

# **LAPORAN TUGAS BESAR**

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

### **Mobilita**


Dipersiapkan oleh:

Kelompok 6

Louis Yanggara	13520063
Jova Andres Riski Sirait	13520072
Samuel Christopher Swandi	13520075
Saul Sayers	13520094
Averrous Saloom	13520100

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-06-02</i>		<i>20</i>
		<i>Revisi</i>	<i>0</i>	<i>18 November 2021</i>

# Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas .....	4
2.1 Save / Load.....	4
2.2 VIP Item .....	4
2.3 Return to Sender.....	5
2.4 Senter Pengecil.....	5
3 Struktur Data (ADT) .....	5
3.1 POINT .....	5
3.2 List Statis.....	5
3.3 List Dinamis .....	5
3.4 Matriks .....	6
3.5 Mesin Karakter dan Mesin Kata.....	6
3.6 Queue .....	6
3.7 Stack .....	6
3.8 Linked List .....	6
3.9 MOBITA .....	6
3.10 MAP.....	7
3.11 Gadget.....	7
3.12 Pesanan .....	7
3.13 DaftarPesanan .....	7
3.14 Effect List .....	7
4 Program Utama .....	8
5 Algoritma-Algoritma Menarik.....	8
5.1 strings not equal .....	8
5.2 DapatDicapai.....	9
5.3 Manajemen Efek .....	9
6 Data Test .....	9
6.1 konfigurasi.txt .....	9
6.2 big.txt.....	10
6.3 tes.txt .....	12
6.4 hm.txt.....	13
7 Test Script .....	14
8 Pembagian Kerja dalam Kelompok .....	17
9 Lampiran .....	17
9.1 Deskripsi Tugas Besar 2.....	17
9.2 Notulen Rapat.....	18
9.3 Log Activity Anggota Kelompok.....	20

# 1 Ringkasan

## A. Deskripsi Umum

Mobilita adalah sebuah permainan berbasis CLI(command-line interface) yang dibuat dengan menggunakan bahasa C dan memanfaatkan Struktur Data (Abstract Data Type) untuk memudahkan pemrograman. Mobilita merupakan permainan dimana terdapat Mobita(player) yang bertugas sebagai seorang kurir. Tugas dari Mobita secara umum adalah menerima pesanan, mengambil dan mengirimkan pesanan.

Tujuan dari permainan Mobilita adalah mengantarkan seluruh pesanan yang ada dengan waktu yang sesingkat mungkin. Mobita memiliki tas yang pada awalnya hanya dapat diisi dengan 3 barang yang kemudian dapat di-upgrade dengan cara tertentu. Setiap melakukan perpindahan tempat, waktu dari Mobita juga bertambah sesuai dengan barang yang sedang diangkut(di dalam tas). Barang yang diantar Mobita terdiri dari 3 yaitu Normal Item yang tidak memiliki efek samping ketika berada di dalam tas Mobita dan berharga 200 Yen, Heavy Item yang memiliki efek yaitu membuat Mobita bergerak lebih lambat, namun jika berhasil diantarkan akan memberi kemampuan Speed Boost kepada Mobita dan berharga 400Yen, Perishable Item yaitu barang yang harus diantarkan dengan cepat karena dapat hangus, jika waktu sudah habis, item akan dikeluarkan otomatis dari tas, jika berhasil diantarkan Mobita akan mendapat ability Increase Capacity dan 400 Yen.

Untuk membuat permainan semakin menarik, terdapat dua ability yaitu Speed Boost dan Increase Capacity(permanent). Speed Boost adalah ability yang diterima Mobita setelah berhasil mengantarkan Heavy Item, efek dari Speed Boost adalah waktu yang diperlukan Mobita untuk berpindah 2 lokasi adalah 1 dan efek ini berlaku untuk perpindahan 10 lokasi, namun ability ini akan hilang jika mobita melakukan pick-up terhadap Heavy Item atau mendapat ability lain. Increase Capacity adalah ability yang akan diperoleh jika Mobita berhasil mengantarkan Perishable Item, efek dari ability ini adalah kapasitas tas akan bertambah 1.

Selain ability, Mobita juga dapat membeli berbagai gadget yang dapat membantu dia yaitu Kain Pembungkus Waktu, Senter Pembesar, Pintu Kemana Saja, dan Mesin Waktu. Semua item ini dapat digunakan untuk mempermudah Mobita melakukan perjalanan.

Permainan akan berakhir jika semua pesanan sudah masuk dan semua pesanan sudah berhasil diselesaikan oleh Mobita.

## B. Isi Umum Laporan

Secara Umum, laporan Tugas Besar IF2110 Kelompok 6 terdiri dari 9 bagian, yaitu :

1. Bagian 1: Berisi deskripsi umum dan penjelasan tentang permainan Mobilita, pemaparan isi laporan secara umum, serta kesimpulan mengenai Tugas Besar IF2110.
2. Bagian 2: Berisi penjelasan tambahan yang diperlukan terkait spesifikasi tugas.
3. Bagian 3: Berisi penjelasan dari setiap Struktur Data yang digunakan serta kegunaannya sesuai dengan spesifikasi yang ada.
4. Bagian 4: Berisi penjelasan mengenai algoritma program utama, dimulai dari New Game hingga Game Over.

STEI- ITB	< IF2110-TB-06-02>	Halaman 3 dari 20 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.		

5. Bagian 5: Berisi penjelasan mengenai algoritma yang dianggap menarik oleh kelompok kami beserta dengan penggunaannya dalam program.
6. Bagian 6: Berisi penjelasan mengenai fitur-fitur yang akan dicoba dan hasil yang diharapkan dari program. Dalam bagian ini juga akan ditampilkan data testnya.
7. Bagian 7: Berisi Tabel Test Script yang terdiri dari fitur yang di tes, tujuan testing, langkah-langkah testing, input data test, hasil yang diharapkan serta hasil yang keluar.
8. Bagian 8: Berisi Tabel pembagian kerja dari Kelompok 6.
9. Bagian 9: Berisi lampiran berupa Deskripsi lengkap Tugas Besar, notulen rapat, dan Log Activity dari setiap anggota kelompok.

### C. Kesimpulan

Pada Tugas Besar IF2110, telah dimanfaatkan semua ADT yang pernah dikerjakan di praktikum serta penerapan pelajaran membuat ADT sendiri seperti Mobita dan banyak lagi. Permainan Mobilita membutuhkan cukup banyak ADT untuk menunjang efisiensi dari pemrograman sehingga setiap pemrogram harus mengerjakan beberapa ADT yang berbeda. Hasil dari Tugas Besar kelompok 6 ini cukup baik, semua program sudah berjalan seperti yang seharusnya.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Save / Load

Save / Load adalah salah satu fungsi bonus yang disediakan oleh spesifikasi tugas besar. Dalam hal ini, save dan load membaca atau menuliskan kepada file .txt yang memiliki direktori yang sama dengan file main.c. Cara kerja fungsi Save adalah menuliskan kembali data map dan data queue ke .txt file baru, kemudian menuliskan data mobita. Save dapat diakses ketika tengah permainan, kita menggunakan command Exit kemudian menginput filename dari tujuan file .txt yang ingin kita simpan.

Cara kerja fungsi load serupa dengan fungsi konfigurasi, namun dengan beberapa langkah tambahan. Untuk data map dan data pesanan cara kerjanya sama, kemudian lanjut membaca data mobita yang sudah dicantumkan oleh fitur save. Fungsi load dan save ini membutuhkan mark berupa newline ataupun mark pada akhir file konfigurasi. Sedikit catatan mengenai fitur save / load, karena konfigurasi dianjurkan menggunakan ADT charmachine dan wordmachine dan tidak menggunakan scanf, maka dianjurkan agar nama file.txt yang ingin dibaca ataupun ditulis tidak terlalu panjang agar tidak terjadi overflow/error apapun saat pembacaan.

### 2.2 VIP Item

VIP Item adalah salah satu fungsi bonus pada spesifikasi tugas besar. Dalam fungsi ini, ketika pemain melakukan *pick-up* pada salah satu pesanan, maka pesanan VIP tersebut yang harus dilayani lebih dahulu. Sebagai konsekuensinya, pemain tidak dapat melakukan *pick-up* pesanan lain sebelum pesanan ini disampaikan ke tempat tujuannya. Sebagai *reward* ketika telah mengantarkan pesanan VIP, pemain akan mendapatkan pesan rahasia dan juga ability *return to sender*. Implementasi fungsi ini di dalam program, ketika pemain ingin melakukan pickup, akan di cek terlebih dahulu apakah sedang membawa item VIP, jika ada item VIP pada tas, maka *pick-up* akan ditolak.

## 2.3 Return to Sender

*Return to Sender* adalah ability yang akan didapatkan pemain jika telah selesai mengatarkan item VIP. Dengan *ability* ini, pemain bisa mengembalikan item teratas pada tas ke tempat asalnya. *Ability* ini tidak berlaku ini VIP item. Jika bagian teratas tas merupakan pesanan berjenis *perishable*, maka waktu *perishable* pesanan tersebut akan di reset ke waktu awalnya. Implementasi fungsi ini adalah dengan menambahkan waktu *perishable* awal untuk pesanan berjenis *perishable*, untuk mencatat dan mengubah sisa waktu ke waktu *perishable* awal.

## 2.4 Senter Pengecil

Senter pengecil merupakan gadget yang dapat dibeli oleh pemain di *headquarters*. Ketika menggunakan gadget ini, maka *move* pemain tidak akan terpengaruh oleh pesanan berjenis *heavy* sehingga pemain dapat melakukan *move* dengan normal dan tetap mendapatkan reward jika telah berhasil mengantarkan pesanan. Implementasi fungsi ini adalah dengan melakukan pengecekan setiap kali pemain akan melakukan *move*, jika pemain memiliki *heavy* item pada tas, maka mekanisme waktu setiap *move* yang dilakukan akan berubah. Akan tetapi program juga akan mengecek apakah pemain sedang menggunakan ability senter pengecil, jika pemain ternyata menggunakan, maka mekanisme pergantian waktunya akan tetap normal.

# 3 Struktur Data (ADT)

## 3.1 POINT

ADT POINT terdiri dari 2 data yaitu nilai x sebagai Absis dan y sebagai Ordinat, ADT ini digunakan untuk menentukan posisi berdasarkan adjacency matriks yang diinput, alasan pemilihan ADT ini adalah mudahnya untuk menentukan lokasi berdasarkan nilai matriks yang memiliki nilai kolom dan baris yang sesuai dengan Absis dan Ordinat. ADT ini diimplementasikan sebagai POINTdi point.c.

## 3.2 List Statis

ADT List Statis merupakan ADT sebuah list yang memiliki kapasitas tetap, ADT ini digunakan untuk mengimplementasikan Inventory atau tempat menyimpan gadget yang dibeli oleh Mobita, karena pada spesifikasi dinyatakan bahwa inventory memiliki kapasitas sebesar 5, maka penggunaan List statis ini sangat cocok. ADT primitif dari list pos diimplementasikan pada listpos.c, sedangkan implementasi untuk ADT inventory diimplementasikan pada inventory.c.

## 3.3 List Dinamis

ADT List Dinamis adalah ADT sebuah list yang memiliki kapasitas yang dapat diubah sesuai dengan kebutuhan program, ADT ini digunakan untuk mengimplementasikan Lokasi yang ada, sesuai spesifikasi, kapasitas maksimum dari List ini adalah 26, namun yang disimpan hanya jumlah elemen efektif yang ada, ADT ini digunakan untuk menghemat penggunaan memori dari program. ADT primitif List Dinamis diimplementasikan sebagai listdin.c sedangkan penggunaannya ada pada Map.c.

### 3.4 Matriks

ADT Matriks adalah list dari list yang bersifat statis, artinya maksimum kapasitas sudah ditentukan namun data dari kolom dan baris efektif tetap disimpan, ADT ini digunakan untuk merepresentasikan matrix adjacency yang ada untuk membuat peta, ADT Matriks digunakan untuk peta karena peta berbentuk segiempat yang masing-masing titik dapat direpresentasikan sebagai baris ke-n dan kolom ke-m. ADT primitif Matriks diimplementasikan sebagai matrix.c sedangkan penggunaannya dalam map.c.

### 3.5 Mesin Karakter dan Mesin Kata

ADT Mesin Karakter dan Mesin Kata adalah ADT yang saling berkaitan, Mesin Karakter digunakan untuk mempermudah penggunaan Mesin Kata, kedua ADT ini digunakan untuk membaca konfigurasi dari file, ADT ini digunakan karena input konfigurasi terdiri dari baris-baris yang menandakan informasi yang berbeda-beda sehingga harus dibaca satu per satu yang sangat cocok dengan ADT Mesin Karakter. ADT primitif Mesin Karakter diimplementasikan dalam charmachine.c, Mesin Kata pada wordmachine.c, sedangkan implementasi kedua ADT ada pada map.c. Selain itu, ADT ini juga digunakan dalam pengerjaan bonus save/load karena harus menuliskan isi dari permainan yang tersimpan dalam ADT lain ke dalam luaran txt file.

### 3.6 Queue

ADT Queue adalah sebuah ADT yang menggunakan metode sama seperti list, namun pada Queue terdapat sifat FIFO(First In First Out) yang artinya elemen yang masuk pertama ke Queue akan selalu diproses pertama. Queue digunakan untuk mengimplementasikan pesanan yang masuk, ADT ini dipilih karena sesuai dengan konsep pesanan yaitu pesanan yang datang duluan juga akan dilayani duluan. ADT primitif Queue diimplementasikan pada queue.c.

### 3.7 Stack

ADT Stack adalah sebuah ADT yang menggunakan metode sama seperti list, namun pada Stack terdapat sifat LIFO(Last In First Out) yang artinya elemen yang masuk terakhir ke Stack akan selalu diproses pertama. Stack digunakan untuk mengimplementasikan barang yang sudah di pick-up. Barang yang dapat diakses hanya barang yang berada paling atas di tas yaitu barang yang terakhir masuk ke tas. ADT primitif Stack diimplementasikan pada stack.c sedangkan ADT Tas diimplementasikan pada tas.c.

### 3.8 Linked List

Linked list merupakan variasi lain dari list dimana setiap list merupakan alamat memori yang saling dihubungkan satu sama lain. Isi sebenarnya dari List terdapat di dalam alamat memori. ADT ini digunakan untuk menyatakan pesanan yang harus dan sedang dikerjakan (TODO dan InProgress), ADT ini digunakan karena untuk mengimplementasikan TODO dan InProgress semua pesanan saling berkaitan dan berurutan urutan masuknya. ADT primitif Linked list diimplementasikan pada list\_linked.c, sedangkan penggunaan ADT ada pada todo.c.

### 3.9 MOBITA

MOBITA adalah ADT yang digunakan untuk menunjuk pemain/user. ADT MOBITA berisi kumpulan ADT lainnya yang berhubungan dengan MOBITA seperti inventory, posisi, toDoList,

inProgress, Tas, Effect, serta waktu dan uang dari MOBITA saat ini. ADT MOBITA berisi fungsi-fungsi untuk menunjukkan keadaan player saat ini dan memiliki fungsi seperti menambah waktu, menambah uang, berpindah posisi, dan mengecek jenis barang bawaan. ADT ini diimplementasikan dalam ADT MOBITA pada file mobita.c.

### **3.10MAP**

MAP adalah ADT yang digunakan untuk menyimpan "peta" dari permainan. ADT map berisi Lokasi, ukuran baris dan kolom, serta Matrix adjacency yang digunakan. Dalam ADT map terdapat beberapa fungsi dasar seperti membentuk map, membentuk lokasi, menampilkan map, load game, dan mengubah nama lokasi menjadi point. ADT ini diimplementasikan sebagai MAP pada file map.c.

### **3.11Gadget**

ADT Gadget adalah ADT yang digunakan untuk merepresentasikan sebuah gadget, pada ADT Gadget, terdapat data nama, id, dan harga dari sebuah gadget. ADT ini akan digunakan untuk menyimpan data dari setiap Gadget yang ada. ADT ini diimplementasikan pada gadget.c.

### **3.12Pesanan**

ADT Pesanan adalah ADT yang digunakan untuk merepresentasikan sebuah pesanan, pada ADT Pesanan terdapat informasi mengenai waktu pesanan, asal, tujuan, dan jenis barang. ADT ini digunakan untuk menyatakan setiap pesanan yang masuk. ADT Pesanan diimplementasikan pada pesanan.c

### **3.13DaftarPesanan**

ADT DaftarPesanan adalah ADT yang fungsinya serupa dengan ADT Queue, yakni ADT yang berisi daftar pesanan yang masuk namun bersifat sebagai list dan mencatat Neff dari banyaknya pesanan. ADT ini hanya digunakan pada awal konfigurasi awal program saja. Alasan ADT ini digunakan adalah karena untuk melakukan sorting pesanan di awal program terlebih dahulu, karena bisa jadi waktu pesanan yang dicantumkan pada txt file konfigurasi tidak urut dari awal. Demi mempertahankan sifat Queue yang memiliki prinsip FIFO (First in First Out) dan hanya bisa mengakses elemen head dan tail, maka kita berusaha mempertahankan karakteristiknya dan melakukan sorting di ADT lain, dan itulah gunanya ADT DaftarPesanan.

### **3.14Effect List**

ADT Effect List adalah ADT yang digunakan untuk merepresentasikan setiap effect yang diberikan oleh gadget yaitu senter pengecil, pintu kemana saja, kain pembungkus waktu, speed boost, dan increase capacity. ADT ini juga memanfaatkan ADT List untuk menyimpan data yang ada. ADT Effect List diimplementasikan pada effect\_list.c.

## 4 Program Utama

Program utama terletak pada file main.c, dalam file ini terdapat beberapa fungsi tambahan untuk menunjang efisiensi program. Permainan dimulai dari pemain memilih untuk memainkan game baru(new game) atau melanjutkan permainan yang sudah ada(load game). Setelah user menentukan pilihan, akan dijalankan fungsi konfigurasi untuk membaca txt file ke program, jika berhasil akan ditampilkan statement berhasil. Jika user memilih untuk new game maka mobita akan diposisikan pada headquarter, jika memilih load game, maka seluruh state yang sudah ada sebelumnya dikembalikan ke program.

Setelah proses inisialisasi selesai, user akan mulai bermain. Pada permainan ini ada 10 command yang dapat dipilih untuk dilakukan oleh mobita. Pertama ada menu move, berguna untuk berpindah dari satu posisi ke posisi lainnya, ketika memilih menu move, player akan diberikan pilihan tempat yang dapat dicapainya dari titik dia saat ini. Kedua Pick Up, berguna untuk mengambil barang dari suatu tempat sesuai dengan To Do List yang ada, setelah dipick-up, barang akan dimasukkan ke dalam tas dan In Progress List. Ketiga Drop Off, berguna untuk mengantarkan item yang ada pada urutan paling atas di tas, setelah barang berhasil diantarkan, barang dihapus dari tas. Keempat display Map, berguna untuk menampilkan map dengan warna yang sesuai dengan spesifikasi. Kelima display Todo, berguna untuk menampilkan pesanan yang sudah masuk dan harus dikerjakan oleh Mobita. Keenam display InProgress, berfungsi untuk menampilkan barang teratas yang ada di tas dan harus dikerjakan oleh Mobita. Ketujuh buyGadget, berfungsi untuk membeli Gadget dengan uang yang sudah dimiliki oleh Mobita, menu ini hanya dapat diakses ketika Mobita berada di Headquarter. Kedelapan inventory, berfungsi untuk melihat daftar gadget yang sudah dibeli oleh Mobita serta menggunakan Gadget tersebut. Kesembilan help, berfungsi untuk menampilkan menu bantuan yaitu penjelasan mengenai kegunaan dari setiap menu yang ada. Kesepuluh, menu untuk save dan exit berfungsi agar player dapat menentukan apakah dia ingin melakukan save terhadap state saat ini atau tidak.

## 5 Algoritma-Algoritma Menarik

### 5.1 *strings not equal*

Algoritma ini terkesan menarik karena ditemukan bahwa untuk mengomparasi word dari ADT wordmachine dengan sebuah string tidak bisa dilakukan secara langsung karena akan selalu menghasilkan false. Dengan demikian, diciptakanlah fungsi ini yang menariknya juga terinspirasi dari mata kuliah lain yakni Praktikum Organisasi dan Arsitektur Komputer. Pada awal algoritma ini, dicek terlebih dahulu apakah length dari kedua kata sama, apabila tidak maka dicatat hasilnya sebagai true di akhir. Untuk length dari Word dapat diakses langsung dengan structnya Word.length, untuk mencari length string dapat digunakan fungsi lain string\_length yang dibuat yakni dengan algoritma traversal menghitung banyaknya char yang ada pada string tersebut. Kemudian apabila lengthnya sama, kita melakukan algoritma searching untuk mengecek dari awal hingga akhir atau hingga ada char yang tidak sama. Apabila ada yang tidak sama maka result langsung dicatat sebagai true, dan apabila sama semua hingga akhir maka result dibiarkan false. Fungsi akan mereturn result true or false tersebut. Algoritma ini digunakan untuk mengecek



apakah input command saat pemilihan menu adalah sama atau tidak dengan daftar command yang sudah dicantumkan dengan fungsi help.

## 5.2 DapatDicapai

Algoritma ini menarik karena sangat vital dalam mencari daftar lokasi yang dapat dicapai oleh suatu lokasi, namun cukup efisien dalam prosesnya. Algoritma ini digunakan dalam proses fungsi move untuk dapat berpindah ke lokasi apa saja yang dapat dicapai, kemudian digunakan pada fungsi displayMap untuk mencetak semua lokasi yang dapat dicapai dengan warna kuning. Algoritma ini pertama – tama mencari indeks lokasi player mobita kemudian mencocokkannya pada matriks adjacency, lalu dengan pencarian traversal mengecek tiap lokasi yang bernilai 1 dan mencatatnya pada list statis dapatdicapai. Dengan demikian, tidak perlu melakukan pencarian sebanyak  $M \times N$  peta, tetapi cukup pencarian sebanyak banyaknya lokasi + 1 sehingga membuatnya efisien.

## 5.3 Manajemen Efek

Spesifikasi meminta kami untuk menjalankan efek yang disebabkan oleh gadget maupun bonus-bonus karena berhasil menyelesaikan pekerjaan. Men-track hal ini dipermudah dengan membuat sebuah EffectList sehingga sistem bisa mengingat apakah suatu efek sedang berjalan, kami menjadikannya sebagai sebuah properti dari struktur data Mobita. Selain harus *tracking*, kita juga harus mengecek apakah suatu efek ada di tiap metode yang bisa mendapatkan efeknya. Hal ini menantang karena perbedaan tahapan operasi yang harus dijalankan, contohnya, terdapat operasi yang tinggal mengubah suatu *instance* seperti efek Kain Pembungkus Waktu dan efek Senter Pembesar, sementara itu efek Pintu Kemana Saja, tidak mengubah sesuatu *instance*, tetapi mengubah metode MOVE menjadi mungkin untuk pergi ke mana pun. Efek-efek yang paling sulit adalah yang memengaruhi waktu dari *perishable item* karena hal ini membutuhkan operasi traversal di dua struktur data berbeda, InProgressList dan Tas.

# 6 Data Test

## 6.1 konfigurasi.txt

konfigurasi.txt adalah txt file yang berisi konfigurasi persis seperti yang ada pada spesifikasi tugas besar. Fitur – fitur yang dites dalam data test ini adalah move, increment time, pick up, drop off. Hasil yang diharapkan melalui data tes ini adalah untuk move, Daftar lokasi yang dapat dicapai yang disediakan oleh command move benar, posisi akhir mobita sesuai input. Untuk increment time, perubahan waktu yang terjadi sudah sesuai dengan yang diharapkan, bergantung dari banyaknya pesanan berjenis Heavy yang dibawa. Untuk Pickup, hanya bisa dilakukan ke lokasi yang di displaymap berwarna merah atau dengan kata lain ada pada todo list. Untuk Dropoff, hanya bisa dilakukan ke lokasi yang di displaymap berwarna biru atau dengan kata lain ada pada inprogress.

```
10 15
1 1
17
A 10 1
B 1 15
```

```

C 1 9
D 1 13
E 2 3
F 3 1
G 3 8
H 3 14
I 4 5
J 5 12
K 6 3
L 7 10
M 8 2
N 8 6
O 8 15
P 9 13
Q 10 3
0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0
0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0
0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0
0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1
0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
10
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
8 Q L P 5

```

## 6.2 big.txt

big.txt adalah txt file yang serupa dengan konfigurasi.txt, namun ukuran petanya ditambahkan dan terdapat beberapa pesanan dan lokasi baru. Dengan kata lain, big.txt adalah peta versi lebih besar dari konfigurasi.txt. Fitur – fitur yang dites dalam data test ini adalah displayMap, todolist, inprogress, dan perishable item. Untuk displayMap, mengecek apakah cetakan matriksnya dan tiap lokasi sesuai seperti pada file konfigurasi, dan apakah tiap warna benar sesuai maknanya dan

```

20 25
13 16
20
A 10 1
B 1 15
C 1 9
D 1 13
E 2 3
F 3 1
G 3 8
H 3 14
I 4 5
J 5 12
K 6 3
L 7 10
M 8 2
N 8 6
O 8 15
P 9 13
Q 10 3
R 18 20
S 16 15
T 17 23
0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0
1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1
0 0 1 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0
0 0 0 0 0 0 1 0 0 1 0 0 1 1 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 0 1
0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0
0 0 0 0 0 1 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1
0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1
0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1
0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 1
15
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
8 Q L V
14 S R N
18 T S N
24 O S N
20 M T H
28 C O H

```

### 6.3 tes.txt

tes.txt adalah txt file yang serupa dengan konfigurasi.txt, namun ukuran petanya dikurangi dan terdapat beberapa pesanan dan lokasi yang dihilangkan juga. Dengan kata lain, tes.txt adalah peta versi lebih kecil dari konfigurasi.txt. Fitur – fitur yang dites dalam data test ini adalah semua efek gadget. Untuk senter pembesar, mengetahui apakah berhasil menambah kapasitas tas. Untuk pintu kemana saja, berhasil berpindah posisi tanpa terikat dengan matriks adjacency. Untuk mesin waktu, mengetahui apakah berhasil mengulang waktu menjadi semula. Untuk senter pengecil, mengetahui apakah efek dari heavy item berhasil diabaikan. Untuk kain pembungkus tas, mengetahui apakah waktu perishable item berhasil kembali ke semula.

```

10 10
5 5
10
A 6 4
B 1 7
C 6 9
D 7 3
E 2 3
F 3 1
G 3 8
H 3 5
I 4 5
J 8 7
0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0 0
1 0 0 0 0 0 1 1 0 1 0
1 0 0 0 0 1 0 0 0 1 0
0 0 0 1 0 1 0 0 1 1 1
0 0 1 1 1 0 0 1 0 0 1
0 0 0 0 0 1 1 1 0 0 1
0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 0 1 0 1 1
0 0 0 0 1 0 0 0 1 1 0
6
1 A H N
3 B I H
2 C F N
1 D A H
5 E B N

```

## 6.4 hm.txt

hm.txt adalah txt file adalah tes.txt yang telah dilakukan beberapa langkah move dan pickup dan menyimpan data mobita. Dengan kata lain, hm.txt adalah hasil dari save konfigurasi.txt di tengah – tengah permainan. Fitur – fitur yang dites dalam data test ini adalah save dan load. Untuk load, mengecek apakah load berhasil membaca data peta, queue, dan player dari savefile. Untuk save, mengecek apakah save berhasil menyimpan data permainan.

```

10 10
5 5
10
A 6 4
B 1 7
C 6 9
D 7 3
E 2 3
F 3 1
G 3 8
H 3 5
I 4 5
J 8 7
0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0 0
1 0 0 0 0 0 1 1 0 1 0
1 0 0 0 0 1 0 0 0 1 0
0 0 0 1 0 1 0 0 1 1 1
0 0 1 1 1 0 0 1 0 0 1
0 0 0 0 0 1 1 1 0 0 1
0 0 0 0 0 0 0 1 1 1 0
0 0 0 0 0 0 0 1 0 1 1
0 0 0 0 1 0 0 0 1 1 0
6
1 A H N
3 B I H
2 C F N
1 D A H
5 E B N
3 F C N

10000
4
7 3
0
6
3 F C N
3 B I H
2 C F N
1 A H N
1 D A H
5 E B N

```

## 7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Move	Mengetahui keberhasilan fitur move, apakah daftar lokasi yang dapat diakses sesuai matriks adjacency, dan apakah perpindahan posisi mobita berhasil.	Melakukan fungsi move berkali – kali, membandingkan titik yang dapat diakses dengan matriks adjacency pada file konfigurasi, melihat posisi akhir mobita.	konfigurasi.txt	Daftar lokasi yang dapat dicapai yang disediakan oleh command move benar, posisi akhir mobita sesuai input	Mobita berhasil berpindah lokasi sesuai input, daftar lokasi yang dapat diakses benar
2	Increment Time	Untuk mengetahui apakah increment time berfungsi sesuai banyaknya heavy item dan efek yang berlaku	Melakukan fungsi move berkali – kali sambil membawa heavy item, lalu mengecek increment time	konfigurasi.txt	Waktu bertambah 1 setiap move, untuk setiap heavy item yang ada di dalam tas mobita	Waktu berhasil bertambah 1 untuk setiap heavy item yang di-pickup oleh mobita. (+2 untuk 1 heavy item, +3 untuk 2 heavy item, dan seterusnya).
3	pick up	Untuk mengetahui apakah barang berhasil di pickup atau tidak	Melakukan fungsi pickup dan drop off untuk melihat apakah barang berhasil di-pick up atau tidak	konfigurasi.txt	Barang diambil dan diantarkan ke tujuan	Barang berhasil diambil dan diantarkan ke tujuan, dan mobita mendapatkan uang yen
4	drop off	Untuk mengetahui apakah barang berhasil di-drop off atau tidak	Melakukan fungsi pickup lalu drop off untuk melihat apakah barang berhasil di drop off	konfigurasi.txt	Barang di-drop off dan uang yang dimiliki mobita bertambah	Barang berhasil di-drop off, dan uang yang dimiliki mobita bertambah
5	displayMap	Mengetahui apakah cetakan matriksnya dan tiap lokasi sesuai seperti pada file konfigurasi, dan apakah tiap warna benar sesuai	Menggunakan command displaymap dan melihat outputnya berkali - kali setelah beberapa langkah move,	big.txt	cetakan peta dan tiap lokasi sesuai seperti pada file konfigurasi, tiap warna benar sesuai maknanya dan	Cetakan peta nya sesuai, dan urutan warnanya sudah sesuai makna tiap lokasi dan urutan prioritas

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
		maknanya dan urutan prioritasnya	pickup, dropoff yang berbeda – beda.		urutan prioritasnya	
6	todolist	Mencoba apakah todolist sudah sesuai dengan pesanan yang masuk	Melihat daftar todolist yang ada pada waktu tertentu dan mengecek apakah sesuai dengan pesanan	big.txt	Isi dari todolist sesuai dengan pesanan yang ada pada waktu tertentu.	Isi dari todolist sesuai dengan pesanan.
7	inprogress	Mencoba apakah daftar inprogress sudah sesuai ketika melakukan pickup dan dropOff	Melakukan pickUp terhadap beberapa pesanan untuk mengecek urutan dalam inprogress dan melakukan dropOff	big.txt	Urutan dalam daftar Inprogress sesuai dengan ketentuan yang ada	Daftar sudah sesuai baik setelah pickUp maupun setelah dropOff
8	perishable	Mencoba apakah perishable item yang ada hilang sesuai dengan waktu yang ada.	Melakukan pickUp terhadap perishable item agar sekarang item berada dalam tas.	big.txt	Setelah periode waktu tertentu, perishable item akan hilang.	Perishable item yang ada di dalam tas hilang sesuai dengan waktu yang dimasukkan.
9	Kain Pembungkus Waktu	Mencoba apakah kain pembungkus waktu berlaku	Menggunakan kain pembungkus waktu ketika ada item perishable di tas	tes.txt	Ketika digunakan, maka waktu dari perishable item teratas di tas akan di reset.	Ketika digunakan, waktu dari perishable item paling atas di atas ter-reset ke waktu item masuk.
10	Senter Pembesar	Mengetes penggunaan senter pembesar untuk menambah kapasitas tas	Melakukan pickup sampai maksimum (3), kemudian menggunakan gadget dan mencoba pickup item lainnya.	tes.txt	Dapat dilakukan pickup setelah ukuran tas bertambah dari kapasitas awal (lebih dari 3 item)	Berhasil pickup lagi setelah menggunakan gadget
11	Pintu Kemaja Saja	Mengetes penggunaan gadget pintu kemana saja untuk berpindah posisi tanpa batasan tempat	Melakukan move setelah menggunakan gadget pintu kemana saja	tes.txt	Pada bagian tempat yang bisa dicapai, semua titik dalam peta terdaftar	Semua titik pada peta terdaftar dan dapat berpindah ke titik mana saja

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
12	Mesin Waktu	Mengetes penggunaan mesin waktu yang bertujuan untuk mengembalikan waktu menjadi waktu semula	Melakukan move beberapa kali sampai waktu lebih besar dari 0, kemudian menggunakan gadget mesin waktu	tes.txt	Waktu berkurang sebanyak lima puluh unit dengan waktu minimal adalah nol.	Waktu berkurang sebanyak lima puluh unit, atau maksimal nol.
13	Senter Pengecil	Mengetes penggunaan senter pengecil untuk mengabaikan efek dari heavy item ketika berpindah ke tempat lain.	Melakukan pickup pada pesanan berjenis heavy, kemudian menggunakan	tes.txt	Senter pengecil berhasil mengabaikan efek heavy item saat increment time pada fungsi move	Increment time yang terjadi berhasil mengabaikan efek heavy item
13	save	Mengetahui apakah save berhasil menyimpan data permainan	Melakukan beberapa langkah move dan pickup pada konfigurasi.txt kemudian menyimpannya pada sebuah txt.file baru.	konfigurasi txt -> hm.txt	Berhasil menuliskan data peta, queue, dan player kepada file .txt baru	Berhasil menyimpan data peta, queue, dan player, namun terdapat beberapa kasus error apabila filename terlalu panjang.
14	load	Mengetahui apakah load berhasil membaca data peta, queue, dan player dari savefile	Memilih opsi load pada awal menyalakan program, kemudian memilih beberapa save file yang berbeda.	hm.txt	Berhasil membaca data peta, queue, dan player dari save file.	Berhasil membaca data pada savestate, namun terdapat beberapa kasus error apabila filename terlalu panjang.



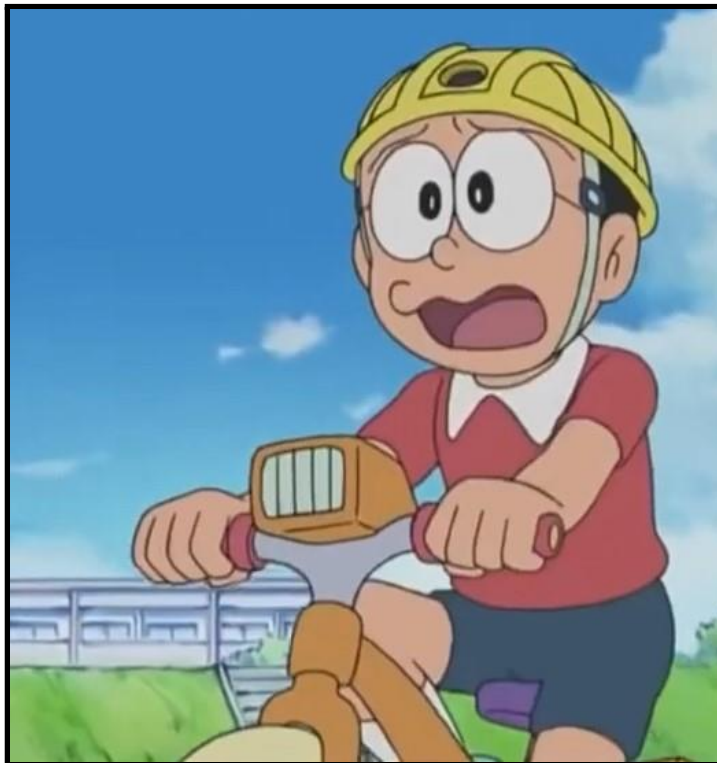
## 8 Pembagian Kerja dalam Kelompok

Pembagian Tugas		
Nama	NIM	Tugas
Averrous Saloom	13520100	Mobita, Tas, Inventory, Manajemen Efek (gadget, reward, perishable), senter pengecil, laporan bagian 5
Jova Andres Riski Sirait	13520072	Todolist, buy, wrap up, final debugging dan fix bug, charmachine, save, load, vip item, return to sender, laporan bagian 2 7
Louis Yanggara	13520063	Inprogress, dropOff, laporan bagian 1,3,4,7,8,9
Samuel Christopher Swandi	13520075	Increment time, pickup item, display todolist dan inprogress
Saul Sayers	13520094	ADT map, pesanan, konfigurasi awal,help,move,displaymap, laporan bagian 5 6 7

## 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 diringkangkannya PPKM, Mobita ingin membantu orang tuanya mendapat penghasilan menjadi seorang kurir. Tugas kalian adalah membantu Mobita membuat aplikasi yang akan membantunya melacak pesanan, navigasi peta, mengambil dan menurunkan barang.

## 9.2 Notulen Rapat

Rapat 1 – 26 Oktober 2021

- Membahas rancangan program utama
- Membahas ADT yang akan digunakan untuk setiap menu
- Membahas pembagian tugas untuk pembuatan ADT

Asistensi 1 – 26 Oktober 2021

Catatan Asistensi:

Q1 : Di dalam peta ada beberapa warna, itu bagaimana membuat warnanya?

A : Pada halaman 6 spesifikasi, ada ketentuan pewarnaan untuk setiap lokasi. Jangan lupa perhatikan prioritas pewarnaan.

Q2 : Semua pesanan ada di konfigurasi ya? Jadi di ingame gabisa nambah?

A : Semua pesanan di konfigurasi termasuk waktu masuk dari pesanan.

Q3 : Kami berlima, jadi bingung mau gimana bagi jadi 5 agar proporsional.

A : Pertama, liat dulu ada berapa tasknya, itu bisa dipecah-pecah. Terus lihat bebannya jadi baginya tergantung kemampuan masing-masing orang. Terus perhatikan keterkaitannya karena ada task yang harus diselesaikan sebelum task lain. Jadi sebenarnya ada banyak cara untuk bagi.

Q4 : Untuk matrix adjacency di contoh konfigurasi, semua elemennya biner antara 0 atau 1. Apakah mungkin ada sebuah hubungan lokasi yang butuh waktu untuk berpindahnya selain angka itu kak ? Misalnya 2 lokasi waktu berpindahnya butuh 2 satuan waktu bukan 1 karena jauh.

A : Tidak, elemen matriksnya tetap biner karena hanya menunjukkan dua lokasi apakah bisa saling berpindah atau tidak. Jadi boleh diasumsikan kalau tiap lokasi butuh 1 satuan waktu saja apabila ada hubungannya.

Rapat 2 – 31 Oktober 2021

- Membahas kelengkapan ADT
- Membahas permasalahan yang dihadapi selama membuat ADT
- Melakukan pembagian tugas untuk menu utama

Asistensi 2 – 9 November 2021

Catatan Asistensi:

STEI- ITB	< IF2110-TB-06-02 >	Halaman 18 dari 20 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.		

Untuk progress, konfigurasi sudah, move sudah, warna sudah, yang belum hanya gadget, ability, dan perishable item saja.

Q1: Buat compile program, bisa langsung import file ke main. Kalau begitu tinggal buat main.c kan. Namun kalau hanya include header harus include semua file c yang ada. Nah itu gapapa kan?

A: Itu lebih ke best practice, tapi kalau hanya c bisa terjadi error, salah satunya error pada variabel karena setiap header bisa saja menginclude file header lain sehingga terjadi double declaration.

Q2: Kalau include header yang circular dependency itu bisa ga?

A: Sudah agak lupa, mungkin harus dicari tahu lagi.

Q3: Di program kami, interaksi dengan user terjadi di terminal, namun di spesifikasi ditulis interaksi menggunakan mesin karakter dan mesin kata

A: Itu bisa dimodifikasi, sehingga ada 2 mode untuk membaca masukkan dari user sesuai dengan spesifikasi.

Q4: Semua pekerjaan kami ada dalam 1 folder, itu harus diubah ga kak?

A: Itu ga terlalu masalah untuk tubes.

Q5: Di spesifikasi, contoh konfigurasi hanya ada 1, jadi takutnya ada error kalau beda, apakah akan diberi konfigurasi lain? Terus untuk demo nanti pakai konfigurasi yang punya kami?

A: Untuk saat ini belum ada, kalau bisa buat aja sendiri untuk mencoba programnya. Untuk penilaian asisten pakai konfigurasinya sendiri.

Q6: Pertanyaan tentang perishable item, misalnya sekarang detik 55 terus item hilang detik ke 60 artinya tersisa 5 detik, terus ada item yang mengurangi waktu, kalau dipakai itu bakal ngurangin waktunya ga ya?

A: Misalnya dia hangus dalam 10 satuan waktu, maka tetap dihitung sesuai dengan waktu yang berjalan

Rapat 3 – 15 November 2021

- Membahas keberjalanan program pertama
- Membahas spesifikasi bonus
- Membagi tugas untuk penyelesaian akhir

### 9.3 Log Activity Anggota Kelompok





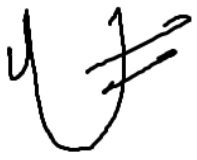
LOG ACT		
Tanggal	Nama	Deskripsi
26/10/2021	Seluruh Anggota	Asistensi dan Bagi Tugas
27/10/2021	Saul Sayers	Membuat ADT map dan pesanan
27/10/2021	Jova Andres Riski Sirait	Membuat ADT Untuk Todo List
27/10/2021	Samuel Christopher Swandi	Menambahkan ADT primitif dan merapikan pada folder
27/10/2021	Louis Yanggara	Merapikan Form Asistensi dan menyelesaikan fungsi untuk InProgress
28/10/2021	Saul Sayers	Membuat konfigurasi read txt file
29/10/2021	Averrous Saloom	Menambahkan ADT Mobita
29/10/2021	Saul Sayers	Membuat main awal, command move dan help
31/10/2021	Seluruh Anggota	Bagi Tugas 2
1/11/2021	Samuel Christopher Swandi	Add increment time pada move, pickup item, display todolist, display inprogress
1/11/2021	Jova Andres Riski Sirait	Membuat fungsi buy dan makefile
1/11/2021	Louis Yanggara	membuat todo list
2/11/2021	Saul Sayers	Membuat displaymap (sudah berwarna)
2/11/2021	Louis Yanggara	Menambahkan menu dropOff
3/11/2021	Jova Andres Riski Sirait	Mencoba buat save game
9/11/2021	Samuel Christopher Swandi	Fix Change Posisi
9/12/2021	Averrous Saloom	Memulai manajemen efek dengan membuat EffectList
9/11/2021	Seluruh Anggota	Asistensi 2
10/11/2021	Averrous Saloom	Menambahkan efek yang bisa bekerja
15/11/2021	Seluruh Anggota	Bagi Tugas 3
15/11/2022	Averrous Saloom	Menambahkan efek yang bisa bekerja
15/11/2021	Jova Andres Riski Sirait	Ngefix bug di todo list
16/11/2021	Jova Andres Riski Sirait	Wrap up seluruh source code
18/11/2021	Louis Yanggara	Finalisasi laporan

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

No. Kelompok/Kelas : Kelompok 6/ K-02  
Nama Kelompok : Hari-Hari Ngetroll  
Anggota Kelompok (Nama/NIM) :  
1. Louis Yanggara/13520063  
2. Jova Andres Riski Sirait/13520072  
3. Samuel Christopher Swandi/13520075  
4. Saul Sayers/13520094  
5. Averrous Saloom/13520100  
  
Asisten Pembimbing : Muhammad Kamal Shafi/13518113

---

# Asistensi I






Tanggal : 26 Oktober 2021	Catatan Asistensi:
Tempat : Google Meet	
<p><b>Kehadiran Anggota Kelompok:</b></p> <p>No NIM Tanda tangan</p> <p>1. 13520063 </p> <p>2. 13520072 </p> <p>3. 13520075 </p> <p>4. 13520094 </p> <p>5. 13520100 </p>	<p>Q1 : Di dalam peta ada beberapa warna, itu bagaimana membuat warnanya? A : Pada halaman 6 spesifikasi, ada ketentuan pewarnaan untuk setiap lokasi. Jangan lupa perhatikan prioritas pewarnaan.</p> <p>Q2 : Semua pesanan ada di konfigurasi ya? Jadi di ingame gabisa nambah? A : Semua pesanan di konfigurasi termasuk waktu masuk dari pesanan.</p> <p>Q3 : Kami berlima, jadi bingung mau gimana bagi jadi 5 agar proporsional. A : Pertama, liat dulu ada berapa tasknya, itu bisa dipecah-pecah. Terus lihat bebannya jadi baginya tergantung kemampuan masing-masing orang. Terus perhatikan keterkaitannya karena ada task yang harus diselesaikan sebelum task lain. Jadi sebenarnya ada banyak cara untuk bagi.</p> <p>Q4 : Untuk matrix adjacency di contoh konfigurasi, semua elemennya biner antara 0 atau 1. Apakah mungkin ada sebuah hubungan lokasi yang butuh waktu untuk berpindah selain angka itu kak ? Misalnya 2 lokasi waktu berpindahnya butuh 2 satuan waktu bukan 1 karena jauh. A : Tidak, elemen matriksnya tetap biner karena hanya menunjukkan dua lokasi apakah bisa saling berpindah atau tidak. Jadi boleh diasumsikan kalau tiap lokasi butuh 1 satuan waktu saja apabila ada hubungannya.</p>
	Tanda Tangan Asisten:

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

No. Kelompok/Kelas : Kelompok 6/ K-02  
Nama Kelompok : Hari-Hari Ngetroll  
Anggota Kelompok (Nama/NIM) :  
1. Louis Yanggara/13520063  
2. Jova Andres Riski Sirait/13520072  
3. Samuel Christopher Swandi/13520075  
4. Saul Sayers/13520094  
5. Averrous Saloom/13520100  
  
Asisten Pembimbing : Muhammad Kamal Shafi/13518113

---

## Asistensi II

Tanggal : 9 November 2021	<p><b>Catatan Asistensi:</b></p> <p>Untuk progress, konfigurasi sudah, move sudah, warna sudah, yang belum hanya gadget, ability, dan perishable item saja.</p> <p><b>Q1:</b> Buat compile program, bisa langsung import file ke main. Kalau begitu tinggal buat main.c kan. Namun kalau hanya include header harus include semua file c yang ada. Nah itu gapapa kan?</p> <p><b>A:</b> Itu lebih ke best practice, tapi kalau hanya c bisa terjadi error, salah satunya error pada variabel karena setiap header bisa saja menginclude file header lain sehingga terjadi double declaration.</p> <p><b>Q2:</b> Kalau include header yang circular dependency itu bisa ga?</p> <p><b>A:</b> Sudah agak lupa, mungkin harus dicari tahu lagi.</p> <p><b>Q3:</b> Di program kami, interaksi dengan user terjadi di terminal, namun di spesifikasi ditulis interaksi menggunakan mesin karakter dan mesin kata</p> <p><b>A:</b> Itu bisa dimodifikasi, sehingga ada 2 mode untuk membaca masukkan dari user sesuai dengan spesifikasi.</p> <p><b>Q4:</b> Semua pekerjaan kami ada dalam 1 folder, itu harus diubah ga kak?</p> <p><b>A:</b> Itu ga terlalu masalah untuk tubes.</p> <p><b>Q5:</b> Di spesifikasi, contoh konfigurasi hanya ada 1, jadi takutnya ada error kalau beda, apakah akan diberi konfigurasi lain? Terus untuk demo nanti pakai konfigurasi yang punya kami?</p> <p><b>A:</b> Untuk saat ini belum ada, kalau bisa buat aja sendiri untuk mencoba programnya. Untuk penilaian asisten pakai konfigurasinya sendiri.</p> <p><b>Q6:</b> Pertanyaan tentang perishable item, misalnya sekarang detik 55 terus item hilang detik ke 60 artinya tersisa 5 detik, terus ada item yang mengurangi waktu, kalau dipakai itu bakal ngurangin waktunya ga ya?</p> <p><b>A:</b> Misalnya dia hangus dalam 10 satuan waktu, maka tetap dihitung sesuai dengan waktu yang berjalan.</p>
Tempat : Google Meet	
<p><b>Kehadiran Anggota Kelompok:</b></p> <p>No NIM Tanda tangan</p> <p>1. 13520063</p>  <p>2. 13520072</p>  <p>3. 13520075</p>  <p>4. 13520094</p>  <p>5. 13520100</p> 	<p><b>Tanda Tangan Asisten:</b></p>