

# **LAPORAN TUGAS BESAR**

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


### **Mobilita**

Dipersiapkan oleh:

Kelompok 2:

1. Hafidz Nur Rahman Ghozali (13520117)
2. Marchotridyo (13520119)
3. Johannes Winson Sukiatmodjo (13520123)
4. Muhammad Gilang Ramadhan (13520137)
5. Steven Gianmarg H. Siahaan (13520145)
6. Mohamad Hilmi Rinaldi (13520149)

**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-02-03</i>		<i>23</i>
		<i>Revisi</i>	<i>1</i>	<i>18 November 2021</i>

## Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas .....	3
2.1 Mekanisme Load Game dan Save Game .....	3
2.2 VIP Item .....	3
2.3 Ability Return To Sender .....	4
2.4 Gadget Senter Pengecil .....	4
3 Struktur Data (ADT) .....	4
3.1 Stack .....	4
3.2 Queue .....	4
3.3 List.....	5
3.4 Map.....	5
3.5 Matrix .....	5
3.6 Mesin Akarakter dan Mesin Kata .....	6
3.7 List Statis.....	6
3.8 List Dinamis dan Bangunan .....	6
3.9 Point .....	7
3.10 File Machine .....	7
3.11 Item.....	7
3.12 Time.....	7
4 Program Utama .....	8
5 Algoritma-Algoritma Menarik.....	8
5.1 Algoritma updatePerishable .....	8
5.2 Algoritma getAdjacentBuildings.....	8
6 Data Test .....	8
6.1 Data Test 1 .....	8
6.2 Data Test 2 .....	9
6.3 Data Test 3 .....	10
6.4 Data Test 4 .....	11
6.5 Data Test 5 .....	11
6.6 Data Test 6 .....	12
6.7 Data Test 7 .....	13
6.8 Data Test 8 .....	13
6.9 Data Test 9 .....	14
6.10 Data Test 10.....	14
6.11 Data Test 11 .....	15
6.12 Data Test 12.....	16
6.13 Data Test 13 .....	16
6.14 Data Test 14.....	16
7 Test Script .....	17
8 Pembagian Kerja dalam Kelompok .....	18
9 Lampiran .....	19
9.1 Deskripsi Tugas Besar 2.....	19
9.2 Notulen Rapat.....	20
9.3 Log Activity Anggota Kelompok.....	20

# 1 Ringkasan

Mobilita adalah sebuah permainan yang bertujuan untuk membantu Mobita dalam melakukan tugasnya sebagai kurir. Di dalam permainan ini, pemain bertindak sebagai Mobita. Pemain bertugas untuk mengantarkan semua pesanan yang ia terima. Mobita Pemain harus menyelesaikan pesanan sebanyak mungkin dan dalam waktu yang sesingkat mungkin.

Laporan tugas besar ini berisi deskripsi umum mengenai game, penjelasan struktur data yang digunakan, penjelasan program utama dari game, algoritma-algoritma yang digunakan, data-data pengetesan game, dan lampiran pendukung.

Permainan ini dibuat dalam bahasa C dan berbasis CLI (*Command Line Interface*). Permainan ini memanfaatkan pengetahuan mengenai Struktur Data yang diperoleh dari kuliah IF2110 – Algoritma dan Struktur Data. Pada saat memulai permainan, pemain dapat memilih untuk melakukan load game, new game, atau exit. Setelah game dimulai, pemain dapat menggunakan berbagai perintah. Perintah MOVE digunakan untuk berpindah posisi ke tempat yang dapat dijangkau. Perintah PICK\_UP digunakan untuk mengambil pesanan di suatu lokasi. Perintah DROP\_OFF digunakan untuk mengantarkan pesanan jika posisinya sesuai dengan posisi pengantaran pesanan. perintah MAP digunakan untuk menampilkan peta permainan. Perintah TO\_DO digunakan untuk menampilkan pesanan yang terdapat di To Do List. Perintah IN\_PROGRESS digunakan untuk menampilkan pesanan yang sedang dikerjakan. Perintah BUY digunakan untuk membeli Gadget di Headquarters. Perintah INVENTORY digunakan untuk menampilkan isi inventory gadget yang dimiliki pemain. Perintah HELP digunakan untuk menampilkan seluruh perintah yang valid beserta penjelasannya. Game ini berakhir ketika semua pesanan telah selesai diantarkan oleh Mobita.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Mekanisme Load Game dan Save Game

Pada program ini terdapat fitur load game dan save game. Pada mekanisme Save Game, pemain dapat menyimpan keadaan pada saat permainan sedang berlangsung. Keadaan permainan akan disimpan ke dalam sebuah file. Sedangkan Load Game digunakan ketika awal permainan dengan memilih file yang disimpan sebelumnya. Dengan adanya mekanisme Load Game, pemain dapat melanjutkan kondisi game sebelumnya.

### 2.2 VIP Item

Dalam permainan ini, terdapat Item berjenis VIP yang harus disegerakan. Ketika pesanan dengan jenis tersebut sudah masuk ke dalam To Do List, Mobita harus segera mengambil dan mengantarkannya. Mobita tidak akan mengambil pesanan item lain sebelum menyelesaikan pesanan VIP Item. Setelah menyelesaikan pesanan VIP Item, Mobita akan mendapatkan ability Return To Sender dan mendapatkan uang sebesar 600 Yen.

## 2.3 Ability Return To Sender

Ability ini didapatkan oleh Mobita ketika berhasil mengantarkan pesanan berjenis VIP Item. Dengan adanya ability ini, Mobita dapat mengembalikan item teratas pada tas ke lokasi pick up-nya. Pesanan tersebut akan dikembalikan ke dalam TO Do List di urutan terakhir. Ability ini akan mereset waktu dari perishable item. Namun, ability ini tidak dapat digunakan untuk VIP Item.

## 2.4 Gadget Senter Pengecil

Senter Pengecil digunakan untuk menghilangkan efek dari heavy item jika berada di tas paling atas. Efek ini bertahan selama Mobita belum melakukan DROP OFF atau RETURN sejak penggunaan gadget ini.

# 3 Struktur Data (ADT)

## 3.1 Stack

- Sketsa struktur data : ADT Stack berisi buffer berupa list of Item dan idxTop yaitu index elemen teratas pada stack. ADT ini memiliki konstruktor untuk membuat stack kosong, dan selektor yang mengembalikan idxTop dan isi pada elemen teratas di stack. Selain itu, terdapat beberapa primitif seperti push untuk menambahkan elemen pada posisi teratas, pop untuk mengambil elemen teratas, isEmpty dan isFull untuk mengecek isi di dalam stack.
- Permasalahan yang diselesaikan : Daftar item di dalam tas.
- Alasan pemilihan : Pesanan yang sudah di pick up memiliki karakteristik yang sama dengan Stack, yaitu LIFO (*Last In First Out*).
- ADT ini diimplementasikan di dalam folder Stack, yaitu pada file stack.h dan stack.c.

## 3.2 Queue

- Sketsa struktur data : ADT Queue terdiri dari idxHead yang berisi index elemen pertama, idxTail yang berisi index elemen terakhir, buffer, yaitu list berbagai Item. Dalam ADT ini terdapat konstruktor yang membuat sebuah Queue kosong. Selain itu, terdapat primitif untuk pemrosesan Queue, contohnya enqueue untuk menambahkan sebuah elemen di akhir barisan, dan dequeue untuk mengambil elemen pertama dari Queue.
- Permasalahan yang diselesaikan : Inventarisasi daftar pesanan yang ada.
- Alasan pemilihan : Daftar pesanan memiliki karakteristik yang sama dengan Queue, yaitu FIFO (*First In First Out*).
- ADT ini diimplementasikan di dalam folder Queue, yaitu pada file queue.h dan queue.c.

### 3.3 List

- Sketsa struktur data : ADT List berisi sebuah alamat dari elemen pertama. Setiap elemen di dalam list terdiri dari info yang berupa Item dan next yang berupa alamat dari elemen selanjutnya. ADT ini memiliki konstruktor untuk membuat list kosong. Selain itu, terdapat primitif-primitif yang digunakan untuk pemrosesan list, contohnya insert dan delet, baik itu di elemen pertama, terakhir, atau di tempat tertentu. Terdapat selektor yang digunakan untuk mengambil nilai seperti getElmt, dan selektor untuk mengubah nilai, yaitu setElmt.
- Permasalahan yang diselesaikan : Inventarisasi pesanan dalam daftar pesanan yang harus dikerjakan dan pesanan yang sedang dikerjakan
- Alasan pemilihan : Daftar pesanan sering berubah karena penambahan atau penghapusan pesanan, sehingga sangat cocok menggunakan ADT List dengan representasi berkait yang cukup efisien.
- ADT ini diimplementasikan di dalam folder List\_linked, yaitu pada file list\_linked.h dan list\_linked.c

### 3.4 Map

- Sketsa struktur data : ADT Map berisi locationMatrix yang bertipe Matrix dan adjacencyMatrix yang juga bertipe Matrix. Selain itu, di dalam ADT ini juga terdapat struktur data adjacentLocation yang digunakan untuk mendapatkan lokasi yang dapat dicapai dari lokasi yang sekarang. ADT ini memiliki konstruktor untuk membuat Map baru sesuai dengan lokasi dan adjacency matriks yang ada. Selain itu, terdapat primitif-primitif untuk pemrosesan Map, seperti displayMap, getLocationName, dan getAdjacentLocation.
- Permasalahan yang diselesaikan : Fitur peta di dalam permainan
- Alasan pemilihan : Di dalam Peta, terdapat banyak bangunan yang saling terhubung. Oleh karena itu, ADT Map bertujuan untuk menggabungkan seluruh info yang berkaitan dengan peta.
- ADT ini diimplementasikan di dalam folder Map, yaitu pada file map.h dan map.c

### 3.5 Matrix

- Sketsa struktur data : ADT Matrix terdiri dari content, yaitu array 2D, rowEff untuk menyimpan banyaknya baris dan colEff untuk menyimpan banyaknya kolom di dalam matrix.
- Permasalahan yang diselesaikan : Letak bangunan di dalam peta, representasi hubungan antarbangunan ke dalam matriks adjacency
- Alasan pemilihan : Posisi bangunan terletak pada koordinat tertentu dalam 2D sehingga cocok dengan representasi Matrix. Hubungan antarbangunan juga lebih mudah dipahami dalam bentuk matriks adjacency.
- ADT ini diimplementasikan di dalam folder Matrix, yaitu pada file matrix.h dan matrix.c

### 3.6 Mesin Akarakter dan Mesin Kata

- Sketsa struktur data : ADT Mesin Kata berisi sebuah kata yang disimpan dalam array of char dan length yang menyimpan panjang dari kata tersebut. ADT ini berisi primitif-primitif seperti startWord untuk mulai membaca kata pertama, advWord untuk membaca kata selanjutnya, copyWord untuk mengakuisisi kata yang dibaca, isWordEqual untuk membandingkan 2 kata, printWord untuk mencetak word dan wordToInt untuk konversi kata yang dibaca menjadi bertipe integer.
- Permasalahan yang diselesaikan : Pembacaan command dan validasi command.
- Alasan pemilihan : ADT ini sesuai dengan cara pembacaan setiap command yang masuk.
- ADT ini diimplementasikan di dalam folder WordMachine, yaitu pada file charmachine.h, charmachine.c, wordmachine.h, dan wordmachine.c.

### 3.7 List Statis

- Sketsa struktur data : ADT ListPos terdiri dari contents sebanyak LP\_CAPACITY yang bertipe LP\_ElType. Dalam ADT ini terdapat konstruktor CreateListPos untuk membentuk sebuah list statis kosong. Selain itu, terdapat juga primitif-primitif lain seperti getEmptySlot untuk mendapatkan indeks slot terkecil yang masih kosong, addToInventory untuk menambahkan gadget ke inventory, dan showInventory untuk menunjukkan isi inventory.
- Permasalahan yang diselesaikan : Inventarisasi gadget yang sudah dibeli.
- Alasan pemilihan : Gadget yang sudah dibeli harus disimpan dalam list tersendiri yang mana list tersebut sudah memiliki kapasitas yang fix.
- ADT ini diimplementasikan di dalam folder ListPos, yaitu pada file listpos.h dan listpos.c

### 3.8 List Dinamis dan Bangunan

- Sketsa struktur data : ADT List Dinamis berisi buffer yang berupa list of Bangunan, nEff untuk mencatat banyaknya bangunan, dan capacity untuk kapasitas maksimal di dalam list. Di dalam bangunan tersimpan POINT lokasi bangunan dan nama bangunan. Di dalam ADT ini terdapat konstruktor untuk membuat bangunan dengan posisi dan nmama yang sudah terdefinisi, dan konstruktor untuk membuat list dinamis yang kosong. Selain itu, terdapat selektor untuk mendapatkan bangunan di dalam list dinamis, panjang list, dan kapasitas list. Terdapat beberapa primitif di dalam ADT ini, yaitu dealocate untuk mendealokasi suatu elemen list, isIdxEff untuk mengecek index yang efektif, dan insertLast untuk menambahkan bangunan ke dalam list dinamis di urutan terakhir.
- Permasalahan yang diselesaikan : Invetarisasi daftar bangunan yang ada di dalam game.
- Alasan pemilihan : Kumpulan bangunan di dalam game harus disatukan dalam sebuah List. List Dinamis cocok karena bisa menghemat memory yang digunakan untuk penyimpanannya.
- ADT ini diimplementasikan di dalam folder ListDin, yaitu pada file listdin.h dan listdin.c.

### 3.9 Point

- Sketsa struktur data : ADT Point terdiri dari posisi absis dan ordinat yang bertipe integer. Di Dalam ADT ini terdapat konstruktor untuk membuat POINT baru dan menuliskan POINT yang terdefinisi. Selain itu terdapat selektor untuk mengembalikan nilai absis atau ordinat dari suatu point.
- Permasalahan yang diselesaikan : Posisi dari setiap bangunan dan posisi dari Mobita
- Alasan pemilihan : Bangunan terletak di dalam matriks 2D, sehingga posisinya dapat direpresentasikan dengan point secara baik.
- ADT ini diimplementasikan di dalam folder Point, yaitu pada file point.h dan point.c

### 3.10 File Machine

- Sketsa struktur data : ADT ini berisi contents, yaitu array of Word dan length yang merupakan panjang array. Di dalam ADT ini terdapat prosedur startLine untuk memulai pembacaan baris, advLine untuk membaca baris selanjutnya, copyLine untuk mengakuisisi baris yang dibaca, printLine untuk mencetak baris yang dibaca.
- Permasalahan yang diselesaikan : Pembacaan setiap kata pada file konfigurasi
- Alasan pemilihan : ADT ini sesuai dengan cara pembacaan kata pada file konfigurasi.
- ADT ini diimplementasikan di dalam folder FileMachine, yaitu pada file filemachine.h dan filemachine.c

### 3.11 Item

- Sketsa struktur data : ADT Item berisi waktu masuk sebuah pesanan, tempat asal dan tujuan pesanan, jenis pesanan, nilai yang dimiliki sebuah pesanan. Selain itu, ADT ini juga memiliki sisa waktu hangus khusus untuk item berjenis *perishable item*. ADT ini memiliki konstruktor untuk membentuk item baru dengan jenis dan lokasi asal dan tujuan yang sudah terdefinisi.
- Permasalahan yang diselesaikan : Pengorganisasian setiap pesanan yang masuk.
- Alasan pemilihan : Setiap pesanan memiliki struktur data yang sama sehingga membuat struktur data baru untuk mengelolanya.
- ADT ini diimplementasikan di dalam folder Item, yaitu pada file item.h dan item.c

### 3.12 Time

- Sketsa struktur data : ADT berisi sebuah variabel bertipe integer untuk menyimpan waktu yang sedang berjalan. ADT ini memiliki konstruktor untuk membuat waktu menjadi posisi inisial. ADT ini juga memiliki primitif increaseTime untuk penambahan waktu.
- Permasalahan yang diselesaikan : mekanisme waktu yang berjalan.
- Alasan pemilihan : Waktu yang disimpan di dalam permainan sering berubah dan terdapat beberapa kombinasi perubahan, sehingga memerlukan pengolahan tersendiri.
- ADT ini diimplementasikan di dalam folder Time, yaitu pada file time.h dan time.c.

## 4 Program Utama

Program utama permainan ini terdapat pada file main.c. Program ini dimulai dengan menampilkan Main Menu yang menawarkan untuk New Game, Exit, atau Load Game. Pemain dapat mengetik '1' untuk New Game, '2' untuk Exit, dan '3' untuk Load Game. Ketika pemain memilih New Game atau Load Game, pemain akan diminta untuk memasukkan nama file konfigurasi yang sudah ada. Setelah itu, permainan dimulai. Pemain dapat memasukkan command HELP untuk mengetahui command apa saja yang diterima oleh game. Setiap command yang dimasukkan oleh pemain akan divalidasi terlebih dahulu.

Pemain dapat memasukkan command MAP untuk menampilkan peta permainan. Untuk berpindah ke lokasi yang terhubung, pemain dapat menggunakan command MOVE. Command PICK\_UP digunakan untuk mengambil pesanan di lokasinya saat ini. Pemain dapat menyelesaikan sebuah pesanan dengan melakukan DROP\_OFF pesanan pada lokasi yang benar. Pemain dapat melihat isi To Do List pesanan yang harus diantarkan dengan command TO\_DO dan menampilkan pesanan yang sedang diantarkan dengan command IN\_PROGRESS. Pemain dapat membeli gadget dengan menggunakan command BUY dan menampilkan gadget yang dimilikinya dengan command INVENTORY. Apabila pemain memiliki ability Return To Sender, pemain dapat menggunakannya dengan command RETURN. Permainan ini akan terus berjalan hingga pemain menyelesaikan tugasnya atau memilih untuk menyimpan game dengan command SAVE\_GAME.

## 5 Algoritma-Algoritma Menarik

### 5.1 Algoritma updatePerishable

Algoritma updatePerishables berguna untuk meng-*update perishable item* yang ada di dalam Bag dan In Progress List. Algoritma ini menarik karena kita mengecek semua pesanan di dalam Bag dan In Progress List dan membuat salinan pesanan tersebut dan mengganti dengan Bag dan In Progress List yang baru. Algoritma ini memanfaatkan kombinasi primitif push dan pop pada Bag serta primitif insertLast pada In Progress List.

### 5.2 Algoritma getAdjacentBuildings

Algoritma getAdjacentLocations berada pada ADT Map. Algoritma ini bertujuan untuk mengetahui lokasi-lokasi yang bisa dicapai dari sebuah lokasi. Algoritma ini memanfaatkan list bangunan dan matriks adjacency antarbangunan pada Map. Algoritma ini mengembalikan list dari index bangunan yang berhubungan dengan lokasinya.

## 6 Data Test

### 6.1 Data Test 1

Test ini dilakukan Untuk memastikan bahwa program sudah dapat dijalankan dan menampilkan tampilan Awal dan Main Menu game. Untuk melakukan kompilasi

STEI- ITB	IF2110-TB-02-03	Halaman 8 dari 23 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		





```
ENTER COMMAND: NEW GAME

Masukkan nama file konfigurasi: konfigurasi2.txt
File konfigurasi ditemukan!
Loading konfigurasi...

=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====
```

**Gambar 6.2** Pembacaan File Konfigurasi

### 6.3 Data Test 3

Test ini digunakan untuk memastikan COMMAND MAP berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
ENTER COMMAND: MAP

*****
*8      C  D B*
* E          *
*F      G    H *
*  I          *
*      J      *
* K          *
*      L      *
* M  N        O*
*      P      *
*A Q          *
*****

=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====
```

**Gambar 6.3** Pembacaan MAP

## 6.4 Data Test 4

Tes ini digunakan untuk memastikan COMMAND MOVE dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====

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: 3

Mobita sekarang berada di titik F (3,1)!
=====
Mobita berada di posisi F (3,1)
Waktu: 2
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 3
=====

ENTER COMMAND: 
```

**Gambar 6.4** Pembacaan Command MOVE

## 6.5 Data Test 5

Tes ini digunakan untuk memastikan COMMAND PICK\_UP dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
ENTER COMMAND: PICK_UP

[!] Pesanan tidak ditemukan!
=====
Mobita berada di posisi F (3,1)
Waktu: 2
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 3
=====

ENTER COMMAND: []
```

**Gambar 6.5** Pembacaan COMMAND PICK\_UP pada pesanan yang tidak ditemukan

```
=====

ENTER COMMAND: PICK_UP

Pesanan berupa Perishable Item berhasil diambil!
[!] Pantau durasi yang Mobita punya melalui command IN_PROGRESS.
=====
Mobita berada di posisi G (3,8)
Waktu: 5
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 7
=====

ENTER COMMAND: []
```

**Gambar 6.6** Pembacaan COMMAND PICK\_UP pada pesanan yang ditemukan

## 6.6 Data Test 6

Tes ini digunakan untuk memastikan COMMAND DROP\_OFF dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====

ENTER COMMAND: DROP_OFF

Pesanan Normal Item berhasil diantarkan.
Uang yang didapatkan: 200 Yen
=====
Mobita berada di posisi N (8,6)
Waktu: 9
Uang yang dimiliki: 15200 Yen
Jumlah pesanan yang harus dikerjakan: 8
=====
```

**Gambar 6.7** Pembacaan COMMAND DROP\_OFF pada pesanan yang diantarkan

```
=====
ENTER COMMAND: DROP_OFF

[!] Tidak ada pesanan yang dapat diantarkan!
=====
Mobita berada di posisi L (7,10)
Waktu: 12
Uang yang dimiliki: 15800 Yen
Jumlah pesanan yang harus dikerjakan: 8
=====
```

**Gambar 6.8** Pembacaan COMMAND DROP\_OFF ketika tidak ada pesanan

## 6.7 Data Test 7

Tes ini digunakan untuk memastikan COMMAND TO\_DO dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
ENTER COMMAND: TO_DO

Pesanan pada To Do List:
1. M -> B (Heavy Item)
2. B -> M (Perishable Item, sisa waktu 10)
3. G -> N (Heavy Item)
4. F -> E (Heavy Item)
5. C -> D (Perishable Item, sisa waktu 10)
6. G -> N (Heavy Item)
=====
Mobita berada di posisi G (3,8)
Waktu: 5
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 7
=====

ENTER COMMAND: 
```

**Gambar 6.9** Pembacaan COMMAND TO\_DO

## 6.8 Data Test 8

Tes ini digunakan untuk memastikan COMMAND IN\_PROGESS dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```

=====
ENTER COMMAND: IN_PROGRESS

Pesanan yang sedang diantarkan:
1. Perishable Item dengan sisa waktu 10 (Tujuan: N)
=====
Mobita berada di posisi G (3,8)
Waktu: 5
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 7
=====

ENTER COMMAND: 

```

**Gambar 6.10** Pembacaan COMMAND IN\_PROGRESS

## 6.9 Data Test 9

Tes ini digunakan untuk memastikan COMMAND BUY dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```

=====
ENTER COMMAND: BUY

Uang Mobita sekarang: 15000 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 kau beli? (ketik 0 jika ingin kembali)

ENTER COMMAND: 1

Kain Pembungkus Waktu berhasil dibeli!
Uang Mobita sekarang: 14200 Yen
=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 14200 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====

ENTER COMMAND: 

```

**Gambar 6.11** Pembacaan COMMAND BUY

## 6.10 Data Test 10

Tes ini digunakan untuk memastikan COMMAND INVENTORY dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
ENTER COMMAND: INVENTORY

1. Kain Pembungkus Waktu
2. -
3. -
4. -
5. -
Gadget mana yang ingin digunakan? (Ketik 0 jika ingin kembali)

ENTER COMMAND: 1

Kain Pembungkus Waktu berhasil digunakan!
[!] Gadget digunakan, tapi tidak ditemukan perishable item pada tas.
=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 14200 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====

ENTER COMMAND: 0
```

**Gambar 6.12** Pembacaan COMMAND INVENTORY

## 6.11 Data Test 11

Tes ini digunakan untuk memastikan COMMAND HELP dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
ENTER COMMAND: HELP

1. MOVE -> Untuk berpindah ke lokasi selanjutnya
2. PICK_UP -> Untuk mengambil item di lokasi sekarang
3. DROP_OFF -> Untuk mengantarkan item ke lokasi sekarang
4. MAP -> Untuk memunculkan peta
5. TO_DO -> Untuk menampilkan pesanan yang masuk ke To Do List
6. IN_PROGRESS -> Untuk menampilkan pesanan yang sedang dikerjakan
7. BUY -> Untuk menampilkan gadget yang dapat dibeli dan membelinya
8. INVENTORY -> Untuk melihat gadget yang dimiliki dan menggunakannya
9. HELP -> Untuk mengeluarkan list command dan kegunaannya
10. RETURN -> Mengembalikan barang teratas di tas apabila memiliki ability return to sender.
11. SAVE_GAME -> Menyimpan state game sekarang agar nanti bisa di-load.
=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====
```

**Gambar 6.13** Pembacaan COMMAND HELP

## 6.12 Data Test 12

Tes ini digunakan untuk memastikan COMMAND SAVE\_GAME dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
ENTER COMMAND: SAVE_GAME

Masukkan nama file penyimpanan (contoh: save.txt): data_test.txt
[!] Progress berhasil disimpan di data_test.txt!
=====
Mobita berada di posisi 8 (1,1)
Waktu: 1
Uang yang dimiliki: 15000 Yen
Jumlah pesanan yang harus dikerjakan: 2
=====

ENTER COMMAND: 
```

**Gambar 6.14** Pembacaan COMMAND SAVE\_GAME

## 6.13 Data Test 13

Tes ini digunakan untuk memastikan COMMAND RETURN dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.

```
=====
ENTER COMMAND: RETURN

Ability return to sender berhasil!
[!] Item jenis P (P -> Q) berhasil dikembalikan.
=====
Mobita berada di posisi Q (10,3)
Waktu: 23
Uang yang dimiliki: 16200 Yen
Jumlah pesanan yang harus dikerjakan: 8
=====

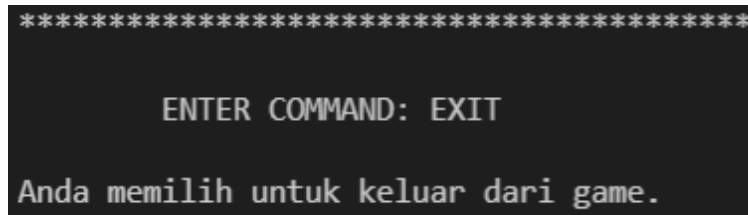
ENTER COMMAND: 
```

**Gambar 6.15** Pembacaan COMMAND RETURN

## 6.14 Data Test 14

Tes ini digunakan untuk memastikan COMMAND EXIT dapat berjalan dan menampilkan tampilan yang sesuai dengan sebagaimana mestinya.





**Gambar 6.16** Pembacaan COMMAND EXIT

## 7 Test Script

Isi dengan skenario test yang dimungkinkan untuk semua fitur yang ada. Bisa dibuat dalam bentuk tabel sebagai berikut.

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Start Game	Memeriksa apakah permainan dapat dijalankan	Melakukan <i>compile</i> pada <i>file</i> main.c dijalankan pada terminal	Data Test 1	Permainan dapat dijalankan dan main menu tertampil pada terminal	Sesuai dengan yang diharapkan
2	Pembacaan file konfigurasi	Memeriksa pembacaan file	Memasukkan nama file konfigurasi, baik untuk NEW GAME maupun LOAD GAME	Data Test 2	File konfigurasi dapat terbaca dengan baik dan menampilkan status terkini dari pemain	Sesuai dengan yang diharapkan
3	Map	Memeriksa fungsionalitas command MAP	Memasukkan <i>command</i> MAP	Data Test 3	Peta permainan dapat terlihat dengan jelas beserta pewarnaan yang benar	Sesuai dengan yang diharapkan
4	Move	Memeriksa fungsionalitas command MOVE	Memasukkan <i>command</i> MOVE	Data Test 4	Menampilkan lokasi-lokasi yang terhubung dengan lokasi saat ini	Sesuai dengan yang diharapkan
5	Pick Up	Memeriksa fungsionalitas command PICK_UP	Memasukkan <i>command</i> PICK_UP	Data Test 5	Jika ada pesanan di lokasi tersebut, pesanan akan terambil. Apabila tidak ada pesanan akan menampilkan pesan error	Sesuai dengan yang diharapkan
6	Drop Off	Memeriksa fungsionalitas command DROP_OFF	Memasukkan <i>command</i> DROP_OFF	Data Test 6	Pesanan berhasil diantarkan jika lokasinya tepat. Jika tidak, akan menampilkan pesan error	Sesuai dengan yang diharapkan
7	To Do	Memeriksa fungsionalitas command TO_DO	Memasukkan <i>command</i> TO_DO	Data Test 7	Pemain dapat melihat To Do List pesanan yang harus diantarkan	Sesuai dengan yang diharapkan

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
8	In Progress	Memeriksa fungsionalitas command IN_PROGRESS	Memasukkan <i>command</i> IN_PROGRESS	Data Test 8	Pemain dapat melihat pesanan apa saja yang sedang ia antarkan	Sesuai dengan yang diharapkan
9	Buy	Memeriksa fungsionalitas command BUY	Memasukkan <i>command</i> BUY dan memilih gadget yang ingin dibeli	Data Test 9	Pemain dapat melihat gadget yang dapat dibeli dan bisa membeli gadget yang ada	Sesuai dengan yang diharapkan
10	Inventory	Memeriksa fungsionalitas command INVENTORY	Memasukkan <i>command</i> INVENTORY dan memasukkan nomor gadget yang ingin digunakan	Data Test 10	Pemain dapat melihat gadget apa saja yang ia miliki dan bisa memilih akan menggunakan agdget yang mana	Sesuai dengan yang diharapkan
11	Help	Memeriksa fungsionalitas command HELP	Memasukkan <i>command</i> HELP	Data Test 11	Pemain dapat melihat semua <i>command</i> yang valid dan fungsinya	Sesuai dengan yang diharapkan
12	Save Game	Memeriksa fungsionalitas command SAVE_GAME	Memasukkan <i>command</i> SAVE_GAME dan memasukkan nama file yang dituju	Data Test 12	State permainan sat ini tersimpan ke dalam file	Sesuai dengan yang diharapkan
13	Return	Memeriksa fungsionalitas command RETURN	Meamsukkan <i>command</i> RETURN	Data Test 13	Mengembalikan pesanan teratas pada Bag jika memiliki ability Return To Sender. Jika tidak punya, akan terlihat pesan error	Sesuai dengan yang diharapkan
14	Exit	Memeriksa fungsionalitas command EXIT	Memasukkan <i>command</i> EXIT di dalam permainan	Data Test 14	Pemain dapat keluar dari permainan	Sesuai dengan yang diharapkan

## 8 Pembagian Kerja dalam Kelompok

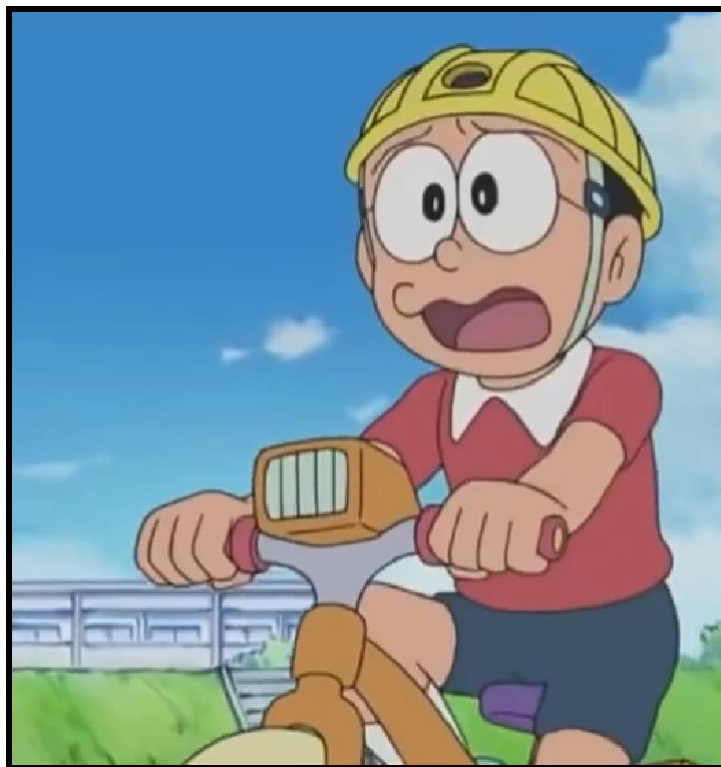
No.	NIM - Nama	Tugas yang dikerjakan
1.	13520117 – Hafidz Nur Rahman Ghozali	ADT Item, ADT Queue, ADT List Linked, <i>command</i> TO_DO, <i>command</i> IN_PROGRESS, Laporan
2.	13520119 – Marchotridyo	ADT Mesin karakter, mesin kata dan file machine, ADT Bangunan, ADT List Dinamis, ADT Matrix, ADT Map, dan Integrasi Main Program
3.	13520123 – Johannes Winson Sukiartmodjo	ADT ListPos, command BUY, command INVENTORY, command HELP, laporan

4.	13520137 – Muhammad Gilang Ramadhan	ADT time , ADT move, ADT bag, ADT ability, command move, laporan, merapikan tulisan program
5.	13520145 – Steven Gianmarg H. Siahaan	Command IN_PROGRESS
6.	13520149 – Mohamad Hilmi Rinaldi	Command DROP_OFF, driver ADT, laporan

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar 2

#### Latar Belakang



“Mobita menjadi seorang kurir demi mencari nafkah untuk keluarganya”

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

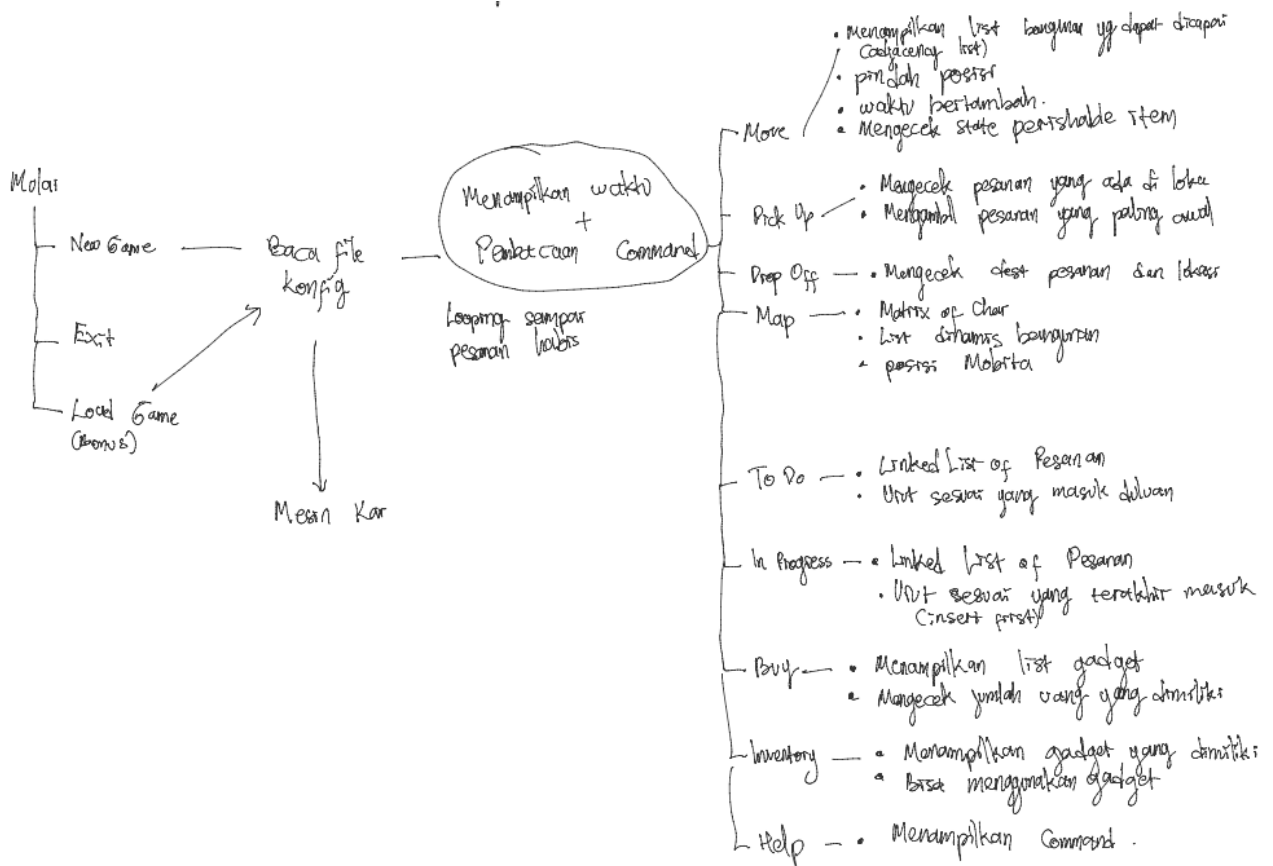
## 9.2 Notulen Rapat

Rapat 1 (23 Oktober 2021) :

- Membahas dan menyamakan persepsi mengenai spesifikasi tugas besar
- Hubungan konfigurasi peta dengan adjacency matriks
- Timbul beberapa pertanyaan yang perlu ditanyakan saat asistensi seperti :
  - a) Bagaimana urutan list isi inventory?
  - b) Apakah ADT yang sudah ada boleh dimodifikasi?
  - c) Apakah pembacaan file konfigurasi selalu benar?

Rapat 2 (27 Oktober 2021) :

- Membahas abstraksi garis besar jalannya program utama
- Rencana membuat ADT tambahan seperti :
  - a) ADT Item
  - b) ADT Bangunan
  - c) ADT Bag
  - d) ADT Mobita
  - e) ADT Mesin kar
- Abstraksi program utama :



- Pembagian tugas sesuai dengan fungsionalitas yang ditentukan

Rapat 3 (8 November 2021) :

- Membahas kekurangan-kekurangan setelah mengerjakan tugas yang sudah ditentukan sebelumnya
- Memberi deadline untuk menyelesaikan tugas masing-masing hingga tanggal 12 November 2021
- Pengintegrasian program utama dilakukan setelah tanggal 12 November 2021

### 9.3 Log Activity Anggota Kelompok

Tanggal	Jam	Tugas/Pekerjaan	Anggota kelompok yang hadir	Hasil
23-10-2021	13.00	Rapat 1	Hafidz, Marcho, Jojo, Gilang, Hilmi	Menyamakan persepsi terhadap spesifikasi tugas besar
27-10-2021	19.00	Asistensi 1	Hafidz, Marcho, Jojo, Gilang, Hilmi	Pertanyaan-pertanyaan sudah terjawab semua
27-10-2021	19.45	Rapat 2	Hafidz, Marcho, Jojo, Gilang, Tepen, Hilmi	Membuat abstraksi program dan membagi tugas per individu
30-10-2021	-	Menambahkan dan mengubah ADT Item, Linked List	Hafidz	ADT sudah bisa digunakan
31-10-2021	13.15	Mengerjakan command BUY, INVENTORY, dan HELP	Jojo	Command BUY dan INVENTORY masih in progress, command HELP sudah finished
31-10-2021	15.30	Mengedit ADT ListPos dan mengerjakan command BUY	Jojo	ADT siap dipakai dan command BUY sudah finished
1-11-2021	23.00	Membuat folder ADT Move beserta file adt move, bag, dan time	Gilang	ADT sudah bisa dipakai
2-11-2021	21.00	Menambahkan command	Hilmi	Command sudah bisa digunakan tetapi ability

		DROP_OFF		setelah mendrop item tertentu belum aktif
8-11-2021	19.00	Asistensi 2	Hafidz, Marcho, Jojo, Gilang, Tepen, Hilmi	Progress report
8-11-2021	19.30	Rapat 3	Hafidz, Marcho, Jojo, Gilang, Tepen, Hilmi	Rencana selama 1 minggu kedepan lebih jelas
8-11-2021	20.30	Menambahkan prosedur updatePesanan	Hafidz	Selesai dan bekerja dengan baik
9-11-2021	08.30	Mengerjakan command BUY dan INVENTORY	Jojo	Validasi untuk input yang tidak valid
9-11-2021	13.00	Mengerjakan command INVENTORY	Jojo	Implementasi gadget senter pembesar, pintu kemana saja, dan mesin waktu
9-11-2021	22.00	Menambahkan ADT ability	Gilang	ADT sudah siap dipakai
12-11-2021	20.00	Mengerjakan command DROP_OFF	Hilmi	Menyelesaikan efek ability saat mendrop item
13-11-2021	10.00	Mengerjakan laporan	Hafidz	Laporan
14-11-2021	11.00	Membuat adt baca file dan membuat command move	Gilang	Implementasi command buat move tetapi masih ada bagian yang belum dilengkapi yaitu pada bagian return to sender
16-11-2021	09.45	Mengerjakan laporan	Jojo	Laporan masih in progress
16-11-2021	20.30	Mengerjakan laporan	Hilmi	Melengkapi laporan (in progress)
17-11-2021	01.40	Mempercantik tulisan pada permainan dan Mengerjakan laporan	Gilang	Tulisan pada permainan sudah lebih rapi dan Beberapa Data test sudah ditambahkan
17-11-2021	11.00	Mengerjakan laporan	Hafidz	Laporan bagian ADT dan program utama sudah selesai
17-11-	15.30-	Melengkapi	Gilang, Jojo, Hilmi,	Driver ADT sudah

2021	16.00	Driver ADT	Marcho, Hafidz	ditambahkan semua
18-11-2021	16.00	Merapihkan laporan	Hilmi	Laporan selesai



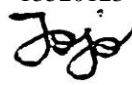



**Form Asistensi Tugas Besar  
IF2110/Algoritma dan Struktur Data  
Sem. 1 2021/2022**

No. Kelompok/Kelas : 02/03  
Nama Kelompok :  
Anggota Kelompok (Nama/NIM) : 1. Hafidz Nur Rahman Ghozali (13520117)  
2. Marchotridyo (13520119)  
3. Johannes Winson Sukiatmodjo (13520123)  
4. Muhammad Gilang Ramadhan (13520137)  
5. Steven Gianmarg H. Siahaan (13520145)  
6. Mohamad Hilmi Rinaldi (13520149)  
  
Asisten Pembimbing : Muhammad Kamal Shafi (13518113)



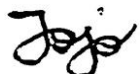




---



# Asistensi I

<b>Tanggal :</b> 27 Oktober 2021	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- File ADT boleh dimodifikasi</li> <li>- Urutan pembacaan file konfigurasi sama dengan contoh. Asumsikan valid</li> <li>- Untuk input harus pakai mesin karakter. Untuk write file dimodifikasi saja</li> <li>- Untuk ability langsung dipakai, tidak bisa disimpan.</li> <li>- Untuk command harus persis dengan contoh di spesifikasi</li> <li>- Harus handle kesalahan pembacaan command</li> <li>- Tidak ada ketentuan buat ekstensi file konfigurasi</li> <li>- Untuk inventory tidak ada ketentuan urutan yang ditampilkan</li> <li>- Kalau tidak mengimplementasi VVIP Item, bisa diabaikan saja di file konfigurasi</li> </ul>
<b>Tempat :</b> <a href="https://meet.google.com/yfg-idow-tvw">https://meet.google.com/yfg-idow-tvw</a>	
<b>Kehadiran Anggota Kelompok:</b> <p>No</p> <p>NIM</p> <p>Tanda tangan</p> <p>1</p> <p>13520117</p>  <p>2</p> <p>13520119</p>  <p>3</p> <p>13520123</p>  <p>4</p> <p>13520137</p>  <p>5</p> <p>13520149</p> 	
	<b>Tanda Tangan Asisten:</b>  <div data-bbox="1242 1354 1372 1470">           Digitally signed            by Muhammad            Kamal Shafi            Date: 2021.10.28            09:50:13 +07'00'         </div>

## Asistensi II

<b>Tanggal :</b> 8 November 2021	<b>Catatan Asistensi:</b> <ul style="list-style-type: none"> <li>- Buy hanya dapat digunakan di headquarter, kalau tidak di headquarter, keluarkan pesan error</li> <li>- Jumlah pesanan yang ditampilkan adalah banyaknya sisa pesanan di to do list dan in progress</li> <li>- Command tetap divalidasi dan menampilkan pesan error jika salah</li> </ul>
<b>Tempat :</b> <a href="https://meet.google.com/rmd-tnyo-xxu">https://meet.google.com/rmd-tnyo-xxu</a>	
<b>Kehadiran Anggota Kelompok:</b> <p>No</p> <p>NIM</p> <p>Tanda tangan</p> <p>1</p> <p>13520117</p>  <p>2</p> <p>13520119</p>  <p>3</p> <p>13520123</p>  <p>4</p> <p>13520137</p>  <p>5</p> <p>13520145</p>  <p>6</p> <p>13520149</p> 	
	<b>Tanda Tangan Asisten:</b>  <div> Digitally signed  by Muhammad  Kamal Shafi  Date:  2021.11.12  18:23:41  +07'00' </div>