

# **LAPORAN TUGAS BESAR**

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


### **AVATAR: WORLD WAR**

Dipersiapkan oleh:

Kelompok 04 K-03

Panawar Hasibuan	13517129
Daffa Pratama Putra	13518033
Dhafin Rayhan Ahmad	13518063
Morgen Sudyanto	13518093
Naufal Dean Anugrah	13518123

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-04-03</i>		<i>20</i>
		<i>Revisi</i>	-	<i>25 November 2019</i>

# Daftar Isi

1 Ringkasan.....	3
2 Penjelasan Tambahan Spesifikasi Tugas .....	3
2.1 Save Game.....	3
2.2 Load Game .....	3
2.3 Skill Attack Up.....	3
2.4 Skill Critical Hit .....	4
2.5 Skill Shield .....	4
3 Struktur Data (ADT) .....	4
3.1 ADT Point .....	4
3.2 ADT Array Dinamis .....	4
3.3 ADT Matriks .....	5
3.4 ADT Mesin Karakter dan Mesin Kata .....	5
3.5 ADT Queue .....	5
3.6 ADT Stack.....	5
3.7 ADT List Linier.....	5
3.8 ADT Graph.....	6
3.9 ADT Bangunan .....	6
3.10 ADT Status .....	6
4 Program Utama .....	6
5 Algoritma-Algoritma Menarik.....	7
5.1 Algoritma Undo.....	7
5.2 Algoritma Interface Permainan .....	8
6 Data Test .....	9
6.1 Attack Tanpa Skill.....	9
6.2 Attack Dengan Skill .....	9
6.3 Save dan Load .....	10
6.4 Mendapatkan Skill.....	10
6.5 Level Up.....	10
6.6 Undo.....	11
6.7 Move.....	11
7 Test Script .....	12
8 Pembagian Kerja dalam Kelompok .....	14
9 Lampiran .....	15
9.1 Deskripsi Tugas Besar 2.....	15
9.2 Notulen Rapat.....	15
9.3 Log Activity Anggota Kelompok.....	15
9.4 Form Asistensi.....	17

# 1 Ringkasan

Tugas besar mata kuliah Algoritma dan Struktur Data ini adalah membuat suatu permainan berbasis bahasa pemrograman C. Secara garis besar, permainan ini adalah permainan turn-based strategy yang mensimulasikan perang dunia pada zaman Avatar. Permainan ini dimainkan oleh 2 orang yang melawan satu sama lain. Segala perintah yang ingin dilakukan pemain, dimasukkan melalui command line interface dengan perintah – perintah yang sudah tersedia. Program yang dibuat memanfaatkan Abstract Data Type atau ADT yang sudah diberikan dari praktikum 1 hingga praktikum 12. Setiap ADT memiliki bagian dan fungsinya masing – masing. Pada tugas besar ini diberikan beberapa kasus bonus. Kami mengimplementasikan semua fitur bonus yang diberikan pada spesifikasi tugas besar. Meskipun permainan ini hanya dijalankan melalui terminal pada komputer, tetapi dalam proses pembuatannya, permainan ini termasuk cukup kompleks, karena harus mengkombinasikan seluruh ADT yang ada.

## 2 Penjelasan Tambahan Spesifikasi Tugas

Pada spesifikasi tugas besar, diberikan fitur - fitur bonus. Berikut adalah fitur bonus yang kami implementasikan.

### 2.1 Save Game

Fitur save game yang kami buat akan menyimpan status - status yang penting dari permainan. Status penting tersebut antara lain adalah ukuran peta (NB x NK), jumlah dan atribut bangunan, matriks keterhubungan, status bangunan (kepemilikan, level, pasukan, dan kondisi serangan bangunan), queue skill pemain 1 dan pemain 2, pemain yang aktif saat itu, dan jumlah turn yang sudah dimainkan. Penyimpanan ini ditulis dalam suatu file dengan direktori penyimpanan bisa diatur sendiri oleh pemain. Implementasi dari fitur save game dapat dilihat pada file console.c dan status.c. Catatan: Save game tidak akan menyimpan stack undo.

### 2.2 Load Game

Fitur load game yang dibuat akan membaca file save game, yang telah dibuat dengan command SAVE, dari direktori yang diisikan oleh pemain. Program akan membaca isi file tersebut yang berisikan semua status permainan yang terakhir kali sebelum permainan disimpan. Status tersebut antara lain adalah ukuran peta (NB x NK), jumlah dan atribut bangunan, matriks keterhubungan, status bangunan (kepemilikan, level, pasukan, dan kondisi serangan bangunan), queue skill pemain 1 dan pemain 2, pemain yang aktif pada terakhir permainan, dan jumlah turn yang sudah dimainkan sebelumnya. Setelah membaca semua status, status tersebut akan dimasukkan pada tempatnya masing - masing dan pemain bisa langsung dilanjutkan permainannya. Implementasi dari fitur load game dapat dilihat pada file console.c dan status.c.

### 2.3 Skill Attack Up

Fitur bonus yang kami implementasikan selanjutnya adalah skill attack up. Jika skill attack up ini aktif, akan menyebabkan pertahanan dan shield musuh menjadi tidak berpengaruh saat penyerangan. Cara mendapatkan skill attack up adalah ketika pemain baru saja menyerang tower musuh dan jumlah tower pemain menjadi 3. Setiap penyerangan yang dilakukan pemain akan dicek terlebih dahulu apakah skill attack up aktif. Jika skill attack up aktif, pertahanan dan skill shield musuh akan di matikan. Jika tidak aktif, penyerangan akan berkurang sesuai dengan

STEI- ITB	IF2110-TB-04-03	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.		

pertahanan yang ada pada bangunan yang diserang. Implementasi skill attack up dapat dilihat pada console.c bagian fungsi AttackCommand.

## **2.4 Skill Critical Hit**

Fitur bonus skill critical hit mirip seperti skill attack up, yaitu dapat meniadakan sementara pertahanan dan shield musuh, tetapi serangan efektif menjadi 2 kali lipat pasukan yang digunakan. Skill critical hit hanya berlaku pada 1 serangan saja. Pemain akan mendapatkan skill critical hit setelah mengaktifkan skill extra turn. Sama seperti skill attack up, saat pemain menyerang lawan menggunakan skill critical hit, program akan mengecek terlebih dahulu apakah skill aktif atau tidak. Jika skill aktif, penyerangan dilakukan dengan melipatgandakan pasukan yang digunakan untuk menyerang. Jika tidak aktif, penyerangan dilakukan seperti biasa/normal. Implementasi skill critical hit dapat dilihat pada file console.c di bagian fungsi AttackCommand.

## **2.5 Skill Shield**

Fitur skill shield menyebabkan semua bangunan yang dimiliki oleh pemain memiliki pertahanan selama 2 turn musuh. Tetapi jika pemain menggunakan skill shield berturut - turut dalam 2 turn, maka durasi tidak akan bertambah, namun menjadi durasi maksimum. Skill shield didapatkan pemain ketika pemain diserang oleh musuh dan bangunan yang dimilikinya berkurang 1 dan bersisa 2 bangunan. Seperti pada skill lainnya, program perlu mengecek apakah pemain menggunakan skill shield ketika diserang musuh. Implementasi dari skill shield dapat dilihat di console.c.

# **3 Struktur Data (ADT)**

Dalam membuat permainan ini, kami menggunakan ADT yang telah diberikan dari praktikum 1. Setiap ADT memiliki fungsi dan perannya masing – masing. Berikut adalah ADT yang digunakan beserta fungsinya.

## **3.1 ADT Point**

ADT point digunakan untuk merepresentasikan lokasi bangunan yang ada pada peta dengan koordinat (X, Y). Tipe data Point berisikan dua elemen, yakni absis dan ordinat. Absis menunjukkan lokasi bangunan pada sumbu X, sedangkan ordinat menunjukkan lokasi bangunan pada sumbu Y. ADT point dipilih untuk menyelesaikan persoalan mengenai lokasi bangunan pada peta. Implementasi dari ADT point dapat dilihat pada file point.c.

## **3.2 ADT Array Dinamis**

ADT array dinamis digunakan untuk menyelesaikan persoalan mengenai penyimpanan bangunan yang ada dalam permainan. Struktur pada ADT array dinamis diberi nama TabBangunan dengan dua elemen pembentuknya, yakni MaxEl dan TI. MaxEl merepresentasikan jumlah maksimum elemen yang dapat ditampung pada array dan TI sebagai container atau tempat menyimpan elemen bangunan tersebut. Implementasi dari ADT array dinamis dapat dilihat pada file array.c.

### 3.3 ADT Matriks

ADT matriks digunakan sebagai representasi peta permainan. Peta permainan berukuran paling kecil sebesar 10 x 10 dan paling besar berukuran 20 x 30. Struktur matriks terdiri dari tiga komponen, yaitu Mem, NBrEff, dan NKolEff. Komponen Mem digunakan untuk menyimpan isi dari matriks baris ke-i dan kolom ke-j, sementara NBrEff dan NKolEff mererrepresentasikan jumlah baris dan kolom yang terisi pada matriks tersebut. Implementasi dari ADT matriks dapat dilihat pada file matriks.c.

### 3.4 ADT Mesin Karakter dan Mesin Kata

ADT mesin karakter dan mesin kata digunakan untuk membaca masukan dari keyboard oleh pengguna, membaca file eksternal, dan membaca hasil save game. ADT mesin karakter dan mesin kata digunakan untuk mem-parsing masukan dari file eksternal dan keyboard, sehingga bisa dibaca oleh program. Implementasi ADT mesin karakter dapat dilihat di mesinkar.c dan ADT mesin kata bisa dilihat di mesinkata.c.

### 3.5 ADT Queue

ADT queue digunakan sebagai representasi antrian skill yang dimiliki oleh pemain. Antrian skill ini berguna saat pemain ingin menggunakan skill dan mendapatkan skill baru. Antrian ini juga seperti antrian pada umumnya, yakni ketika menggunakan skill, kepala/head dari antrian akan dihapus, dan saat mendapatkan skill, ekor/tail dari antrian dimasukkan oleh skill baru. Struktur queue terdiri dari empat komponen, yaitu T, Head, Tail, dan MaxEl. T digunakan sebagai representasi container atau tempat menyimpan skill. Head dan Tail sebagai representasi kepala dan ekor dari antrian. MaxEl adalah elemen maksimum yang dapat ditampung oleh queue, dalam hal ini adalah 10 elemen maksimum. Implementasi dari ADT queue dapat dilihat di queue.c.

### 3.6 ADT Stack

ADT stack digunakan untuk menyimpan status sementara. Setiap berubahnya status, akan disimpan di bagian atas stack. Ketika command "UNDO" dilakukan, isi stack teratas akan dikeluarkan atau di pop. Selama command "END TURN" belum dimasukkan, isi stack akan terus bertambah. Jumlah elemen maksimum yang dapat disimpan di stack sebanyak 25 elemen. Struktur dari Stack terdiri dari komponen T dan Top. T merepresentasikan stack penyimpanan status yang terdiri dari TabBangunan, TabSkill pemain 1, dan TabSkill pemain 2, sedangkan Top merepresentasikan elemen teratas dari stack. Implementasi ADT stack dapat dilihat di stack.c.

### 3.7 ADT List Linier

ADT list linier digunakan untuk menyimpan indeks bangunan yang dimiliki oleh tiap pemain. ADT ini diperlukan karena penomoran indeks bangunan dilakukan tidak berdasarkan kepemilikan bangunan dari tiap pemain, sehingga perlu ADT list untuk menyimpan indeks kepemilikannya. Struktur tipe bentukan List terdiri dari komponen First. First menunjuk ke alamat pertama dari list. Selain itu, elemen list disimpan dalam array ElmtList yang terdiri dari Info dan Next. Info adalah isi elemen dari list dan Next menunjuk ke alamat elemen setelohnya. Implementasi dari ADT list linier dapat dilihat di file listlinier.c.

### 3.8 ADT Graph

ADT graph digunakan untuk menunjukkan keterhubungan antar bangunan yang ada pada peta. ADT ini diperlukan untuk melihat bangunan tersebut terhubung oleh bangunan apa saja di sekitarnya. Command seperti ATTACK dan MOVE sangat bergantung pada ADT ini, sebab untuk kedua command ini, perlu melihat ada bangunan apa saja yang terhubung di sekitarnya. Tidak bisa menggunakan command ATTACK dan MOVE untuk bangunan yang berada tidak terhubung dengan bangunan asal. Untuk implementasi dari ADT graph, dapat dilihat pada graph.c.

### 3.9 ADT Bangunan

ADT bangunan digunakan untuk merepresentasikan atribut – atribut yang terdapat pada tiap bangunan. Struktur data bentukan ADT bangunan adalah atribut bangunan itu sendiri, yang terdiri dari ID, Tipe, Lokasi, Kepemilikan, N, Level, A, M ,P ,U, S. ID bangunan adalah indeks dari bangunan, Tipe menunjukkan jenis bangunan (Castle, Tower, Fort, Village), Lokasi menunjukkan letak bangunan tersebut di peta dalam bentuk koordinat (X,Y), Kepemilikan menunjukkan bangunan tersebut dimiliki oleh pemain 1, pemain 2, atau tidak dimiliki siapapun, N merepresentasikan jumlah pasukan yang terdapat pada bangunan, Level menunjukkan level dari bangunan tersebut yaitu dari level 1-4, A adalah nilai penambahan pasukan pada setiap turn, M adalah maksimum penambahan pasukan, P adalah pertahanan yang dimiliki oleh bangunan tersebut, U adalah pasukan awal dari bangunan, dan S adalah nilai boolean yang menunjukkan apakah bangunan sudah menyerang atau belum. Implementasi ADT bangunan dapat dilihat di bangunan.c.

### 3.10 ADT Status

ADT status digunakan untuk menyimpan status yang berubah selama permainan. Struktur tipe data Status terdiri dari beberapa komponen, termasuk penggunaan ADT yang lain di dalamnya. Komponen tersebut antara lain NBPeta, NKPeta, NBangunan, T (TabBangunan), Q1 dan Q2 (Queue), Peta (Matriks), Adjacency (Graph), StatusPemain (Stack), S1 dan S2 (TabSkill), Active Player, dan Turn. NBPeta dan NKPeta merepresentasikan jumlah baris dan kolom pada peta. NBangunan menunjukkan jumlah bangunan yang tersedia pada permainan. T adalah TabBangunan yang digunakan untuk menyimpan data/atribut dari bangunan. Q1 dan Q2 adalah antrian skill yang tersedia di setiap pemain. Angka 1 dan 2 merepresentasikan pemain 1 dan pemain 2. Peta menggunakan ADT matriks untuk merepresentasikan bentuk peta. Adjacency menggunakan ADT graph untuk menunjukkan keterhubungan antar setiap bangunan di peta. StatusPemain menggunakan ADT stack untuk menyelesaikan persoalan undo dengan menyimpan status pemain pada stack. S1 dan S2 adalah TabSkill, yakni digunakan untuk menunjukkan skill yang aktif pada saat itu. ActivePlayer menunjukkan pemain yang aktif pada turn tersebut, yaitu pemain 1 atau pemain 2. Turn merepresentasikan urutan turn yang pada saat permainan berlangsung. Implementasi ADT status dapat dilihat di status.c.

## 4 Program Utama

Program utama yang kami buat dapat dilihat pada file main.c dan console.c. Kami meringkas program utama dalam fungsi AppExecute pada console.c. Isi dari program utama ini, yang pertama adalah mencetak teks inisiasi game status pada saat memulai permainan. Teks ini digunakan untuk memberikan pesan kepada pemain bahwa permainan telah berhasil di inisiasi dan siap untuk

dimainkan. Pesan error akan ditampilkan ketika setup konfigurasi permainan tidak berhasil. Selanjutnya, isi dari program utama ini adalah berisi interface permainan, seperti mencetak turn yang dimainkan saat itu, mencetak jumlah pasukan dan bangunan yang terdapat pada permainan saat itu dengan format (Pemain 1 – Netral – Pemain 2), giliran pemain, skill yang sedang aktif saat itu, peta, dan masukan command.

Tampilan peta dibedakan menjadi 4 warna dengan biru sebagai border peta, kuning adalah bangunan yang netral, hijau adalah bangunan milik pemain 1 (Radiant), dan merah adalah bangunan milik pemain 2 (Dire). Kami memilih nama Radiant dan Dire untuk pemain 1 dan pemain 2 karena terinspirasi oleh permainan DoTA 2. Pada masukan command, kami membedakan warna untuk pemain 1 dan 2. Ketika giliran pemain 1, tulisan "Your Command" akan menjadi warna hijau, begitu juga untuk giliran pemain 2. Setiap turn yang berlangsung, program akan mengecek apakah permainan telah berakhir atau belum. Ketika permainan berakhir, yakni salah satu pemain sudah tidak memiliki bangunan di peta, akan dimunculkan pesan "Radiant / Dire Victory".



Gambar 1. Tampilan awal permainan

## 5 Algoritma-Algoritma Menarik

Pada pengerjaan tugas besar ini, ada beberapa algoritma menarik. Algoritma menarik tersebut adalah sebagai berikut.

### 5.1 Algoritma Undo

Algoritma undo kami sebut sebagai algoritma yang menarik. Pada dasarnya, command undo ini hanya memanfaatkan ADT Stack yang telah diberikan dengan sedikit penyesuaian terhadap komponen dalam permainan. Dengan menerapkan ADT Stack, ketika pemain melakukan command, maka pada stack akan di push kondisi setelah command tersebut dilakukan. Pada saat

pemain melakukan undo, elemen teratas stack, yakni kondisi terakhir, akan dipop atau dikeluarkan dari stack. Algoritma ini menarik karena penyimpanan elemen pada stack bukan hanya kondisi bangunan saja, tetapi hampir seluruh komponen yang berubah dalam game, seperti kondisi bangunan, tab skill, dan queue skill. Terlebih lagi, yang disimpan bukan hanya kondisi pemain yang melakukan undo saja, tetapi kondisi lawan juga harus disimpan, sehingga menyebabkan elemen stack menjadi sangat kompleks. Pada awalnya kami juga mendapati bug pada command undo ini, yaitu bug pada queue skill. Namun bug tersebut bisa ditangani dengan baik dan command undo bisa berjalan sebagaimana seharusnya.

## 5.2 Algoritma Interface Permainan

Interface permainan menjadi salah satu hal yang penting dalam membuat game, sebab interface juga mempengaruhi daya minat dari pemain. Algoritma ini terbilang cukup menarik, sebab kami menggunakan warna – warna dan penataan layar permainan. Warna yang kami gunakan dalam permainan ini adalah warna biru untuk border peta, ungu untuk border antar turn, hijau untuk pemain 1 (Radiant), merah untuk pemain 2 (Dire), dan kuning untuk bangunan netral. Kami menggunakan warna tersebut untuk membedakan antar komponen dalam permainan, sehingga tampilan menjadi lebih jelas. Selain itu, ketika pemain 1 yang berwarna hijau melakukan end turn, warna dari perintah masukan dan border "Next Move" menjadi berwarna merah yang menandakan giliran pemain 2, begitu juga sebaliknya. Kami juga membuat penataan layar menjadi agak ke tengah bagian, agar pemain bisa lebih fokus dan jelas saat memainkannya. Untuk penataan, kami mengira-ngira ukuran yang pas untuk border agar pas berada di tengah. Untuk mengakalnya, diberikan spasi sebanyak beberapa kali. Jumlah spasi yang diberikan dapat dihitung dengan suatu rumus tertentu.



Gambar 2 Tampilan permainan saat berubah giliran



## 6 Data Test

Untuk menguji keberjalanan program, kami menguji dengan beberapa data test. Data test tersebut adalah sebagai berikut :

### 6.1 Attack Tanpa Skill

Pengujian ini bertujuan untuk mengecek apakah command ATTACK dapat berjalan dengan baik. Attack yang dilakukan tanpa menggunakan skill apapun atau normal attack. Mekanisme attack sesuai dengan spek yang diberikan, yaitu untuk normal attack, jika pemain 1 menyerang dengan X pasukan, terhadap bangunan dengan Y pasukan, maka

- Jika  $X \geq Y$ , bangunan menjadi milik pemain 1 dengan jumlah pasukan di bangunan baru sebanyak  $(X-Y)$  pasukan.
- Jika  $X < Y$ , bangunan tidak akan berpindah kepemilikan, namun jumlah pasukan di bangunan yang diserang akan menjadi sebanyak  $(Y-X)$  pasukan.
- Jika bangunan yang diserang memiliki pertahanan, jumlah pasukan yang menyerang akan berkurang menjadi  $\frac{3}{4}$  nya.

Kami mencoba dengan 3 kasus seperti di atas, yaitu :

- Castle level 1 menyerang dengan 40 pasukan terhadap Tower level 1 dengan 30 pasukan.
  - o Hasil yang seharusnya : Tower berpindah kepemilikan dengan 0 pasukan.
- Castle level 1 menyerang dengan 10 pasukan terhadap Castle level 1 dengan 40 pasukan.
  - o Hasil yang seharusnya : Castle yang diserang tidak berpindah kepemilikan dengan pasukan menjadi 30
- Castle level 1 menyerang dengan 40 pasukan terhadap Castle level 1 dengan 40 pasukan.
  - o Hasil yang seharusnya : Castle berpindah kepemilikan dengan 0 pasukan.

### 6.2 Attack Dengan Skill

Pengujian ini digunakan untuk mengecek apakah command ATTACK dengan skill dapat berjalan dengan baik. Attack dilakukan dengan beberapa skill yang menyebabkan attack modifier. Skill tersebut antara lain adalah Shield, Attack Up, Critical Hit.

- Skill Shield menyebabkan seluruh bangunan mempunyai pertahanan selama 2 turn.
- Skill Attack Up akan menonaktifkan shield maupun pertahanan lawan.
- Skill Critical Hit menyebabkan serangan menjadi 2x pasukan efektif.

Kami mencoba dengan beberapa kasus, yaitu :

- Tower level 1 menyerang dengan 20 pasukan terhadap Castle level 2 dengan 60 pasukan ditambah Shield.
  - o Hasil yang seharusnya : Castle tidak berpindah kepemilikan dengan pasukan menjadi 45
- Tower level 1 menyerang dengan 20 pasukan ditambah Attack Up terhadap Tower level 1 dengan 30 pasukan.
  - o Hasil yang seharusnya : Tower tidak berpindah kepemilikan dengan pasukan menjadi 10.
- Fort level 1 menyerang dengan 20 pasukan ditambah Critical Hit terhadap Castle level 1 dengan 40 pasukan.
  - o Hasil yang seharusnya : Castle berpindah kepemilikan dengan 0 pasukan.

### 6.3 Save dan Load

Pengujian ini dilakukan untuk mengetahui apakah command SAVE dan LOAD dapat berfungsi dengan baik. Pengujian ini dilakukan dengan memasukkan command SAVE setelah game berjalan, lalu memasukkan command LOAD setelahnya. Game akan disimpan pada file eksternal dengan nama bisa disesuaikan oleh pemain.

- Command SAVE :
  - o Hasil yang seharusnya : Program akan menyimpan game pada file eksternal
- Command LOAD :
  - o Hasil yang seharusnya : Program dapat membaca file save pada file eksternal dan melanjutkan game sesuai dengan isi file save

### 6.4 Mendapatkan Skill

Pengujian ini dilakukan untuk mengecek apakah terjadi bug pada saat pemain mendapatkan skill. Tiap skill memiliki kondisi tertentu untuk mendapatkan skill tersebut. Kondisi – kondisi tersebut antara lain :

- Instant Upgrade : Skill ini hanya bisa didapatkan di awal permainan
- Shield : Skill ini didapatkan ketika pemain diserang lawan, sehingga bangunan pemain berkurang menjadi 2
- Extra Turn : Skill ini didapatkan ketika bangunan fort pemain diambil oleh lawan
- Attack Up : Skill didapatkan ketika pemain mendapatkan tower lawan dan tower pemain menjadi 3
- Critical Hit : Skill ini didapatkan pemain ketika lawan mengaktifkan skill extra turn
- Instant Reinforcement : Skill didapatkan ketika bangunan pemain seluruhnya memiliki level 4
- Barrage : Skill didapatkan ketika lawan baru saja memiliki bangunan sebanyak 10 bangunan

Untuk menguji apakah semua skill didapatkan pada kondisi tersebut, kami membuat beberapa kasus untuk mendapatkan skill.

### 6.5 Level Up

Pengujian ini dilakukan untuk mengecek apakah command LEVEL\_UP dapat berjalan dengan benar. Bangunan bisa dinaikkan levelnya jika memenuhi beberapa syarat :

- Jumlah pasukan pada bangunan sudah mencukupi untuk minimum kebutuhan level up
- Level pada bangunan belum mencapai maksimum (4)

Kami melakukan beberapa kasus uji :

- Castle level 2 dengan 60 pasukan, dinaikkan levelnya (minimum level up : 30 pasukan).
  - o Hasil yang seharusnya : Level up berhasil. Castle level 3 dengan 30 pasukan.
- Castle level 3 dengan 30 pasukan, dinaikkan levelnya (minimum level up : 40 pasukan).
  - o Hasil yang seharusnya : Level up gagal. Castle level 2 dengan 30 pasukan.
- Tower level 4 dengan 50 pasukan, dinaikkan levelnya.
  - o Hasil yang seharusnya : Level up gagal. Tower level 4 dengan 50 pasukan.

## 6.6 Undo

Pengujian ini dilakukan untuk mengecek apakah command UNDO dapat berjalan dengan benar. Pemain tidak mendapatkan batasan jumlah undo untuk tiap turn, tetapi undo tidak dapat dilakukan pada kondisi berikut :

- Pemain menggunakan skill dalam turn
- Pemain sudah end turn
- Pemain baru saja memulai permainan
- Pemain baru saja melakukan load game

Kami melakukan beberapa kasus uji :

- Undo dilakukan setelah mendapatkan skill.
  - o Hasil yang seharusnya : Undo berhasil. Skill yang baru saja didapatkan, dihapus.
- Undo dilakukan setelah mendapatkan bangunan.
  - o Hasil yang seharusnya : Undo berhasil. Bangunan kembali satu kondisi sebelumnya.
- Undo dilakukan setelah menggunakan skill.
  - o Hasil yang seharusnya : Undo gagal, karena sudah menggunakan skill.
- Undo dilakukan setelah end turn.
  - o Hasil yang seharusnya : Undo gagal, karena sudah end turn.
- Undo dilakukan saat baru saja memulai permainan :
  - o Hasil yang seharusnya : Undo gagal, karena tidak ada yang dikembalikan kondisinya.
- Undo dilakukan setelah load game.
  - o Hasil yang seharusnya : Undo gagal, karena tidak ada yang dikembalikan kondisinya.

## 6.7 Move

Pengujian dilakukan untuk memastikan command MOVE dapat berjalan dengan baik. Command Move berfungsi memindahkan pasukan dari bangunan satu ke bangunan lainnya. Move dapat dilakukan dengan beberapa syarat, yaitu :

- Bangunan asal dan bangunan tujuan terhubung langsung
- Jumlah pasukan dari bangunan asal harus lebih besar dari 0

Kami akan menguji dengan beberapa kasus :

- Castle level 1 dengan 0 pasukan, memindahkan 0 pasukan ke Fort level 1 dengan 0 pasukan.
  - o Hasil yang seharusnya : Move gagal, karena jumlah pasukan yang dipindahkan adalah 0.
- Tower level 2 dengan 30 pasukan, memindahkan -10 pasukan ke Castle level 1 dengan 40 pasukan.
  - o Hasil yang seharusnya : Move gagal, karena jumlah pasukan yang dipindahkan adalah negatif.
- Castle level 2 dengan 60 pasukan, memindahkan 10 pasukan ke Castle level 1 dengan 40 pasukan.
  - o Hasil yang seharusnya : Move berhasil. Castle tujuan pasukannya bertambah 10, dan Castle asal pasukannya berkurang 10.

## 7 Test Script

Untuk menguji data test pada Bab 6 di atas, berikut adalah test script yang digunakan untuk menunjukkan hasil uji yang didapatkan :

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Normal ATTACK	Mengetahui apakah command normal ATTACK (tanpa skill) dapat bterfungsi	<ul style="list-style-type: none"> <li>- Mengetikkan command ATTACK</li> <li>- Memilih bangunan untuk menyerang</li> <li>- Memilih bangunan yang diserang</li> </ul>	<ul style="list-style-type: none"> <li>- Castle (1:40) attack → Tower (1:30)</li> <li>- Castle (1:10) attack → Castle (1:40)</li> <li>- Castle (1:40) attack → Castle (1:40)</li> </ul>	<ul style="list-style-type: none"> <li>- Tower berpindah kepemilikan (0 pasukan)</li> <li>- Castle tidak berpindah kepemilikan (sisa 30 pasukan)</li> <li>- Castle berpindah kepemilikan (0 pasukan)</li> </ul>	<ul style="list-style-type: none"> <li>- Tower berpindah kepemilikan (0 pasukan)</li> <li>- Castle tidak berpindah kepemilikan (sisa 30 pasukan)</li> <li>- Castle berpindah kepemilikan (0 pasukan)</li> </ul>
2	ATTACK Modifier	Mengetahui apakah command ATTACK modifier (dengan skill) dapat berfungsi	<ul style="list-style-type: none"> <li>- Menggunakan skill</li> <li>- Mengetikkan command ATTACK</li> <li>- Memilih bangunan untuk menyerang</li> <li>- Memilih bangunan yang diserang</li> </ul>	<ul style="list-style-type: none"> <li>- Tower (1:20) attack → Castle (1:60)</li> <li>- Tower (1:20) attack → Tower (1:30)</li> <li>- Fort (1:20) attack → Castle (1:40)</li> </ul>	<ul style="list-style-type: none"> <li>- Castle tidak berpindah kepemilikan (sisa 45 pasukan)</li> <li>- Tower tidak berpindah kepemilikan (sisa 10)</li> <li>- Castle berpindah kepemilikan (0 pasukan)</li> </ul>	<ul style="list-style-type: none"> <li>- Castle tidak berpindah kepemilikan (sisa 45 pasukan)</li> <li>- Tower tidak berpindah kepemilikan (sisa 10)</li> <li>- Castle berpindah kepemilikan (0 pasukan)</li> </ul>
3	SAVE	Mengecek apakah command SAVE dapat berjalan dengan baik	<ul style="list-style-type: none"> <li>- Mengetikkan command SAVE</li> <li>- Mengetikkan nama save file</li> </ul>	<ul style="list-style-type: none"> <li>- Save dilakukan dengan command SAVE di tengah permainan</li> </ul>	<ul style="list-style-type: none"> <li>- Save game berhasil</li> </ul>	<ul style="list-style-type: none"> <li>- Save game berhasil</li> </ul>
4	LOAD	Mengecek apakah command LOAD dapat berjalan dengan baik	<ul style="list-style-type: none"> <li>- Mengetikkan command LOAD</li> <li>- Mengetikkan nama save file untuk membuka kembali permainan</li> </ul>	<ul style="list-style-type: none"> <li>- Load dilakukan dengan command LOAD di tengah permainan</li> </ul>	<ul style="list-style-type: none"> <li>- Load game berhasil</li> </ul>	<ul style="list-style-type: none"> <li>- Load game berhasil</li> </ul>
5	SKILL	Mengecek apakah pemain dapat menggunakan skill dan mendapat skill baru	<ul style="list-style-type: none"> <li>- Menggunakan skill dengan mengetikkan command SKILL</li> <li>- Memainkan game hingga mendapatkan kondisi pemain</li> </ul>	<ul style="list-style-type: none"> <li>- Menggunakan skill SHIELD</li> <li>- Mengurangi Tower lawan hingga bersisa 2</li> <li>- Level up semua bangunan seorang pemain</li> </ul>	<ul style="list-style-type: none"> <li>- Serangan pemain lawan pada turn kedua setelah pemakaian Shield menjadi kurang efektif (berhasil)</li> <li>- Pemain lawan mendapat skill Shield</li> </ul>	<ul style="list-style-type: none"> <li>- Serangan pemain lawan pada turn kedua setelah pemakaian Shield menjadi kurang efektif (berhasil)</li> <li>- Pemain lawan mendapat skill Shield</li> </ul>

No.	Fitur yang Dites	Tujuan Testing	Langkah- Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
			mendapat skill baru	hingga level 4 dan melakukan END_TURN	- Pemain mendapatkan skill Instant Reinforcement	- Pemain mendapatkan skill Instant Reinforcement
6	LEVEL UP	Mengetahui apakah fitur LEVEL UP berfungsi sebagaimana seharusnya dengan beberapa kondisi	<ul style="list-style-type: none"> <li>- Memainkan game hingga mendapat kondisi : <ul style="list-style-type: none"> <li>- Jumlah pasukan cukup untuk level up</li> <li>- Jumlah pasukan tidak cukup untuk level up</li> <li>- Bangunan sudah mencapai level maksimum</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Castle (2:60) level up</li> <li>- Castle (3:30) level up</li> <li>- Tower (4:50) level up</li> </ul>	<ul style="list-style-type: none"> <li>- Level up berhasil → Castle (3:30)</li> <li>- Level up gagal → Castle (3:30)</li> <li>- Level up gagal → Tower (4:50)</li> </ul>	<ul style="list-style-type: none"> <li>- Level up berhasil → Castle (3:30)</li> <li>- Level up gagal → Castle (3:30)</li> <li>- Level up gagal → Tower (4:50)</li> </ul>
7	UNDO	Mengecek apakah fungsi UNDO dapat berfungsi dengan baik	<ul style="list-style-type: none"> <li>- Mengetikkan command UNDO pada beberapa kondisi permainan yaitu <ul style="list-style-type: none"> <li>- Pemain mendapat skill baru</li> <li>- Pemain baru saja mendapatk an bangunan</li> <li>- Pemain baru saja menggunak an skill</li> <li>- Pemain baru saja END TURN</li> <li>- Pemain baru saja memulai permainan</li> <li>- Pemain baru saja</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Undo pada saat mendapatkan skill</li> <li>- Undo pada saat mendapatkan bangunan</li> <li>- Undo pada saat menggunakan skill</li> <li>- Undo pada saat End Turn</li> <li>- Undo pada saat memulai permainan</li> <li>- Undo pada saat Load game</li> </ul>	<ul style="list-style-type: none"> <li>- Undo berhasil. Skill yang didapatkan hilang</li> <li>- Undo berhasil. Bangunan kembali ke status sebelumnya</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> </ul>	<ul style="list-style-type: none"> <li>- Undo berhasil. Skill yang didapatkan hilang</li> <li>- Undo berhasil. Bangunan kembali ke status sebelumnya</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> <li>- Undo gagal. Karena tidak ada yang dikembalikan</li> </ul>

No.	Fitur yang Dites	Tujuan Testing	Langkah- Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
			LOAD game			
8	MOVE	Mengecek apakah command MOVE dapat berjalan dengan baik	<ul style="list-style-type: none"> <li>- Memasukkan command MOVE</li> <li>- Memilih pasukan dari bangunan asal dan bangunan tujuan</li> </ul>	<ul style="list-style-type: none"> <li>- Castle (1:0) move 0 pasukan → Fort (1:0)</li> <li>- Tower (2:30) move -10 pasukan → Castle (1:40)</li> <li>- Castle (2:60) move 10 pasukan → Castle (1:40)</li> </ul>	<ul style="list-style-type: none"> <li>- Move gagal. Karena jumlah pasukan yang dipindahkan adalah 0</li> <li>- Move gagal. Karena jumlah pasukan yang dipindahkan adalah negatif</li> <li>- Move berhasil. Castle tujuan bertambah 10 pasukan.</li> </ul>	<ul style="list-style-type: none"> <li>- Move gagal. Karena jumlah pasukan yang dipindahkan adalah 0</li> <li>- Move gagal. Karena jumlah pasukan yang dipindahkan adalah negatif</li> <li>- Move berhasil. Castle tujuan bertambah 10 pasukan.</li> </ul>

Tabel 1. Test Script

## 8 Pembagian Kerja dalam Kelompok

Berikut adalah pembagian kerja dalam kelompok selama pengerjaan tugas besar ini.

NIM	Nama	Pembagian Kerja
13517129	Panawar Hasibuan	Membuat driver beberapa ADT
13518033	Daffa Pratama Putra	Membuat ADT matriks dan ADT list, membuat laporan, mencari bug pada program.
13518063	Dhafin Rayhan Ahmad	Mencari bug pada program, mengidentifikasi letak bug pada <i>source code</i>
13518093	Morgen Sudyanto	Membuat ADT graf, membuat program utama (modul attack, skill, cheat), memperbaiki bug pada program.
13518123	Naufal Dean Anugrah	Membuat ADT Array, Bangunan, Kata, Mesinkata, Point, Queue, Stack, dan Status. Membuat sistem config, save, load game, serta UI game.

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar 2

Saat avatar Aang belum muncul, dunia menjadi kacau dan terjadi perang dunia. Perang ini diikuti oleh 4 negara - Api, Air, Tanah, dan Udara. Pada mulanya, keempat negara berada di keempat penjuru dunia. Mereka memiliki pasukan masing-masing dan jumlahnya terus bertambah. Tiap negara dapat bergerak dan menduduki sebuah kota. Untuk menduduki sebuah kota, sebuah negara harus menyerang dengan pasukan lebih banyak dari penghuni kota tersebut. Ketika sebuah kota diduduki, kota tersebut akan menghasilkan pasukan untuk negara yang berhasil mendudukinya. Pemenangnya adalah negara yang berhasil menduduki seluruh kota.

Game Avatar World War adalah game turn-based strategy yang mensimulasikan perang dunia yang sudah diceritakan di atas. Game ini dimainkan dengan cara memasukkan perintah melalui command line interface dengan command yang akan dijelaskan pada masing-masing fitur game. Game ini dibuat dengan bahasa C.

### 9.2 Notulen Rapat

Berikut adalah hasil notulen rapat yang telah dilakukan.

Hari Kamis, 24 Oktober 2019 :

1. Menentukan pengerjaan tiap ADT
2. Menentukan rancangan program

Hari Rabu, 20 November 2019 :

1. Menentukan pengerjaan driver tiap ADT
2. Menentukan pencarian bug pada program / bug hunt
3. Menentukan pembagian kerja laporan

Hari Kamis, 21 November 2019 :

1. Membuat interface program
2. Mengetes fitur yang ada pada program

### 9.3 Log Activity Anggota Kelompok

Berikut adalah log activity tiap anggota kelompok selama pengerjaan tugas besar ini.

NIM	Nama	Log Activity
13517129	Panawar Hasibuan	22-24 November 2019: Membuat driver beberapa ADT.
13518033	Daffa Pratama Putra	22 Oktober 2019 : Membuat ADT Matriks 23 Oktober 2019 : Membuat ADT List 18 November 2019 : Mencicil laporan 22 - 23 November 2019 : Mencari bug pada program

		20 - 24 November 2019 : Mengerjakan dan menyelesaikan laporan
13518063	Dhafin Rayhan Ahmad	20 Oktober 2019: Merancang ADT Mesin Karakter dan Mesin Kata 23 Oktober 2019: Menambahkan ADT Mesin Karakter dan Mesin Kata 15 November 2019: Testing program pertama setelah jadi 18 November 2019: Menemukan tiga bug 20 November 2019: Menemukan satu bug, merancang desain map 21 November 2019: Merapikan dan menkonversi laporan 24 November 2019: Menambahkan <i>test script</i> pada laporan
13518093	Morgen Sudyanto	30 Oktober 2019: Merancang ADT Graph 1 November 2019: Membuat ADT Graph 2 November 2019: Membuat modul tambah pasukan 13 November 2019: Membuat implementasi attack command 14 November 2019: Mengubah implementasi undo command 14-15 November 2019: Membuat implementasi semua skill 19 November 2019: Memfix beberapa bug (save, inialisasi variabel, level restart, dll) 20 November 2019: Membuat winning condition, memfix bug attack modifier, shield dan tambah pasukan, menambah command cheat.



		22 November 2019: Menambah icon untuk skill dan fitur autosave 23 November 2019: Mengubah tampilan UI (sedikit)
13518123	Naufal Dean Anugrah	22 Oktober 2019: Membuat ADT Point, Bangunan, Array, dan Queue. 23 Oktober 2019: Membuat ADT Kata dan Stack. 1 November 2019: Membuat ADT Status, menambah fungsi ADT Mesinkar dan Mesinkata untuk input stdin. Membuat config game. 2 November 2019: Menambah fungsi CopyGraph, bug fix. Menambah fungsi di ADT Kata. 14 November 2019: Membuat command MOVE, LEVEL_UP, dan HELP. Bug fix command UNDO. 15 November 2019: Membuat sistem load dan save game. 22 November 2019: Membenahi UI game, bug fix saat load game, menambahkan QueueSkill ke ADT Stack.

#### 9.4 Form Asistensi

Berikut adalah lampiran form asistensi.







**Form Asistensi Tugas Besar**  
**IF2110/Algoritma dan Struktur Data**  
**Sem. I 2019/2020**

No. Kelompok/Kelas : 4/K03  
 Nama Kelompok :  
 Anggota Kelompok (Nama/NIM) :  
 1. Daffa Pratama Putra / 13510033  
 2. Naufal Dean Anugrah / 13518123  
 3. Mogen Sidyoto / 13518093  
 4. Dhaifa Rakhma Akmal / 13518063  
 5. Panamar Hasbiyuan / 13517129  
 6.








Asisten Pembimbing : Shewalda Gracielira (13516134)

---

Gambar 3. Form asistensi halaman 1

Asistensi I	
Tanggal : 24 Oktober 2019	Catatan Asistensi:
Tempat : Lab Pemrograman	1. Typo ukuran pete, pack spek (contoh file eksternal) ✓
Kehadiran Anggota Kelompok:	2. Mekanisme perbandingan (3/4)
No	3. Skill instant reinforcement ✓ (pelama level semua, dapat skill)
NIM	4. Mesin karakter & mesin kata ✓
Tanda tangan	5. Mekanisme undo ✓ (bisa undo selama belum end turn)
1	6. Skill selama masih ada di queue ✓ (bisa dipake di turn (queue max 10))
13510033	7. Mekanisme penambahan pasukan
	8. Kritical Hit (revisi)
2	9. Batasan stack (aksi) ✓ ⇒ asumsi
13518123	
	
3	
13510033	
	
4	
13518063	
	
5	
13510033	
	
6	
	Tanda Tangan Asisten: 

Gambar 4. Form asistensi halaman 2

Asistensi II	
Tanggal : 20 November 2019	<b>Catatan Asistensi:</b> 1. Folder pengumpulan (duplikasi tiap ART dan ada driver) 2. Scan hasil asistensi & dimajukan ke laporan 3. Rptk direktur, diarsipkan tanpa spasi 4. Tes Script jangan panjang panjang dan yang sekiranya penting aja 5. Pengisian form penelitian (nant dihubungin)
Tempat : Lab Pemrograman	
Kehadiran Anggota Kelompok:	
No	
NIM	
Tanda tangan	<b>Tanda Tangan Asisten:</b> 
1 13519033 	
2 13513063 	
3 13518123 	
4 13518093 	
5 14517129 	
6 	

Gambar 5. Form asistensi halaman 3