

LAPORAN TUGAS BESAR

IF2110 / Algoritma dan Struktur Data


BATTLE FOR OLYMPIA

Dipersiapkan oleh:

KELOMPOK 6

13516016 Adylan Roaffa Ilmy
13516022 Albert Sahala Theodore
13516076 Haifa Fadhila Ilma
13516127 M. Azka Widyanto
13516130 Nadija Herdwina Putri S

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2110-TB-06-01</i>		<i>16</i>
		<i>Revisi</i>	<i>01</i>	<i>31 Oktober 2017</i>

Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Spesifikasi Unit White Mage	4
2.2 Spesifikasi Fitur Next Unit.....	4
2.3 Akuisisi Village.....	4
3 Struktur Data (ADT)	5
3.1 ADT Point	5
3.2 ADT Matriks	5
3.3 ADT Stack.....	5
3.4 ADT Queue	5
3.5 ADT List	6
3.6 ADT Mesin Kata	6
4 Program Utama	7
5 Data Test	8
5.1 Menu Utama.....	8
5.2 Movement Point	8
5.3 Undo	9
5.4 Recruit New Unit	11
5.5 Attack	11
5.6 Command InfoCell.....	12
5.7 1 Unit Player Berada di Sebelah White Mage.....	13
5.8 Memilih Unit yang akan digunakan oleh player	14
6 Test Script	16
7 Pembagian Kerja dalam Kelompok	19
8 Lampiran.....	20
8.1 Deskripsi Tugas Besar 2.....	20
8.2 Notulen Rapat.....	20
8.3 Log Activity Anggota Kelompok.....	22
8.4 Lembar Asistensi.....	24

1 Ringkasan

Program yang kami buat merupakan sebuah *game* berjudul Battle for Olympia. *Game* ini merupakan *game* yang memerlukan strategi dan aksi dilakukan oleh *user* melalui *command prompt*. Cara memenangkan *game* ini dengan membunuh raja pada sisi musuh juga dengan kerajaannya. Pemain memiliki pasukan yang dapat melawan pasukan musuh dan juga rajanya. Pasukan yang tersisa dalam permainan akan mati jika raja mereka juga mati karena dibunuh pasukan lawan.

Pada laporan yang kami buat, terdapat beberapa bagian diantaranya struktur data yang digunakan oleh program, algoritma yang dipakai pada program utama, masukan atau data yang dicoba pada program serta keluaran atau hasilnya, fitur-fitur pada program, dan pembagian kerja serta notulensi dari anggota kelompok.

Kesimpulan dari program yang kami buat bahwa *Battle for Olympia* adalah permainan yang bertujuan untuk menghancurkan kerajaan musuh dengan membunuh raja dari sisi lawan, permainan ini merupakan permainan yang dibuat dengan bahasa C dengan memanfaatkan Tipe Data Abstrak yang penerapannya dapat menambahkan fitur-fitur pada permainan ini.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Spesifikasi Unit White Mage

White mage adalah salah satu unit yang dapat direkrut oleh King. Tipe serangan unit ini adalah *Melee*. Selain itu, keberadaan unit ini dalam suatu petak menyebabkan unit-unit lain yang berada pada petak yang *adjacent* (satu petak diatas, bawah, kanan atau kiri) dengannya bertambah health sebanyak 5 point untuk tiap awal turn.

2.2 Spesifikasi Fitur Next Unit

Fitur ini memungkinkan pemain untuk langsung berpindah unit (otomatis) sesuai unit yang dimilikinya, berbeda dengan CHANGE_UNIT yang harus terlebih dahulu memilih unit sesuai daftar yang ada. Unit yang muncul saat command NEXT_UNIT hanyalah yang memiliki Movement Point (tidak 0) atau yang bisa melakukan Attack. Jika unit tersebut memiliki movement point 0 dan belum bisa melakukan Attack, unit tersebut tidak bisa terpilih pada command NEXT_UNIT.

2.3 Akuisisi Village

Sebagai tambahan, akuisisi Village hanya dapat dilakukan pada village yang belum berkepemilikan, jika sudah, player yang berada dalam petak yang sama dengan Village yang sudah diakuisisi oleh player lain tidak akan di 'rebut' oleh player tersebut.

3 Struktur Data (ADT)

Struktur data yang kami gunakan ada beberapa tipe diantaranya ADT Matriks, ADT Stack, ADT Queue, ADT List, dan ADT Point. Struktur data merupakan komponen penting dalam membentuk fitur dan tampilan pada program ini. Kami hanya memilih struktur data yang sesuai dengan kebutuhan dari program ini, sehingga tidak semua ADT digunakan dalam program.

3.1 ADT Point

ADT Point kami implementasikan dalam beberapa file program yang digunakan untuk pembuatan tugas besar ini, terutama "map.h". ADT Point disini kami gunakan untuk merepresentasikan koordinat atau posisi yang ada pada peta dalam permainan tersebut, selain itu Point juga digunakan untuk merepresentasikan lokasi para pemain dan lokasi *building* saat itu pada peta. Melalui penggunaan ADT Point, pemain dapat melihat posisi yang tepat dari *building* dan pasukan-pasukan dalam permainan tersebut.

3.2 ADT Matriks

ADT Matriks kami implementasikan dalam file "map.h" dimana penggunaannya untuk merepresentasikan peta permainan secara keseluruhan. Peta yang dibuat digambarkan sebagai matriks berukuran $m \times n$ yang tiap elemen matriks, yaitu $M[i][j]$ merepresentasikan satu petak pada peta. Koordinat sumbu Y pada peta merupakan baris pada matriks, dan koordinat sumbu X pada peta merupakan kolom pada matriks. Pada peta, semakin ke utara (jika dilihat di gambar berarti keatas), maka indeks baris akan berkurang 1, sebaliknya, jika pemain semakin ke selatan (ke bawah), maka indeks baris akan bertambah 1.

3.3 ADT Stack

ADT Stack kami implementasikan dalam file "command_list.h" dimana penggunaannya untuk prosedur undo yaitu dengan cara setiap koordinat suatu unit saat berpindah disimpan kedalam stack, lalu saat dipanggil prosedur undo tersebut maka koordinat terakhir akan di-*pop* dari stack sehingga unit berpindah ke koordinat sebelumnya. Koordinat yang dimasukkan terakhir ke dalam stack akan di-*pop* pertama kali.

3.4 ADT Queue

ADT Queue kami implementasikan dalam file "playerQueue.h" dimana penggunaannya untuk mengatur giliran permainan. Data pemain disimpan dalam struktur Queue yaitu data yang masuk pertama akan keluar pertama (FIFO).

3.5 ADT List

ADT List kami implementasikan dalam file "unitList.h" dan "buildingList.h" dimana penggunaannya untuk menyimpan list unit dan list building yang sudah dibuat dalam program. Implementasi dari program unitList dan buildingList menggunakan konsep ADT List yang di modifikasi sesuai kebutuhan penggunaan. Setiap pemain memiliki *List of Units* dan *List of Buildings* nya masing-masing, untuk menyimpan sejumlah unit yang dimiliki pemain, atau *building* yang dimiliki, seperti Castle, Tower, dan juga Village yang telah pemain tersebut akuisisi. Selain itu, program pada buildingList.h juga digunakan dalam pembuatan List of Villages, yang menyimpan sejumlah village yang belum punya kepemilikan.

3.6 ADT Mesin Kata

ADT Mesin Kata kami implementasikan dalam file "mesinlogo.h" dan "mesinkar.h" dimana penggunaannya untuk menampilkan logo dari Game Battle of Olympia yang telah kami buat pada file bernama "logo.txt". Di program utama, program akan memanggil prosedur TulisLogo() yang implementasinya telah dibuat di "mesinlogo.c" sehingga logo akan ditampilkan ke layar.

4 Program Utama

Program utama dari Game Battle For Olympia yang kami buat terdiri dari dua variabel yaitu M untuk Map dan Q untuk Queue. Pada saat program dibuka, akan keluar logo yang telah kami buat menggunakan mesin kata dan pemain diminta untuk mengisi ukuran peta dari game yang akan dimainkan. Setelah itu, program memanggil prosedur NewGame sehingga game akan dimulai. Game ini akan terus berjalan selama jumlah element dari variabel Q adalah 2, yaitu merepresentasikan banyaknya pemain yang ada, yakni Player 1 dan Player 2. Hal ini akan dilakukan terus-menerus dengan memanggil prosedur playTurn sebagai pergantian giliran pemain selama jumlah pemain masih 2. Apabila jumlah elemen sudah berkurang menjadi 1, dimana artinya salah satu pemain sudah kalah, maka game akan berakhir dan *loop* selesai. Setelah itu, akan muncul pesan berupa ucapan selamat kemenangan kepada pemain yang tersisa sebagai elemen dari variabel Q.

5 Data Test

5.1 Menu Utama

Fitur : clearscreen, TulisLogo, NbElmt, newGame, playTurn, InfoHead

Hasil yang diberikan saat menjalankan program adalah program menampilkan tampilan menu utama yang akan meminta ukuran dari peta yang akan digunakan untuk permainan. Peta itu di tampilkan pada layar sekaligus meminta command yang akan dijalankan oleh player dan menampilkan info player tersebut.



Gambar 1.1 Tampilan Menu Utama

5.2 Movement Point

Fitur : Move

Hasil yang diberikan saat menjalankan command move pada program maka unit tersebut akan berpindah ketika memenuhi syarat untuk berpindah. Pada awalnya movement point King adalah 1, yang artinya hanya bisa berpindah 1 petak yang adjacent dengan lokasinya. Setelah berpindah, movement point dari King menjadi 0. Perlu pergantian turn agar movement point dari suatu unit kembali seperti semula.


```

* * * * *
*****
* * * * * C * T * C *
1 * * * * * * K * *
* * * * * * * *
*****
* * * * * C * *
2 * * * * * * *
* * * * * * *
*****
* * * * * * *
3 * * * * * * *
* * * * * * *
*****
* * * * * * *
4 * * * * * * *
* * * * * * *
*****
* * C * * * V * * *
5 * * * * * * *
* * * * * * *
*****
* C * T * C * * * *
6 * * K * * * * *
* * * * * * *
*****
* * C * * * V * * V *
7 * * * * * * *
* * * * * * *
*****2

Player 1's Turn
Cash : 100G | Income : 0G | Upkeep : 5G
Unit : KING (1,6) | Health : 500 | MELEE | Movement Point : 1 | Can Attack : NO
Your input : MOVE

```

Gambar 1.2 Peta dan Command Move

```

Please enter cell's coordinate x y: 1 5

```

Gambar 1.3 Masukan Koordinat

```

Player 1's Turn
Cash : 100G | Income : 0G | Upkeep : 5G
Unit : KING (1,5) | Health : 500 | MELEE | Movement Point : 0 | Can Attack : NO
Your input : 

```

Gambar 1.4 Tampilan Info Unit King dan Movement Point

5.3 Undo

Fitur : Undo

Hasil yang diberikan saat menjalankan command Undo adalah program mengembalikan unit tersebut ke state sebelumnya, dimana command ini hanya berlaku jika command sebelumnya adalah move, map, dan info.

```
0 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
1 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
2 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
3 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
4 * * * * * * * * *
  * * * * * * * * *
  *****
  * * C * * * * V * * *
5 * * K * * * * * * *
  * * * * * * * * *
  *****
  * C * T * C * * * * *
6 * * * * * * * * *
  * * * * * * * * *
  *****
  * * C * * * * V * * V *
7 * * * * * * * * *
  * * * * * * * * *
  *****2

Player 2's Turn
Cash : 95G | Income : 0G | Upkeep : 5G
Unit : KING (6,2) | Health : 500 | MELEE | Movement Point : 0 | Can Attack : YES
Your input : UNDO
```

Gambar 2.1 Masukan Command Undo

```
*****
0 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
1 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
2 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
3 * * * * * * * * *
  * * * * * * * * *
  *****
  * * * * * * * * *
4 * * * * * * * * *
  * * * * * * * * *
  *****
  * * C * * * * V * * *
5 * * K * * * * * * *
  * * * * * * * * *
  *****
  * C * T * C * * * * *
6 * * * * * * * * *
  * * * * * * * * *
  *****
  * * C * * * * V * * V *
7 * * * * * * * * *
  * * * * * * * * *
  *****2

Player 2's Turn
Cash : 95G | Income : 0G | Upkeep : 5G
Unit : KING (6,1) | Health : 500 | MELEE | Movement Point : 1 | Can Attack : YES
Your input :
```

Gambar 2.2 Hasil Command Undo

5.4 Recruit New Unit

Fitur : Recruit

Hasil yang diberikan saat menjalankan command Recruit adalah king mampu menambah unit bila terdapat bagian kastil yang kosong. Selain itu, Unit hanya bisa di Recruit saat King berada pada tower.

```
Player 2's Turn
Cash : 95G | Income : 0G | Upkeep : 5G
Unit : KING (6,1) | Health : 500 | MELEE | Movement Point : 1 | Can Attack : YES
Your input : RECRUIT

Enter coordinate of empty castle: 6 2
=== List of Recruits ===
1. Archer | Health 200 | Attack Type Ranged | Attack Point 20 | Price 20
2. Swordsman | Health 300 | Attack Type Melee | Attack Point 20 | Price 30
3. White Mage | Health 100 | Attack Type Melee | Attack Point 20 | Price 35
Enter no. of unit you want to recruit: 2
```

Gambar 3.1 Rekrutmen Unit

5.5 Attack

Fitur : Attack

Hasil yang diberikan saat menjalankan program adalah player dapat melakukan attack kepada musuh bila musuh berada pada adjacency point dan bila type serang keduanya sama maka musuh dapat membalas serangan player secara langsung. Setelah attack dilakukan, maka health dari player atau musuh akan berkurang sesuai dengan besarnya attack point.

```

* * * * *
*****
* * * * * C * T * C *
1 * * * * *
* * * * *
*****
* * * * * C * *
2 * * * * *
* * * * *
*****
* V * * V * * * * *
3 * * * * * K * * * *
* * * * *
*****
* * * * * * * V *
4 * * * * * K * * * *
* * * * *
*****
* * C * * * * * V *
5 * * * * *
* * * * *
*****
* C * T * C * * * *
6 * * * * *
* * * * *
*****
* * C * * * * *
7 * * * * *
* * * * *
*****2

Player 2's Turn
Cash : 55G | Income : 0G | Upkeep : 5G
Unit : KING (4,3) | Health : 500 | MELEE | Movement Point : 1 | Can Attack : YES
Your input : ATTACK
1. King (4,4) | Health 500/500 (Retaliates)
Select the enemy you want to attack:

```

Gambar 4.1 Pilihan Attack Unit

```

Player 1's Turn
Cash : 50G | Income : 0G | Upkeep : 5G
Unit : KING (4,4) | Health : 490 | MELEE | Movement Point : 1 | Can Attack : YES
Your input :

```

Gambar 4.2 Hasil Health King Setelah Terkena Attack

5.6 Command InfoCell

Fitur : Infocell
 Hasil yang diberikan saat menjalankan command Infocell pada program adalah program menampilkan info keadaan player pada point tertentu yg ingin diketahui.

```

Player 1's Turn
Cash : 70G | Income : 0G | Upkeep : 5G
Unit : KING (2,4) | Health : 500 | MELEE | Movement Point : 1 | Can Attack : YES
Your input : INFO

Enter the coordinate of the cell: 5 3

== Unit Info ==
King
Owned by Player 2
Health 500/500 | ATK 10

Player 1's Turn
Cash : 70G | Income : 0G | Upkeep : 5G
Unit : KING (2,4) | Health : 500 | MELEE | Movement Point : 1 | Can Attack : YES
Your input : 

```

Gambar 5.1 Pemanggilan Fungsi InfoCell

5.7 1 Unit Player Berada di Sebelah White Mage

Fitur : UnitInfo

Hasil yang diberikan saat menjalankan program adalah unit player yang berada pada adjacency point dengan white mage maka helathnya akan menambah sebanyak 5. Program mengecek apakah terdapat unit-unit yang berada di adjacency point dengan white mage. Pada contoh dibawah, attack dari swordsman adalah 20, artinya health dari King dari 490 menjadi 470. Setelah end turn, Health dari King menjadi 475 karena mendapat tambahan 5 poin dari White Mage. Posisi map bisa dilihat pada peta dibawahnya.

```

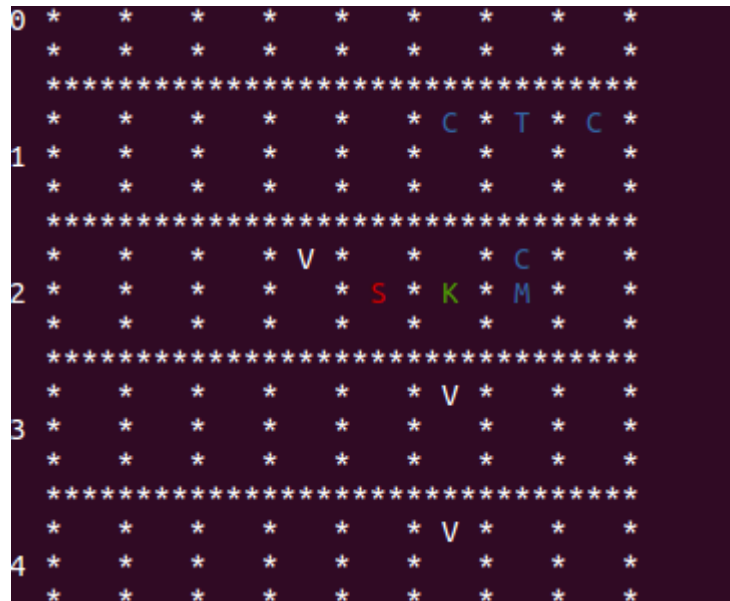
Player 1's Turn
Cash : 10G | Income : 0G | Upkeep : 15G
Unit : SWORDSMAN (4,2) | Health : 280 | MELEE | Movement Point : 3 | Can Attack : YES
Your input : ATTACK
1. King (5,2) | Health 490/500 (Retaliates)

Select the enemy you want to attack:

Player 2's Turn
Cash : 10G | Income : 0G | Upkeep : 10G
Unit : KING (5,2) | Health : 475 | MELEE | Movement Point : 1 | Can Attack : YES
Your input : 

```

Gambar 6.1 dan 6.2 Tampilan saat King telah di Heal oleh White Mage

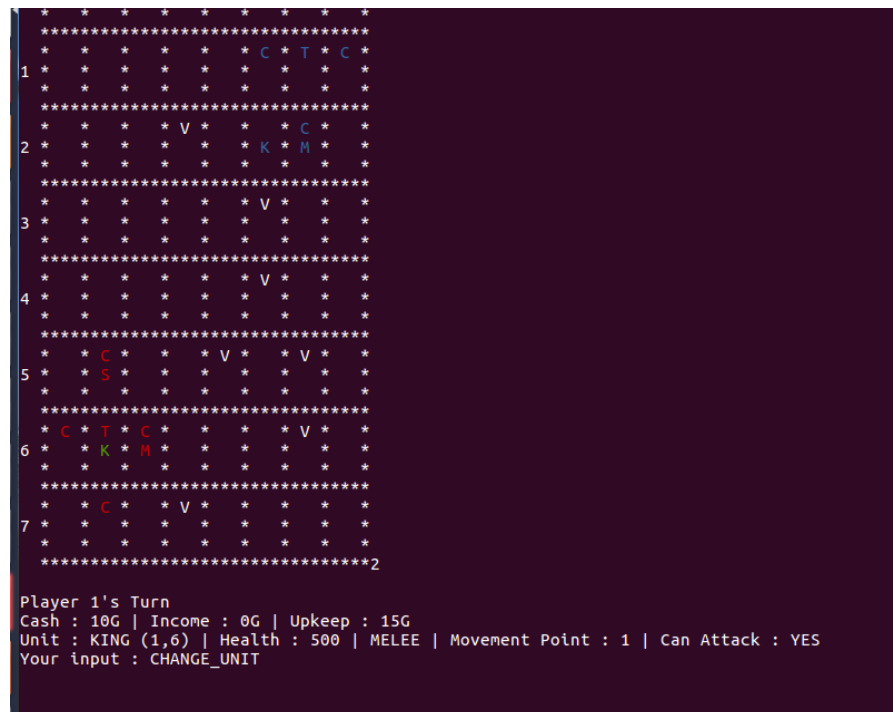


Gambar 6.2 Tampilan Map dengan King dikelilingi White Mage dan Swordsman Lawan

5.8 Memilih Unit yang akan digunakan oleh player

Fitur : ChangeUnit

Hasil yang diberikan saat menjalankan program adalah program menampilkan data unit-unit yang ada secara rinci pada player dan juga player dapat memilih unit yang akan digunakan oleh player



Gambar 7.1 Tampilan Command Change Unit

```

4 * * * * *
  * * * * *
  *****
  * * C * * V * V *
5 * * S * * * * *
  * * * * *
  *****
  * C * T * C * * V *
6 * * K * H * * * *
  * * * * *
  *****
  * * C * * V * *
7 * * * * *
  * * * * *
  *****2

Player 1's Turn
Cash : 10G | Income : 0G | Upkeep : 15G
Unit : SWORDSMAN (1,5) | Health : 300 | MELEE | Movement Point : 3 | Can Attack : YES
Your input : 

```

Gambar 7.2 Tampilan Command Change Unit

6 Test Script

Tabel 1.1 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1.	Move – 0 movement point	Mengecek apakah jika movement point suatu unit = 0, unit tidak bisa berpindah tempat	<ol style="list-style-type: none"> Pastikan movement point = 0 Panggil fitur <i>move</i> 	Movement point	Keluar pesan kesalahan “ <i>The unit doesn't have any move point for this turn.</i> ”	Sesuai dengan yang diharapkan
2.	Move - Healing Village	Mengecek apakah jika suatu unit berada di village yang dimiliki, <i>health</i> unit tersebut tidak melebihi batas <i>maximal health</i>	<ol style="list-style-type: none"> Move suatu unit ke <i>village</i> yang dimiliki Perhatikan <i>health</i> unit tersebut 	Movement point	Health unit menjadi: <ol style="list-style-type: none"> King = 500 Archer = 200 Swordsman = 300 White Mage = 100 	Sesuai dengan yang diharapkan
3.	Move - Akuisisi village	Mengecek apakah jika suatu unit bergerak ke <i>village</i> yang kosong, maka <i>village</i> diakuisisi oleh player tersebut dan tidak bisa melakukan fitur move & undo	<ol style="list-style-type: none"> <i>Move</i> unit ke <i>village</i> yang kosong print map dan perhatikan warna dari <i>village</i> tersebut lakukan fitur undo lakukan fitur move 	Movement point ...	<ol style="list-style-type: none"> Warna <i>village</i> tersebut sewarna dengan player mengakuisisinya Saat melakukan fitur <i>undo</i>, unit tidak berpindah tempat ke tempat sebelumnya Saat melakukan fitur <i>move</i>, muncul pesan kesalahan tidak punya <i>movement point</i> 	Sesuai dengan yang diharapkan

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
4.	Recruit – No move no attack	Mengecek apakah jika suatu unit baru saja di <i>recruit</i> tidak bisa melakukan fitur <i>move</i> maupun <i>attack</i>	<ol style="list-style-type: none"> 1. <i>Change unit</i> ke <i>King</i> 2. Pilih fitur <i>recruit</i> dan <i>recruit</i> suatu unit 3. <i>Change unit</i> ke unit yang baru saja di <i>recruit</i> 4. Lakukan fitur <i>move</i> dan <i>attack</i> 	Recruit new unit	<ol style="list-style-type: none"> 1. Saat melakukan fitur <i>move</i>, akan keluar pesan kesalahan bahwa <i>movement point</i> tidak ada 2. Tidak bisa melakukan <i>attack</i> 	Sesuai dengan yang diharapkan
5.	Recruit – only King in Tower	Mengecek apakah jika unit yang sedang digunakan bukan King, maka unit tersebut tidak bisa melakukan fitur <i>recruit</i> . dan mengecek apakah jika King tidak berada di Tower, King tidak bisa merekrut unit baru	<ol style="list-style-type: none"> 1. <i>Change unit</i> ke unit selain King 2. Lakukan fitur <i>recruit</i> <p>Untuk kasus ke dua,</p> <ol style="list-style-type: none"> 1. <i>Change unit</i> ke King 2. Pastikan King tidak berada di Tower 3. Lakukan fitur <i>recruit</i> 	Recruit new unit	Keluar pesan kesalahan “ <i>Only king can recruit a new unit</i> ” atau keluar pesan kesalahan “ <i>You’re not in your Tower!</i> ”	Sesuai dengan yang diharapkan
6.	Undo	Mengecek apakah jika setelah melakukan <i>move</i> ke beberapa tempat, lalu memilih fitur <i>attack</i> / <i>change unit</i> / <i>recruit</i> unit yang sebelumnya sudah tidak bisa melakukan <i>undo</i> setelahnya	<ol style="list-style-type: none"> 1. <i>Move</i> beberapa tempat 2. Pilih salah satu fitur <i>attack</i> / <i>change unit</i> / <i>recruit</i> 3. Gunakan unit yang sama dengan sebelum melakukan fitur lalu lakukan fitur <i>undo</i> 	Undo	Keluar pesan kesalahan “ <i>No moves</i> ”	Sesuai dengan yang diharapkan

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
7.	Dead King	Mengecek apakah jika King dari suatu Player mati, maka Player lainnya memenangkan permainan		Attack	...	Sesuai dengan yang diharapkan
	Attack – only once	Mengecek apakah jika suatu unit sudah melakukan <i>attack</i> terhadap musuh, unit tersebut idak dapat melakukannya lagi di <i>turn</i> yang sama	<ol style="list-style-type: none"> 1. Pilih salah satu unit musuh yang akan diserang 2. Serang musuh setelah menentukan musuh yg akan diserang 3. Lakukan attack kembali 	Attack	Attack yang kedua tidak dpaat dilakukan dikarrenakan movement point player telah menajdi 0.	Sesuai dengan yang diharapkan
	Attack – ketika ada musuh lebih dari 1 di adjacency point	Mengecek apakah jika ada banyak unit musuh di adjacency point player 1 dapat memilih untuk menyerang salah satu unit tersebut	<ol style="list-style-type: none"> 1. Pilih salah satu unit musuh yang akan diserang dan berada pada 2. Serang musuh yang telah ditentukan 	Attack	Attack dapat dilakukan bila memang unit musuh adjacent dengan unit kita.	Sesuai dengan yang diharapkan
	White mage	Mengecek kondisi health unit yang id playernya sama dengan white mage dan juga adjacent dengan white mage	<ol style="list-style-type: none"> 1. Mengecek apakah ada unit yang adjacent dengan white mage apa tidak. 2. Jika iya, apakah health unit tersebut setiap turnnya bertambah apa tidak 	White mage		Sesuai dengan yang diharapkan

7 Pembagian Kerja dalam Kelompok

Tabel 1.2 Pembagian Kerja

NAMA	PEMBAGIAN KERJA
Adylan Roaffa	Membuat “building.h”, “buildinglist.h”, “unit.h”, “unitlist.h”, “player.h” “printfmt.h”, dan “playerQueue.h” beserta implementasinya, dan membuat main program.
Albert Sahala	Membuat “gameplay.h” dan implementasinya, fitur PrintBbyID, PrintUbyID, dan InfoCell pada “command_list.h”, dan sebagian laporan.
Haifa Fadhila	Membuat “map.h” dan implementasinya, fitur untuk unit White Mage, fitur attack dan PrintBbyID yang ada pada “command_list.h”, dan sebagian laporan.
M. Azka Widyanto	Membuat fitur attack dan infocell yang ada pada “command_list.h” dan sebagian laporan.
Nadija Herdwina	Membuat “movestack.h” dan “mesinlogo.h” beserta implementasinya, fitur Move, ChangeUnit, Recruit, dan Undo pada “command_list.h”, serta sebagian laporan.

8 Lampiran

8.1 Deskripsi Tugas Besar 2

Battle for Olympia adalah permainan dengan genre *turn-based strategy* yang dimainkan dengan cara memasukkan perintah melalui *command line interface*. Tujuan utama dari permainan ini adalah menghancurkan kerajaan musuh dengan cara membunuh raja yang memimpin seluruh unit musuh. Seorang raja dapat merekrut beberapa pasukan, dan pasukan dapat dikerahkan untuk membunuh pasukan dan raja kerajaan musuh. Apabila raja mati, maka pemain selesai bermain dan seluruh pasukan yang tersisa ikut mati.

8.2 Notulen Rapat

- Kamis, 2 November 2017 : Pemahaman dan diskusi bersama spesifikasi tugas besar antar anggota kelompok.
- Selasa, 7 November 2017 : Pembuatan header dan pembagian tugas pada setiap anggota kelompok. Hasil dari rapat pembagian tugas pada setiap anggota adalah sebagai berikut:
- Modul Building, BuildingList, Unit, UnitList, Player, Main Program = Ody (sudah dibuat dari awal)
 - Modul Map keseluruhan = Haifa
 - Modul Gameplay keseluruhan = Albert
 - Modul CommandList = Azka dan Nadija
Nadija → dari fungsi countMove sampai Undo (berurutan)
Azka → dari fungsi Attack sampai EndTurn (berurutan)
- Rabu, 8 November 2017 : Asistensi 1, spek tambahan: pas recruit, tidak bisa langsung menyerang. Player diubah spesifikasinya jadi 2. Turn tidak harus pake queue, tetapi pastikan semua ADT terpakai (boleh ganti-ganti type). Di Map menggunakan type building dan unit.
- Senin, 13 November 2017 : Rapat progress pengisian Modul.
- Catatan umum: Tiap-tiap modul pisahkan file ekstensi .h dan .c nya (.h hanya untuk header dan .c implementasi header fungsi tersebut)
- Modul gameplay = 20% (Albert)
Catatan: Hanya perlu dilengkapi.
 - Modul Map = 50% (Haifa)
Catatan: di UpdateMap harus check tiap list building dan unit, disini handling penggunaan pcolor.c juga di Modul Map. Ada perubahan beberapa parameter di modul Building dan modul Unit untuk kebutuhan inialisasi dan fungsi lain pada Map. (Lengkapi printMap dan updateMap)
 - Modul CommandList Azka = 50%

Catatan: Penggunaan `playerQueue.c` pada fungsi `endTurn` masih belum tepat, perlu diperbaiki, pada fungsi `attack` pengecekan musuh sekitar masih belum tepat.

- Modul `CommandList` Nadija = 70%

Catatan: Secara keseluruhan implementasi pada fitur `Move`, `Undo`, `Recruit`, dan `ChangeUnit` sudah selesai dibuat, namun masih ada beberapa yang terlewat seperti kepemilikan *village* saat unit *move* ke koordinat yang terdapat *village*. Dan parameter fitur tersebut masih bertipe `Unit` belum `Player`.

- Modul `Building` = fungsi `createVillage`, `createCastle` dan sejenisnya diganti parameternya jadi `playerID` dan `location` untuk menyesuaikan saat dipanggil pada fungsi `InitMap` (didalam modul `Map`).

Kamis, 16 November 2017 : Asistensi 2. Catatan: Untuk command undo movement, yang di undo adalah movement terakhir (jadi kalau sudah change unit, stacknya kembali kosong dan tidak bisa di undo setelah itu). ADT Jam hanya digunakan untuk save and load (jadi kalau tidak bikin bonus, ADT Jam tidak digunakan).

Jumat, 17 November 2017 : Pengerjaan semua modul bersama-sama dari pukul 16.00 Jumat s.d. 00.00 Sabtu. Semua modul dilengkapi dan yang masih kurang tepat diperbaiki. Pengerjaan laporan hingga 20%.

- Modul `UnitList` = terdapat kesalahan jenis penggunaan variasi list yang digunakan (harusnya list sirkuler tapi handling elemen listnya menggunakan skema list linier), sempat berpengaruh pada `updateMap`, sudah diganti.
- Modul `Map` = Dilengkapi hingga 95%, masih kurang printing perbedaan warna untuk current unit. Terdapat tambahan fungsi untuk men-generate angka random untuk menentukan jumlah dan letak *village-village* baru pada peta, tambahan membuat type data baru di map yaitu `villageList` yang berisi *village* baru tersebut.
- Pengerjaan laporan masih 20% karena masih belum bisa mulai test case dan sebagainya.
- Modul `gamePlay` = terdapat perubahan-perubahan pada modul ini, sudah selesai.
- Modul `CommandList` (bag 1) = Setiap fungsi program sudah dilengkapi semua (sudah ter compile), tetapi masih ada beberapa fungsi yang jalannya belum tepat (yang sudah tepat ada 65%)
- Modul `CommandList` (bag 2) = Beberapa fungsi masih belum tepat & ada yang belum lengkap. Sehingga dibetulkan hingga 50%.

Minggu, 19 November 2017 : Pengerjaan semua modul bersama-sama kembali, dari pukul 14.30 Minggu hingga 07.30 Senin. Semua modul dilengkapi dan

yang masih kurang diperbaiki. Progress laporan telah mencapai 50%.

- Modul `gamePlay` = tidak terdapat perubahan pada modul ini, sudah selesai
- Modul `command list` (bag 1) = setiap fungsi telah tercompile dan sudah diperbaiki fungsi yang masih jalannya belum tepat, dan sedang dalam tahap testing case sehingga modul ini tidak mengalami
- Modul `command list` (bag 2) = setiap fungsi telah tercompile, dan beberapa fungsi telah diperbaiki sehingga pengerjaannya sedang dalam tahap testing case,
- Modul `Map` = melengkapi bagian printing pewarnaan pada current unit
- Modul `unitList` = menyamakan penggunaan variasi list pada seluruh fungsi
- Pengerjaan laporan telah mencapai tahap 50%

Kamis, 25 November 2017 :

Asistensi 3. Catatan: Penggunaan mesin kata untuk penulisan pembukaan "Battle of Olympia" maupun *command*. Apa bila baru merecruit suatu unit tidak bisa melakukan *attack*, kalo unit mati tidak bisa *retaliate* lagi, dan saat *move* tidak boleh menabrak musuh.

8.3 Log Activity Anggota Kelompok

Adylan Roaffa (13516016) :

- Jumat, 10 November 2017 ADT `unit.h`, `unitlist.h`, `commandlist.h`, `gameplay.h`, `building.h`, `buildinglist.h`, `map.h` dan kemudian diberikan kepada kelompok untuk dilengkapi
- Rabu, 16 November 2017 penyatuan main program dengan adt lainnya
- Jumat, 17 November 2017 debugging
- Minggu, 19 November 2017 selesai debugging dan fitur sudah bisa digunakan
- Jumat, 24 November 2017 menggabungkan dan memperbaiki beberapa bug kecil

Albert Sahala (13516022) :

- Sabtu, 11 November 2017 melanjutkan "`gameplay.h`" dan mulai mengerjakan fungsi dan prosedur di dalam `gameplay.c`
- Minggu – Selasa, 12-14 November 2017 menyelesaikan pembuatan source code dari `gameplay.c`
- Jumat, 18 November 2017 melakukan debugging dari program dan mengerjakan laporan bagian awal
- Minggu, 19 November 2017 mengerjakan fungsi `InfoCell`, `PrintBbyID`, dan `PrintUbyID` dalam `command_list.c`
- Rabu, 22 November 2017 melanjutkan pengerjaan laporan
- Minggu, 26 November 2017 melanjutkan pengerjaan laporan

Haifa Fadhila Ilma (13516076) :

- Minggu, 12 November 2017 : Membuat implementasi fungsi `getUnit()`, `getBuilding()`, `createMap()`, dan `initMap()` pada Modul Map.
- Selasa, 14 November 2017 : Membuat prosedur `printMap()`.
- Jumat, 17 November 2017 : Membuat prosedur `updateMap()`, melengkapi `initMap()` dengan inisiasi `village random`, melengkapi `printMap()`, debugging, modul Map sudah dapat digunakan.
- Minggu, 19 November 2017 : Membuat fungsi `attack()` dan implementasinya (sudah dapat digunakan), membantu mengarahkan pembuatan fungsi `printByID()`, debugging modul Map, membuat sebagian laporan.
- Senin, 20 November 2017 : Debugging fungsi `attack()`, membuat fitur bonus White Mage (sudah dapat digunakan).
- Jumat, 24 November 2017 : Debugging (minor bugs) fungsi `attack()` dan white mage, testing program.
- Sabtu, 25 November 2017 : Testing beberapa fitur pada program dan debugging.
- Minggu&Senin, 26&27 November 2017 : Melengkapi laporan.

M. Azka Widyanto (13516127) :

- Minggu 12 November 2017 memulai mengerjakan fitur attack dan Infocell pada “`command_list.h`”
- Senin – Kamis 13-16 November 2017 menyelesaikan fitur attack dan info cell, lalu membetulkan bug yang masih ada pada program
- Minggu 19 November 2017 debugging dan fitur pada program sudah dapat digunakan lalu mengerjakan sebagian laporan
- Senin 20 November 2017 melanjutkan pengerjaan laporan
- Minggu 26 November 2017 mengerjakan laporan
- Senin 27 November 2017 melengkapi laporan

Nadija Herdwina (13516130) :

- Sabtu, 11 November 2017 membuat “`movestack.h`” dan mulai mengerjakan fitur recruit, move, change unit dan undo pada “`command_list.h`”
- Minggu – Selasa, 12-14 November 2017 menyelesaikan pembuatan source code fitur
- Jumat, 18 November 2017 debugging
- Minggu, 19 November 2017 selesai debugging dan fitur sudah bisa digunakan
- Rabu, 22 November 2017 mengerjakan beberapa subbab pada laporan
- Jumat, 24 November 2017 membuat “`mesinlogo.h`” dan implementasinya serta melengkapi beberapa spek pada fitur move
- Sabtu, 25 November 2017 mengerjakan beberapa subbab pada laporan


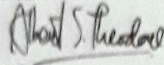



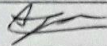
8.4 Lembar Asistensi

**Form Asistensi Tugas Besar
IF2110/Algoritma dan Struktur Data
Sem. 1 2017/2018**

No. Kelompok/Kelas :
Nama Kelompok :
Anggota Kelompok (Nama/NIM) :
1. Adylan Roaffa Ilmy
2. Albert Sanala Theodore
3. Haifa Fadwila Ilma
4. Nadiya Herdwina P.S.
5. M. Azka Widyanto
6.



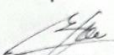

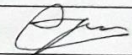
Asisten Pembimbing : Muhammad Gumilang

Asistensi I

Tanggal: 8 November 2019	Catatan Asistensi:
Tempat: Lab. IIR	
Kehadiran Anggota Kelompok:	Spk tambahan: far reant gabisa nyerang juga.
No	
NIM	
Tanda tangan	
1	
13516016	→ di Map typenya building sama
	Ura, udah benar / belum? Udah.
2	
13516022	→ Turn harus pake queue? terserah
	aga.
3	→ Tapi wajib pake ADT → bda
13516096	ganti-ganti tipe
	
4	→ Player jadi 2 aga (gansak 2-4)
13516130	
	
5	
13516127	
	
6	
	Tanda Tangan Asisten: 

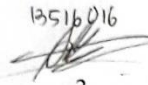




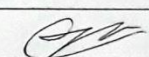
Notulen : Haifa Fadhila Ilma

Asistensi II

Tanggal: Kamis, 16 November 2017 Tempat: Lab IRK		Catatan Asistensi: ▶ "undo" → misal ul' movement, yg terjadi yg di undo adalah movement terakhir. ↳ kalo Change unit, stacknya kosong lagi, gabisa undo abis itu. ▶ ADT sam cuma buat save & load
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 13516016  2 13516022 Albert S. Theodore 3 13516076  4  13516127 5 6  13516130		
		Tanda Tangan Asisten: 

Notulen : Haifa Fadhila Ilma

Asistensi III

Tanggal : 28 NOV 2017	Catatan Asistensi:
Tempat : Lab IRK	
Kehadiran Anggota Kelompok:	
No	
NIM	
Tanda tangan	
1	
13516016	
	
2	
13516022	
	
3	
13516076	
	
4	
13516127	
	
5	
13516130	
	
6	
	Tanda Tangan Asisten: 

- * Penggunaan mesinkata ?
 - Load tulisan battle of olympia / untuk command.
- * Village → kalo udh ke village stacknya jd empty, movement point = 0
- * Kalo baru recruit = gabisa move (Gabisa attack.
- * Kalo mati ga bisa retaliates lagi.
- * Move : Gaboleh "nabrak" musuh

Notulen : Haifa Fadhila Ilma