

# **LAPORAN TUGAS BESAR**

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

### **Willy Wangky's World**

Dipersiapkan oleh:

Kelompok 1

Ronggur Mahendra Widya Putra (13519008)

Kinantan Arya Bagaspati (13519044)

Syamil Cholid Abdurrasyid (13519052)


Girvin Junod (13519096)

Jonathan Christoper Jahja (13519144)

Jauhar Wibisono (13519160)

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	<b>Sekolah Teknik Elektro dan Informatika ITB</b>	Nomor Dokumen		Halaman
		IF2110_TB_01_04		20
		Revisi	0	30 November 2020

# Daftar Isi

1 Ringkasan	4
2 Penjelasan Tambahan Spesifikasi Tugas	5
2.1 ADT Wahana	5
2.2 ADT N-ary Tree	5
2.3 ADT Pengunjung	5
2.4 ADT Material	5
2.5 Penggunaan globalvariable.h	5
2.6 Pembuatan File main phase dan preparation phase	6
2.7 ADT Aksi	6
2.8 Perintah yang Memakan Waktu	6
2.9 Perintah yang Membutuhkan Uang	6
2.10 Syarat Kesabaran Berkurang	6
3 Struktur Data (ADT)	6
3.1 Point	8
3.2 Jam	8
3.3 List Implementasi Array	8
3.4 Matriks	8
3.5 Mesin Karakter + Mesin Kata	8
3.6 Queue (Priority Queue)	9
3.7 Stack	9
3.8 List Implementasi List Berkait	9
3.9 Tree	9
3.10 Graph (Variasi Multilist)	9
3.11 Wahana	10
3.12 Pengunjung.	10
3.13 ADT Aksi	10
3.14 ADT Material	10
4 Program Utama	11
5 Algoritma-Algoritma Menarik	11
6 Data Test	11
6.1 Data Test Menu	11
6.2 Data Test Movement	12
6.3 Data Test Preparation Phase	12
6.3.1 Data Test Buy	12

6.3.2 Data Test Build	12
6.3.3 Data Test Undo	12
6.3.4 Data Test Upgrade	12
6.3.5 Data Test Execute	13
6.3.6 Data Test Main	13
6.4 Data Test Main Phase	13
6.4.1 Data Test Serve	13
6.4.2 Data Test Office	13
6.4.3 Data Test Detail	13
6.4.4 Data Test Repair	14
6.4.5 Data Test Prepare	14
6.5 Data Test Wahana Rusak	14
6.6 Data Test Antrian Pengunjung	14
8 Pembagian Kerja dalam Kelompok	17
9 Lampiran	17
9.1 Deskripsi Tugas Besar 2	17
9.2 Notulen Rapat	19
9.3 Log Activity Anggota Kelompok	19
9.4 Form Asistensi	20

# 1 Ringkasan

Dalam tugas besar ini diberikan tugas untuk merancang suatu program yang menyimulasikan suatu permainan membangun dan mengelola wahana Willy Wangky dengan pemain berperan sebagai pemilik wahana.

Permainan dibagi menjadi *main menu*, *preparation phase*, dan *main phase*. Ketika game pertama kali mulai terbuka main menu yang ada pilihan NEW GAME dan EXIT. Jika EXIT dipilih, maka program akan berhenti. Jika NEW GAME dipilih, maka permainan akan mulai.

Ketika permainan mulai akan langsung diterjunkan ke *preparation phase*. Permainan akan terlihat melalui *map* yang dituliskan. Akan ada 4 *map* yang dihubungkan sehingga pemain bisa pindah *map*. Pemain bisa bergerak dengan *input* 'w', 'a', 's', 'd' untuk bergerak sama dengan gim pada umumnya. Di *preparation phase* pemain bisa melakukan aksi membuat wahana, meng-*upgrade* wahana, membeli bahan untuk membuat dan meng-*upgrade* wahana, melakukan undo untuk hal-hal yang telah dilakukan di *preparation phase* itu, dan mengeksekusi aksi2 itu lalu lanjut ke *main phase*. Untuk membeli bahan itu dibutuhkan uang dan jika uang pemain tidak cukup maka maka pembelian wahana akan gagal. Untuk setiap aksi ini memakan waktu kecuali *undo* dan *execute*. Jika total waktu yang dimakan oleh aksi melebihi batas waktu *preparation phase*, maka *execute* akan gagal dan harus dilakukan *undo*. Dengan *undo* aksi yang terakhir dilakukan akan di-*undo* dan waktu yang termakan akan kembali.

Setelah *preparation phase*, permainan lanjut ke *main phase*. Dalam *main phase*, akan ter-generate pengunjung-pengunjung seiring waktu wahana buka. Setiap pengunjung akan ada daftar wahana yang ingin dinaikinya. Seluruh pengunjung ada dalam suatu *priority queue* yang berlaku untuk keempat map dan prioritasnya berdasarkan kesabaran pengunjung. Pengunjung yang datang ini harus di-*serve* oleh pemain agar mereka bisa memainkan wahana. *Command serve* akan memunculkan opsi wahana yang ingin dinaiki pengunjung. Ketika mereka main wahana, mereka akan keluar dari *queue* dan berada di wahana untuk waktu tertentu. Setelah selesai naik wahana, mereka akan kembali ke *queue* dengan prioritas lebih tinggi. Ketika pengunjung tidak di *serve*, kesabaran mereka akan berkurang dan jika kesabaran habis, maka mereka akan keluar dari antrian. Wahana dapat rusak ketika dinaiki pengunjung. Kerusakan ini *random* dan jika ada pengunjung yang sedang menaiki wahana yang rusak maka mereka akan dikembalikan ke antrian dengan prioritas tertinggi. Jika antrian penuh maka pengunjung akan keluar dari wahana. Pengunjung tidak dapat menaiki wahana yang rusak. Pemain dapat memperbaiki wahana dengan mendekati wahana tersebut dan melakukan *command repair*. Uang akan bertambah untuk setiap *serve* yang dilakukan pemain. Jika pengunjung sudah tidak ada wahana yang ingin dinaiki lagi maka akan keluar dari antrian. Ada juga *command detail* dan *office* yaitu untuk melihat detail wahana yang ada di samping pemain dan untuk membuka *menu office*. *Menu office* sendiri isinya daftar dan laporan dari tiap wahana. Ketika waktunya habis, maka seluruh pengunjung akan dikeluarkan dari wahana dan lanjut ke *preparation phase* lagi. Ini berlangsung terus-menerus tanpa akhir.

Laporan ini berisi ringkasan persoalan tugas besar mata kuliah IF2110 Algoritma dan Struktur Data ITB semester 1 2020/2021, penjelasan spesifikasi tugas besar tambahan, penjelasan struktur-struktur data (ADT) yang digunakan di tugas, penjelasan tentang algoritma program utama, penjelasan algoritma menarik yang digunakan di tugas, data test untuk *testing* program, pengetesan program dengan *data test* beserta penjelasan fitur yang di *test*, langkah *testing*, hasil, dan penjelasan hasilnya, pembagian kerja tugas, dan lampiran berupa deskripsi tugas besar dari spesifikasi tugas yang dibagikan, notulen rapat, dan *log activity* kelompok.

Dari tugas besar ini dapat kami simpulkan bahwa hasilnya sukses dan program dapat berjalan dengan baik sesuai spesifikasi setelah kami tes namun ada beberapa *bug-bug* yang *minor*. Perencanaan pengerjaan tugas kami kurang tersebar rata mengingat durasi pengerjaan tugas namun tugasnya tetap dapat selesai dengan cukup baik.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 ADT Wahana

ADT wahana dibuat untuk mempermudah representasi wahana di program sehingga pembuatan ADT *tree* dan *main* program tidak sulit karena sudah disederhanakan ADT wahana.

### 2.2 ADT N-ary Tree

ADT *tree* yang kami gunakan untuk pohon *upgrade* wahana bukan *binary tree* seperti yang diajarkan di kelas. ADT *tree* kami untuk tiap node dapat memiliki lebih dari 2 anak sehingga tidak *binary*.

### 2.3 ADT Pengunjung

ADT pengunjung kami buat untuk merepresentasikan tiap pengunjung di permainan. ADT ini berisi data-data pengunjung seperti wahana yang ingin dinaiki, kesabaran, dll.

### 2.4 ADT Material

ADT material ini digunakan untuk membantu ADT *stack* sebagai representasi dari bahan-bahan dan toko bahan yang diperlukan untuk membangun dan meng-*upgrade* wahana.

### 2.5 Penggunaan *globalvariable.h*

STEL-ITB	IF2110_TB_01_04	Halaman 5 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		

Dibuat file *globalvariable.h* yang isinya adalah include untuk semua file ADT dan semua variabel global. File ini gunanya untuk memudahkan *run* program dan untuk menghindari konflik antar ADT dan mempermudah program utama. Variabel global yang ada di *globalvariable.h* adalah :

## 2.6 Pembuatan File *main phase* dan *preparation phase*

Dalam program ini dibuat file *mainphase.h*, *mainphase.c*, *prepphase.h*, *prepphase.c*. File ini dibuat untuk membantu jalannya *main* program dengan file *mainphase* untuk bagian *main phase* dan file *prepphase* untuk bagian *preparation phase*.

## 2.7 ADT Aksi

ADT wahana dibuat untuk mempermudah pembuatan *stack* aksi di *preparation phase*. Setiap elemen *stack* dibuat menjadi ADT aksi ini.

## 2.8 Perintah yang Memakan Waktu

Di spesifikasi ada beberapa spesifikasi perintah yang memakan waktu yang tidak jelas seperti DETAILS dan OFFICE. Dalam program kami, kami membuat perintah bergerak hanya memakan waktu di *main phase* dan perintah DETAILS dan OFFICE beserta opsi OFFICE tidak memakan waktu.

## 2.9 Perintah yang Membutuhkan Uang

Di spesifikasi ada beberapa spesifikasi perintah yang membutuhkan uang tidak jelas seperti UPGRADE dan REPAIR. Dalam program kami, kami membuat perintah UPGRADE dan REPAIR membutuhkan uang sesuai dengan tiap wahana.

## 2.10 Syarat Kesabaran Berkurang

Di spesifikasi tidak jelas kapan kesabaran berkurang. Kami menentukan kesabaran berkurang setiap kali pengunjung tidak di-SERVE dan suatu aksi yang memakan waktu dilakukan. Kami juga menentukan kesabaran maksimum pengunjung 100.

# 3 Struktur Data (ADT)

Berikut adalah ADT-ADT yang kami gunakan dalam program. *Main phase* dan *prep phase* di gambar di bawah ini adalah bagian dari program utama. Program utama kami bagi menjadi dua file yaitu *main phase* dan *prep phase* ini untuk memudahkan pembagian kerja. Yang dimaksud dengan diagram ini adalah jika ADT A → ADT B maka ADT A itu digunakan dalam ADT B. Diagram ini tidak menunjukkan hubungan ADT ke program utama namun hanya keterhubungan antar ADT.

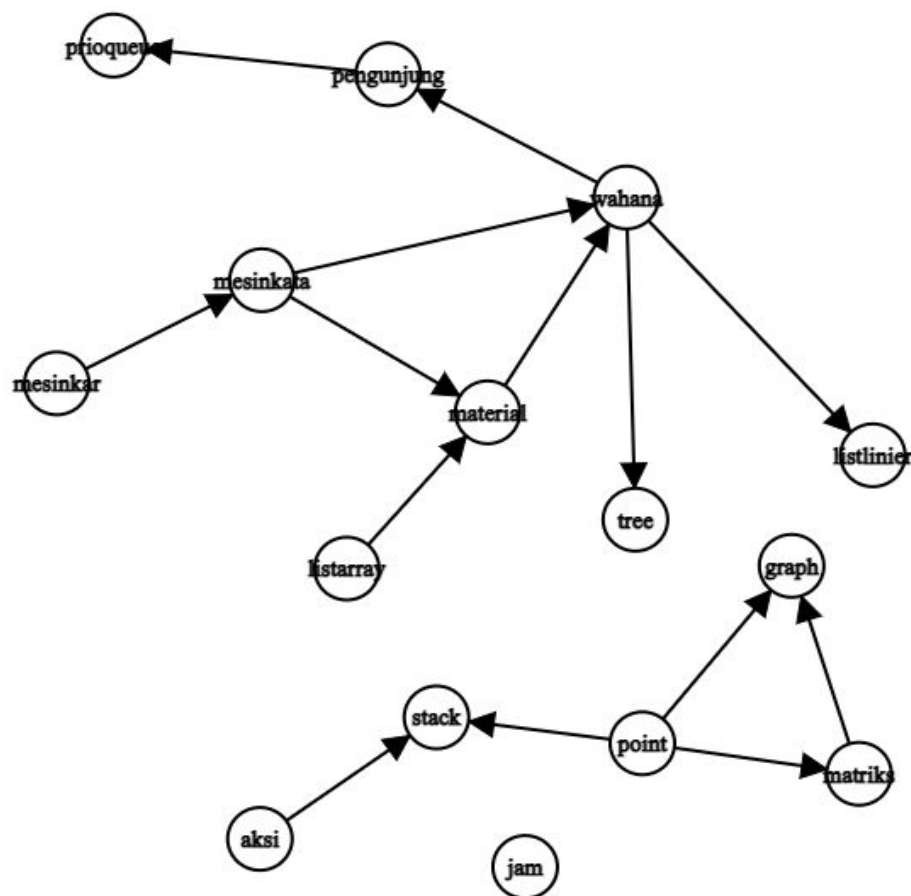
STEL-ITB	IF2110_TB_01_04	Halaman 6 dari 20 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB		

Dari diagram ini dapat dilihat ADT mesinkar digunakan dalam ADT mesinkata, lalu ADT mesinkata ini digunakan di ADT material dan wahana. ADT material menggunakan ADT mesinkata dan listarray dan digunakan juga di ADT wahana. ADT wahana sendiri digunakan dalam ADT *tree*, listlinier, dan pengunjung. ADT pengunjung dipakai untuk ADT *priority queue*.

Juga dapat dilihat ADT point digunakan untuk ADT matriks, *graph*, dan *stack*. ADT matriks sendiri juga digunakan dalam ADT *graph*. ADT *stack* juga menggunakan ADT aksi selain ADT point.

ADT jam sendiri tidak digunakan atau menggunakan ADT lainnya.

Semua ADT ini akan digunakan dalam program utama. Program utama sendiri dibagi menjadi *main phase* dan *preparation phase*. Dalam program kami, kami membuat sebuah *globalvariable.h* yang menampung semua variabel global dan include untuk semua ADT. Ini dilakukan sehingga untuk program utama hanya perlu *include globalvariable.h* saja.



Gambar 3.1 Diagram keterhubungan ADT

### 3.1 Point

ADT point terdiri dari absis dan ordinat yang bernilai integer, nilai absis dan ordinat ini berfungsi untuk melambangkan titik sebagai suatu koordinat kartesius. Dengan adanya ADT point ini maka ADT point digunakan untuk melambangkan pemain, wahana, dan gerbang di map. ADT point dipakai karena sesuai dengan kebutuhannya karena pemain, wahana, dan gerbang merupakan titik-titik di peta. Dalam ADT Point kami, kami membuat semua nilai point sebagai nilai positif dan nilai -1 sebagai nilai tidak terdefinisi. ADT point ini di program diimplementasikan sebagai ADT dalam file point.h dan point.c di folder point.

### 3.2 Jam

ADT jam terdiri dari 3 integer yaitu integer jam, menit, dan detik. ADT jam ini berfungsi untuk merepresentasikan jam dan waktu. Jalannya wahana di program berdasarkan waktu, aksi memakan waktu dan wahana tutup jika melewati waktu tertentu, semua yang berhubungan dengan waktu ini diselesaikan dengan ADT jam. Oleh karena itu, kami menggunakan ADT jam untuk menyelesaikan masalah waktu di program kami. ADT jam kami implementasikan sebagai ADT dalam file jam.h dan jam.c di folder jam.

### 3.3 List Implementasi Array

ADT list implementasi array terdiri dari array yang berisi char dan integer. Penggunaan list array ini berguna untuk membuat daftar seperti daftar bahan yang dibutuhkan dan bisa dibeli untuk membangun dan meng-upgrade wahana. List implementasi array ini dipilih karena dapat digunakan untuk tujuan penggunaannya dan tidak rumit. ADT list implementasi array ini diimplementasikan sebagai ADT dalam file listarray.h dan listarray.c di folder listarray.

### 3.4 Matriks

ADT matriks terdiri dari char legenda, jumlah maksimum baris, jumlah maksimum kolom, dan titik lokasi pemain. ADT matriks ini digunakan untuk pembuatan dan *display map*. Ini karena struktur map dan matriks itu mirip sehingga map dapat direpresentasikan dalam matriks. Karena itu ADT matriks digunakan untuk membuat dan menuliskan map. Dalam ADT matriks ini kami buat baris dan kolom maksimal tiap peta itu 50. ADT matriks ini kami implementasikan dalam ADT map.h dan map.c di folder map.

### 3.5 Mesin Karakter + Mesin Kata

ADT mesin karakter terdiri dari char untuk karakter, boolean global EOP, dan definisi MARK sebagai ' '. ADT mesin karakter dipakai untuk mesin kata yang terdiri dari string kata, definisi BLANK sebagai ' '

STEL-ITB	IF2110_TB_01_04	Halaman 8 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		



atau newline, integer panjang kata yaitu 50, boolean global EndKata, dan variabel kata global CKata. ADT mesin karakter dan mesin kata ini berguna untuk membaca *input* yang berupa kata. Ini banyak dibutuhkan di program karena program menerima banyak *input* dari file yang berupa kata. Karena ADT-ADT ini menyelesaikan permasalahan ini maka kami menggunakan ADT mesin karakter dan mesin kata. ADT - ADT ini diimplementasikan dalam file mesinkar.h, mesinkar.c di folder mesinkata.

### 3.6 *Queue (Priority Queue)*

ADT *queue* terdiri dari *address head queue*, *tail queue*, *max element queue*, dan elemen. Elemen *queue* ini terdiri dari integer prioritas, dan pengunjuk yang diatur ADT pengunjuk. ADT *queue* ini berfungsi untuk memprogramkan antrian pengunjuk, karena memang didesain untuk merepresentasikan antrian. Karena kecocokannya, kami menggunakan ADT *queue* ini, menggunakan *priority queue* karena antrian pengunjuk memang ada prioritasnya berdasarkan kesabaran. ADT ini diimplementasikan di file prioqueuePengunjuk.h dan prioqueuePengunjuk.c di folder queue.

### 3.7 *Stack*

ADT *stack* ini terdiri dari *address stack* dan struct aksi. ADT *stack* ini digunakan untuk masa *preparation phase* sebagai *stack* aksi. ADT *stack* ini digunakan karena cocok dengan kebutuhan program yaitu kemampuan untuk undo karena kemampuan *stack* untuk meng-undo sesuatu aksi di *preparation phase* dengan pop *stack*-nya. ADT ini diimplementasikan dalam file stack.h dan stack.c.

### 3.8 *List Implementasi List Berkait*

ADT *list* implementasi *list* berkait ini terdiri dari *address-address* yang dikaitkan menjadi *list*. Jadi suatu *list* hanya diketahui *address* dari *first* elemennya. Dari *first* elemennya akan tahu nilai elemennya dan *address* selanjutnya dan seterusnya. ADT ini berguna dan digunakan untuk membuat *list history upgrade* dari wahana. ADT *list* ini dipakai karena efektif dan efisien untuk membuat *list history upgrade* wahana. ADT ini diimplementasikan dalam file listlinier.h dan listlinier.c di folder list linier.

### 3.9 *Tree*

ADT *tree* terdiri dari *address-address* node yang dihubungkan melalui hubungan *parent* dan *child*. Ada juga integer jumlah maksimum anak dari parent dan elemen dari tiap *address* yang berupa wahana. Ini dikarenakan ADT *tree* ini digunakan untuk membuat *tree upgrade* wahana. ADT *tree* berguna untuk membuat *tree upgrade* wahana karena bentuk *upgrade* wahana itu mirip *tree* sehingga cocok untuk digunakan ADT *tree*. ADT *tree* ini diimplementasikan dalam file tree.h dan tree.c di folder tree.

### 3.10 *Graph (Variasi Multilist)*

STEL-ITB	IF2110_TB_01_04	Halaman 9 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		

ADT *graph* variasi *multilist* terdiri dari graf yang terdiri dari *address* first simpul, *list* simpul, dan *list* simpul-simpul yang berdampingan untuk tiap simpul di list simpul. Untuk tiap node atau simpul ada integer nama ID untuk map, map yang ada di node, integer jumlah simpul yang terhubung ke simpul itu, address ke list simpul berdampingan ke simpul itu, dan address ke simpul selanjutnya. ADT graf ini berguna untuk membuat peta yang saling berhubungan karena di spesifikasi tugas disebutkan ada 4 peta yang berhubungan. Oleh karena itu, ADT graf dipakai agar keempat peta dapat ditempatkan dalam graf itu. ADT graph ini diimplementasikan dalam file *graph.h* dan *graph.c* di folder *graph*.

### 3.11 Wahana

ADT wahana adalah ADT yang dibuat untuk merepresentasikan wahana-wahana. ADT ini terdiri dari informasi tiap wahana yaitu nama, ID, waktu pembangunan, bahan yang dibutuhkan, ukuran, durasi, kapasitas, harga, kesempatan rusak, posisi, total jumlah penghasilan, total jumlah pengunjung, jumlah pengunjung hari ini, dan jumlah penghasilan hari ini. ADT ini berguna untuk merepresentasikan wahana dan membuat program wahana lebih sederhana. ADT ini diimplementasikan dalam file *wahana.h* dan *wahana.c* di folder *wahana*.

### 3.12 Pengunjung.

ADT pengunjung adalah ADT yang dibuat untuk merepresentasikan pengunjung. ADT ini terdiri dari informasi tiap pengunjung yaitu kesabaran, wahana yang ingin dinaiki, dan wahana yang sedang dinaiki. ADT ini berguna untuk merepresentasikan pengunjung dalam program sehingga programnya lebih sederhana. ADT ini diimplementasikan dalam file *pengunjung.h* dan *pengunjung.c* di folder *pengunjung*.

### 3.13 ADT Aksi

ADT aksi ini merupakan ADT yang merepresentasikan aksi-aksi yang dilakukan pemain di *preparation phase*. ADT aksi ini dipakai dalam stack aksi di *preparation phase*. ADT terdiri dari integer identitas aksi, struct *build*, struct *upgrade*, dan struct *buy*. Struct *build* terdiri dari titik koordinat bangunan, ID bangunan, harga bangunan, dan durasi membangun. Struct *upgrade* terdiri dari titik koordinat bangunan, ID bangunan, harga upgrade, dan durasi upgrade. Struct *buy* terdiri dari ID barang, jumlah barang, harga pembelian, durasi pembelian. ADT aksi ini membuat pemrograman stack aksi lebih mudah karena aksi telah menjadi suatu ADT. Karena kemudahan ini maka kami membuat dan menggunakan ADT aksi ini. ADT aksi ini diimplementasikan di file *stack.h* dan *stack.c*

### 3.14 ADT Material

ADT material terdiri dari material yang merupakan *array of integer*. Ini digunakan untuk merepresentasikan material di *inventory* dan di *shop*. Kami menggunakan ADT ini karena ADT-nya

STEL-ITB	IF2110_TB_01_04	Halaman 10 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		

berguna dalam pemrograman wahana, *inventory*, dan toko di *preparation phase*. File ADT material ini diimplementasikan di file *material.h* dan *material.c* di folder *material*.

## 4 Program Utama

Program utama kami bagi menjadi file *prepphase*, *mainphase*, dan *main*. File *prepphase* adalah file untuk *preparation phase* dari permainan, file *mainphase* adalah file untuk *main phase* dari permainan, dan file *main* adalah file untuk menjalankan permainan.

File *preparation phase* akan menyediakan beberapa *command* yang hanya bisa digunakan pada *preparation phase*. Pada prosedur ini, *input* yang bisa dibaca hanya *movement command*, *execute*, *build*, *buy*, *upgrade*, dan *main* untuk melanjutkan ke *main phase*. Semua kegiatan yang dipilih *player* akan disimpan terlebih dahulu pada sebuah *stack* dan akan dijalankan ketika *command execute* dijalankan.

File *main phase* dapat menerima beberapa *command* yang berbeda dengan *preparation phase*, seperti *serve*, *repair*, *office*, dan *detail*. Tapi *command preparation phase* tidak akan berfungsi pada *main phase*. Perlu diperhatikan beberapa kegiatan - kegiatan dari *preparation phase* dan *main phase* memakan durasi yang beragam. Jadi *player* harus bisa mengalokasikan waktu dengan baik agar dapat menjalankan setiap aktivitas dengan efektif.

File *main* dari program utama berfungsi untuk menginisiasi semua variabel global yang dibutuhkan, contohnya uang dan map. Selain itu file *main* juga mengandung prosedur *main* yang akan menggunakan *loop while* untuk mengulang prosedur *preparation phase* dan *main phase*. Setiap 1 *loop* akan menambah 1 hari pada variabel *day*.

## 5 Algoritma-Algoritma Menarik

### 5.1 *N-ary Tree*

ADT *tree* kami menggunakan *N-ary Tree*, meskipun dari asisten disarankan menggunakan Binary Tree. Tree ini digunakan untuk menunjukan pilihan *upgrade* untuk tiap wahana, jika tersedia. Karena kami menggunakan *N-ary tree*, pilihan *upgrade* bisa lebih dari 2.

## 6 Data Test

### 6.1 *Data Test Menu*

STEL-ITB	IF2110_TB_01_04	Halaman 11 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		

Fitur yang dites adalah *main menu* dan *input* saat masuk *main menu*. Jika *input* 1 maka seharusnya masuk ke permainan. Jika *input* 2 maka seharusnya keluar permainan. Jika diluar ini maka ada validasi. Dites dengan menjalankan program lalu *input* 1, 2 atau *input* lain

## 6.2 Data Test Movement

Fitur yang dites adalah pergerakan pemain saat di *preparation phase* dan *main phase*. Hasilnya harusnya pemain dapat bergerak ke atas, kanan, kiri, bawah, berpindah map jika bergerak ke gerbang, dan tidak bergerak jika bergerak ke dinding atau wahana. Dites dengan *input* 'w', 'a', 's', dan 'd' saat bermain.

## 6.3 Data Test Preparation Phase

### 6.3.1 Data Test Buy

Fitur yang dites adalah perintah BUY pada *preparation phase*. Jika perintah dieksekusi maka program akan memberikan daftar bahan yang bisa dibeli lalu pemain akan menginput bahan yang ingin dibeli, kemudian program akan memvalidasi apakah pemain memiliki uang yang cukup untuk membeli bahan maka bahan akan dibeli, selain itu akan diberikan pesan bahan gagal dibeli. Juga ada validasi waktu yang dimakan BUY, jika melebihi waktu sisa dari *preparation phase*. Dites dengan melakukan perintah BUY di *preparation phase*.

### 6.3.2 Data Test Build

Fitur yang dites adalah perintah BUY pada *preparation phase*. Jika perintah dieksekusi maka program akan memberikan opsi wahana yang bisa dibangun, pemain akan menginput ID wahana yang hendak dibangun, lalu memasukan perintah tersebut ke stack beserta waktu yang termakan karena aksi. Ada dilakukan validasi command dalam bentuk pengecekan biaya BUILD, bahan yang dimiliki pemain, waktu yang dimakan, dan bahan yang dibutuhkan wahana yang ingin dibangun. Jika melebihi maka akan gagal. Dites dengan melakukan perintah BUILD di *preparation phase*.

### 6.3.3 Data Test Undo

Fitur yang dites adalah perintah UNDO pada *preparation phase*. Jika perintah dieksekusi maka program akan menghilangkan perintah teratas pada stack dan waktu yang termakan jadi kembali. Ada validasi jika stack aksi kosong, maka tidak akan terjadi apa-apa. Dites dengan melakukan perintah UNDO di *preparation phase*.

### 6.3.4 Data Test Upgrade

Fitur yang di tes adalah perintah UPGRADE pada *preparation phase*. Jika perintah dieksekusi maka program akan memberikan list upgrade dari wahana tersebut. *Player* dapat memilih wahana yang ingin di

STEL-ITB	IF2110_TB_01_04	Halaman 12 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		

upgrade pada wahana yang bersangkutan. Perintah ini akan ada validasi waktu yang dimakan, biaya, dan bahan yang dimiliki pemain. Jika tidak memenuhi syarat maka perintah akan gagal. Fitur ini dites dengan menggunakan perintah UPGRADE dengan pemain berada di *tile adjacent* ke suatu wahana lalu memilih *upgrade* yang diinginkan.

### 6.3.5 Data Test Execute

Fitur yang dites adalah perintah BUY pada *preparation phase*. Jika perintah dieksekusi maka program akan mengeksekusi semua program yang ada pada *stack*. Dites dengan menggunakan perintah EXECUTE di *preparation phase* saat pada berbagai tingkat *stack* aksi.

### 6.3.6 Data Test Main

Fitur yang dites adalah perintah MAIN pada *preparation phase*. Jika perintah dieksekusi maka program akan mengubah state ke *main phase*. Dites dengan menggunakan perintah MAIN saat *preparation phase*.

## 6.4 Data Test Main Phase

### 6.4.1 Data Test Serve

Fitur yang dites adalah perintah SERVE <nama wahana> pada *main phase*. Jika perintah dieksekusi maka program akan mengecek jika pemain ada di samping antrian, jika tidak maka perintah gagal, jika ya dan nama wahana di perintah cocok dengan wahana yang ditunjukkan di display antrian maka program akan mengeluarkan pengunjung terdepan antrian dan memasukkannya ke wahana yang diinput di perintah. Jika nama wahana tidak cocok dengan display antrian atau wahananya sedang rusak maka perintah SERVE akan gagal dan tidak terjadi apa-apa.

### 6.4.2 Data Test Office

Fitur yang dites adalah perintah OFFICE pada *main phase*. Jika perintah dieksekusi saat pemain ada di *tile office*, maka program akan menampilkan menu office dengan opsi *Details*, *Report*, dan *Exit*. Opsi *Details* menunjukkan informasi detail tiap wahana, opsi *Report* menunjukkan laporan tiap wahana, dan opsi *Exit* mengeluarkan dari menu *office*. Dilakukan validasi posisi pemain, jika pemain tidak di *tile office* maka perintah gagal. Harusnya semua opsi ini tereksekusi dengan baik. Fitur dites dengan input perintah OFFICE di *tile office* lalu input semua opsi.

### 6.4.3 Data Test Detail

Fitur yang dites adalah perintah DETAIL pada *main phase*, jika perintah dieksekusi perintah ini akan memberikan data mengenai wahana yang *adjacent* dengan pemain. Dites dengan menggunakan perintah DETAIL pada *main phase* dan harus di *tile office* atau di sebelah *tile* wahana.

STEL-ITB	IF2110_TB_01_04	Halaman 13 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		

#### 6.4.4 Data Test Repair

Fitur yang dites adalah perintah REPAIR pada *main phase*, perintah ini dieksekusi di sebelah suatu wahana yang rusak, jika perintah dieksekusi maka wahana yang rusak tersebut akan kembali berfungsi. Ada validasi harga *repair*, posisi pemain, dan status wahana sehingga jika uang pemain tidak cukup atau pemain tidak *adjacent* dengan wahana atau wahana tidak rusak maka perintah gagal. Dites dengan melakukan perintah REPAIR di tile *adjacent* wahana yang rusak.

#### 6.4.5 Data Test Prepare

Fitur yang dites adalah perintah PREPARE pada *main phase*. Jika perintah dieksekusi maka program akan mengubah state ke *preparation phase*. Harusnya perintah ini tereksekusi otomatis jika durasi *main phase* habis. Dites dengan menggunakan perintah PREPARE saat *main phase* dan menghabiskan waktu di *main phase*.

#### 6.5 Data Test Wahana Rusak

Fitur yang dites adalah wahana rusak. Harusnya wahana akan ada kesempatan untuk rusak tiap kali dinaiki pengunjung. Ketika wahana rusak maka semua pengunjung dikembalikan ke antrian dengan prioritas tertinggi. Wahana rusak juga tidak bisa dinaiki oleh pengunjung. Fitur ini dites dengan terus menerus me-SERVE pengunjung ke wahana sampai wahana rusak, lalu mencoba untuk SERVE pengunjung lagi ke wahana yang rusak itu.

#### 6.6 Data Test Antrian Pengunjung

Fitur yang dites adalah antrian pengunjung. Harusnya antrian pengunjung bertambah terisi secara random sepanjang *main phase* atau ketika pengunjung selesai naik wahana dan masih ada wahana yang ingin dinaiki, antrian berkurang ketika ada pengunjung di-SERVE atau pengunjung yang kesabarannya habis, dan antrian terurut secara prioritas yang berupa kesabaran pengunjung yang berkurang untuk tiap pengunjung untuk tiap aksi yang tidak SERVE pengunjung itu. Dites dengan melakukan aksi SERVE berkali-kali di *main phase* dan melakukan aksi lain sehingga kesabaran pengunjung berkurang sampai habis.

### 7 Test Script

No.	Fitur yang dites	Tujuan Testing	Langkah-langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	<i>New Game</i>	Berhasil memulai <i>game</i> baru	Menjalankan game, <i>input 1</i>	Data Test Menu	<i>Game</i> mulai	Sesuai harapan

STEL-ITB	IF2110_TB_01_04	Halaman 14 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		

2	<i>Exit</i>	Berhasil keluar dari <i>game</i>	Menjalankan <i>game</i> , <i>input 2</i>	Data Test Menu	Keluar dari <i>game</i>	Sesuai harapan
3	Bergerak (w, a, s, d)	Berhasil bergerak ke atas, kiri, bawah, kanan	Saat di permainan, <i>input w, a, s, d</i>	Data Test Movement	Pemain bergerak 1 tile menurut arah w,a,s,d	Sesuai harapan
4	BUILD	Berhasil membuat wahana, jika tidak memenuhi syarat gagal	<i>Input command BUILD</i> saat <i>preparation phase</i> . Pilih wahana.	Data Test Build	Wahana yang terpilih terbuat jika syarat terbangun terpenuhi, gagal jika tidak	Sesuai harapan
5	UPGRADE	<i>Berhasil upgrade</i> wahana jika memenuhi syarat, gagal jika tidak memenuhi	Input command UPGRADE saat <i>preparation phase</i> dan pemain <i>adjacent</i> wahana.	Data Test Upgrade	Wahana yang dipilih ter- <i>upgrade</i> jika syarat terpenuhi, gagal jika tidak	Sesuai harapan
6	BUY	Berhasil beli bahan jika syarat terpenuhi, gagal jika tidak	<i>Input command BUY</i> saat <i>preparation phase</i> , pilih bahan dan jumlah	Data Test Buy	Bahan bangunan bertambah jika syarat terpenuhi	Sesuai harapan
7	UNDO	Berhasil membatalkan perintah	<i>Input command UNDO</i> saat <i>preparation phase</i> , ketika ada perintah di stack	Data Test Undo	Perintah terakhir terbatalan	Sesuai harapan
8	EXECUTE	Berhasil menjalankan <i>stack</i> aksi	<i>Input command EXECUTE</i> saat <i>preparation phase</i> , ketika ada perintah di <i>stack</i>	Data Test Execute	<i>Stack</i> aksi pada <i>preparation phase</i> dilaksanakan	Sesuai harapan
9	MAIN	Berhasil masuk ke <i>main phase</i>	<i>Input command MAIN</i> saat <i>preparation phase</i>	Data Test Main	<i>Phase</i> berlanjut dari <i>preparation</i>	Sesuai harapan

					<i>phase ke main phase</i>	
10	SERVE	Berhasil melayani antrian pengunjung jika syarat terpenuhi	<i>Input command SERVE &lt;nama wahana&gt; saat main phase dan pemain adjacent antrian</i>	Data Test Serve	Antrian berkurang dan kesabaran pengunjung lainnya berkurang jika syarat terpenuhi	Sesuai harapan
11	REPAIR	Berhasil memperbaiki wahana jika syarat terpenuhi	<i>Input Command REPAIR saat main phase, dan adjacent dengan wahana yang rusak</i>	Data Test Repair	Wahana menjadi tidak rusak jika memenuhi syarat	Sesuai harapan
12	DETAIL	Melihat rincian dari wahana jika memenuhi syarat	<i>Input Command DETAIL saat main phase dan adjacent dengan wahana.</i>	Data Test Detail	Memberi rincian info dari wahana jika syarat terpenuhi	Sesuai harapan
13	OFFICE	Berhasil mengakses menu opsi <i>office</i> dan menjalankan perintahnya jika syarat terpenuhi	<i>Input Command OFFICE pada tile office, input opsi Details, Report, dan Exit</i>	Data Test Office	Membuka <i>menu office</i> dan opsi <i>Details, Report, dan Exit</i> jika syarat terpenuhi	Sesuai harapan
14	PREPARE	Berhasil kembali ke <i>preparation phase</i>	<i>Input Command PREPARE pada main phase atau menunggu waktu main phase habis</i>	Data Test Prepare	Memulai <i>preparation phase day</i> berikutnya	Sesuai harapan
15	Wahana rusak	Berhasil menunjukkan wahana bisa rusak dan interaksi pengunjung dan	<i>Input command SERVE &lt;nama wahana&gt; saat main phase sampai ada wahana rusak</i>	Data Test Wahana Rusak	Ada wahana yang rusak dan interaksi pengunjung dengan wahana	Sesuai harapan



		wahana rusak sesuai syarat			rusak benar	
16	Antrian Pengunjung	Berhasil menunjukkan antrian <i>priority queue</i> pengunjung berfungsi sesuai spesifikasi	<i>Input command</i> SERVE <nama wahana> berkali-kali dan <i>input command</i> lain untuk mengetes antrian	Data Test Antrian Pengunjung	Antrian pengunjung berfungsi sesuai spesifikasi	Sesuai harapan

Tabel 7.1 Tabel Pengetesan Program

## 8 Pembagian Kerja dalam Kelompok

Nama - NIM	Pembagian Kerja
Ronggur - 13519008	Driver, ADT Prioqueue, ADT pengunjung, debugging, laporan, form penilaian
Kinantan - 13519044	Main phase, ADT wahana, ADT tree, ADT pengunjung, main
Syamil - 13519052	ADT Material, debugging, ADT listarray, ADT listlinier
Girvin - 13519096	ADT Graph, ADT jam, laporan, notula, debugging, form penilaian, driver
Jonathan - 13519144	ADT Matriks, debugging, driver, laporan, ADT wahana, ADT Map, ADT Point
Jauhar - 13519160	Preparation phase, driver, ADT stack, debugging, main

Tabel 8.1 Pembagian Kerja

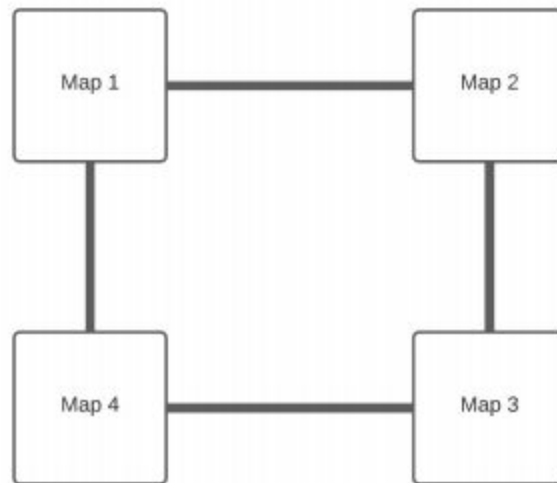
## 9 Lampiran

### 9.1 Deskripsi Tugas Besar 2

Willy Wangky telah menerima program yang kamu dan teman-temanmu telah buat untuk mengurus tiket taman bermainnya. Namun, taman bermain yang dibuat oleh Oompa Loompa dianggap kurang memuaskan oleh Willy Wangky. Maka dari itu, ia menginginkan kamu untuk membuat program dalam bahasa C untuk mensimulasikan taman bermain sehingga ia dapat mendesain taman bermainnya terlebih dahulu sebelum dibangun oleh Oompa Loompa.

Wahana bermain Willy Wangky memiliki 4 petak lahan yang dihubungkan oleh sebuah lorong. Salah satu dari 4 petak lahan yang dimiliki merupakan lahan khusus. Pada lahan tersebut, terdapat sebuah antrian dan sebuah office di mana Willy Wangky dapat melihat hal-hal berikut:

1. Melihat detail dari wahana.
2. Melihat laporan penjualan wahana.



Gambar 9.1 Contoh peta taman bermain

Sebelum hari dimulai, Willy Wangky ingin mempersiapkan segala sesuatu untuk wahana bermainnya. Kegiatan yang akan dilakukan saat persiapan adalah sebagai berikut:

1. Membeli bahan-bahan bangunan apabila uang mencukupi.
2. Membangun wahana baru di taman bermain untuk menambah pemasukan apabila bahan bangunan mencukupi.
3. Melakukan upgrade pada wahana yang sudah ada di taman bermain apabila bahan bangunan mencukupi.

Karena Willy Wangky juga sibuk mengurus pabrik coklatnya (selain mengurus wahana bermainnya), program simulasi harus mampu menyimpan kegiatan yang akan dilakukan saat persiapan. Kegiatan terakhir harus dimasukkan sebelum kegiatan sebelumnya dikarenakan Willy Wangky ingin melakukan Undo apabila aksi terakhir yang dia lakukan kurang memuaskan.

Selain mensimulasikan persiapan, program juga harus dapat mensimulasikan kegiatan pada saat hari kerja (saat taman bermain dibuka). Pada saat taman bermain dibuka:

1. Terdapat pengunjung yang datang setiap waktu tertentu dan mengantri untuk masuk ke dalam taman bermain dengan daftar jenis wahana yang ingin diikuti.
2. Setelah mengikuti sebuah wahana, pengunjung akan mengantri lagi dalam antrian dengan prioritas lebih tinggi. Dengan kata lain, pengunjung yang sudah dilayani harus dilayani terlebih dahulu.

Program simulasi juga harus memungkinkan Willy Wangky untuk:

1. Berjalan mengelilingi wahana bermain yang telah dibangun (wahana yang dibangun sendiri).
2. Melayani pengunjung yang berada dalam antrian.
3. Dikarenakan wahana memiliki kemungkinan untuk rusak, maka Willy Wangky harus mendatangi wahana yang bersangkutan untuk memperbaiki wahana tersebut.
4. Masuk ke dalam office dan melihat detail atau laporan wahana.

Berdasarkan deskripsi game di atas, buatlah sebuah program yang dapat mensimulasikan game tersebut menggunakan bahasa C. Interaksi pengguna dan komputer dilakukan melalui command line interface (CLI). Pengguna memasukkan perintah-perintah yang akan dijelaskan pada bagian game mechanics dan command.

## 9.2 Notulen Rapat

Berikut notulensi rapat kami yang diambil dari log chat grup Line kami dalam dokumen ini [Notulensi Rapat Kelompok 1 Kelas 4](#).

## 9.3 Log Activity Anggota Kelompok

Di commit history repository ini terlihat log activity tiap anggota kelompok:

- [Ronggur Mahendra Widya Putra 13519008](#)
  - Initial Commit repository Github
  - Push ADT queue dan prioqueue, ADT Point, dan ADT Pengunjung
  - Push Driver List array, Driver Prioqueue, Driver Listlinier, Driver Tree, Driver Graph, Driver List array, Driver pengunjung, Driver\_Jam,
  - Debugging global\_variable, Debugging ADT stack
- [Kinantan Arya Bagaspati 13519044](#)
  - Push ADT Wahana
  - Push Preperation phase
  - Adding Serve and Wahana
  - Mengintegrasikan ADT - ADT
  - Final Merge
- [Syamil Cholid Abdurrasyid 13519052](#)

STEL-ITB	IF2110_TB_01_04	Halaman 19 dari 20 halaman
<p>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</p>		

- Debugging Listlinier
- Push ADT Material dan listarray
- Menghubungkan ADT Material dengan Wahana
- [Girvin Junod 13519096](#)
  - Mengisi readme dengan identitas.
  - Push ADT jam dan matriks
  - Menambah ke ADT stack
  - Push ADT graph + map lalu dilanjutkan orang lain
  - Susun ulang repository
  - Push driver jam
  - Debugging
  - Pembuatan dan push laporan
- [Jonathan Christopher Jahja 13519144](#)
  - Push ADT Point, list array, dan list linier yang kemudian dilanjutkan anggota lain
  - Push ADT Map dengan matriks
  - Menggabungkan ADT Graph dan Map
  - Penambahan dan debugging fungsi-fungsi primitif Graph
  - Push Driver Graph dan Map
- [Jauhar Wibisono 13519160](#)
  - Pembuatan skeleton main
  - Push ADT Stack dan Aksi
  - Pembuatan Prep Phase
  - Pembuatan main khususnya Prep Phase

## 9.4 *Form Asistensi*

Berikut adalah link untuk form asistensi yang telah dikomentari asisten [IF2110\\_FormAsistensiTB\\_01\\_04](#).

STEL-ITB	IF2110_TB_01_04	Halaman 20 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		