*(nLoc+1) matrix adjacency
nOrder
Orders sebanyak nOrder
nOrderedOrders
Ordered orders
current loc
Current time
Current money
Current bagcapacity
Ntodolist
Todolist
nInProgress
InProgress
nInventory
Inventory
boostCount
speedBoost
bag
efekVIP
efekHeavyItem

8.3 Log Activity Anggota Kelompok

No	Nama/NIM	Hari, Tanggal	Kondisi	Progress
1	Bariza Haqi / 13520018	28/10/2021	Rumah	Sistem perhitungan money, time, dan bag capacity. Sistem command move.
		9/11/2021	Rumah	Sistem Inventory dan Gadget
		10/11/2021	Rumah	Implementasi gadget Mesin waktu dan Senter pengecil
		16/11/2021	Meet	Debugging
		17/11/2021	Meet	Laporan
2	Maharani Ayu	24/10/2021	Rumah	Mengumpulkan ADT-ADT yang telah

	Putri Irawan / 13520019	– 25/10/2021		dibuat dalam praktikum yang telah dilaksanakan dan memodifikasi sesuai keperluan tugas besar.
		27/10/2021	Rumah	Menyelesaikan fungsi StartGame dan membuat penyesuaian pada ADT Word Machine dan Char Machine untuk membaca file.
		29/10/2021	Rumah	Memodifikasi ADT Word Machine dan Char Machine untuk menerima input dari console.
		4/11/2021	Meet	Melanjutkan debug modifikasi ADT Word Machine dan Char Machine.
		9/11/2021	Rumah	Men-debug modifikasi ADT Word Machine dan Char Machine.
		12/11/2021	Rumah	Mengimplementasikan Save Game (belum benar 100%)
		13/11/2021	Rumah	Melanjutkan debug implementasi Save Game
		15/11/2021	Rumah	Koreksi Start Game, Load Game, Save Game. Implementasi ability speed boost.
		16/11/2021	Meet	Debugging
		17/11/2021	Meet	Debugging, laporan
		18/11/2021	Meet	Debugging, laporan
3	Damianus Clairvoyance Diva P. / 13520035	28/10/2021	Rumah	Sistem pewarnaan map. Sistem perhitungan money, time, dan bag capacity. Sistem command move.
		9/11/2021	Rumah	Sistem main sementara. Efek penambahan money untuk dropoff. Efek dropoff perishable.
		10/11/2021	Rumah	Efek pengurangan money untuk buy gadget. Error handling untuk dropoff dan buy gadget.
		15/11/2021	Meet	Efek increment time pada heavy item. Efek gadget senter pembesar dan gadget pintu ke mana saja.
		16/11/2021	Meet	Laporan.
		17/11/2021- 18/11/2021	Meet	Debugging. Laporan.
4	Adiyansa Prasetya Wicaksana / 13520044	27/10/2021	Rumah	Implementasi ADT Map
		15/11/2021	Meet	Menambahkan fungsi untuk menyelesaikan game
		16/11/2021	Rumah	Menambahkan beberapa command pada main dan merapihkan command
		17/11/2021	Rumah	Debugging, Merapihkan struktur program dan membagi ke dalam folder


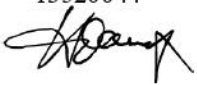





		15/11/2021 – 17/11/2021	Meet dan Rumah	Laporan
5	Farrel Farandieka Fibriyanto / 13520054	26/10/2021	Rumah	Implementasi Implementasi tas, To Do List dan In Progress
		5/11/2021	Rumah	Debugging implementasi sendiri
		10/11/2021	Rumah	Implementasi Return To Sender dan Kain Pembungkus Waktu
		14/11/2021	Rumah	Debugging
		15/11/2021 – 17/11/2021	Meet dan Rumah	Laporan

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


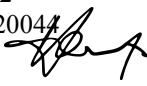




No. Kelompok/Kelas : 11 / K01
Nama Kelompok : AL-JAYA
Anggota Kelompok (Nama/NIM) : 1. Maharani Ayu Putri Irawan / 13520019
2. Adiyansa Prasetya Wicaksana / 13520044
3. Farrel Farandieka Fibriyanto / 13520054
4. Damianus Clairvoyance Diva P. / 13520035
5. Bariza Haqi / 13520018

Asisten Pembimbing : Muhammad Kamal Shafi / 13518113

Asistensi I

Tanggal : 29 Oktober 2021	Catatan Asistensi:
Tempat : Google Meet (https://meet.google.com/fky-doiw-oud)	<ol style="list-style-type: none"> 1. Tidak ada konvensi urutan save/load 2. Tidak diperbolehkan menggunakan scanf, semua harus pakai charmachine/wordmachine 3. Boleh memodifikasi ADT 4. Bangunan tetap hanya pakai list dinamis meskipun statis dianggapnya 5. Waktu → pesanan ada waktu datangnya, kalau pakai mesin waktu, berkurang 50 unit, pesanan yang udah masuk to do list (sudah dikerjakan), di waktu yang tertera nanti tidak muncul lagi (tidak dobel) 6. Contoh Tampilan Game di halaman 12 Spesifikasi Tugas Besar typo, seharusnya bukan V(4,5) setelah command MAP, tetapi tetap di I(4,5) sebelum dan setelah command MAP
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 13520019  2 13520044  3 13520054  4 13520035  5 13520018 	<p>Dokum:</p> 
	Tanda Tangan Asisten:  Digitally signed by Muhammad Kamal Shafi Date: 2021.10.30 10:50:08 +07'00'

Asistensi II

Tanggal : 10/11/2021	Catatan Asistensi:
Tempat : Google Meet (https://meet.google.com/cfi-ibyx-fxw)	<ol style="list-style-type: none"> 1. Perishable count waktu? Kalau waktu nya di 15 ya diambil di 15 2. Boleh pakai memset dan memcpy untuk nama file input berulang? Boleh selama masih termasuk library yang boleh di include. 3. Boleh pakai getc untuk baca char satu persatu dari file? Boleh. 4. Apakah harus pakai makefile? Tidak 5. [Kak Kamal] Di github repo masih include .c, kenapa tidak .h? Belum diubah, akan diubah. [Kak Kamal] Saran: pakai script, di Windows .bat, di Linux .bash
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 13520019  2 13520044  3 13520054  13520035  5 13520018 	
	Tanda Tangan Asisten:  Digitally signed by Muhammad Kamal Shafi Date: 2021.11.15 18:16:30 +07'00'