

LAPORAN TUGAS BESAR

IF2110/Algoritma dan Struktur Data

MOBILITA




Dipersiapkan oleh:

Kelompok 1 – K01

Vionie Novencia Thanggestyo	13520006
Gagas Praharsa Bahar	13520016
Muhammad Garebaldhie Er Rahman	13520029
Taufan Fajarama Putrawansyah R	13520031
Jeremy S.O.N. Simbolon	13520042
Arik Rayi Arkananta	13520048

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-01-01</i>	<i>32</i>
			<i>Revisi</i>	<i>18-11-2021</i>

Daftar Isi

1 Ringkasan.....	4
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 VIP Item	4
2.2 Ability Return to Sender	4
2.3 Gadget Senter Pengecil	4
2.4 Save Game.....	4
2.5 Load Game	5
3 Struktur Data (ADT)	5
3.1 ADT Point	5
3.2 ADT List Statis.....	5
3.3 ADT List Dinamis	5
3.4 ADT Matriks	5
3.5 ADT Mesin Karakter dan Mesin Kata	5
3.6 ADT Queue	6
3.7 ADT Stack.....	6
3.8 ADT Linked List	6
3.9 ADT Game	6
3.10 ADT Utility.....	6
3.11 ADT Item.....	6
3.12 ADT Gadget	6
3.13 ADT Ability.....	7
3.14 ADT Lokasi	7
3.15 ADT PColor.....	7
4 Program Utama	7
5 Algoritma-Algoritma Menarik.....	8
5.1 Mekanisme Mesin Kata.....	8
6 Data Test	8
6.1 Data Test NEW_GAME/LOAD_GAME.....	8
6.2 Data Test MOVE.....	9
6.3 Data Test PICK_UP	11
6.4 Data Test DROP_OFF	11
6.5 Data Test MAP.....	12
6.6 Data Test TO_DO	13
6.7 Data Test IN_PROGRESS	14
6.8 Data Test BUY	14
6.9 Data Test INVENTORY	15
6.10 Data Test HELP	18
6.11 Data Test RETURN.....	18
6.12 Data Test SAVE	19
6.13 Data Test QUIT	19
7 Test Script	20
8 Pembagian Kerja dalam Kelompok	26
9 Lampiran	26

9.1	Deskripsi Tugas Besar 2.....	26
9.2	Notulen Rapat.....	27
9.3	Log Activity Anggota Kelompok.....	27
9.4	Form Asistensi Tugas Besar.....	30

1 Ringkasan

Pada tugas besar IF2110 Algoritma dan Struktur Data 2021, kami mendapat tantangan untuk membuat permainan berbasis *command-line interface* (CLI) tentang pengantaran pesanan dan mengumpulkan uang hingga mencapai target. Pemain dapat menggunakan *ability* yang didapat setelah membeli *gadget* atau mengantar pesanan tertentu. Pemain juga dapat menggunakan fitur lain yang tersedia sesuai spesifikasi tugas besar.

Laporan ini dibuat untuk merangkum hasil tugas besar yang kami lakukan dan mencakup beberapa penjelasan. Penjelasan tersebut di antaranya, spesifikasi tambahan untuk fitur bonus, struktur data (ADT) yang digunakan, alur program utama, algoritma yang menurut kami menarik, hasil *testing* program, dan beberapa lampiran.

Secara garis besar, hasil dari tugas besar kami memenuhi semua spesifikasi tugas yang diberikan dan berhasil dijalankan pada mesin Windows-based dan UNIX-based. Tugas besar yang diberikan melatih kami untuk menyelaraskan berbagai struktur data (ADT) yang selama ini dipelajari dalam kuliah IF2110 Algoritma dan Struktur Data menjadi sebuah program permainan yang dapat melakukan berbagai *command* tertentu.

2 Penjelasan Tambahan Spesifikasi Tugas

Pada tugas besar ini, kami mengerjakan beberapa fitur bonus. Berikut adalah spesifikasi dari fitur bonus yang dikerjakan.

2.1 VIP Item

VIP *item* akan memiliki efek berbeda dengan *item* lain ketika masuk ke To Do List. Pesanan di To Do List dengan jenis *item* VIP harus segera diantarkan dan pemain tidak dapat melakukan command PICK_UP untuk jenis *item* lainnya. Setelah berhasil mengantar pesanan VIP *item*, pemain akan mendapat ability Return to Sender.

2.2 Ability Return to Sender

Ability Return to Sender dapat diaktifkan dengan command RETURN dan akan mengembalikan *item* teratas pada tas (kecuali VIP *item*) ke lokasi pick up, pesanan dengan *item* tersebut akan dimasukkan ke To Do List pada urutan terakhir. Khusus untuk *perishable item*, time-limitnya akan direset sebelum dikembalikan.

2.3 Gadget Senter Pengecil

Gadget Senter Pengecil dapat dibeli melalui command BUY ketika berada di Headquarters dan dapat digunakan melalui command INVENTORY jika sudah membeli sebelumnya. Senter Pengecil akan menghilangkan efek *heavy item* jika *item* tersebut terdapat pada tumpukan teratas tas (tidak akan berdampak apa-apa jika *item* teratas tas bukan *heavy item* dan gadget akan terbuang sia-sia). Efek dari senter pengecil akan berlangsung sampai dilakukan command DROP_OFF pertama kali setelah gadget digunakan.

2.4 Save Game

Pada *loop* permainan utama, ditambahkan command SAVE_GAME yang dipanggil untuk melakukan save state dari permainan yang sedang dijalankan. Hasil dari save game akan

disimpan pada file yang kemudian dapat dilakukan load pada awal permainan dimulai lagi di waktu yang lain.

2.5 Load Game

Pada implementasi save game, program dapat menyimpan file konfigurasi baru yang menyimpan state dari game pada waktu save game. Fungsi load game berfungsi untuk memanggil kembali state dari game yang telah disave. Setelah file dibaca, maka game akan dilanjutkan dari state pada penyimpanan game sebelumnya.

3 Struktur Data (ADT)

Kami menggunakan beberapa struktur data untuk menyusun tugas besar. Berikut adalah struktur data yang kami gunakan dalam program kami.

3.1 ADT Point

ADT Point terdiri atas *tuple* yang terdiri atas nilai absis dan ordinat dalam tipe *integer*. Struktur data ini dipilih untuk merepresentasikan lokasi bangunan pada peta permainan. Implementasi ADT ini meliputi selektor, interaksi *input/output*, dan operasi relasional yang dapat ditemukan pada *file point.c*.

3.2 ADT List Statis

ADT List Statis merupakan *array* dengan ukuran yang bersifat statis. Struktur data ini dipilih untuk merepresentasikan ADT Inventory pada *file gadget.c* yang digunakan sebagai tempat penyimpanan *gadget* yang digunakan oleh Mobita. Implementasi ADT ini meliputi konstruktor, interaksi *input/output*, dan operasi fungsional lain yang melibatkan jenis-jenis *gadget*.

3.3 ADT List Dinamis

ADT List Dinamis merupakan *array* dengan ukuran yang bersifat dinamis. Struktur data ini dipilih untuk merepresentasikan daftar bangunan yang berubah sesuai masukan konfigurasi permainan. Implementasi ADT ini meliputi konstruktor, selektor, interaksi *input/output*, dan operasi penambahan elemen yang dapat ditemukan pada *file listdin.c*. Listdin berisi banyaknya elemen efektif, serta array of lokasi yang nantinya akan dialokasi secara dinamis.

3.4 ADT Matriks

ADT Matriks merupakan *array* bertingkat dengan ukuran yang bersifat statis. Struktur data ini dipilih untuk merepresentasikan peta permainan dan *adjacency matrix* dari bangunan. Implementasi ADT ini mencakup konstruktor dari matriks yang dapat ditemukan pada *file matrix.c*.

3.5 ADT Mesin Karakter dan Mesin Kata

ADT Mesin Karakter dan Mesin Kata adalah basis untuk membaca input dari user. ADT ini dibuat agar aplikasi yang kita buat dapat menerima masukan dari pengguna lalu diproses sesuai dengan masukan dari penggunaanya. ADT ini memiliki dua source code, yaitu *charmachine.c* dan *wordmachine.c*. Charmachine adalah ADT yang memproses per karakter, sedangkan wordmachine memproses per kata dan dibuat dengan dasar charmachine.

3.6 ADT Queue

ADT Queue merupakan *array* statis yang akses elemennya mengikuti aturan *first in, first out* (FIFO). Struktur data ini dipilih untuk merepresentasikan urutan pesan yang diterima oleh Mobita dengan prioritas tertentu pada *file prioqueue.c*. Implementasi ADT ini meliputi konstruktor, selektor, operasi pemeriksaan, interaksi *input/output*, dan operasi manipulasi elemen. Pada ADT Queue ini juga terdapat sub ADT lainnya yaitu ADT Pesanan yang menyimpan pesan ke dalam bentuk abstraknya. ADT Pesanan ini memiliki komponen item, waktu tiba pesan serta lokasi pick up dan dropoff.

3.7 ADT Stack

ADT Stack merupakan *array* statis yang akses elemennya mengikuti aturan *last in, first out* (LIFO). Struktur data ini dipilih untuk merepresentasikan tas sebagai tempat penyimpanan item yang telah diambil. Implementasi ADT ini meliputi konstruktor, operasi pemeriksaan, operasi manipulasi elemen, dan operasi manipulasi kapasitas yang dapat ditemukan pada *file stack.c*. ADT ini berisi *array of* pesan yang terdapat pada ADT Queue.

3.8 ADT Linked List

ADT Linked List merupakan *array* dinamis yang struktur elemennya merupakan *node*. Struktur data ini dipilih untuk merepresentasikan daftar pesan *to do* dan *in progress*. Implementasi ADT ini meliputi konstruktor, operasi pemeriksaan, interaksi *input/output*, dan operasi manipulasi elemen yang dapat ditemukan pada *file list_linked.c*.

3.9 ADT Game

ADT Game merupakan gabungan dari banyak adt lainnya dan berisi komponen yang sangat penting dalam game. ADT Game berisi fungsi fungsi yang nantinya akan dijalankan pada program utama. Implementasinya meliputi konstruktor, operasi pemeriksaan, interaksi *input/output*, operasi manipulasi pointer dan elemen komponen game. ADT ini dapat ditemukan pada *file game.c*. Pada ADT Game ini juga ada fungsi yang menarik yaitu *save/load game* yang mengutilisasikan mesin kata dengan lebih lanjut lagi.

3.10 ADT Utility

ADT Utility bertujuan sebagai ADT yang berisi fungsi pembantu yang nantinya akan digunakan di berbagai ADT Lainnya maupun program utama. ADT ini meliputi *input file*, *start game*, *help* dan lainnya. Implementasi ADT ini kebanyakan menggunakan mesin kata karena membaca file.

3.11 ADT Item

ADT Item merupakan ADT yang mendeskripsikan item dengan tipe, deskripsinya, waktu expirednya apabila dia perishable, dan harga item tersebut. ADT ini diimplementasikan dengan tipe *char*, *string* dan *integer*. Terdapat konstruktor dan operasi operasi lainnya yang dapat diaplikasikan kepada adt ini seperti *displayItem*, *isItemEqual* serta *minusExpiry*.

3.12 ADT Gadget

ADT Gadget merupakan dasar dari ADT Listpos. ADT Listpos merupakan ADT yang memiliki *array of* gadget yang nantinya akan dipakai pada *inventory* dan *toko*. Setiap gadget memiliki fungsi unik nya masing masing yang akan membuat game lebih terasa hidup.

3.13 ADT Ability

ADT Ability merupakan gabungan dari ADT Stack, ADT Pesanan, serta ADT Game. ADT ini bertujuan untuk membuat ability serta melakukan pengecekan terhadap ability ketika pemain melakukan dropoff pesanan, setiap pesanan memiliki efek uniknya masing masing ketika di dropoff. Pesanan dengan tipe Heavy akan memberikan efek speedboost yang membuat mobita bisa berjalan 2x lebih cepat dari sebelumnya, Pesanan dengan tipe Perishable akan memberikan efek increase capacity sehingga pemain dapat mengangkut lebih banyak pesanan dari sebelumnya. Terak

3.14 ADT Lokasi

ADT Lokasi merupakan ADT yang menggunakan ADT Point yang berbentuk kordinat dan diberi nama koor. Bedanya, pada ADT ini juga terdapat char tipeBangunan yang merupakan nama lokasi tersebut, seperti 'A', 'B', atau 'C'. Terdapat dua fungsi pada ADT ini, yaitu tulisLokasi untuk meng-output koor lokasi juga tipe dari bangunan lokasi tersebut dan fungsi isLocationEqual untuk mengecek apakah koor dari dua lokasi bernilai sama dan juga apakah tipeBangunan-nya sama. ADT ini akan dipakai untuk mencatat lokasi terkini pemain dan juga lokasi pickup dan dropoff tiap pesanan.

3.15 ADT PColor

ADT Pcolor merupakan ADT yang berfungsi untuk memberi warna untuk char. Terdapat 8 warna yang dapat dipanggil, yaitu warna normal (warna biasa pada command line), merah, kuning, hijau, biru, magenta, dan cyan. Terdapat beberapa fungsi untuk tiap warna di atas yang menerima char c dan akan meng-output char c tersebut dengan warna sesuai fungsinya. ADT ini nantinya akan berfungsi untuk memberi warna pada lokasi-lokasi pada peta sesuai dengan kondisinya, seperti apakah lokasi tersebut bisa dituju, ada barang yang bisa di-pick up, dan lainnya.

4 Program Utama

Pada aplikasi buatan kami, alur program utama dimulai dengan file utama main.c yang dijalankan dengan *batchfile* run.bat di sistem operasi Windows-based dan run.sh di sistem operasi UNIX-based. Program utama kemudian akan menampilkan menu untuk memulai game dengan pilihan New Game, Load Game, atau Exit.

Apabila memilih New Game, maka program akan meminta input nama file yang belum dikonfigurasi untuk game ini. Setelah diberikan file konfigurasi yang valid, maka game akan dimulai dengan kriteria: lokasi awal di HQ, uang awal 0 Yen, time awal 0, gadget kosong, ability tidak ada, dan tas kosong.

Apabila memilih Load Game, maka program akan meminta input nama file yang sudah dikonfigurasi sebelumnya. Setelah diberikan file konfigurasi yang valid, maka game akan dimulai dengan state-state yang ada saat penyimpanan terakhir ke dalam file tersebut. Setelah dilakukan New/Load, *game* akan melakukan *looping* dengan beberapa pilihan command:

- MOVE : Bergerak ke tempat lain
- PICK_UP : Untuk mengambil item yang ada di lokasi sekarang
- DROP_OFF : Untuk mengantarkan item ke lokasi jika item di tumpukan teratas tas sesuai dengan pesanan.

- MAP : Untuk memunculkan peta.
- TO_DO : Untuk menampilkan pesanan yang masuk ke dalam To Do List.
- IN_PROGRESS : Untuk menampilkan pesanan yang sedang diantar
- BUY : Untuk membeli gadget (Hanya dapat dilakukan di HQ)
- INVENTORY : Untuk menampilkan isi inventory.
- HELP : Menampilkan menu Help yang berisi daftar command.
- SAVE_GAME : Untuk menyimpan state game saat ini.
- QUIT : Keluar dari game.

Selain untuk melakukan command, *loop* program utama juga akan memunculkan posisi terkini dari mobita, uang dari mobita, dan waktu dalam game yang telah berjalan pada setiap iterasinya. Dilakukan juga pengecekan terhadap pesanan-pesanan yang ada pada file konfigurasi, apabila waktu muncul pesanan telah lewat maka akan dimasukkan ke dalam in progress list.

Game akan berakhir ketika sudah memenuhi kriteria endGame, yaitu ketika semua pesanan sudah selesai diantar dan Mobita berada di Headquarters.

5 Algoritma-Algoritma Menarik

5.1 Mekanisme Mesin Kata

Dikarenakan penggunaan string dan fungsi scanf dibatasi, kami menggunakan modifikasi dari mesin kata dan mesin karakter untuk penanganan input dari user. Kami membuat ADT Word yang berada di dalam mesin kata untuk menjadi pengganti string pada program kami, lengkap dengan beberapa primitif yang berguna seperti concat, equals, dan lain sebagainya.

Adapun poin menariknya adalah alur dari penanganan input itu sendiri, yang diawali oleh fungsi startWord yang meminta masukan dari user lalu di-copy ke dalam currentWord. CurrentWord inilah yang nantinya akan diolah oleh program. Selain itu, terdapat juga modifikasi sedikit untuk menangani input dari file, yaitu startFromFile.

Untuk menangani save/load, kami juga menggunakan mesin kata untuk load sebagaimana load config default, namun untuk save kami menggunakan salah satu fungsi di bahasa C yaitu fopen.

6 Data Test

6.1 Data Test NEW_GAME/LOAD_GAME

Saat memulai permainan, pemain dapat memilih untuk memulai permainan baru, melanjutkan permainan sebelumnya, atau keluar dari game. Berikut tampilan jika pemain memanggil command NEW_GAME atau LOAD_GAME.


```
=====THE GAME=====
1. NEW GAME
2. LOAD GAME
3. EXIT
NEW GAME
Selamat datang di Mobilita!
Silahkan masukkan nama config file:
config.txt
=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 0
Uang yang dimiliki: 0 Yen
ENTER COMMAND: █
```

Gambar 6.1.1 Tampilan saat game baru dimulai

```
=====THE GAME=====
1. NEW GAME
2. LOAD GAME
3. EXIT
LOAD GAME
Selamat datang di Mobilita!
Silahkan masukkan nama load file:
1241.txt
=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 46
Uang yang dimiliki: 400 Yen
```

Gambar 6.1.2 Tampilan saat load game

6.2 Data Test MOVE

Pemain dapat berpindah ke bangunan terdekat menurut *adjacency matrix* di config file menggunakan command MOVE. Pemain akan berpindah dan waktu akan bertambah apabila input posisi yang diberikan dapat dicapai (Gambar 6.2.1). Pemain dapat melakukan input 0 apabila tidak ingin berpindah (Gambar 6.2.2). Program akan menampilkan pesan error apabila input tidak valid (Gambar 6.2.3).

```

=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 0
Uang yang dimiliki: 0 Yen
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: 1
=====MENU=====
Mobita sekarang berada di titik C (1,9)
Waktu: 1
Uang yang dimiliki: 0 Yen

```

Gambar 6.2.1 Tampilan ketika berhasil MOVE

```

=====MENU=====
Mobita sekarang berada di titik G (3,8)
Waktu: 2
Uang yang dimiliki: 0 Yen
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: 0
=====MENU=====
Mobita sekarang berada di titik G (3,8)
Waktu: 2
Uang yang dimiliki: 0 Yen

```

Gambar 6.2.2 Tampilan ketika batal melakukan MOVE

```

=====MENU=====
Mobita sekarang berada di titik C (1,9)
Waktu: 1
Uang yang dimiliki: 0 Yen
ENTER COMMAND: MOVE
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: 5
Input tidak valid. Mengembalikan ke menu utama.
=====MENU=====
Mobita sekarang berada di titik C (1,9)
Waktu: 1
Uang yang dimiliki: 0 Yen

```

Gambar 6.2.2 Tampilan ketika input posisi tidak valid

6.3 Data Test PICK_UP

Pemain dapat memanggil command PICK_UP saat ingin mengambil pesanan. Command ini hanya bekerja jika pemain berada pada bangunan yang memiliki pesanan, tas tidak penuh, dan pemain tidak sedang membawa VIP item. Jika pemain berhasil mengambil pesanan, akan ditampilkan tampilan seperti pada gambar 6.3.1. Program akan menampilkan pesan khusus, jika salah satu dari ketiga syarat diatas tidak terpenuhi (lihat gambar 6.3.2, gambar 6.3.3, gambar 6.3.4)

```
ENTER COMMAND: 3
=====MENU=====
Mobita sekarang berada di titik G (3,8)
Waktu: 2
Uang yang dimiliki: 0 Yen
ENTER COMMAND: PICK_UP
Pesanan Normal berhasil diambil.
```

Gambar 6.3.1 Tampilan ketika pesanan berhasil diambil

```
ENTER COMMAND: 4
=====MENU=====
Mobita sekarang berada di titik I (4,5)
Waktu: 3
Uang yang dimiliki: 0 Yen
ENTER COMMAND: PICK_UP
Tidak ada item pada bangunan ini.
```

Gambar 6.3.2 Tampilan ketika memanggil command PICK_UP pada bangunan yang tidak tepat

```
ENTER COMMAND: 4
=====MENU=====
Mobita sekarang berada di titik Q (10,3)
Waktu: 91
Uang yang dimiliki: 800 Yen
ENTER COMMAND: PICK_UP
Antarkan dulu VIP Itemnya!
```

Gambar 6.3.3 Tampilan ketika memanggil command PICK_UP saat ada VIP item pada tas

```
ENTER COMMAND: 3
=====MENU=====
Mobita sekarang berada di titik G (3,8)
Waktu: 10
Uang yang dimiliki: 0 Yen
ENTER COMMAND: PICK_UP
Tas sudah penuh tidak bisa mengambil barang lagi.
=====MENU=====
```

Gambar 6.3.4 Tampilan ketika memanggil command PICK_UP saat tas penuh

6.4 Data Test DROP_OFF

Pemain dapat mengantarkan pesanan teratas pada tas di lokasi tujuan dengan menggunakan command DROP_OFF dan pemain akan mendapatkan reward sesuai item yang diantarkan (Gambar 6.4.1). Pemain tidak dapat mengantarkan pesanan apabila tidak berada di lokasi tujuan (Gambar 6.4.2) atau kondisi tas kosong/tidak ada pesanan (Gambar 6.4.3).

```

ENTER COMMAND: 4
=====MENU=====
Mobita sekarang berada di titik N (8,6)
Waktu: 6
Uang yang dimiliki: 0 Yen
ENTER COMMAND: DROP_OFF
Pesanan Normal berhasil diantarkan.
Uang yang didapatkan: 200 Yen
=====MENU=====

```

Gambar 6.4.1 Tampilan ketika berhasil DROP_OFF

```

=====MENU=====
Mobita sekarang berada di titik I (4,5)
Waktu: 3
Uang yang dimiliki: 0 Yen
ENTER COMMAND: DROP_OFF
Bukan lokasi yang tepat!
=====MENU=====

```

Gambar 6.4.2 Tampilan ketika lokasi DROP_OFF tidak sesuai pesanan

```

=====MENU=====
Mobita sekarang berada di titik N (8,6)
Waktu: 6
Uang yang dimiliki: 200 Yen
ENTER COMMAND: DROP_OFF
Tidak ada pesanan yang dapat diantarkan!
=====MENU=====

```

Gambar 6.4.3 Tampilan ketika DROP_OFF saat tas kosong/tidak ada pesanan

6.5 Data Test MAP

Untuk melihat peta saat ini, pemain dapat memanggil command MAP, program akan menampilkan warna yang berbeda untuk beberapa bangunan. Bangunan dengan warna merah menunjukkan ada pesanan yang dapat di ambil pada bangunan tersebut. Warna hijau menunjukkan lokasi mana yang dapat dituju pemain, Warna kuning menunjukkan lokasi pemain saat ini, sedangkan warna biru menunjukkan lokasi tujuan pengantaran item teratas. Tampilan map akan berbeda-beda tergantung lokasi pemain saat ini.

```

=====MENU=====
Mobita sekarang berada di titik N (8,6)
Waktu: 6
Uang yang dimiliki: 200 Yen
ENTER COMMAND: MAP
* * * * *
*8      C  D B*
*  E      *
*F      G  H *
*  I      *
*      J  *
*  K      *
*      L  *
* M  N      O*
*      P  *
*A Q      *
* * * * *

```

Gambar 6.5.1 Tampilan map saat pemain berada pada titik N

```

=====MENU=====
Mobita sekarang berada di titik M (8,2)
Waktu: 7
Uang yang dimiliki: 200 Yen
ENTER COMMAND: MAP
* * * * *
*8      C  D B*
*  E      *
*F      G  H *
*  I      *
*      J  *
*  K      *
*      L  *
* M  N      O*
*      P  *
*A Q      *
* * * * *

```

Gambar 6.5.2 Tampilan map saat pemain berada pada titik M dan tujuan pesanan pada titik B

6.6 Data Test TO_DO

Tampilan seperti pada gambar 6.6.1 akan ditampilkan jika pemain memanggil command TO_DO. Program akan menampilkan daftar item yang dapat di pick-up oleh pemain.

```

=====MENU=====
Mobita sekarang berada di titik M (8,2)
Waktu: 7
Uang yang dimiliki: 200 Yen
ENTER COMMAND: TO_DO
Pesanan pada To Do List:
1. B -> M (Normal Item)
2. G -> N (Heavy Item)
3. F -> E (Normal Item)
4. C -> D (Normal Item)
5. G -> N (Perishable Item, sisa waktu 10)

```

Gambar 6.6.1 Tampilan program saat memanggil command TO_DO

6.7 Data Test IN_PROGRESS

Command IN_PROGRESS akan menampilkan daftar pesanan yang ada pada tas saat ini.

```

=====MENU=====
Mobita sekarang berada di titik M (8,2)
Waktu: 7
Uang yang dimiliki: 200 Yen
ENTER COMMAND: IN_PROGRESS
Pesanan yang sedang diantarkan:
1. Heavy Item (Tujuan: B)

```

Gambar 6.7.1 Tampilan program ketika memanggil command IN_PROGRESS

6.8 Data Test BUY

Pada tes ini, akan dicoba membeli gadget dari toko. Command BUY hanya bisa dilakukan jika pemain berada pada HQ. Jika pemain tidak berada pada HQ, program akan menampilkan pesan seperti pada gambar 6.8.1.

```

=====MENU=====
Mobita sekarang berada di titik M (8,2)
Waktu: 7
Uang yang dimiliki: 200 Yen
ENTER COMMAND: BUY
Tidak bisa membeli item karena tidak di HQ.

```

Gambar 6.8.1 Tampilan jika memanggil command BUY saat pemain tidak berada di HQ

Apabila pemain sedang berada pada titik HQ, toko dapat diakses, kemudian pemain dapat memilih gadget yang ingin dibeli. Jika uang pemain tidak cukup untuk membeli gadget, akan keluar pesan seperti pada gambar 6.8.2. Apabila pemain berhasil membeli gadget, akan keluar pesan seperti pada gambar 6.8.3


```

=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 23
Uang yang dimiliki: 200 Yen
ENTER COMMAND: BUY
Uang anda sekarang: 200 Yen
Gadget yang tersedia:
1. Kain Pembungkus Waktu (800 Yen)
2. Senter Pembesar (1200 Yen)
3. Pintu Kemana Saja (1500 Yen)
4. Mesin Waktu (3000 Yen)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin Anda beli? (Ketik 0 jika ingin kembali)

ENTER COMMAND: 1
Uang Anda tidak cukup untuk membeli gadget!

```

Gambar 6.8.2 Tampilan ketika uang tidak cukup untuk membeli gadget

```

=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 63
Uang yang dimiliki: 53200 Yen
ENTER COMMAND: BUY
Uang anda sekarang: 53200 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)
5. Senter Pengecil (800 Yen)
Gadget mana yang ingin Anda beli? (Ketik 0 jika ingin kembali)

ENTER COMMAND: 1
Kain Pembungkus Waktu berhasil dibeli!
Uang Anda sekarang: 52400 Yen

```

Gambar 6.8.3 Tampilan ketika pemain berhasil membeli gadget

6.9 Data Test INVENTORY

Jika memanggil command INVENTORY akan muncul tampilan program seperti pada gambar 6.9.1, pemain dapat memilih gadget apa yang ingin digunakan, atau dapat memasukkan 0 jika ingin kembali dan batal menggunakan gadget.

```

Uang Anda sekarang: 52400 Yen
=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 63
Uang yang dimiliki: 52400 Yen
ENTER COMMAND: INVENTORY
1. Pintu Kemana Saja
2. Pintu Kemana Saja
3. Mesin Waktu
4. Kain Pembungkus Waktu
5. Kain Pembungkus Waktu
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 

```

Gambar 6.9.1 Tampilan saat command INVENTORY dipanggil

Jika pemain berhasil menggunakan gadget, akan muncul pesan seperti pada gambar 6.9.2, gambar 6.9.3, gambar 6.9.4, dan gambar 6.9.6 bergantung pada jenis gadget yang digunakan pemain. Gambar 6.9.7 akan ditampilkan jika pemain memilih slot kosong pada inventory untuk digunakan.

```
=====MENU=====
Mobita sekarang berada di titik Q (10,3)
Waktu: 13
Uang yang dimiliki: 52400 Yen
ENTER COMMAND: INVENTORY
1. -
2. -
3. -
4. Kain Pembungkus Waktu
5. Kain Pembungkus Waktu
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 4
Kain pembungkus waktu berhasil digunakan!
```

Gambar 6.9.2 Tampilan saat kain pembungkus waktu berhasil digunakan

```
=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 61
Uang yang dimiliki: 52000 Yen
ENTER COMMAND: INVENTORY
1. Pintu Kemana Saja
2. Pintu Kemana Saja
3. Mesin Waktu
4. Kain Pembungkus Waktu
5. Senter Pembesar
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 5
Senter Pembesar berhasil digunakan!
Kapasitas tas saat ini: 6
```

Gambar 6.9.3 Tampilan saat senter pembesar berhasil digunakan

```
=====MENU=====
Mobita sekarang berada di titik Q (10,3)
Waktu: 63
Uang yang dimiliki: 52400 Yen
ENTER COMMAND: INVENTORY
1. -
2. -
3. Mesin Waktu
4. Kain Pembungkus Waktu
5. Kain Pembungkus Waktu
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 3
Mesin Waktu berhasil digunakan!
=====MENU=====
Mobita sekarang berada di titik Q (10,3)
Waktu: 13
Uang yang dimiliki: 52400 Yen
```

Gambar 6.9.4 Tampilan saat mesin waktu berhasil digunakan


```

Mobita sekarang berada di titik 8 (1,1)
Waktu: 63
Uang yang dimiliki: 52400 Yen
ENTER COMMAND: INVENTORY
1. -
2. Pintu Kemana Saja
3. Mesin Waktu
4. Kain Pembungkus Waktu
5. Kain Pembungkus Waktu
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 2
Posisi yang dapat dicapai:
1. 8 (1,1)
2. A (10,1)
3. B (1,15)
4. C (1,9)
5. D (1,13)
6. E (2,3)
7. F (3,1)
8. G (3,8)
9. H (3,14)
10. I (4,5)
11. J (5,12)
12. K (6,3)
13. L (7,10)
14. M (8,2)
15. N (8,6)
16. O (8,15)
17. P (9,13)
18. Q (10,3)
Masukkan lokasi yang ingin dituju: 18
Pintu Kemana Saja berhasil digunakan!
=====MENU=====
Mobita sekarang berada di titik Q (10,3)

```

Gambar 6.9.5 Tampilan saat pintu kemana saja berhasil digunakan

```

=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 26
Uang yang dimiliki: 200 Yen
ENTER COMMAND: INVENTORY
1. Senter Pengecil
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 1
Senter pengecil berhasil digunakan

```

Gambar 6.9.6 Tampilan saat senter pengecil berhasil digunakan

```

=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 38
Uang yang dimiliki: 200 Yen
ENTER COMMAND: INVENTORY
1. Kain Pembungkus Waktu
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (ketik 0 jika ingin kembali)

ENTER COMMAND: 2
Tidak ada gadget yang dapat digunakan!

```

Gambar 6.9.7 Tampilan saat ingin menggunakan gadget pada slot kosong inventory

6.10 Data Test HELP

Saat dipanggil, command HELP akan menampilkan informasi mengenai fitur-fitur lainnya. Command HELP dapat dipanggil kapan pun saat permainan berlangsung.

```

=====MENU=====
Mobita sekarang berada di titik 8 (1,1)
Waktu: 0
Uang yang dimiliki: 0 Yen
ENTER COMMAND: HELP
=====
==+ HELP MENU +==
=====
1. MOVE -> Untuk berpindah ke lokasi lainnya
2. PICK_UP -> Untuk mengambil item yang ada di lokasi sekarang
3. DROP_OFF -> Untuk mengantarkan item ke lokasi jika item di tumpukan teratas tas sesuai dengan pesanan.
4. MAP -> Untuk memunculkan peta.
5. TO_DO -> Untuk menampilkan pesanan yang masuk ke dalam To Do List
6. IN_PROGRESS -> Untuk menampilkan pesanan yang sedang diantar
7. BUY -> Untuk membeli gadget (Hanya dapat dilakukan di HQ)
8. INVENTORY -> Untuk menampilkan isi inventory gadget dan menggunakannya.
9. HELP -> Menampilkan menu Help yang berisi daftar command.
10. SAVE_GAME -> Menyimpan state game terkini.
11. QUIT -> Keluar dari game.

```

Gambar 6.10.1 Tampilan saat command HELP dipanggil

6.11 Data Test RETURN

Ability return to sender hanya dapat digunakan pada item selain VIP item. Saat dipanggil, program akan mengembalikan item paling atas pada tas ke daftar to do paling akhir. Berikut adalah tampilan jika pemain mencoba menggunakan command RETURN saat membawa VIP item.

```

Bukan lokasi yang tepat.
=====MENU=====
Mobita sekarang berada di titik Q (10,3)
Waktu: 91
Uang yang dimiliki: 800 Yen
ENTER COMMAND: RETURN
Mobita tidak bisa memakai Ability Return To Sender!
=====MENU=====

```

Gambar 6.11.1 Tampilan saat gagal menggunakan ability return to sender

```

=====MENU=====
Mobita sekarang berada di titik P (9,13)
Waktu: 14
Uang yang dimiliki: 600 Yen
ENTER COMMAND: RETURN
Perishable berhasil dikembalikan ke Pick Up Point Q
=====MENU=====

```

Gambar 6.11.2 Tampilan saat berhasil menggunakan ability return to sender

6.12 Data Test SAVE

Command SAVE_GAME dapat dipanggil untuk menyimpan data riwayat permainan. Pemain dapat melanjutkan permainan dengan memanggil command LOAD_GAME nantinya, jika data riwayat permainan berhasil disimpan.

```

ENTER COMMAND: SAVE_GAME
Masukkan nama file penyimpanan: test.txt
Game successfully saved.

```

Gambar 6.12.1 Tampilan saat command SAVE_GAME dipanggil

6.13 Data Test QUIT

Jika pemain ingin berhenti bermain, pemain dapat memanggil command QUIT, program kemudian akan menanyakan apakah pemain ingin menyimpan riwayat permainannya. Jika pemain ingin menyimpan riwayat permainannya, program akan menampilkan tampilan seperti pada gambar 6.13.2. Tampilan seperti pada gambar 6.13.1 akan ditampilkan jika pemain tidak ingin menyimpan riwayat permainannya.

```

ENTER COMMAND: QUIT
Apakah anda ingin menyimpan permainan sebelum quit? (Y/N): N
Sampai jumpa di lain waktu!

```

Gambar 6.13.1 Tampilan saat pemain ingin berhenti dari permainan dan tidak ingin menyimpan riwayat permainannya

```

Apakah anda ingin menyimpan permainan sebelum quit? (Y/N): Y
Masukkan nama file penyimpanan: test.txt
Game successfully saved.
Sampai jumpa di lain waktu!

```

Gambar 6.13.2 Tampilan saat pemain ingin berhenti dari permainan dan ingin menyimpan riwayat permainannya

6.14 Data Test ENDGAME

Apabila pemain telah berhasil menyelesaikan permainan dengan berhasil mengantarkan semua pesanan dan/atau semua item perishable sudah expired atau sudah berhasil diantar, dan mobita sudah kembali ke headquarter, maka program akan mengeksekusi tampilan apabila sudah mencapai akhir dari game. Pengeksekusian ini berjalan secara otomatis ketika kondisi telah memenuhi, tanpa harus melakukan command.

SELAMAT! Anda berhasil menamatkan game MOBILITA!
 Dengan ini, kamu berhasil pensiun dari pekerjaanmu sebagai kurir!
 Pandemi telah usai, mobita pun kembali masuk kampus..

Finished stats:
 Pesanan yang telah diantar: 10
 Waktu yang terlampau: 41
 Total uang yang didapat: 3600
 Sampai jumpa di lain waktu!

Gambar 6.14.1 Tampilan saat pemain sudah berhasil menyelesaikan game

7 Test Script

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	NEW_GAME	Memeriksa apakah game bisa dimulai dari kondisi awal	Ketik NEW GAME pada awal mula permainan	NEW GAME	Game berhasil dijalankan dengan konfigurasi config.txt	Gambar 6.1.1
2	LOAD_GAME	Memeriksa apakah game bisa dimulai dari state yang sudah pernah disimpan	Mengetik LOAD GAME pada awal mula permainan	LOAD GAME	Game berhasil dijalankan dengan hasil save file	Gambar 6.1.2
3	MOVE	Memeriksa apakah pemain bisa berpindah	Mengetik MOVE ketika permainan sudah di mulai lalu mengetik angka 1 untuk berpindah dari HQ ke posisi C	MOVE 1	Pemain berhasil pindah dari posisi HQ ke posisi C	Gambar 6.2.1
4	MOVE	Memeriksa apakah pemain bisa	Mengetik MOVE ketika	MOVE 0	Pemain berhasil kembali dari	Gambar 6.2.2

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		membatalkan command	permainan sedang berjalan lalu mengetik angka 0 untuk kembali		MOVE command ke menu utama	
5	MOVE	Memeriksa apakah pemain gagal berpindah jika input tidak valid	Mengetik MOVE ketika permainan sedang berlangsung lalu mengetik angka yang tidak ada pada list	MOVE 5	Pemain gagal berpindah lokasi	Gambar 6.2.3
6	PICK_UP	Memeriksa apakah pemain bisa mengambil pesanan	Mengetik PICK_UP pada posisi bangunan yang memiliki pesanan	PICK_UP	Pemain berhasil pick up pesanan dan pesanan dimasukkan ke in progress dan tas	Gambar 6.3.1
7	PICK_UP	Memeriksa apakah pemain gagal mengambil pesanan jika tidak berada di posisi yang tepat	Mengetik PICK_UP pada posisi bangunan yang tidak memiliki pesanan	PICK_UP	Pemain tidak berhasil pick up pesanan karena tidak ada pesanan yang bisa diambil	Gambar 6.3.2
8	PICK_UP	Memeriksa apakah pemain gagal mengambil pesanan jika ada VIP item di tas	Mengetik PICK_UP ketika pada tas terdapat VIP Item	PICK_UP	Pemain tidak berhasil pick up pesanan karena harus mengantarkan dulu VIP item terlebih dahulu	Gambar 6.3.3
9	PICK_UP	Memeriksa apakah	Mengetik PICK_UP	PICK_UP	Pemain tidak berhasil	Gambar 6.3.4

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		pemain gagal mengambil pesanan jika tas penuh	ketika tas sudah penuh		mengambil pesanan karena tas sudah penuh dan tidak bisa mengambil pesanan	
10	DROP_OFF	Memeriksa apakah pemain bisa mengantar pesanan	Mengetik DROP_OFF pada lokasi yang sesuai	DROP_OFF	Pemain berhasil mengembalikan pesanan dan mendapat ability serta uang yang sesuai dengan jumlahnya	Gambar 6.4.1
11	DROP_OFF	Memeriksa apakah pemain gagal mengantar pesanan jika tidak berada di lokasi yang tepat	Mengetik DROP_OFF pada lokasi yang tidak sesuai	DROP_OFF	Pemain tidak berhasil mengembalikan pesanan. Pesanan masih ada di dalam tas dan inprogress list	Gambar 6.4.2
12	DROP_OFF	Memeriksa apakah pemain gagal mengantar pesanan jika tas kosong	Mengetik DROP_OFF ketika tas kosong	DROP_OFF	Pemain tidak berhasil melakukan pemengembalian item karena pemain tidak memiliki item pada tas	Gambar 6.4.3
13	MAP	Memeriksa apakah program dapat menampilkan peta	Mengetik MAP pada sembarang lokasi untuk menampilkan peta	MAP	Map berhasil ditampilkan. Gambar 1 menunjukkan ketika kondisi tas kosong dan gambar kedua menunjukkan ketika ada	Gambar 6.5.1 Gambar 6.5.2

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
					barang yang harus diantarkan	
14	TO_DO	Memeriksa apakah program dapat menampilkan To Do List	Mengetikkan TO_DO	TO_DO	Program akan menampilkan daftar pesanan yang dapat diambil	Gambar 6.6.1
15	IN_PROGRES S	Memeriksa apakah program dapat menampilkan In Progress List	Mengetik IN_PROGRES S pada posisi sembarang	IN_PROGRES S	Program akan menampilkan daftar pesanan yang sedang aktif	Gambar 6.7.1
16	BUY	Memeriksa apakah pemain gagal membeli gadget jika tidak berada di HQ	Memanggil command BUY di sembarang lokasi	BUY	Pemain tidak dapat mengakses toko dikarenakan pemain tidak berada pada titik HQ	Gambar 6.8.1
17	BUY	Memeriksa apakah pemain gagal membeli gadget jika uang tidak cukup	Mengetik BUY saat pemain di HQ, memilih gadget yang lebih mahal dari pada uang yang tersedia	BUY	Gadget tidak dapat dibeli karena uang yang dimiliki pemain tidak cukup	Gambar 6.8.2
18	BUY	Memeriksa apakah pemain bisa membeli gadget	Mengetik BUY saat pemain di HQ, memilih gadget yang diinginkan	BUY	Gadget berhasil dibeli, uang berkurang sesuai harga gadget, gadget dimasukkan ke inventory	Gambar 6.8.3
19	INVENTORY	Memeriksa	Ketikkan	INVENTORY	Menampilkan	Gambar 6.9.1

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		apakah program dapat menampilkan inventory	INVENTOR Y		daftar gadget yang ada pada inventory	
20	INVENTORY	Memeriksa apakah pemain dapat menggunakan gadget yang ada di inventory	Ketikkan inventory, kemudian pilih gadget yang tersedia pada inventory (bukan slot kosong)	INVENTORY	Gadget berhasil digunakan sesuai dengan fungsinya	Gambar 6.9.2 Gambar 6.9.3 Gambar 6.9.4 Gambar 6.9.5 Gambar 6.9.6
21	INVENTORY	Memeriksa apakah pemain gagal menggunakan gadget jika memilih slot inventory kosong	Ketikkan INVENTOR Y, kemudian pilih nomor slot inventory yang kosong	INVENTORY	Tidak ada gadget yang dapat digunakan	Gambar 6.9.7
22	HELP	Memeriksa apakah program dapat menampilkan Help Menu	Ketikkan HELP di sembarang lokasi	HELP	Program menampilkan isi dari HELP	Gambar 6.10.1
23	RETURN	Memeriksa apakah pemain gagal menggunakan ability Return to Sender jika tidak memilikinya	Ketikkan RETURN	RETURN	Program akan menampilkan pesan khusus, item gagal dikembalikan ke pick up point dan daftar to do	Gambar 6.11.1
24	RETURN	Memeriksa apakah pemain bisa mengaktifkan ability Return	Ketikkan RETURN	RETURN	Item berhasil dikembalikan ke pick up point, item kembali ke	Gambar 6.11.2

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		to Sender			daftar to do di urutan terakhir	
25	SAVE_GAME	Memeriksa apakah pemain bisa menyimpan riwayat permainan ke file eksternal	Ketikkan SAVE_GAME di sembarang lokasi, ketikkan nama file tempat penyimpanan data permainan	SAVE_GAME test.txt	Data permainan berhasil disimpan pada file yang ditentukan	Gambar 6.12.1
26	QUIT	Memeriksa apakah pemain dapat keluar dari game tanpa menyimpan riwayat permainan	Ketikkan QUIT di sembarang lokasi, ketik N ketika program menanyakan apakah data permainan ingin disimpan	QUIT N	Pemain berhasil keluar dari permainan dengan data permainan tidak disimpan	Gambar 6.13.1
27	QUIT	Memeriksa apakah pemain dapat keluar dari game dengan menyimpan riwayat permainan terlebih dahulu	Ketikkan QUIT di sembarang lokasi, ketik Y ketika program menanyakan apakah data permainan ingin disimpan	QUIT Y test.txt	Pemain berhasil keluar dari permainan dengan data permainan tersimpan pada file yang ditentukan	Gambar 6.13.2
28	ENDGAME	Memeriksa apakah saat kondisi permainan selesai terpenuhi,	Mainkan game sampai kondisi endgame terpenuhi (semua	-	Pemain berhasil mengakhiri game dengan ditampilkannya <i>a statistics</i>	Gambar 6.14.1

No .	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		pemain dapat melihat tampilan akhir game	pesanan sudah selesai, mobita berposisi di headquarters.		akhir dari game.	

8 Pembagian Kerja dalam Kelompok

Nama (NIM)	Pembagian Tugas
Vionie Novencia Thanggestyo (13520006)	All of gadget implementation & Inventory gadget
Gagas Praharsa Bahar (13520016)	Start menu and main menu, new game, exit, load/save, mesin kata & karakter, move, map, help
Muhammad Garebaldhie Er Rahman (13520029)	Start game, game flow, pickup, dropoff, map, bag, item, ability, VIP Item, Return to sender, TO DO list, inprogress list
Taufan Fajarama Putrawansyah R (13520031)	Daftar Pesanan, TO DO list, inprogress list
Jeremy S.O.N. Simbolon (13520042)	Bag, Normal item, Heavy item, Perishable item.
Arik Rayi Arkananta (13520048)	Speed boost, increase capacity, map, dropoff, return to sender

9 Lampiran

9.1 Deskripsi Tugas Besar 2

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

Buatlah sebuah permainan berbasis **CLI** (command-line interface) tentang pengantaran barang. Permainan ini dibuat dalam **bahasa C** dengan menggunakan struktur data yang sudah

STEI- ITB	IF2110-TB-01-01	Halaman 26 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

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

Hari, Tanggal	NIM	Notulen Rapat
Sabtu, 23 Oktober 2021	13520006 13520016 13520029 13520031 13520042 13520048	Diskusi gambaran besar program Mengurai ADT dan fitur yang perlu dikerjakan Pembagian tugas awal sebelum asistensi
Kamis, 28 Oktober 2021	13520006 13520016 13520029 13520031 13520042 13520048	Asistensi I: notulen terlampir pada bagian 9.4 Meet kelompok: Menyusun rencana pengerjaan tugas besar
Senin, 1 November 2021	13520006 13520016 13520029 13520031 13520042 13520048	Progress pengerjaan tugas besar Membahas kendala dalam pengerjaan tugas besar
Selasa, 9 November 2021	13520006 13520016 13520029 13520031 13520042 13520048	Asistensi II: notulen terlampir pada bagian 9.4 Meet kelompok: Debugging Testing flow program Membahas fitur yang belum selesai
Rabu, 17 November 2021	13520006 13520016 13520029 13520031 13520042 13520048	Debugging Gladi bersih demo Finishing kode dan laporan

9.3 Log Activity Anggota Kelompok

Vionie Novencia <13520006>:

5 November: added inventory and gadget

6 November: Edited item, edited stack, edited gadget

8 November 2021: added gadget_driver.c

15 November: fixed gadget, edited gadget

18 November: fixed gadget, finishing laporan.

Gagas Praharsa Bahar <13520016>:

STEI- ITB	IF2110-TB-01-01	Halaman 27 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

23 Oktober: adding placeholders
 28 Oktober: Adding start menu
 29 Oktober: validasi input
 30 Oktober: moved c and h to lib and separated the folders, merapihkan lib
 31 Oktober: repaired dependencies and duplicate function names
 1 November: small fix in headers, edit menu
 2 November: input from file (to ADT Game)
 4 November: made mainmenu and little fix
 5 November: help, move function, comment. Changed bangunan list from static to dynamic, delete vector and upgrade map.
 10 November: fix dependencies
 16 November: add pintu kemana saja
 17 November: made save function and load function, update todo and merge branch
 18 November: adding endgame and quit function, fix load game and adding some decoration, finishing laporan.

Muhammad Garebaldhie <13520029>:

21 Oktober: Initial Commit
 23 Oktober: adding headerfile, adding pcolor, adding doc folder and files; adding c files and correct path and merge.
 27 Oktober: adding linked list, made some changes in point and time adt. Adding game.h and game.c
 30 Oktober: several changes and merge
 31 Oktober: fix some changes, delete adt time, and adding chanes in adt peta
 1 November: finished adt peta
 5 November: Updated TODO, Main.c, spesifikasi, game, ability, stackdriver, implement in progress and to do list
 13 November: fixed circular import, utility, tempdriver and main
 14 November: peta sync pickup, pickup, fix return to sender, finished pick_up, fixed dropoff and working map, fixed perishable item.
 15 November: adding driver to adt
 16 November: fixed ability, integrated gadget with main.
 17 November: fixed minor bugs and adding support for UNIX
 18 November: finishing laporan

Taufan Fajarrama Putrawansyah R <13520031>:

16 November: update readme.md
 17 November: test driver & save game and laporan
 18 November: Laporan fix

Jeremy S.O.N Simbolon <13520042>:

3 November: implement stack capacity and operation, change ElType to item, implement item data type.
 4 November: Refactor item ADT
 17 November : laporan.

STEI- ITB	<i>IF2110-TB-01-01</i>	Halaman 28 dari 32 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

Arik Rayi Arkananta <13520048>:

10 November: edit ability.c and ability.h

14 November: edit prioqueue.c, update stack.h, add dropoff, activate ability, add return to sender, update todo and fixed return to sender

17 November: Updated todo and laporan

18 November: finishing laporan.

9.4 Form Asistensi Tugas Besar

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

No. Kelompok/Kelas : 01/K01
Nama Kelompok :
Anggota Kelompok (Nama/NIM) :
1. Vionie Novencia Thanggestyo/13520006
2. Gagas Praharsa Bahar/13520016
3. Muhammad Garebaldhie E R/13520029
4. Taufan Fajarama P R/13520031
5. Jeremy S O N Simbolon/13520042
6. Arik Rayi Arkananta/13520048

Asisten Pembimbing : Michel Fang/13518137

Asistensi I

Tanggal : Kamis, 28 Oktober 2021

Tempat : Google Meet

Kehadiran Anggota Kelompok:

No
NIM
Tanda tangan

1



2



3



4



5



6









- Catatan Asistensi: Mesin kata dimodif biar nerima newline
- Laporan gaada notal
- ADT dipisah perfolder aja
- Boleh bikin tambahan asal ga dilarang di spek
- Ga boleh freestyle, ADT hrs dipake sesuai spek, kalo ga nilai ADT tsb ngilang
- Pake makefile dan tes compile dulu
- Bagi tugas per sistem aja jangan per ADT, karena ada beberapa sistem yang pake beberapa matriks.
- Invite kakaknya ke github biar kali aja bisa dicek sekilas :D

Tanda Tangan Asisten:



Asistensi II

Tanggal :9 November 2021	Catatan Asistensi: <ul style="list-style-type: none"> - Progress udah lumayan tp masih banyak yang harus dikejar - Next coba dikerjain aja biar keliatan dimana salahnya - Struktur informasi dibebaskan - Sabar ya tubesnya banyak
Tempat :Google Meet	
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1  2  3  4  5  6 	
	Tanda Tangan Asisten: 