

# **LAPORAN TUGAS BESAR**

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


### **MOBILITA**

Dipersiapkan oleh:

Kelompok 5 / K02

Eiffel Aqila Amarendra	13520074
Claudia	13520076
Ghebyon Tohada Nainggolan	13520079
Sarah Azka Arief	13520083
Firizky Ardiansyah	13520095

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

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>IF2110-TB-05-02</i>		<i>42</i>
		<i>Revisi</i>	<i>1</i>	<i>18/11/2021</i>

# Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas .....	3
2.1 Item.....	3
2.2 Return To Sender Ability .....	4
2.1 Map.....	4
2.2 Gadget .....	4
2.3 Inventory Gadget.....	4
2.4 Save/Load Game .....	4
3 Struktur Data (ADT) .....	5
3.1 ADT Point .....	5
3.2 ADT List Statis.....	6
3.3 ADT List Dinamis .....	7
3.4 ADT Matriks .....	8
3.5 ADT Mesin Karakter dan Mesin Kata .....	9
3.6 ADT Queue (Priority Queue).....	11
3.7 ADT Stack.....	12
3.8 ADT Linked List .....	12
3.9 ADT Pesanan.....	13
3.10 ADT Peta .....	14
4 Program Utama .....	15
5 Data Test .....	15
5.1 Data Test 1 .....	16
5.2 Data Test 2 .....	17
5.3 Data Test 3 .....	19
5.4 Data Test 4 .....	20
5.5 Data Test 5 .....	20
5.6 Data Test 6 .....	21
5.7 Data Test 7 .....	22
5.8 Data Test 8 .....	23
5.9 Data Test 9 .....	26
5.10 Data Test 10.....	27
5.11 Data Test 11 .....	27
5.12 Data Test 12.....	30
5.13 Data Test 13.....	30
6 Test Script .....	31
7 Pembagian Kerja dalam Kelompok .....	36
8 Lampiran .....	37
8.1 Deskripsi Tugas Besar 2.....	37
8.2 Notulen Rapat.....	37
8.3 Log Activity Anggota Kelompok.....	38
8.4 Form Asistensi Tugas Besar.....	40

# 1 Ringkasan

Pada tugas besar IF2110 Algoritma dan Struktur Data ini, kelompok kami mendapat tugas untuk merealisasikan permainan Mobilita. Mobilita merupakan *simulation game* dimana sang pemain yang dalam permainan ini adalah Mobita berperan sebagai seorang kurir yang bertugas mengantarkan barang-barang pesanan. Sebagai seorang kurir, pemain akan dibekali aplikasi untuk melacak pesanan, navigasi, mengambil dan menurunkan barang, hingga berbagai macam ability dan gadget untuk membantunya mengantarkan barang-barang pesanan tersebut.

Berdasarkan spesifikasi, kami memperoleh tugas untuk merealisasikan aplikasi (permainan) Mobilita dengan berbasis CLI (*command-line interface*) dan empat belas buah *Game Mechanics*, yakni main menu, mekanisme waktu, daftar pesanan, to do list, in progress list, tas, jenis item, ability, gadget, inventory gadget, peta, lokasi dan command, game flow, dan save/load game. Aplikasi ini dibuat dalam bahasa pemrograman C dengan memanfaatkan berbagai struktur data (ADT) yang sudah dipelajari di mata kuliah IF2110 Algoritma dan Struktur Data beserta beberapa modifikasi/tambahan, di antaranya ADT Point, ADT List Statis, ADT List Dinamis, ADT Matriks, ADT Mesin Karakter dan Mesin Kata, ADT Queue (Priority Queue), ADT Stack, ADT Linked List, ADT Pesanan, dan ADT Peta.

Laporan berisi ringkasan laporan secara keseluruhan, penjelasan tambahan mengenai spesifikasi tugas besar, struktur data (ADT) yang digunakan, penjelasan mengenai program utama, data-data dan script uji, pembagian kerja dalam kelompok, dan lampiran yang berisi deskripsi tugas besar, notulen rapat, log activity kelompok, dan lampiran tambahan lainnya.

Permainan akan dimulai dengan menampilkan main menu yang berisi pilihan terhadap player untuk NEW\_GAME, LOAD\_GAME, atau EXIT. Kemudian, ketika permainan dimulai, pemain akan memulai permainan di Headquarter, mendapatkan jumlah pesanan yang harus diselesaikan, dan mulai mendapatkan pesanan yang masuk ke To Do List. Untuk mencapai tujuan dari permainan ini, yakni menyelesaikan semua pesanan dengan waktu secepat mungkin, pemain dapat memanfaatkan command-command yang tersedia dalam permainan ini, yakni MOVE, PICK\_UP, DROP\_OFF, MAP, TO\_DO, IN\_PROGRESS, BUY, INVENTORY, RETURN, HELP, SAVE, dan EXIT. Permainan akan berakhir ketika pemain berhasil menyelesaikan semua pesanan dan telah kembali ke Headquarters.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Item

- Perishable item yang sudah melewati batas waktu akan dihapus dari In Progress List dan tidak bisa dikerjakan kembali. Oleh karena itu, Perishable Item tidak akan masuk ke dalam To Do List lagi.
- Apabila pada To Do List sudah tampil VIP item, maka player tidak bisa pick up jenis item selain VIP item. Dengan kata lain, player masih bisa melakukan pick up pada item lain dengan jenis yang sama, yaitu VIP Item(apabila di To Do List sudah dimunculkan kembali). Diimplementasikan dengan membuat variabel yang menghitung ada berapa VIP item dalam

STEI- ITB	IF2110-TB-05-02	Halaman 3 dari 42 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.		

bag. Apabila variabel tersebut bernilai lebih dari 0, maka PICK\_UP untuk jenis item lain akan di *disable*

## 2.2 Return To Sender Ability

- *Return To Sender Ability* merupakan ability yang memiliki kuantitas. Dengan kata lain, ability ini dapat dimiliki lebih dari sekali oleh player. Contoh kasus, player sudah melakukan DROP\_OFF pada beberapa VIP Item namun belum menggunakan RETURN sama sekali.

## 2.1 Map

- Map adalah leburan antara ADT matriks dan ADT list dinamis (yang menyimpan building). Pada modul ini, dibangun sebuah struktur data berisi matriks dari karakter yang digunakan untuk memperlihatkan peta pada pemain. Modul ini sangat erat kaitannya dengan list dinamis, sebab sangat berhubungan dengan struktur lokasi yang abstraksinya ada pada ADT list dinamis. Inisialisasi struktur data ini sama persis dengan inisialisasi pada matriks biasa, bedanya elemen yang disimpan bertipe data karakter dan elemennya ditentukan berdasarkan objek yang berada pada indeks matriks ini.

## 2.2 Gadget

- Apabila kain pembungkus waktu digunakan, namun item paling atas pada bag tidak berjenis perishable item, maka gadget akan dihapus dan terpakai sia-sia
- Apabila senter pengecil digunakan, namun item paling atas pada bag tidak berjenis heavy item, maka gadget akan dihapus dan terpakai sia-sia

## 2.3 Inventory Gadget

- Inventory berkapasitas 5 dan dapat diisi oleh jenis gadget yang sama.

## 2.4 Save/Load Game

- Save dapat dilakukan kapanpun saat game berlangsung dan akan secara otomatis disarankan pada pemain sebelum keluar dari program saat melakukan in-game EXIT. Save dilakukan dengan menerima input berupa nama file yang akan ditulis dengan progress pemain. Setelah menerima nama file, proses save dilanjut dengan mengisi file tersebut dengan nama dari file konfigurasi yang digunakan pada permainan tersebut serta nilai yang ditampung oleh variabel global yang mengindikasikan progress dan current state dari pemain (curTime, curMoney, curSpeed, speedBoost, countHeavyItem, countMove, countVIP, countReturn, countDelivered, curBag, curInventory, curProgress, dan curToDoList, dan daftarPesanan). Setelah penulisan selesai, file tersebut disimpan di dalam folder saved-file. Perlu dicatat bahwa pada saat pemain memilih nama file, program akan mencari terlebih dahulu apakah sudah terdapat file dengan nama tersebut pada folder saved-file. Apabila ada, pemain akan diberi pilihan untuk membatalkan proses save atau tetap melanjutkan save alias overwriting file yang sudah ada.

- Proses load akan mencari file saved game pada folder saved-file. Apabila folder saved-file kosong (belum ada saved game), maka load tidak dapat dilakukan. Apabila folder saved-file tidak kosong, load akan menampilkan nama file saved game yang dapat di-load. Load dilakukan dengan menerima input berupa nama file yang dipilih oleh pemain dan dilanjutkan dengan membaca isi dari file tersebut dan menginisialisasi variabel global dengan nilai yang terdapat pada file tersebut. Load juga akan mengekstrak nama file konfigurasi yang terdapat pada file saved game dan menggabungkan nama file konfigurasi tersebut dengan path menuju folder original-config-file. Setelah mendapatkan path dari file konfigurasi, load melakukan fungsi initConfig terhadap filepath tersebut sehingga map dan adjacency matrix terinisialisasi.

## 3 Struktur Data (ADT)

### 3.1 ADT Point

- Sketsa struktur data  
ADT Point memiliki struktur data sebagai berikut (dalam C):

```
typedef struct{
    int X;
    int Y;
} Point;
```

Berikut beberapa primitif di dalam ADT Point:

1. Selektor
    - a. Absis(P)
    - b. Ordinat(P)
  2. Konstruktor
    - a. Point MakePoint (int X, int Y)
    - b. void TulisPoint (Point P)
  3. Operator Relasional  
boolean EQ (Point P1, Point P2)
  4. Operasi Lain  
void PlusDelta (Point\* P, int deltaX, int deltaY)
- Persoalan yang dapat diselesaikan  
ADT ini berfungsi untuk menunjukkan posisi bangunan dan posisi Mobita.
  - Alasan pemilihan  
Alasan memilih ADT ini sebagai wadah menyimpan daftar bangunan adalah kemampuannya dalam menyimpan koordinat suatu lokasi dan hal tersebut sangat dibutuhkan dalam penyimpanan posisi bangunan dan Mobita. Pemilihan ADT ini juga sesuai dengan spesifikasi yang diinginkan.
  - Implementasi sebagai ADT (nama file)

Nama file: *point.h*, *point.c*, dan *driver\_point.c*

### 3.2 ADT List Statis

- Sketsa struktur data

ADT List Statis memiliki struktur data sebagai berikut (dalam C) :

```
typedef struct {
    int idGadget;
    int harga;
}Gadget;

typedef struct{
    Gadget contents[INVENTORY_CAP];
}Inventory;
```

- Berikut beberapa primitif di dalam ADT List Statis:

1. Selektor

- a. IDGADGET(g)
- b. HARGAGADGET(g)
- c. INVIDGADGET(I, idx)
- d. INVHARGAGADGET(I, idx)

2. Konstruktor

void CreateInventory(Inventory \*inv)

3. Test Kosong/Penuh

- a. boolean isInventoryEmpty(Inventory inv)
- b. boolean isInventoryFull(Inventory inv)

4. Menambah dan Menghapus Gadget

- a. void addGadget(Inventory \*inv, Gadget g)
- b. void deleteGadget(Inventory \*inv, int i, Gadget \*val)

5. Operator lain

void displayGadgetinInventory(Inventory inv)

- Persoalan yang dapat diselesaikan

ADT ini digunakan untuk menyelesaikan permasalahan pembelian dan penyimpanan Gadget

- Alasan pemilihan

Alasan memilih ADT ini sebagai Inventory (tempat penyimpanan Gadget) karena kemampuannya dalam menyimpan Gadget. Pembelian Gadget dapat dihandle dengan menambahkan Gadget pada bagian inventory yang kosong dan pertama kali ketemu saat traversal. Penggunaan Gadget juga dapat dihandle, pemain bebas memilih Gadget dalam inventory. Pemilihan ADT ini juga sesuai dengan spesifikasi yang diinginkan.

- Implementasi sebagai ADT (nama file)

Nama file: *liststat.c*, *liststat.h*, dan *driver\_listat.c*

### 3.3 ADT List Dinamis

- Sketsa struktur data

ADT List Dinamis memiliki struktur data sebagai berikut (dalam C):

```
typedef struct {
    char charLoc;
    Point coorLoc;
} Location;

typedef struct{
    Location *buffer
    int nEff;
    int capacity;
} ListDin;
```

Berikut beberapa primitif di dalam ADT List Dinamis:

1. Selektor
  - a. NEFF(l)
  - b. BUFFER(l)
  - c. ELMT(l,i)
  - d. CAPACITY(l)
  - e. CHARELMT(l,i)
  - f. COORELMT(l,i)
  - g. CHARLOC(L)
  - h. COORLOC(L)
2. Konstruktor
  - a. void CreateListDin(ListDin \*l, int capacity)
  - b. void dealocate(ListDin \*l)
3. Selektor (tambahan)
  - a. int length(ListDin l)
  - b. IdxType getLastIdx(ListDin l)
4. Test Kosong/Penuh
  - a. boolean isListEmpty(ListDin l)
  - b. boolean isListFull(ListDin l)
5. Baca/Tulis
  - a. void readList(ListDin \*l)
  - b. void displayList(ListDin l)
6. Operator Relasional  
boolean isListEqual(ListDin l1, ListDin l2)
7. Searching
  - a. IdxType indexOfCharLoc(ListDin l, char charloc)
  - b. IdxType indexOfCoorLoc(ListDin l, int X, int Y)

8. Operasi Lain
  - `void copyList(ListDin lIn, ListDin *lOut)`
9. Menambah dan Menghapus Elemen di Akhir
  - a. `void insertLast(ListDin *l, char charloc, int X, int Y)`
  - b. `void deleteLast(ListDin *l, char *charloc, int *X, int *Y)`
10. Mengubah Ukuran Array
  - a. `void growList(ListDin *l, int num)`
  - b. `void shrinkList(ListDin *l, int num)`
  - c. `void compactList(ListDin *l)`
11. Operasi Tambahan Lainnya
  - a. `void setLocComponent(Location *L, int x, int y, char c)`
  - b. `void setLoc(Location *L, Location L1)`
  - c. `void displayLoc(Location L);`
  - d. `boolean IsHQ (Location P);`
  - e. `boolean isSpace(Location P);`
  - f. `boolean isDropOffPoint(Location P);`
  - g. `boolean isPickUpPoint(Location P);`
  - h. `boolean isMobitaPoint(Location P);`
  - i. `ListDin getNeighbor(Location L);`
  - j. `void displayBuilding(ListDin P);`
  - k. `boolean isAdjacent(Location L1, Location L2);`

- Persoalan yang dapat diselesaikan  
ADT ini berfungsi untuk menyimpan daftar bangunan yang tersedia pada peta.
- Alasan pemilihan  
Alasan memilih ADT ini sebagai wadah menyimpan daftar bangunan adalah kemampuannya dalam menyimpan elemen-elemen bertipe sama dan juga kemampuannya untuk memperbesar ukuran List. Hal tersebutlah yang sangat dibutuhkan dalam penyimpanan bangunan yang lebih bersifat dinamis. Pemilihan ADT ini juga sesuai dengan spesifikasi yang diinginkan.
- Implementasi sebagai ADT (nama file)  
Nama file: *listdin.c*, *listdin.h*, dan *driver\_listdin.c*

### 3.4 ADT Matriks

- Sketsa struktur data  
ADT Matriks memiliki struktur data sebagai berikut (dalam C):

```
typedef struct{
    int contents[ROW_CAP][COL_CAP];
    int rowEff;
    int colEff;
} Matrix;
```



Berikut beberapa primitif di dalam ADT Matriks:

1. Selektor
  - a. ROWS(M)
  - b. COLS(M)
  - c. MAT(M,I,j)
2. Konstruktor  
void CreateMatrix(int nRow, int nCol, Matrix \*m)
3. Selektor "Dunia Matrix"  
boolean isMatrixIdxValid(int i, int j)
4. Selektor: Untuk sebuah matriks m yang terdefinisi
  - a. IdxType getLastIdxRow(Matrix m)
  - b. IdxType getLastIdxCol(Matrix m)
  - c. boolean isMatrixIdxEff(Matrix m, IdxType i, IdxType j)
  - d. int getElmtDiagonal(Matrix m, IdxType i)
5. Assignment Matrix  
void copyMatrix(Matrix mIn, Matrix \*mRes)
6. Baca/Tulis
  - a. void readMatrix(Matrix \*m, int nRow, int nCol)
  - b. void displayMatrix(Matrix m)

- Persoalan yang dapat diselesaikan  
ADT ini berfungsi untuk merepresentasikan tampilan dari peta serta hubungan jalan antarbangunan (menggunakan adjacency matriks).
- Alasan pemilihan  
Alasan memilih ADT ini sebagai representasi tampilan peta dan hubungan jalan antarbangunan adalah dimensinya yang berupa dua dimensi dan kemudahannya dalam pengaksesan koordinat/indeks baris dan kolomnya sehingga sangat membantu proses alokasi posisi bangunan-bangunan representatif ke dalam matriks dan menjadi penunjuk hubungan jalan antarbangunan. Pemilihan ADT ini juga sesuai dengan spesifikasi yang diinginkan.
- Implementasi sebagai ADT (nama file)  
Nama file: *matrix.h*, *matrix.c*, dan *driver\_matrix.c*

### 3.5 ADT Mesin Karakter dan Mesin Kata

- Sketsa struktur data  
ADT Mesin Karakter dan Mesin Kata mendefinisikan BLANK sebagai ' ' dan memproses input dari file dan command secara berbeda. Pada pembacaan file, mark berupa EOF dan semua newline dikonversi menjadi blank untuk memudahkan pembacaan. Pada pembacaan command, mark berupa newline. Secara umum, ADT Mesin Karakter dan Kata memiliki struktur data sebagai berikut (dalam C):

STEI- ITB	IF2110-TB-05-02	Halaman 9 dari 42 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.		

```
typedef struct {
    char contents[MAXWORD]; /* container penyimpan kata, indeks yang
dipakai [0..CAPACITY-1] */
    int length;
} Word;
```

Berikut beberapa primitif pada ADT Mesin Karakter dan Mesin Kata

1. Operasi terhadap Karakter (File/Command-Line Input)

- a. start\_file(char \*filename)
- b. adv()
- c. start\_command()
- d. adv\_command()

2. Operasi terhadap Kata (File/Command-Line Input)

- a. copyWord\_file()
- b. copyWord\_command()
- c. startword\_file(char \*filename)
- d. startword\_command()
- e. advWord\_file()
- f. advWord\_command()

3. Operasi Pembacaan Input dan Penulisan Output

- a. inputWord()
- b. readFile(Word filename)
- c. printWordFile(Word kata)
- d. printWord(Word kata)

4. Konversi Kata ke Integer

- wtoi(Word kata)

5. Operasi Lainnya

- a. ignoreBlank()
- b. emptyWord()
- c. isWordEqual(Word kata1, Word kata2)

- Persoalan yang dapat diselesaikan

Membaca command dari interaksi user di main menu/in-game, membaca informasi dari file eksternal (*config* atau *saved* file), menulis output ke suatu file eksternal (save file).

- Alasan Pemilihan

Pada proses pembacaan informasi dari file eksternal, diperlukan proses akuisisi kata dari file eksternal agar dapat dikonversi menjadi bentuk variabel global (map, inventory, dll.). Selain itu, input dari command-line juga disimpan dalam bentuk string sehingga dapat diolah menggunakan mesin kata dan karakter agar dapat disimpan dalam variabel global dan disimpan dalam file eksternal baru apabila dilakukan proses save.

- Implementasi sebagai ADT (nama file)

<b>STEI- ITB</b>	<b>IF2110-TB-05-02</b>	<b>Halaman 10 dari 42 halaman</b>
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.		

Nama file: *wordmachine.c*, *wordmachine.h*, *charmachine.c*, *charmachine.h*, *driver\_machine.c*, dan *driver\_charmachine.c*

### 3.6 ADT Queue (Priority Queue)

- Sketsa struktur data

```
typedef struct {
    Pesanan buffer[QUEUE_CAP];
    int idxHead;
    int idxTail;
} Queue;
```

Berikut primitif dalam ADT Queue:

1. Selektor
    - a. IDX\_HEAD(q)
    - b. IDX\_TAIL(q)
    - c. HEAD(q)
    - d. TAIL(q)
  2. Konstruktor  
void CreateQueue(Queue \*q)
  3. Prototype
    - a. boolean isEmptyQueue(Queue q)
    - b. boolean isFullQueue(Queue q)
    - c. int lengthQueue(Queue q)
  4. Add/Delete
    - a. void enqueue(Queue \*q, Pesanan val)
    - b. void dequeue(Queue \*q, Pesanan \*val)
  5. Operasi lain  
displayQueue(Queue q)
- Persoalan yang dapat diselesaikan  
ADT ini digunakan untuk menyimpan daftar pesanan atau urutan pesanan yang nantinya akan masuk ke To Do List
  - Alasan pemilihan  
Pada dasarnya ADT Queue memiliki prinsip FIFO (First In First Out), namun pada enqueue dimodifikasi menjadi enqueue pada priority queue, yaitu memasukkan pesanan secara terurut membesar berdasarkan waktu pesanan masuk dari configuration file. Saat waktu tertentu, maka pasti pesanan yang berada di urutan paling depan yang akan keluar terlebih dahulu (didequeue) untuk dimasukkan ke dalam To Do List.
  - Implementasi sebagai ADT (nama file)  
Nama file: *queue.c*, *queue.h*, dan *driver\_queue.c*

### 3.7 ADT Stack

- Sketsa struktur data  
ADT Stack memiliki struktur data sebagai berikut (dalam C):

```
typedef struct {  
    Pesanan buffer[STACK_CAP];  
    int idxTop;  
    int curCap;  
} Stack;
```

Berikut beberapa primitif di dalam ADT Stack:

1. Selektor
    - a. IDX\_TOP(s)
    - b. TOP(s)
    - c. CURCAP(s)
  2. Konstruktor  
void CreateStack(Stack \*s)
  3. Prototype
    - a. boolean isEmpty(Stack s)
    - b. boolean isStackFull(Stack s)
    - c. void capMultiplier(Stack\* s, int mult)
    - d. void capInc(Stack\* s, int inc)
  4. Menambahkan dan Menugurangi Sebuah Elemen Stack
    - a. void push(Stack \*s, Pesanan val)
    - b. void pop(Stack \*s, Pesanan \*val)
- Persoalan yang dapat diselesaikan  
ADT ini berfungsi untuk menjadi tas/ransel untuk menyimpan item yang di pick up oleh Mobita.
  - Alasan pemilihan  
Alasan memilih ADT ini sebagai representasi tas/ransel adalah sifatnya yakni *Last In First Out* (LIFO). Hal ini sesuai dengan spesifikasi tas karena hanya item di tumpukan terataslah yang dapat di-drop off. Pemilihan ADT ini juga sesuai dengan spesifikasi yang diinginkan.
  - Implementasi sebagai ADT (nama file)  
Nama file: *stack.h*, *stack.c*, dan *driver\_stack.c*

### 3.8 ADT Linked List

- Sketsa struktur data  
ADT Linked List memiliki struktur data sebagai berikut (dalam C)

```
typedef struct node* Address
```

```
typedef struct node {
    Pesanan info;
    Address next;
} Node;
typedef Address LinkedList;
```

Berikut primitif dalam ADT Linked List:

1. Konstruktor:

```
void CreateList(LinkedList *l)
```

2. Operasi:

- a. boolean isEmpty(LinkedList l)
- b. Pesanan getElmt(LinkedList l, int idx)
- c. void setElmt(LinkedList \*l, int idx, Pesanan val)
- d. int indexOf(LinkedList l, Pesanan val)
- e. int lengthLL(LinkedList l)
- f. Address fSearch(LinkedList l, Pesanan P)

3. Penambahan Elemen:

- a. void insertFirst(LinkedList \*l, Pesanan val)
- b. void insertLastLL(LinkedList \*l, Pesanan val)
- c. void insertAt(LinkedList \*l, Pesanan val, int idx)

4. Penghapusan Elemen:

- a. void deleteFirst(LinkedList \*l, Pesanan \*val)
- b. void deleteLastLL(LinkedList \*l, Pesanan \*val)
- c. void deleteAt(LinkedList \*l, int idx, Pesanan \*val)

5. Operasi tambahan:

- a. int indexOfPesananBuilding(LinkedList l, char building)
- b. int indexOfVIP(LinkedList l, char building)
- c. void DecrementAllPerishableItem(LinkedList \*l, int decr)

- Persoalan yang dapat diselesaikan

ADT ini digunakan untuk menyimpan daftar pesanan yang harus dikerjakan (To Do List) dan sedang dikerjakan (In Progress List)

- Alasan pemilihan

Karena pada daftar pesanan yang harus dikerjakan dan sedang dikerjakan akan dilakukan banyak operasi penyisipan dan penghapusan elemen, salah satunya seperti penghapusan elemen pertama di In Progress List ketika item di DROP OFF, jika menggunakan ADT List Statis maka harus dilakukan penggeseran seluruh elemen sehingga ADT yang tepat digunakan adalah ADT Linked List

- Implementasi sebagai ADT (nama file)

Nama file: *list\_linked.c*, *list\_linked.h*, *node.c*, *node.h*, dan *driver\_list\_linked.c*

### 3.9 ADT Pesanan

- Sketsa struktur data

STEI- ITB	IF2110-TB-05-02	Halaman 13 dari 42 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.		

ADT Pesanan memiliki struktur data sebagai berikut (dalam C):

```
typedef struct pesanan {
    int time;
    char pickup;
    char dropoff;
    char type;
    int ptime;
} Pesanan;
```

Berikut primitif dalam ADT Linked List:

1. Konstruktor:

Pesanan CreatePesanan(int T, char PU, char DO, char TP, int PT)

2. Selektor

- a. TIMEPESANAN(p)
- b. PICKUPPESANAN(p)
- c. DROPOFFPESANAN(p)
- d. TIPEITEM(p)
- e. PTIME(p)

- Persoalan yang dapat diselesaikan

ADT ini dibuat untuk menyimpan data-data yang dimiliki sebuah pesanan yaitu waktu pesanan masuk, bangunan lokasi pick up, bangunan lokasi drop off, jenis item, dan waktu hangus bila itemnya adalah perishable item

- Alasan pembuatan

Alasan pembuatan ADT Pesanan yaitu untuk mempermudah operasi yang akan dilakukan karena semua informasi tersimpan dalam satu struktur data.

- Implementasi sebagai ADT (nama file)

Nama file: *pesanan.c*, *pesanan.h*, dan *driver\_pesanan.c*

### 3.10 ADT Peta

- Sketsa struktur data

ADT Peta memiliki struktur data sebagai berikut:

```
typedef struct{
    char Tile[ROW_CAP+2][COL_CAP+2]
    int REff;
    int CEff;
} Map;
```

Berikut primitif dalam ADT Linked List:

1. Konstruktor:

void CreateMap(Map \*m, int rows, int cols);

2. Selektor:

- a. TILE(M,i,j)

- b. REFF(M)
  - c. CEFF(M)
- 3. Operasi lain:
  - a. void BuildingToMap(Map \*m, ListDin L)
  - b. void DisplayMap(Map m)
- Persoalan yang dapat diselesaikan  
ADT ini dibuat untuk menyimpan data-data yang dimiliki map yaitu tile (berupa ' ' atau '\*\*' atau karakter building), ukuran panjang efektif dan lebar efektif dari peta.
- Alasan pembuatan  
Alasan pembuatan ADT Map untuk mempermudah proses pembacaan peta dari file konfigurasi serta command MAP
- Implementasi sebagai ADT(nama file)  
  
Nama file: *peta.c*, *peta.h*, dan *driver\_peta.c*

## 4 Program Utama

Pada program utama, pertama-tama, program akan menerima input kata, yakni 'NEW\_GAME' untuk memulai permainan atau 'EXIT' untuk keluar dari permainan. Jika pemain menginput kata yang salah, program kami akan mencetak pesan kesalahan dan meminta input kata kembali hingga input benar.

Selanjutnya, jika program menerima kata "NEW\_GAME", program akan menerima input nama file konfigurasi berupa kata. Jika nama file konfigurasi salah/tidak ditemukan, program akan mencetak pesan bahwa file tidak ditemukan dan meminta kembali input kata kembali hingga benar. Setelah itu, program akan melakukan load pada folder eksternal untuk mendapatkan layout Map dan daftar urutan pesanan, kemudian permainan pun dimulai.

Selanjutnya, program akan menerima input command berupa kata, command yang diterima antara lain "MOVE" untuk berpindah ke lokasi selanjutnya, "PICK\_UP" untuk mengambil item di lokasi sekarang, "DROP\_OFF" untuk mengantarkan item ke lokasi sekarang, "MAP" untuk memunculkan peta, "TO\_DO" untuk menampilkan pesanan yang masuk ke To Do List, "IN\_PROGRESS" untuk menampilkan pesanan yang sedang dikerjakan, "BUY" untuk menampilkan gadget yang dapat dibeli, "INVENTORY" untuk menampilkan isi inventory, "RETURN" untuk mengembalikan item teratas di bag ke lokasi tempat pick-up item tersebut, "HELP" untuk menampilkan command yang dapat dimasukan beserta deskripsinya, "SAVE" untuk menyimpan progres game, dan "EXIT" untuk keluar dari permainan. Jika pemain menginput kata yang salah, program kami akan mencetak pesan kesalahan dan meminta input kata kembali hingga input benar. Permainan akan berakhir ketika seluruh pesanan selesai diantar dan pemain sudah kembali ke Headquarters.

## 5 Data Test

Untuk memastikan program dapat berjalan dengan lancar dan sesuai dengan spesifikasi, dibutuhkan suatu *testing*. Berikut data test yang digunakan pada *testing* terhadap program kami

STEI- ITB	IF2110-TB-05-02	Halaman 15 dari 42 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.		



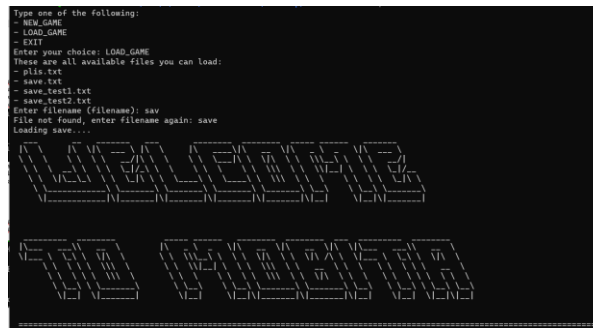


Seperti yang dilihat, load game dibatalkan karena belum adanya saved file. *Testing* dilanjutkan dengan menambahkan saved files secara manual pada folder saved-file yang hanya digunakan demi keperluan *testing*. Setelah menambahkan file tersebut, dilakukan make run agar kembali ke main menu dan memasukkan input LOAD\_GAME untuk melihat apakah kali ini load dapat dilakukan atau tidak.

```

Type one of the following:
- NEW_GAME
- LOAD_GAME
- EXIT
Enter your choice: LOAD_GAME
There are all available files you can load:
- pls.txt
- save.txt
- save.test1.txt
- save.test2.txt
Enter filename (filename): sav
File not found, enter filename again: save
Loading save....

```



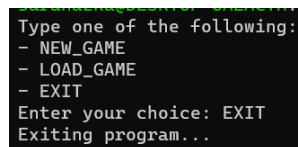
Seperti yang dilihat, terdapat tampilan nama saved file yang dapat di-load dan permainan dapat dimulai dengan melakukan *load* terhadap saved file yang dipilih.

- c. Pada main menu, dilakukan tes untuk melihat apakah EXIT berjalan dengan baik atau tidak. Output yang baik adalah apabila pemain keluar dari program.

```

Type one of the following:
- NEW_GAME
- LOAD_GAME
- EXIT
Enter your choice: EXIT
Exiting program...

```



Seperti yang dilihat, exit berjalan sebagaimana harusnya.

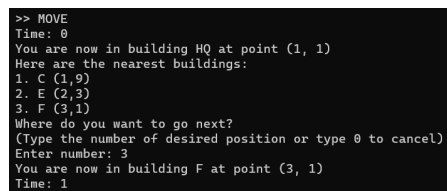
## 5.2 Data Test 2

Untuk melihat apakah mekanisme waktu berjalan dengan baik, dilakukan *testing* melalui command MOVE. Apabila keadaan sedang normal (tidak ada heavy item dan tidak menggunakan speed boost), waktu akan bertambah satu seiring pemain berjalan.

```

>> MOVE
Time: 0
You are now in building HQ at point (1, 1)
Here are the nearest buildings:
1. C (1, 9)
2. E (2, 3)
3. F (3, 1)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 3
You are now in building F at point (3, 1)
Time: 1

```



Saat player sedang membawa suatu *heavy item*, mekanisme waktu akan berubah yakni waktu akan bertambah 1 bagi setiap heavy item yang dibawa oleh pemain. Untuk menguji fitur ini, *testing* dilakukan dengan melakukan pick-up terhadap dua heavy item dan kemudian mencoba command MOVE sembari membawa kedua heavy item tersebut.

```

>> IN_PROGRESS
Here are your on-going orders:
1. Heavy Item (Destination: B)
2. Heavy Item (Destination: N)

>> MOVE
Time: 8
You are now in building M at point (8, 2)
Here are the nearest buildings:
1. A (10,1)
2. K (6,3)
3. N (8,6)
4. Q (10,3)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 1
You are now in building A at point (10, 1)
Time: 11

```

Seperti yang terlihat, waktu bertambah sebanyak 3 karena adanya dua heavy item. Fitur lain yang berhubungan dengan waktu adalah speed boost yang didapat dengan melakukan drop-off heavy item. Saat speed boost teraktivasi, waktu akan bertambah 1 unit setiap pemain berpindah 2 lokasi (maksimal 10 lokasi). Namun, apabila pemain masih membawa heavy item lainnya saat speed boost diaktivasi, maka ability dari speed boost teranulir dan waktu tetap bertambah sejumlah banyaknya heavy item yang dibawa. Maka, pada tahap *testing* berikutnya dilakukan drop-off terhadap salah satu dari dua heavy item.

```

>> DROP_OFF
Order in the form of Heavy Item succesfully delivered!
You got 400 Yen. Great job.
You were rewarded the ability 'Speed Boost'. Ability activated.

>> MOVE
Time: 32
You are now in building B at point (1, 15)
Here are the nearest buildings:
1. D (1,13)
2. H (3,14)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 1
You're carrying Heavy Item. Speed Boost deactivated.
You are now in building D at point (1, 13)
Time: 34

```

Dapat dilihat pada contoh di atas bahwa speed boost tidak teraktivasi karena masih ada satu heavy item sehingga waktu tetap bertambah 2. Namun, apabila heavy item tersebut didrop-off juga, speed boost teraktivasi. *Testing* dilanjutkan dengan melihat apakah speed boost tersebut berjalan dengan baik atau tidak.

```

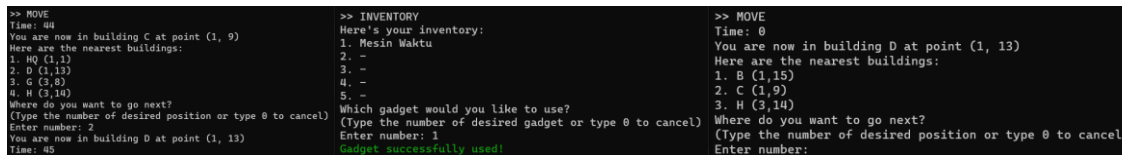
>> DROP_OFF
Order in the form of Heavy Item succesfully delivered!
You got 400 Yen. Great job.
You were rewarded the ability 'Speed Boost'. Ability activated.

>> MOVE
Time: 42
You are now in building N at point (8, 6)
Here are the nearest buildings:
1. K (6,3)
2. L (7,10)
3. M (8,2)
4. Q (10,3)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 1
You are now in building K at point (6, 3)
Time: 42

>> MOVE
Time: 42
You are now in building K at point (6, 3)
Here are the nearest buildings:
1. F (3,1)
2. I (4,5)
3. L (7,10)
4. M (8,2)
5. N (8,6)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 2
You are now in building I at point (4, 5)
Time: 43

```

Seperti yang dapat dilihat, waktu hanya bertambah setiap dua langkah sehingga speed boost bekerja dengan baik. Fitur terakhir yang berhubungan dengan waktu adalah adanya gadget Mesin Waktu yang dapat mengurangi waktu sebanyak 50 detik. *Testing* dilakukan dengan membeli dan menggunakan gadget tersebut dan melihat apakah waktu berkurang sebanyak 50 atau tidak (apabila pengurangan mencapai nilai negatif, waktu dijadikan 0).

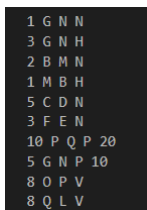


The first screenshot shows the player in building C at point (1, 9) with a time of 44. The second screenshot shows the player's inventory, including the 'Mesin Waktu' gadget. The third screenshot shows the player in building D at point (1, 13) with a time of 0, indicating the gadget was successfully used to reduce the time by 50 units.

Seperti yang dilihat pada ketiga foto di atas, waktu dimulai pada detik 45. Kemudian, setelah digunakan gadget Mesin Waktu, waktu berkurang menjadi 0.

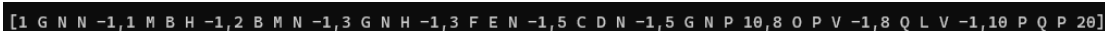
### 5.3 Data Test 3

Untuk *testing* daftar pesanan dan to-do list, digunakan command TO\_DO untuk melihat apakah daftar pesanan yang diekstrak dari konfigurasi benar dan dapat disortir di dalam to-do list dengan baik dan benar. Data yang digunakan untuk *testing* berupa 10 item yang ditulis dalam file konfigurasi secara tidak teratur seperti berikut



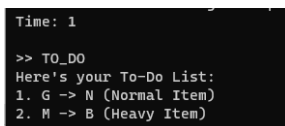
The configuration file contains the following items: 1 G N N, 3 G N H, 2 B M N, 1 M B H, 5 C D N, 3 F E N, 10 P Q P 20, 5 G N P 10, 8 O P V, and 8 Q L V.

Untuk melihat apakah daftar pesanan terekstrak secara benar, fungsi displayQueue digunakan demi keperluan *testing* saja. Apabila berhasil, daftar pesanan akan berupa queue yang berisi informasi 10 item tersebut dengan catatan item tersebut telah diurut berdasarkan waktu pesanan masuk. Berikut hasil daftar pesanan yang didapat:



The output shows the items sorted by their time value: [1 G N N -1, 1 M B H -1, 2 B M N -1, 3 G N H -1, 3 F E N -1, 5 C D N -1, 5 G N P 10, 8 O P V -1, 8 Q L V -1, 10 P Q P 20].

Seperti yang terlihat, daftar pesanan sudah dalam bentuk teratur sesuai harapan. *Testing* dilanjut untuk melihat apakah to-do list berfungsi dengan baik dan benar dengan cara menggunakan command TO\_DO.



The terminal shows the command >> TO\_DO being entered, resulting in a to-do list with two items: 1. G -> N (Normal Item) and 2. M -> B (Heavy Item).

Saat current time adalah 1, to-do list berisi kedua item yang waktu pesanannya muncul pada waktu 1 sehingga didapat bahwa to-do list bekerja dengan baik.

## 5.4 Data Test 4

*Testing* terhadap In-Progress List dilakukan dengan melihat apakah suatu item akan masuk ke list ini apabila di-pick up dan apakah item tersebut akan terhapus dari list ini apabila di-dropoff pada lokasi yang sesuai. Oleh karena itu, *testing* dimulai dengan menggunakan command IN\_PROGRESS untuk melihat isi list in-progress.

```
>> IN_PROGRESS
Yay! There is no order in progress
>>
```

In-progress list dimulai dengan menampilkan pesan bahwa tidak ada pesanan in-progress. Setelah itu, dilakukan pick-up yang dilanjut dengan penggunaan command IN\_PROGRESS untuk melihat apakah item sudah berada pada in-progress list atau tidak.

```
>> PICK_UP
Normal Item successfully picked up!
Order destination: N

>> IN_PROGRESS
Here is the order that you're currently delivering:
1. Normal Item (Destination: N)
```

Seperti yang terlihat, item sudah masuk ke dalam in-progress list. Selanjutnya, item di-drop off dan command IN\_PROGRESS digunakan lagi untuk melihat apakah item sudah dihapus dari list in-progress atau belum.

```
>> DROP_OFF
Order in the form of Normal Item succesfully delivered!
You got 200 Yen. Nice.

>> IN_PROGRESS
Yay! There is no order in progress
>>
```

List in-progress kembali kosong, mengindikasikan bahwa item telah berhasil terhapus.

## 5.5 Data Test 5

Untuk *testing* tas, pertama akan dilihat apakah pada awal kapasitas tas benar hanya dapat membawa 3 item sekaligus atau tidak dengan cara mencoba melakukan pick-up pada 4 item di awal permainan melalui command PICK\_UP dan IN\_PROGRESS.

```
>> PICK_UP
Bag is full!

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: D)
2. Normal Item (Destination: M)
3. Normal Item (Destination: N)
```

Seperti yang dilihat, tas hanya dapat membawa 3 item pada awal permainan. Selain itu, tas hanya akan memilih item yang urutan pesanan masuknya lebih dulu untuk di-pick up apabila terdapat lebih dari satu item pada satu lokasi. Hal ini terbukti berfungsi seperti pada gambar berikut, dimana terdapat dua item yang dapat di-pickup dari lokasi G:

```
>> TO_DO
Here's your 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)

>> PICK_UP
Normal Item successfully picked up!
Order destination: N
```

Kapasitas tas juga dapat bertambah dengan menggunakan ability increase capacity yang didapat dari mengantarkan Perishable Item. *Testing* dilakukan dengan mengantarkan perishable item dan melihat apakah kapasitas tas bertambah sebanyak satu.

```
>> DROP_OFF
Order in the form of Perishable Item succesfully delivered!
You got 400 Yen. Good work.
You were also given the 'Increase Capacity' ability. Ability activated.

>> PICK_UP
Bag is full!

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: D)
2. Normal Item (Destination: E)
3. Perishable Item (Destination: N, Remaining Time: 3)
4. Normal Item (Destination: N)
```

Seperti yang dapat dilihat, setelah mengantarkan Perishable Item kapasitas tas bertambah sebanyak 1. Hal ini ditunjukkan melalui kapasitas tas yang menjadi 4 setelah drop-off.

## 5.6 Data Test 6

Terdapat 4 jenis item dengan karakteristik masing-masing. *Testing* dilakukan untuk memastikan agar karakteristik setiap item berfungsi sebagaimana harusnya.

- Untuk Normal Item, *testing* cukup dengan melakukan drop-off pada normal item untuk menguji apakah item bernilai 200 yen atau tidak.

```
>> DROP_OFF
Order in the form of Normal Item succesfully delivered!
You got 200 Yen. Nice.
```

Seperti yang dilihat, item memberikan reward sebesar 200 yen.

- Untuk Heavy Item, *testing* telah dilakukan pada Data Test 2 untuk membuktikan bahwa pergerakan pemain melambat ketika membawa heavy item dan menunjukkan bahwa speed boost akan teraktivasi setelah pemain mengantarkan item ini. Oleh karena itu, *testing* yang tersisa adalah melihat apakah item seharga 400 yen atau tidak.

```
>> DROP_OFF
Order in the form of Heavy Item succesfully delivered!
You got 400 Yen. Great job.
You were rewarded the ability 'Speed Boost'. Ability activated.
```

Seperti yang dilihat, heavy item memiliki harga sebesar 400 yen.

- Untuk Perishable Item, *testing* pada Data Test 5 telah membuktikan bahwa Increase Capacity akan didapatkan setelah mengantarkan item ini. Maka, *testing* dilanjut dengan melihat apakah item bernilai 400 yen dan menguji apakah mekanisme waktu dari perishable item berfungsi dengan baik atau tidak.

```

>> DROP_OFF
Order in the form of Perishable Item succesfully delivered!
You got 400 Yen. Good work.
You were also given the 'Increase Capacity' ability. Ability activated.

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: D)
2. Normal Item (Destination: E)
3. Perishable Item (Destination: N, Remaining Time: 1)
4. Normal Item (Destination: N)

>> MOVE
Time: 15
You are now in building H at point (3, 14)
Here are the nearest buildings:
1. B (1,15)
2. C (1,9)
3. D (1,13)
4. G (3,8)
5. J (5,12)
6. O (8,15)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 4
You are now in building G at point (3, 8)
Time: 16

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: D)
2. Normal Item (Destination: E)
3. Normal Item (Destination: N)

```

Seperti yang dilihat, perishable item memiliki harga sebesar 400 yen. Selain itu, mekanisme waktu pada perishable item berfungsi sebagaimana harusnya dengan terjadinya penghapusan pesanan dari in-progress list serta pengeluaran item dari tas saat batas waktu dari perishable item telah habis.

- d. Untuk VIP item, *testing* dilakukan dengan melihat apakah ketika ada item VIP, item lainnya dapat di-pickup dan di-dropoff atau tidak.

```

>> TO_DO
Here's your To-Do List:
1. M -> B (Heavy Item)
2. B -> M (Normal Item)
3. P -> Q (Perishable Item, required delivery time: 20)
4. O -> P (VIP Item)
5. Q -> L (VIP Item)

>> PICK_UP
There are no VIP items here!
You have VIP item(s) to be done first, simp.

```

Seperti yang dilihat, saat terdapat VIP item pada to-do list pemain harus mengambil VIP item terlebih dahulu. Selain itu, perlu diuji bahwa item ini bernilai 600 yen dan pemain akan mendapatkan ability setelah mengantarkan item ini.

```

>> DROP_OFF
Order in the form of VIP Item succesfully delivered!
You got 600 Yen and the ability 'Return to Sender'. Amazing.

>> RETURN
Normal Item successfully returned to Pick Up Point C

```

Seperti yang dilihat, setelah mengantarkan VIP item pemain dapat melakukan Return. Selain itu, dapat dilihat juga bahwa VIP item akan memberikan reward sebesar 600 yen.

## 5.7 Data Test 7

Terdapat 3 ability pada program ini. *Testing* dilakukan pada masing-masing ability untuk menjamin bahwa ability tersebut berfungsi dengan baik dan benar. Perlu dicatat bahwa *testing* ability Speed Boost telah dilakukan secara terpisah pada Data Test 2 dan ability Increase Capacity telah diuji pada Data Test 5, sehingga Data Test 7 hanya akan menguji ability Return to Sender. *Testing* dilakukan dengan menggunakan command RETURN.

STEI- ITB	IF2110-TB-05-02	Halaman 22 dari 42 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.		

```
>> RETURN
You don't have this ability...
```

Seperti yang dilihat, terdapat pesan eror yang diakibatkan oleh belum aktifnya ability Return to Sender. Oleh karena itu, *testing* dilanjutkan dengan mengantarkan item VIP sehingga mengaktifkan ability Return to Sender, dilanjut dengan command RETURN.

```
>> DROP_OFF
Order in the form of VIP Item succesfully delivered!
You got 600 Yen and the ability 'Return to Sender'. Amazing.

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: E)
2. Normal Item (Destination: N)

>> RETURN
Normal Item successfully returned to Pick Up Point F

>> IN_PROGRESS
Here is the order that you're currently delivering:
1. Normal Item (Destination: N)

>> TO_DO
Here's your To-Do List:
1. M -> B (Heavy Item)
2. B -> M (Normal Item)
3. P -> Q (Perishable Item, required delivery time: 20)
4. C -> D (Normal Item)
5. F -> E (Normal Item)
```

Seperti yang dilihat, return berhasil dan item di atas tas dikembalikan ke lokasi pick-up

## 5.8 Data Test 8

Terdapat 5 gadget yang dapat digunakan pemain pada program. *Testing* dilakukan dengan cara mencoba setiap gadget untuk memastikan bahwa efek dari setiap gadget berfungsi dengan benar. Perlu dicatat bahwa gadget Mesin Waktu telah diuji pada Data Test 2.

- Kain Pembungkus Waktu digunakan untuk membuat perishable item teratas pada tas agar kembali ke durasi semula

```
>> IN_PROGRESS
Here is the order that you're currently delivering:
1. Perishable Item (Destination: N, Remaining Time: 7)

>> INVENTORY
Here's your inventory:
1. Kain Pembungkus Waktu
2. Pintu Kemana Saja
3. Mesin Waktu
4. Senter Pengecil
5. -
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Gadget successfully used!

>> IN_PROGRESS
Here is the order that you're currently delivering:
1. Perishable Item (Destination: N, Remaining Time: 10)
```

Seperti yang dilihat, durasi dari perishable item teratas pada tas kembali ke semula. Selain itu, dilakukan *testing* untuk memastikan bahwa apabila gadget digunakan saat item paling atas bukan Perishable Item maka gadget akan digunakan secara sia-sia.

```

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: M)
2. Heavy Item (Destination: N)

>> INVENTORY
Here's your inventory:
1. Kain Pembungkus Waktu
2. -
3. -
4. -
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Gadget successfully used!

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: M)
2. Heavy Item (Destination: N)

```

- b. Senter Pembesar digunakan untuk meningkatkan kapasitas tas sebesar dua kali lipat, tetapi tidak bisa melebihi batas maksimum. Untuk keperluan *testing*, ditaruh fungsi print yang menampilkan current capacity dari tas.

```

>> INVENTORY
Here's your inventory:
1. Senter Pembesar
2. Pintu Kemana Saja
3. Mesin Waktu
4. Senter Pengecil
5. -
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Gadget successfully used!
Current Capacity: 6

```

Seperti yang dilihat, kapasitas tas yang pada awalnya adalah 3 membesar sebanyak dua kali lipat menjadi 6. *Testing* dilanjut dengan mencoba menggunakan senter pembesar hingga melampaui kapasitas.

```

>> INVENTORY
Here's your inventory:
1. Senter Pembesar
2. Pintu Kemana Saja
3. Mesin Waktu
4. Senter Pengecil
5. Senter Pembesar
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 5
Gadget successfully used!
Current Capacity: 100

>> INVENTORY
Here's your inventory:
1. Senter Pembesar
2. Pintu Kemana Saja
3. Mesin Waktu
4. Senter Pengecil
5. -
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Gadget successfully used!
Current Capacity: 100
>>

```

Seperti yang dilihat, kapasitas tas hanya dapat diperbesar hingga batas maksimum.

- c. Pintu Kemana Saja digunakan untuk berpindah ke lokasi yang diinginkan tanpa menambahkan unit waktu.



```

>> MOVE
Time: 22
You are now in building HQ at point (1, 1)
Here are the nearest buildings:
1. C (1,9)
2. E (2,3)
3. F (3,1)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 1
You are now in building C at point (1, 9)
Time: 24

>> INVENTORY
Here's your inventory:
1. -
2. Pintu Kemana Saja
3. -
4. -
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 2
Gadget successfully used!

Gadget successfully used!
Here's the Map:
*****
* E   C   B *
* F   G   H *
*   I   J   *
* K           *
*   L       *
* M   N   O *
*   P       *
* A Q       *
*****
(you're the yellow point)
Where do you want to go?
Type "HQ" to go to HQ, type a letter between A to Q that represents
the desired location : HQ

>> MOVE
Time: 24
You are now in building HQ at point (1, 1)
Here are the nearest buildings:
1. C (1,9)
2. E (2,3)
3. F (3,1)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number:

```

Seperti yang dilihat, perpindahan lokasi dari C ke HQ terjadi dengan waktu tetap 24.

- d. Senter Pengecil digunakan untuk menghilangkan efek dari satu heavy item jika terdapat pada tumpukan teratas tas. Efeknya berlangsung sampai melakukan drop-off atau return pertama kali setelah penggunaannya. *Testing* dimulai dengan melihat apakah mekanisme waktu menjadi normal walau ada Heavy Item.

```

>> INVENTORY
Here's your inventory:
1. -
2. Pintu Kemana Saja
3. Mesin Waktu
4. Senter Pengecil
5. -
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 4
Gadget successfully used!

>> IN_PROGRESS
Here are your on-going orders:
1. Heavy Item (Destination: B)
2. Perishable Item (Destination: N, Remaining Time: 4)

>> MOVE
Time: 11
You are now in building G at point (3, 8)
Here are the nearest buildings:
1. C (1,9)
2. E (2,3)
3. H (3,14)
4. I (4,5)
5. J (5,12)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 1
You are now in building C at point (1, 9)
Time: 12

```

Seperti yang dilihat, efek dari Heavy Item teranulir dengan penggunaan gadget ini sehingga waktu berjalan dengan normal. Dilakukan *testing* untuk melihat apakah yang teranulir hanyalah salah satu Heavy Item saja.

```

>> IN_PROGRESS
Here are your on-going orders:
1. Heavy Item (Destination: N)
2. Normal Item (Destination: M)
3. Heavy Item (Destination: N)

>> INVENTORY
Here's your inventory:
1. Pintu Kemana Saja
2. Pintu Kemana Saja
3. -
4. Senter Pengecil
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 4
Gadget successfully used!

>> MOVE
Time: 14
You are now in building G at point (3, 8)
Here are the nearest buildings:
1. C (1,9)
2. E (2,3)
3. H (3,14)
4. I (4,5)
5. J (5,12)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 4
You are now in building I at point (4, 5)
Time: 16

```

Seperti yang dilihat, hanya salah satu Heavy Item yang teranulir efek waktunya. Terakhir, dilakukan *testing* untuk memastikan bahwa apabila item paling atas bukanlah Heavy Item maka gadget digunakan secara sia-sia.

```

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: M)
2. Heavy Item (Destination: N)

>> INVENTORY
Here's your inventory:
1. Pintu Kemana Saja
2. Pintu Kemana Saja
3. Senter Pengecil
4. Senter Pengecil
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 3
Gadget successfully used!

>> MOVE
Time: 8
You are now in building B at point (1, 15)
Here are the nearest buildings:
1. D (1,13)
2. H (3,14)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 1
You are now in building D at point (1, 13)
Time: 10

```

Seperti yang dilihat, gadget berakhir sia-sia karena efek Heavy Item tidak teranulir.

## 5.9 Data Test 9

Testing terhadap gadget dan inventory memastikan bahwa keduanya berfungsi dengan baik. Pertama, perlu diuji bahwa setiap gadget yang terbeli akan langsung masuk ke inventory.

```

>> BUY
Current money: 100000 Yen
Check out these gadgets...!
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)
Which gadget would you like to buy?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Gadget successfully bought!
Current money: 99200 Yen

>> INVENTORY
Here's your inventory:
1. Kain Pembungkus Waktu
2. -
3. -
4. -
5. -
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: |

```

Seperti yang terlihat, gadget yang dibeli akan masuk ke inventory. Perlu dipastikan juga bahwa inventory tidak bisa menampung gadget lebih dari kapasitasnya yakni 5.

```

>> INVENTORY
Here's your inventory:
1. Kain Pembungkus Waktu
2. Senter Pembesar
3. Pintu Kemana Saja
4. Mesin Waktu
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 0

>> BUY
Current money: 92700 Yen
Check out these gadgets...!
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)
Which gadget would you like to buy?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Inventory full. Unable to buy gadgets.

```

Seperti yang dilihat, ketika inventory sudah penuh pemain tidak dapat membeli gadget. Perlu dipastikan bahwa setiap gadget hanya dapat digunakan tepat sekali dan setelah digunakan gadget tersebut akan hilang dari inventory.

```

>> INVENTORY
Here's your inventory:
1. Kain Pembungkus Waktu
2. Senter Pembesar
3. Pintu Kemana Saja
4. Mesin Waktu
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 1
Gadget successfully used!

>> INVENTORY
Here's your inventory:
1. -
2. Senter Pembesar
3. Pintu Kemana Saja
4. Mesin Waktu
5. Senter Pengecil
Which gadget would you like to use?
(Type the number of desired gadget or type 0 to cancel)
Enter number: |

```

Seperti yang dilihat, gadget tersebut langsung hilang dari inventory sehingga hanya dapat digunakan tepat sekali.

## 5.10 Data Test 10

*Testing* dilakukan terhadap fitur map. Apabila berhasil, program akan menunjukkan sebuah peta dengan ukuran sesuai file konfigurasi dengan batasan berupa karakter ‘\*’ dan karakter berwarna oranye untuk posisi pemain, merah untuk posisi pick-up pesanan, biru untuk posisi drop-off pesanan, hijau untuk posisi yang dapat dicapai, dan hitam untuk posisi lainnya.

```

>> MAP
Here's the Map:
*****
*8   C D B*
*   E   *
*F   G   H *
*   I   *
*   J   *
*   K   *
*   L   *
* M N   O*
*   P   *
*A Q   *
*****
(you're the yellow point)

You are now in building E at point (2, 3)
Time: 1

>> MAP
Here's the Map:
*****
*8   C D B*
*   E   *
*F   G   H *
*   I   *
*   J   *
*   K   *
*   L   *
* M N   O*
*   P   *
*A Q   *
*****
(you're the yellow point)

```

Di atas merupakan hasil *testing* dimana foto paling kiri merupakan kondisi dimana pemain baru saja memulai permainan, foto di tengah merupakan kondisi dimana pemain mulai mendapatkan pesanan dengan bagian atasnya menunjukkan koordinat terakhir pemain berdasarkan command MOVE, dan foto paling kanan merupakan kondisi dimana pemain sudah melakukan pick-up pesanan. Warna hitam untuk posisi lainnya berubah menjadi putih karena menyesuaikan dengan latar dari terminal. Seperti yang dilihat, warna berfungsi dengan baik dan map menunjukkan pergerakan pemain.

## 5.11 Data Test 11

Terdapat 11 *in-game command* pada program ini. Untuk memastikan setiap command berjalan dengan lancar, kita melakukan *testing* pada setiap command. Catatan bahwa beberapa command *ditesting* secara terpisah, yakni command TO\_DO dilakukan pada Data Test 3, command IN\_PROGRESS dilakukan pada Data Test 4, command INVENTORY dilakukan pada Data Test 9, command MAP dilakukan pada Data Test 10, command SAVE dilakukan pada Data Test 13, dan command RETURN dilakukan pada Data Test 7.

- a. Command MOVE dites dengan menuliskan MOVE pada input. Apabila berhasil, maka program akan menampilkan gedung terdekat yang dapat dikunjungi oleh player, menerima input gedung yang ingin dikunjungi, dan memindahkan player ke tempat tersebut. Waktu akan bertambah seiring pemain berjalan sesuai pada Data Test 1.

```
>> MOVE
Time: 0
You are now in building HQ at point (1, 1)
Here are the nearest buildings:
1. C (1,9)
2. E (2,3)
3. F (3,1)
Where do you want to go next?
(Type the number of desired position or type 0 to cancel)
Enter number: 3
You are now in building F at point (3, 1)
Time: 1
```

- b. Command PICK\_UP dites dengan menuliskan command PICK\_UP. Apabila berhasil, pemain akan mengambil item jika terdapat pesanan yang harus diambil pada lokasi.

```
>> PICK_UP
Normal Item successfully picked up!
Order destination: N
```

Jika tidak terdapat pesanan yang harus diambil pada lokasi, maka diberikan pesan error.

```
>> PICK_UP
There seems to be no items here... Pickup unsuccessful.
```

Apabila tas penuh, akan dikeluarkan pesan bahwa tas sudah penuh seperti berikut:

```
>> PICK_UP
Bag is full!
```

Saat ada item VIP pada to-do list, pemain tidak dapat melakukan pick-up pada order lainnya sebelum menyelesaikan order VIP tersebut dan pada kasus dimana tas sudah penuh, pemain harus melakukan drop-off terhadap barang teratas sebelum melakukan pick-up VIP item yang telah dijadikan prioritas.

```
>> TO_DO
Here's your To-Do List:
1. M -> B (Heavy Item)
2. G -> N (Heavy Item)
3. F -> E (Normal Item)
4. G -> N (Perishable Item, required delivery time: 10)
5. O -> P (VIP Item)
6. Q -> L (VIP Item)

>> PICK_UP
Bag is full!
```

Seperti yang dapat dilihat, walaupun terdapat VIP item, pemain tetap tidak bisa melakukan pick-up dikarenakan tas penuh. *Testing* dilanjut dengan melakukan drop-off terhadap item paling atas di tas dan dilanjut dengan melakukan pick-up terhadap item lain yang bukan VIP item untuk melihat apakah program akan menyuruh pemain untuk melakukan pick-up terhadap VIP item terlebih dahulu atau tidak.

```
>> DROP_OFF
Order in the form of Normal Item succesfully delivered!
You got 200 Yen. Nice.

>> PICK_UP
There are no VIP items here!
You have VIP item(s) to be done first, simp.

>> IN_PROGRESS
Here are your on-going orders:
1. Normal Item (Destination: M)
2. Normal Item (Destination: N)
```

Seperti yang dapat dilihat, pick-up VIP item diprioritaskan karena tas tidak penuh.

- c. Command `DROP_OFF` dites dengan menuliskan `DROP_OFF`. Drop-off hanya dapat dilakukan pada item yang berada di tumpukan teratas tas. Apabila lokasi pesanan bukanlah lokasi drop-off dari item paling atas di tas, akan ditampilkan pesan error.

```
>> DROP_OFF
This is not the drop-off location of the top-most item in your bag!
```

Apabila ada pemain melakukan drop-off pada lokasi dari destinasi pesanan paling atas pada tas, maka pesanan didrop-off dan pemain mendapatkan upah serta reward sesuai tipe item yang diantarkan.

```
>> DROP_OFF
Order in the form of VIP Item succesfully delivered!
You got 600 Yen and the ability 'Return to Sender'. Amazing.
```

Apabila pemain melakukan drop-off pada kondisi dimana tidak ada pesanan apapun untuk di drop-off, ditampilkan pesan.

```
>> DROP_OFF
There are no items to be dropped off from your bag!
```

- d. Command `BUY` dites dengan menuliskan `BUY`. Apabila berhasil, program akan menampilkan gadget yang dapat dibeli.

```
>> BUY
Current money: 1000000000 Yen
Check out these gadgets...!
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)
Which gadget would you like to buy?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 5
Gadget successfully bought!
Current money: 999999200 Yen
```

Perlu dicatat bahwa `BUY` hanya dapat dilakukan di HQ, sehingga apabila pemain menulis `BUY` pada saat posisinya tidak berada di HQ, keluar pesan sebagai berikut:

```
>> BUY
You can only buy when you're in HQ.
```

Apabila pemain tidak memiliki uang yang cukup, program akan menampilkan pesan dan pemain dikembalikan ke tempat input command.

```
>> BUY
Current money: 0 Yen
Check out these gadgets...!
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)
Which gadget would you like to buy?
(Type the number of desired gadget or type 0 to cancel)
Enter number: 3
Oops... Seems like you don't have enough money!

>> |
```

- e. Command HELP digunakan untuk menampilkan seluruh command beserta deskripsinya dan dipesan dengan menuliskan HELP.

```
>> HELP
- MOVE: move to next location
- PICK_UP: pickup item in current location
- DROP_OFF: deliver top-most item to current location
- MAP: display map
- TO_DO: display list of to-do order(s)
- IN_PROGRESS: display list of in-progress order(s)
- BUY: purchase gadgets (only available when in HQ)
- INVENTORY: display list of inventory
- RETURN: return top-most item in bag to its pick-up location
- HELP: display list of available command(s)
- EXIT: exit game
- SAVE_GAME: save current progress
```

## 5.12 Data Test 12

Game Flow dari program ini diuji dengan memainkan program hingga selesai. Seperti yang dilihat pada Data Test 1, start game telah diuji pada main menu dengan melakukan load game atau memulai game baru. Saat semua pesanan telah selesai diantar alias sudah tidak ada lagi item pada to-do list maupun in-progress list, pemain akan mendapat pesan yang menyuruhnya untuk pergi ke HQ. Setelah itu, akan ditampilkan waktu yang ditempuh, item yang berhasil diantarkan, dan uang yang didapat.

```
Seems like there are no orders left for you to deliver. Nice work today.
Head back to HQ immediately to end the day and get your reward!
=====
Congratulations, You've finished the game!!!! YYYYYYYYYY!!!!!!
=====
ACHIEVEMENT
Total Delivered Items : 1
Time elapsed          : 4
Earned money          : 200
```

## 5.13 Data Test 13

Save game diuji dengan menggunakan command SAVE\_GAME untuk menyimpan progress dari permainan. Untuk *testing*, permainan disimpan dalam file bernama 'testingspeed' seperti berikut.

```
>> SAVE_GAME
Save file as (filename):
testingspeed
Saving testingspeed...
...
Your progress has been saved!
```

Seperti yang dilihat, program menampilkan bahwa progress telah berhasil disimpan. Untuk melihat apakah penyimpanan berhasil dilakukan dengan baik atau tidak, *testing* dilanjutkan dengan

STEI- ITB	IF2110-TB-05-02	Halaman 30 dari 42 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.		

membuka folder saved-file dan melihat apakah ada file testingspeed.txt serta konten dari txt file tersebut.

```
data > saved-file > testingspeed.txt
1 tes_config
2 0 1 13
3 38
4 3000
5 0
6 1
7 0
8 0
9 3
10 0
11 5
12 -1 -1
13 -1 -1
14 -1 -1
15 -1 -1
16 -1 -1
17 0
18 4
19 2 B M N -1
20 3 F E N -1
21 5 C D N -1
22 5 G N P 10
23
```

Seperti yang dilihat, testingspeed.txt berhasil dibuat. Selain itu, isi dari testingspeed.txt adalah progress dari permainan sebelumnya. *Testing* dilanjut untuk melihat apakah fungsi dapat menangani kasus save yang menulis ulang suatu file yang sudah ada. Dilakukan save dengan input filename yang sama yakni 'testingspeed'.

```
>> SAVE_GAME
Save file as (filename):
testingspeed
File already exists.
Would you like to overwrite changes (yes/no)? : no
Save cancelled.

>>
```

Seperti yang dilihat, save dibatalkan dan isi dari testingspeed.txt tetap sama. *Testing* terakhir dilanjutkan dengan melakukan overwrite terhadap file yang sudah ada.

```
>> SAVE_GAME
Save file as (filename):
testingspeed
File already exists.
Would you like to overwrite changes (yes/no)? : yes
Saving testingspeed....
...
Your progress has been saved!
```

Dapat dilihat bahwa konten dari testingspeed.txt telah berubah dengan progres terbaru.

## 6 Test Script

No.	Bagian	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Main Menu	New Game	Memeriksa apakah permainan baru dapat dijalankan	Melakukan compile dan memasukkan command NEW_GAME serta	Data Test 1	File <i>config</i> berhasil dibaca dan muncul <i>interface</i> di awal game	Sama seperti hasil yang diharapkan

No.	Bagian	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
				nama file <i>config</i>			
		Load Game	Memeriksa apakah permainan yang sudah disimpan dapat dijalankan	Melakukan compile dan memasukkan command LOAD_GAME dilanjut nama <i>saved</i> file		Apabila tidak ada saved file, load tidak dilakukan. Apabila ada, saved file yang dipilih berhasil dibaca dan muncul <i>interface</i>	Sama seperti hasil yang diharapkan
		Exit	Memeriksa apakah pemain dapat keluar dari main menu	Melakukan compile dan memasukkan command EXIT		Pemain dikeluarkan dari permainan	Sama seperti hasil yang diharapkan
2	Mekanisme Waktu	Normal	Memeriksa apakah mekanisme waktu pada setiap kondisi berjalan sesuai spesifikasi	Melakukan perpindahan lokasi dan melihat pertambahan waktu yang terjadi	Data Test 2	Waktu bertambah 1 setiap MOVE	Sama seperti hasil yang diharapkan
		Efek Heavy Item				Waktu bertambah 1 dan ditambah 1 lagi untuk setiap Heavy Item di tas	Sama seperti hasil yang diharapkan
		Speed Boost				Waktu bertambah 1 setiap pemain berpindah 2 lokasi (ability habis setelah berpindah 10 lokasi)	Sama seperti hasil yang diharapkan
		Speed Boost + Efek Heavy Item				Mekanisme waktu seperti pada Heavy Item, Speed Boost teranulir	Sama seperti hasil yang diharapkan
3	Daftar Pesanan		Melihat apakah daftar pesanan bekerja dengan baik atau tidak	Menjalankan program dan menaruh print di awal program (khusus testing saja) untuk melihat isi daftar pesanan	Data Test 3	Daftar pesanan dari file <i>config</i> dan terurut berdasarkan waktu pesanan masuk	Sama seperti hasil yang diharapkan
4	To Do List		Memeriksa apakah to-do list bekerja dengan baik atau tidak	Menuliskan command TO_DO dan melihat isi to-do list		Isi item pada To-Do list adalah pesanan yang waktu masuknya telah lewat	Sama seperti hasil yang diharapkan
5	In Progress List		Memeriksa apakah in-progress list berjalan dengan	Melakukan pick-up dan drop-off, dilanjut dengan	Data Test 4	Item masuk ke in-progress list setelah pick-up	Sama seperti hasil yang



No.	Bagian	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
			baik	command IN_PROGRESS		dan terhapus setelah di drop-off	diharapkan
6		Tas	Memeriksa apakah tas dapat bekerja sesuai ketentuan	Melakukan pick-up, drop-off, dan melihat in-progress serta to-do list untuk melihat tas	Data Test 5	Item yang dipick-up akan masuk ke tas dan dihapus dari tas saat drop-off dan tidak dapat menampung lebih dari kapasitasnya	Sama seperti hasil yang diharapkan
7	Jenis Item	Normal Item	Memeriksa apakah karakteristik setiap item berfungsi sebagaimana harusnya	Melakukan drop-off pada setiap item (khusus VIP item, dilakukan pick-up pada item lain saat ada VIP item di to-do list)	Data Test 6	Setelah drop-off, pemain mendapat 200 yen	Sama seperti hasil yang diharapkan
		Heavy Item				Pergerakan pemain melambat. Setelah drop-off, pemain mendapat 400 yen dan Speed Boost	Sama seperti hasil yang diharapkan
		Perishable Item				Item menghilang dari to-do list apabila melebihi batas waktu. Setelah drop-off, pemain mendapat 400 yen dan Increase Capacity	Sama seperti hasil yang diharapkan
		VIP Item				Pengantaran item diprioritaskan. Setelah drop-off, pemain mendapat 600 yen dan Return to Sender.	Sama seperti hasil yang diharapkan
8	Ability	Speed Boost	Memastikan setiap ability berfungsi dengan baik	Berpindah lokasi dengan setelah drop-off Heavy Item alias saat Speed Boost teraktivasi	Data Test 2	Waktu bertambah 1 setiap pemain berpindah 2 lokasi (ability habis setelah berpindah 10 lokasi)	Sama seperti hasil yang diharapkan
		Increase Capacity		Melihat bag capacity setelah drop-off Perishable Item	Data Test 5	Kapasitas tas bertambah 1	Sama seperti hasil yang diharapkan
		Return to Sender		Mencoba command RETURN setelah	Data Test 7	Return dapat dilakukan, item di	Sama seperti

No.	Bagian	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
				drop-off VIP item		atas tas kembali ke lokasi pick-up	hasil yang diharapkan
9	Gadget	Kain Pembungkus Waktu	Memastikan gadget bekerja dengan baik dan benar	Membeli setiap gadget dan menggunakannya	Data Test 8	Batas waktu Perishable Item teratas pada tas kembali ke durasi semula. Apabila item teratas bukan tipe perishable maka gadget digunakan sia-sia	Sama seperti hasil yang diharapkan
		Senter Pembesar				Kapasitas tas meningkat sebesar dua kali lipat (tidak bisa melebihi batas max. yakni 100)	Sama seperti hasil yang diharapkan
		Pintu Kemana Saja				Pemain berpindah lokasi tanpa menambah unit waktu	Sama seperti hasil yang diharapkan
		Senter Pengecil				Menghilangkan efek dari heavy item pada tumpukan teratas tas. Apabila item teratas tas bukan heavy item, gadget digunakan sia-sia	Sama seperti hasil yang diharapkan
		Mesin Waktu			Data Test 2	Waktu berkurang sebanyak 50. Apabila hasilnya negatif, waktu jadi 0	Sama seperti hasil yang diharapkan
10	Inventory Gadget		Memastikan inventory berfungsi sesuai spesifikasi	Membeli gadget dan mencoba command INVENTORY	Data Test 9	Gadget yang terbeli langsung masuk inventory dan dihapus setelah digunakan. Tidak bisa menampung lebih dari 5 gadget.	Sama seperti hasil yang diharapkan
11	Peta		Memastikan peta dapat menampilkan posisi pemain/bangunan dengan warna	Mencoba command MAP	Data Test 10	Posisi berwarna oranye untuk pemain, merah untuk pick-up, biru untuk drop-off, hijau untuk	Sama seperti hasil yang diharapkan

No.	Bagian	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
			sesuai spesifikasi			posisi yang dapat dicapai, dan hitam untuk lainnya	
12	Lokasi dan Command	MOVE	Memeriksa apakah command dapat berfungsi sesuai spesifikasi	Menuliskan command pada terminal dengan kasus-kasus yang sesuai spesifikasi	Data Test 11	Pemain berpindah lokasi	Sama seperti hasil yang diharapkan
		PICK_UP				Item dipick-up	Sama seperti hasil yang diharapkan
		DROP_OFF				Item didrop-off	Sama seperti hasil yang diharapkan
		HELP				Menjelaskan command yang tersedia	Sama seperti hasil yang diharapkan
		BUY				Pemain dapat membeli gadget. Hanya dapat dilakukan saat pemain berada di Headquarters	Sama seperti hasil yang diharapkan
		TO_DO			Data Test 3	Menampilkan to-do list	Sama seperti hasil yang diharapkan
		IN_PROGRESS			Data Test 4	Menampilkan in-progress list	Sama seperti hasil yang diharapkan
		RETURN			Data Test 7	Item di tumpukan teratas pada tas Kembali ke lokasi pick-up. Hanya dapat diakses saat memiliki ability Return to Sender	Sama seperti hasil yang diharapkan
		INVENTORY			Data Test 9	Menampilkan isi list inventory dan pemain dapat menggunakan gadget yang dimiliki	Sama seperti hasil yang diharapkan
		MAP			Data	Menampilkan	Sama

No.	Bagian	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
					Test 10	peta	seperti hasil yang diharapkan
		SAVE_GAME			Data Test 13	Menyimpan progress pemain	Sama seperti hasil yang diharapkan
13	Game Flow		Melihat apakah alur permainan sudah sesuai atau tidak	Memainkan permainan dari awal hingga end game	Data Test 12	Menampilkan pesan bahwa game selesai apabila semua pesanan selesai diantar dan pemain sudah balik ke HQ.	Sama seperti hasil yang diharapkan
14	Save Game		Melihat apakah penyimpanan progress pada file eksternal bekerja dengan lancar atau tidak	Menuliskan command SAVE_GAME	Data Test 13	State permainan disimpan dalam file. Apabila sebelumnya file tersebut sudah ada, pemain dapat memilih untuk tetap overwrite atau tidak	Sama seperti hasil yang diharapkan

## 7 Pembagian Kerja dalam Kelompok

No	NIM	Nama	Pembagian Kerja
1	13520074	Eiffel Aqila Amarendra	ADT Point, ADT List Dinamis, ADT Matrix, command PICK_UP, command DROP_OFF, command RETURN, mekanisme ability, laporan
2	13520076	Claudia	ADT Queue, ADT Linked List, ADT Pesanan, command TO__DO, command IN_PROGRESS, laporan
3	13520079	Ghebyon Tohada Nainggolan	ADT List, Buy, Inventory, mekanisme penggunaan gadget, mekanisme jenis-jenis item, laporan
4	13520083	Sarah Azka Arief	ADT Mesin Karakter, ADT Mesin Kata, inialisasi <i>config</i> , save/load game, ability speed boost, validasi command, laporan
5	13520095	Firizky Ardiansyah	ADT Stack, ADT Peta, program utama, command MOVE, command MAP, mekanisme waktu, VIP Item, gadget senter pengecil, mekanisme kompilasi,

			laporan
--	--	--	---------

## 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. Tugas kalian adalah membantu Mobita membuatkan aplikasi yang akan membantunya melacak pesanan, navigasi peta, mengambil dan menurunkan barang.

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

### 8.2 Notulen Rapat

#### Rapat Kelompok 5 K02

Hari, tanggal : Minggu, 24 Oktober 2021

Waktu : 13.00 WIB - 15.25 WIB

Pembahasan :

#### Objek Program

- Daftar pesanan
- To do list
- In Progress List
- Tas
- Jenis Item
- Ability
- Gadget
- Peta
- Duid
- Waktu

#### Command

- Command dibaca dengan Mesin karakter, kata
- ADT dan implementasinya
- Move
- Pick\_up
- drop\_off → Termasuk mekanisme ability
- Map → ADT Point, List dinamis, Matriks
- To\_do → Queue, linked list
- In\_progress → Linked list
- Buy
- Inventory → List Statis
- Help
- Save\_game

- Return
- Load game
- Laporan (bareng bareng)
- Buat konfigurasi file

#### **Pertanyaan yang mau diajukan**

- Bedanya end game sama exit game?
- Command dibaca oleh mesin karakter? marknya apa?
- Linux
- Senter pembesar meningkatkan kapasitas tas. Yang ditingkatkan apanya?
- Mapnya satu atau banyak?
- Beda konfigurasi level dan permainan apa? + Jumlahnya?

### **8.3 Log Activity Anggota Kelompok**

#### 13520074 / Eiffel Aqila Amarendra

31 Oktober 2021 : Menambahkan file ADT Point, ADT List Dinamis, dan ADT Matriks  
 5 November 2021 : Merevisi ADT  
 11 November 2021 : Menambahkan laporan bagian ringkasan dan program utama  
 12 November 2021 : Menambahkan command Pick\_Up  
 15 November 2021 : Menambahkan command Drop\_Off  
 16 November 2021 : Merevisi command Pick\_Up dan Drop\_Off, menambahkan laporan bagian Struktur Data (ADT) Point, List Dinamis, Matriks, dan Stack  
 17 November 2021 : Menambahkan command Return dan merevisi beberapa syntax yang salah  
 18 November 2021 : Menambahkan driver ADT Point, ADT List Dinamis, dan ADT Matriks  
 Merevisi bagian laporan yang masih ambigu, kurang lengkap, dan salah

#### 13520076 / Claudia

27 Oktober 2021 : Menambahkan spesifikasi, pedoman, form asistensi  
 8 November 2021 : Membuat ADT Pesanan  
 9 November 2021 : Melakukan revisi pada ADT Pesanan, memodifikasi ADT Queue, ADT Linked List  
 10 November 2021 : Melakukan revisi ADT Pesanan  
 11 November 2021 : Melakukan revisi ADT Pesanan  
 12 November 2021 : Menambahkan command TO\_DO, command IN\_PROGRESS  
 13 November 2021 : Melakukan revisi pada ADT Queue  
 16 November 2021 : Melakukan revisi pada command, menambahkan driver ADT Pesanan dan ADT Queue, menambahkan laporan bagian Struktur Data (ADT) Pesanan, Queue, Linked List, lampiran  
 17 November 2021 : Merapikan program, menambahkan driver ADT Linked List, menambahkan dan melengkapi laporan  
 18 November 2021 : Finalisasi laporan

#### 13520079 / Ghebyon Tohada Nainggolan

10 November 2021 : Menambahkan "liststat.c" dan "liststat.h", membuat kerangka implementasi Inventory pada "command.c", Membuat kerangka Buy

15 November 2021 : Menambahkan reward dari pengantaran Perishable item ,  
Pengerjaan Laporan.  
16 November 2021 : Memperbaiki validasi pada pemakaian gadget pada inventory,  
Memperbaiki ADT liststat.c  
17 November 2021 : Pengerjaan Laporan Bagian 2 Spesifikasi Tambahan  
18 November 2021 : Finalisasi Laporan

13520083 / Sarah Azka Arief

28 Oktober 2021 : Menambahkan ADT Mesin Karakter, ADT Mesin Kata, driver ADT  
Mesin Karakter dan Kata, serta dummy *config* file  
2 November 2021 : Melakukan revisi mesin karakter, menambahkan inisialisasi *config*  
5 November 2021 : Merevisi fungsi getLoc pada inisialisasi *config*  
9 November 2021 : Menambahkan ability Speed Boost dan membuat kerangka command  
15 November 2021 : Merevisi inisialisasi *config*, memperbaiki mesin kata, menambahkan  
save dan load function  
16 November 2021 : Melakukan penyesuaian save/load state serta inisialisasi *config*,  
menambah laporan bagian Struktur Data (ADT) Mesin Kata/Karakter  
17 November 2021 : Melakukan *testing* untuk mengisi laporan bagian Data Test dan Test  
Script, memperbaiki minor bugs serta Speed Boost, menambah  
handling *overwrite file* pada save function, menyesuaikan init/save/load  
function  
18 November 2021 : Merapikan driver ADT Mesin Karakter/Kata dan laporan

13520095 / Firizky Ardiansyah

24 Oktober 2021 : Membuat repository di Github  
31 Oktober 2021 : Membuat Makefile dan struktur program  
1 November 2021 : Debugging file-reader, merapikan struktur folder  
3 November 2021 : Membuat variabel global uang, waktu, dan map serta membuat cetak  
biru algoritma  
9 November 2021 : Membuat ADT stack dengan tipe elemen yang disesuaikan  
10 November 2021 : Membuat interaksi move dan membuat ADT peta, membuat interface  
untuk display peta, membuat interface awal program, dan  
menambahkan perintah *Exit*  
11 November 2021 : Debug makefile  
12 November 2021 : Memperbaiki bug dalam program, terutama terkait perintah  
14 November 2021 : Menambah fitur pada Adjacency matriks dan membangun prosedur-  
prosedur pembantu yang berhubungan dengan lokasi player  
16 November 2021 : Memperbaiki ADT Move dan menambahkan mekanisme waktu  
berdasarkan kasus-kasus yang diberikan di spesifikasi, menambahkan  
mekanisme VIP item, menghubungkan ADT ke dalam perintah move  
17 November 2021 : Memperbaiki perintah buy, memperbaiki dan menambahkan efek dari  
gadget, memperbaiki inventory, menambahkan mekanisme senter  
pengecil, memperbaiki pintu kemana saja, dan mewarnai interface  
18 November 2021 : Menambahkan driver, menambahkan final rewards, dan menambahkan  
makefile pada driver

## 8.4 Form Asistensi Tugas Besar



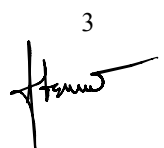

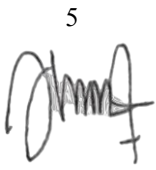

**Form Asistensi Tugas Besar**  
**IF2110/Algoritma dan Struktur Data**  
**Sem. 1 2021/2022**

No. Kelompok/Kelas : 5/K02  
Nama Kelompok : Alstrukyey  
Anggota Kelompok (Nama/NIM) :  
1. Eiffel Aqila Amarendra/13520074  
2. Claudia/13520076  
3. Ghebyon Tohada Nainggolan/13520079  
4. Sarah Azka Arief/13520083  
5. Firizky Ardiansyah/13520095  
  
Asisten Pembimbing : Ricky Fernando/13518062







---



## Asistensi I

<b>Tanggal : 27 Oktober 2021</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"><li>- Bedanya end game &amp; exit game: ga terlalu dipentingin, kalo mau ditambahin boleh tapi ga jadi prioritas utama</li><li>- Command untuk mesin karakter itu marknya apa setelah selesai baca command? Pake new line (enter) aja marknya</li><li>- Di speknya harus bisa dijalanin di linux? Asalkan tidak ada yang aneh seharusnya aman dijalankan di linux, tapi pastiin aja bisa dijalankan di linux dan windows</li><li>- Di tas kapasitas maksimum 100, di senter pembesar itu yang ditingkatkan apanya? Tas by default awalnya cuma 3 tapi kalo pake senter pembesar berulang-ulang maksimumnya cuma bisa sampe 100. Misalnya kapasitasnya sampe 99, trus dipakein senter pembesar mentoknya di 100</li><li>- Bikin di main menunya bisa milih map atau cuma 1? Mapnya itu input dari konfigurasi, untuk new game kan diminta konfigurasi level itu ada mapnya juga</li><li>- Untuk new game, disuruh input konfigurasi level. Jumlah dan isi tiap konfigurasi itu dibebaskan ke kami? Buat file konfigurasi sesuai spek atau buat file konfigurasi sendiri, tetapi pas demo seharusnya filenya dari asisten</li></ul>
<b>Tempat : Google Meet</b>	
<b>Kehadiran Anggota Kelompok:</b> <div><div>1</div><div></div><div>13520074/Eiffel Aqila Amarendra</div></div>	
<div><div>2</div><div></div><div>13520076/Claudia</div></div>	
<div><div>3</div><div></div><div>13520079/Ghebyon Tohada Nainggolan</div></div>	
<div><div>4</div><div></div><div>13520083/Sarah Azka Arief</div></div>	
<div><div>5</div><div></div><div>13520095/Firizky Ardiansyah</div></div>	
	<b>Tanda Tangan Asisten:</b> <div></div>

## Asistensi II

<b>Tanggal : 8 November 2021</b>	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"><li>- Di spesifikasi terbaru ditulis kalau save dimasukkan ke configuration file? Maksudnya bukan disave di configuration file, tapi update configuration filenya. Jadi kalau mau diloat, pakai 1 file yang sama.</li><li>- Kita pake makefile, kalo di windows harus pake gitbash, apakah boleh? Kayaknya ga masalah nanti tulis penjelasan tambahan aja di README.</li><li>- Setelah pesanan diantar selesai semuanya, Mobita harus kembali ke headquarter, itu tetap pakai command MOVE ya berarti? Iya tetap pakai command MOVE.</li></ul>
<b>Tempat : Google Meet</b>	
<b>Kehadiran Anggota Kelompok:</b> <div><div>1</div><div></div><div>13520074/Eiffel Aqila Amarendra</div></div> <div><div>2</div><div></div><div>13520076/Claudia</div></div> <div><div>3</div><div></div><div>13520079/Ghebyon Tohada Nainggolan</div></div> <div><div>4</div><div></div><div>13520083/Sarah Azka Arief</div></div> <div><div>5</div><div></div><div>13520095/Firizky Ardiansyah</div></div>	
<b>Tanda Tangan Asisten:</b> <div></div>	