

**LAPORAN TUGAS BESAR**  
**IF2111/Algoritma dan Struktur Data**

**Permainan Mobitangga**


Dipersiapkan oleh:

Kelompok 16

Muhammad Davin Dzimar	18220046
Theodore Justin Lionar	18220011
Fikri Muhammad Fahreza	18220012
Bryan Eagan	18220041
Verawati Esteria S. Simatupang	18220002

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>IF2111-TB-16</i>		<i>17</i>
		<i>Revisi</i>	<i>1</i>	<i>28 November 2021</i>

# Daftar Isi

<b>Ringkasan</b>	<b>3</b>
<b>Penjelasan Tambahan Spesifikasi Tugas</b>	<b>4</b>
Spesifikasi Skill: Mesin Waktu	4
Spesifikasi Skill: Baling-Baling Jambu	4
<b>Struktur Data (ADT)</b>	<b>5</b>
ADT Array of Integer	5
ADT Array of Character	5
ADT Mesin Karakter	5
ADT Mesin Kata	6
ADT List	6
ADT Player	6
ADT State (Stack)	7
<b>Program Utama</b>	<b>8</b>
<b>Algoritma-Algoritma Menarik</b>	<b>9</b>
Algoritma UseSkill1 dan UseSkill2	9
Algoritma readConfig	9
<b>Pembagian Kerja dalam Kelompok</b>	<b>10</b>
<b>Lampiran</b>	<b>11</b>
Deskripsi Tugas Besar	11
Form Asistensi	12
Notulen Rapat	15
Log Activity Anggota Kelompok	16

# 1 Ringkasan

Permainan Mobitangga adalah *board game digital* yang menggabungkan permainan ular tangga dengan modifikasi-modifikasi yang dapat mengganggu lawan. Permainan ini dilakukan oleh 2 pemain di suatu papan satu dimensi yang memiliki panjang N. Para pemain akan memulai permainan dari petak 1 dan berakhir saat mencapai petak ke-N. Saat permainan berlangsung, pemain dapat menggunakan teleporter dan skill untuk mencapai tujuan baik oleh pemain, maupun lawan. Permainan akan berakhir jika sudah ada satu pemain yang mencapai petak ke-N. Peringkat pemain lain akan dilihat berdasarkan posisi pemain saat pemenang ditemukan. Permainan akan dilakukan dalam beberapa ronde, dengan satu ronde menandakan semua pemain telah menyelesaikan gilirannya.

Berdasarkan deskripsi *game* tersebut, kami membuat program permainan dengan *game mechanics* yang terdiri atas 7 komponen, yaitu main menu, peta, roll, teleporter, skill, buff, dan mekanisme permainan. Laporan ini berisi deskripsi umum permainan, penjelasan tambahan terkait spesifikasi game, struktur data (ADT) yang digunakan, dan algoritma menarik. Adapun ADT yang terdapat pada laporan ini adalah ADT array, mesin karakter, mesin kata, list, player, dan state. Sebagai pelengkap laporan, dilampirkan juga beberapa dokumen antara lain deskripsi tugas besar, form asistensi, notulen rapat, dan log activity.

Permainan Mobitangga dibuat dalam bahasa C yang ditujukan kepada pemain menggunakan struktur data yang telah dipelajari pada mata kuliah Algoritma dan Struktur Data. Pada awal game, ditampilkan main menu yang memberikan pilihan kepada pemain untuk bermain game atau keluar dari game. Setelah permainan dimulai, pemain akan melakukan command secara bergiliran atau bergantian. Command yang valid pada game ini adalah NEW GAME, EXIT, SKILL, MAP, BUFF, INSPECT, ROLL, ENDTURN, dan UNDO. Permainan akan berakhir jika sudah ada pemain yang mencapai petak ke-N dan akan ditampilkan pesan bahwa permainan sudah berakhir. Kelompok kami berhasil menyelesaikan fitur-fitur utama yang wajib dimiliki permainan. Selain itu, kelompok kami berhasil menyelesaikan fitur bonus yaitu skill mesin waktu dan baling-baling jambu.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Spesifikasi Skill: Mesin Waktu

Ketika skill Mesin Waktu digunakan, pemain dapat membuat pemain lain mundur. Banyaknya langkah mundur merupakan angka *random* antara 1 sampai maksimal roll yang dapat diperoleh. Pemain lawan yang bergerak mundur tetap akan mengalami teleportasi jika berhenti pada petak dengan teleporter. Akan tetapi, pemain tidak bisa mengirim pemain lawan ke petak terlarang ('#') atau ke luar map permainan. Ketika hal tersebut terjadi, maka skill gagal digunakan dan pemain lawan tidak terpengaruh.

### 2.2 Spesifikasi Skill: Baling-Baling Jambu

Ketika mengaktifkan skill Baling-Baling Jambu, pemain dapat membuat pemain lain melangkah maju. Jumlah langkah maju yang terjadi adalah angka *random* antara 1 sampai jumlah roll maksimal. Pemain lawan yang bergerak maju akan tetap mengaktifkan teleporter apabila bertemu dengan petak dengan teleporter, namun tetap dapat menggunakan imunitas teleport. Namun, pemain tidak dapat membuat pemain lawan bergerak ke petak terlarang ('#') atau ke luar map permainan. Ketika hal tersebut terjadi, pemain lawan tidak akan bergerak dan skill gagal digunakan.

## 3 Struktur Data (ADT)

### 3.1 ADT Array of Integer

- Sketsa struktur data: Struktur array terdiri dari struktur TabInt yang terdiri atas array of integer dan Neff (integer). *Prototype* pada ADT meliputi konstruktor pembentuk tabel kosong, *selector* banyaknya elemen, mengirimkan elemen, dan menghasilkan sebuah elemen, dan *selector* mengeset nilai elemen pada tabel.
- Persoalan yang diselesaikan: Menyimpan pintu masuk dan pintu keluar teleporter yang ada pada file konfigurasi.
- Alasan pemilihan: Informasi teleporter, baik pintu masuk dan pintu keluar harus disimpan ke dalam struktur selama permainan berjalan. Informasi disimpan ke dalam tabel berisi integer, karena setiap petak pada peta digambarkan posisinya dengan integer.
- Diimplementasikan sebagai ADT Array dengan nama *file header* “array.h”

### 3.2 ADT Array of Character

- Sketsa struktur data: Struktur array terdiri dari struktur TabChar yang terdiri atas array of character dan Neff (integer). *Prototype* pada ADT meliputi konstruktor pembentuk tabel kosong, *selector* banyaknya elemen, mengirimkan elemen, dan menghasilkan sebuah elemen, dan *selector* mengeset nilai elemen pada tabel.
- Persoalan yang diselesaikan: Menyimpan peta permainan (map)
- Alasan pemilihan: Informasi peta permainan beserta posisi player harus disimpan selama permainan. Informasi ini disimpan ke dalam tabel untuk menyimpan karakter yang ada di konfigurasi peta.
- Diimplementasikan sebagai ADT Array dengan nama *file header* “arrayChar.h”

### 3.3 ADT Mesin Karakter

- Sketsa struktur data: Struktur mesin karakter terdiri dari sebuah variabel extern bertipe char dengan nama CC dan boolean dengan nama EOP. *Prototype* pada ADT meliputi START() , STARTFILE (), dan ADV().
- Persoalan yang diselesaikan: ADT Mesin Kata
- Alasan pemilihan: ADT Mesin Kata memerlukan ADT Mesin Karakter dan primitif-primitifnya dalam pemrosesan kata.
- Diimplementasikan sebagai ADT Mesin Karakter dengan nama *file header* “mesin\_kar.h”

### 3.4 ADT Mesin Kata

- Sketsa struktur data: Struktur mesin kata terdiri dari struktur Kata yang terdiri atas array of char dan Length (integer). Blank yang didefinisikan dalam ADT ini adalah ' ' (spasi). *Prototype* pada ADT meliputi prosedur untuk mengabaikan blank, mengakuisisi kata, menyalin kata, dan fungsi-fungsi untuk membaca file configure, seperti fungsi untuk konversi kata ke string dan mitigasi input nama direktori sehingga space tetap diakuisisi.
- Persoalan yang diselesaikan: Membaca informasi peta (map) dari *file external* konfigurasi, membaca command dari player.
- Alasan pemilihan: Informasi yang ada pada file eksternal harus dibaca satu per satu per 'kata' sehingga dapat ditranslasikan ke bentuk peta (map) atau bentuk-bentuk lainnya yang ada di dalam program.
- Diimplementasikan sebagai ADT Mesin Kata dengan nama *file header* "mesin\_kata.h"

### 3.5 ADT List

- Sketsa struktur data: Struktur list terdiri dari address First dan struktur elemen (tElmntList) terdiri atas info bertipe data integer dan next bertipe data *address*. *Prototype* pada ADT meliputi creator list kosong, manajemen memori (alokasi dan dealokasi), pengecekan list kosong, pencarian elemen dalam list, penambahan dan pengurangan elemen.
- Persoalan yang diselesaikan: ADT Player
- Alasan pemilihan: Skill yang dimiliki player akan terus bertambah setiap turn, oleh sebab itu dibutuhkan struktur yang mampu menampung skill sesuai urutan didapatkannya skill tersebut.
- Diimplementasikan sebagai ADT List dengan nama *file header* "listlinier.h"

### 3.6 ADT Player

- Sketsa struktur data: Struktur player terdiri dari name (*array of character*), position (integer), skill (List), dan buff (*array of boolean*). *Prototype* pada ADT meliputi inisiasi pembentuk player kosong dan mengirimkan nama dari player.
- Persoalan yang diselesaikan: Menentukan posisi player sesaat, menyimpan skill yang dimiliki player, menyimpan buff yang sedang aktif, menyimpan nama player, dan ADT State.
- Alasan pemilihan: Diperlukan suatu struktur data yang dapat menyimpan informasi tentang player, mulai dari nama, posisi player di dalam map, serta skill dan buff yang sedang aktif. Struktur ini memungkinkan penyimpanan data player saat itu sebagai satu kesatuan.
- Diimplementasikan sebagai ADT Player dengan nama *file header* "player.h"

### 3.7 ADT State (Stack)

- Sketsa struktur data: Struktur State terdiri atas *array of* Player dan *address* dari alamat teratas State. State menggunakan struktur serupa dengan ADT Stack. ADT State memiliki prosedur umum seperti stack, yaitu Push dan Pop kondisi Player ke dalam data State.
- Persoalan yang diselesaikan: Fitur undo untuk mengembalikan state permainan ke akhir turn sebelumnya.
- Alasan pemilihan: Untuk menyimpan kondisi dari permainan, dibutuhkan lokasi yang dapat di-*update* terus menerus tanpa menghilangkan kondisi sebelumnya. Kondisi permainan yang harus disimpan adalah data bentukan Player.
- Diimplementasikan sebagai ADT State dengan nama file header “state.h”.

## 4 Program Utama

Program utama dengan nama file “main.c” meng-include semua file ADT dan fitur-fitur permainan. Program utama akan menampilkan main menu berupa pilihan *new game* dan *exit*. Program akan melakukan *looping* untuk meminta user memasukkan pilihan. Program keluar dari *loop* ketika *user* memberikan *input* “NEWGAME” atau “EXIT”. Kemudian, program akan memulai permainan dengan melakukan beberapa proses awal, yaitu meminta file konfigurasi permainan yang akan dibaca dan nama-nama pemain. Selanjutnya program akan kembali melakukan *looping* fungsi *play* dari file “gameRound.c” untuk menampilkan menu yang dapat dipilih oleh pemain (*roll*, *skill*, *buff*, *map*, *inspect*, dan *undo*) dan meminta masukan dari *user*. Input dari *user* akan dibaca menggunakan mesin kata.

Jika *user* atau pemain memasukkan *input* “ROLL”, akan dilakukan pemanggilan prosedur *roll* dari file “mapOperate.c”, yaitu pemutaran dadu dan pergerakan pemain pada *map* jika dapat bergerak menuju petak bukan petak terlarang. Jika pemain memasukkan *input* “SKILL”, program akan memanggil prosedur SKILL dari file “skill.c” dan menampilkan *skill* yang dimiliki pemain. Pemain dapat menggunakan *skill* jika memberikan input sesuai nomor urut *skill*, membuang *skill* dengan memberikan bilangan negatif, dan keluar dari menu *skill* dengan memberi input ‘0’. Jika pemain memberikan *input* “BUFF”, prosedur BUFF dari file “buff.c” akan dipanggil untuk menampilkan daftar *buff* yang sedang dimiliki pemain.

Sementara jika pemain memasukkan *input* “MAP”, prosedur *printMap* dari file “mapOperate.c” akan dipanggil untuk mencetak *map* permainan beserta posisi masing-masing pemain. Jika *input* yang dimasukkan pemain adalah “INSPECT”, program utama akan memanggil prosedur *inspectTile* dari file “mapOperate.c” untuk melakukan inspeksi keberadaan teleporter pada petak yang ingin diketahui pemain. Jika pemain memberikan *input* “UNDO”, prosedur *undo* dari file “gameRound.c” akan dipanggil untuk mengembalikan state permainan ke *ronde* sebelumnya.

Setelah pemain melakukan *roll* dengan melakukan *input* “ROLL”, program akan menampilkan menu selanjutnya yang dapat dipilih pemain, yaitu *end turn* dan *undo*. Ketika pemain memasukkan *input* “ENDTURN”, *turn* dari pemain tersebut berakhir dan berlanjut ke *turn* pemain lain dalam *loop*. Setiap akhir dari *round*, program akan memeriksa apakah salah satu pemain sudah berada di ujung *map* permainan. Apabila ditemukan, program akan berhenti melakukan *looping* dan menampilkan pemenang permainan dan ranking akhir.



## 5 Algoritma-Algoritma Menarik

### 5.1 Algoritma UseSkill1 dan UseSkill2

Algoritma UseSkill1 dan UseSkill2 merupakan prosedur yang didefinisikan pada modul Skill.

```
void UseSkill1 (Player *P, int input);  
void UseSkill2 (Player *P1, Player *P2, int input);
```

Kedua algoritma ini digunakan untuk mengeksekusi penggunaan skill ketika dipanggil oleh seorang player. Mengapa algoritma ini menarik ialah karena terdapat dua versi berbeda dari jenis permasalahan yang nampak serupa. Pada dasarnya, UseSkill1 digunakan untuk menggunakan skill yang hanya akan berdampak kepada player yang memanggilnya, contohnya untuk mengaktifkan buff. Sedangkan UseSkill2 digunakan karena terdapat beberapa skill yang juga mempengaruhi player lain selain player yang memanggilnya, yakni untuk menukar posisi dengan pemain lawan atau mendorong lawan untuk mundur ataupun maju secara paksa.

### 5.2 Algoritma readConfig

Algoritma readConfig ini terdefinisi pada modul Read Config yang digunakan untuk membaca file dengan format .txt.

```
void readConfig (char *filepath, TabChar *map, int *maxRoll, int  
*TeleporterCount, TabInt *TeleportIn, TabInt *TeleportOut);
```

Algoritma ini membaca file .txt yang merupakan file konfigurasi permainan. Alasan algoritma ini menarik adalah prosedur yang terealisasi merupakan prosedur file reader/parser yang khusus digunakan untuk membaca file konfigurasi untuk melakukan set up program. ADT mesin kata yang digunakan pun menarik karena “pita” yang merupakan objek pembacaan mesin biasanya merupakan hasil input keyboard, namun pada algoritma ini yang menjadi objek pembacaan utama adalah file eksternal.

## 6 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1	Program Utama	18220046	18220046
2	ADT Array of Character dan Array of Integer	18220046	18220002
3	ADT Mesin Karakter dan ADT Mesin Kata	18220046	18220012
4	ADT List	18220002, 18220041, 18220046	18220011
5	ADT Player	18220041, 18220046	18220041
6	ADT State (Stack)	18220011, 18220046	18220002
7	Skill	18220002, 18220041, 18220046	18220041
8	Map	18220046	18220011
9	Buff	18220012, 18220041, 18220046	18220012
10	Inspect	18220046	18220011
11	Roll	18220046	18220041
12	Endturn	18220046	18220046
13	Undo	18220046	18220046

## 7 Lampiran

### 7.1 Deskripsi Tugas Besar

Sebuah Institut Teknologi tertentu sedang mengadakan lomba *game dev* dengan tema membuat board game digital terbaik se-kecamatan. Mendengar hal tersebut, Borakemon dan Mobita bersekongkol untuk membuat sebuah board game digital terasyik. Mereka kepikiran untuk menggabungkan game ular tangga dengan modifikasi-modifikasi yang dapat mengganggu lawan sehingga lahirlah ide Mobitangga.

Namun sayangnya, Mobita tidak memiliki kemampuan maupun niat memprogram Mobitangga. Borakemon, kucing robot Mobita, juga belum memiliki kemampuan untuk memprogram karena belum belajar terlalu *deep*. Oleh karena itu, Borakemon menculik sekelompok *programmer* dari dimensi lain agar dapat membantu mereka membuat program Mobitangga agar dapat memenangkan lomba *game dev* itu.

Permainan ini berbasis CLI (*command-line interface*). Permainan ini dibuat dalam bahasa C dengan menggunakan atau memodifikasi struktur data yang sudah dipelajari di mata kuliah Algoritma dan Struktur data.

## 7.2 Form Asistensi

### 7.2.1. Asistensi I







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

No. Kelompok/Kelas : 16/01  
Nama Kelompok : Kelompok 16  
Anggota Kelompok (Nama/NIM) :  
1. Muhammad Davin Dzimar/18220046  
2. Theodore Justin Lionar/18220011  
3. Fikri Muhammad Fahreza/18220012  
4. Bryan Eagan/18220041  
5. Verawati Esteria S. Simatupang/18220002

Asisten Pembimbing : Jun Ho Choi Hedyatmo/13518044






---

Asistensi I

<b>Tanggal : 08 - 11 - 2021</b>	<b>Catatan Asistensi:</b>
<b>Tempat : Google Meet</b>	<ol style="list-style-type: none"> <li>1. Jika kerjain bonus, bisa masukin player sampai 4. Kalau ga hanya 2</li> <li>2. Nge-draw skill-nya otomatis</li> <li>3. Menyimpan skill bisa dengan list</li> <li>4. Bisa menggunakan library random</li> <li>5. Jika terdapat 3 pemain dan mendapatkan skill baling - baling jambu, maka yang mendapatkan dapat memilih pemain lawan yang akan dikenakan</li> <li>6. Modifikasi ADT</li> <li>7. Tampilan awal boot game bebas</li> <li>8. Peta hanya 1-dimensional</li> <li>9. Jika tidak mengerjakan skill bonus maka chance skill lain menyesuaikan sehingga total 100%</li> <li>10. Game mengakses file konfigurasi terlebih dahulu dengan menggunakan mesin kata untuk membaca peta, max roll, dan teleporter</li> <li>11. Setiap giliran akan di-write ke txt file agar jika kita meng-load saved file kita dapat undo giliran</li> <li>12. Setelah setiap giliran pemain meng-end turn secara manual</li> <li>13. Pembagian tugas baiknya secara seri bukan paralel sehingga tidak kesulitan meng-merge fungsi-fungsi di akhir dan tidak banyak bug</li> </ol>
<b>Kehadiran Anggota Kelompok:</b> <div style="text-align: center;"> <p>No</p> <p>NIM</p> <p>Tanda tangan</p>   <p>1</p> <p>18220046</p>    <p>2</p> <p>18220012</p>    <p>3</p> <p>18220002</p>    <p>4</p> <p>18220011</p>    <p>5</p> <p>18220041</p>  </div>	<b>Tanda Tangan Asisten:</b> 

## 7.2.2. Asistensi II

### Asistensi II

<b>Tanggal : 19 - 11 - 2021</b>	<b>Catatan Asistensi:</b> <ol style="list-style-type: none"> <li>1. Untuk revisi dan no revisi pada cover laporan abaikan aja dulu buat aja revisi 1</li> <li>2. Tanggal release adalah tanggal laporan selesai</li> <li>3. Log activity tidak ada format jadi terserah mau buatnya gimana. Biasanya dibuat dalam bentuk tabel terdiri dari NIM, Nama, Tanggal dan deskripsi tugas</li> <li>4. Deskripsi sistem pada tugas besar sampai spesifikasi umum. Jangan semua, tapi deskripsikan.</li> <li>5. Kalau load filenya, state awalnya harus di-save dulu</li> <li>6. Kalau misalnya ada 3 pemain, pemain kedua bebas mau nge-undo</li> </ol>
<b>Tempat : Google Meet</b>	
<b>Kehadiran Anggota Kelompok:</b>  <div> <div>No</div> <div>NIM</div> <div>Tanda tangan</div> </div> <div> <div>1</div> <div>18220046</div> <div></div> </div> <div> <div>2</div> <div>18220002</div> <div></div> </div> <div> <div>3</div> <div>18220041</div> <div></div> </div> <div> <div>4</div> <div>18220012</div> <div></div> </div>	
	<b>Tanda Tangan Asisten:</b>  

### 7.3 Notulen Rapat

7.2.1 Minggu, 14 November 2021

Pembagian tugas:

Tugas	Nama
Skill (mekanisme random number dan implementasi list)	Verawati Esteria S. Simatupang
Buff (penggunaan skill dan efek)	Fikri Muhammad Fahreza
ADT Player (implementasi dan data yang disimpan)	Bryan Eagan
ADT State (menyimpan state permainan untuk fitur undo)	Theodore Justin Lionar
Undo dan implementasi stack/state	Muhammad Davin Dzimar

7.2.2 Rabu, 24 November 2021

Rencana berikutnya: Merapikan kode dan menyatukan semua fungsi.

## 7.4 Log Activity Anggota Kelompok

NIM	Deskripsi	Tanggal
18220046	Membuat <i>repository</i> github	5 November 2021
18220002	Mengisi form asistensi 1	8 November 2021
18220041	Melengkapi form asistensi 1	9 November 2021
18220046	Membuat ADT Mesin Kata, Mesin Karakter, <i>prototype reader file</i> konfigurasi	13 November 2021
18220046	Membuat <i>prototype file</i> utama “main.c”, <i>sample file</i> konfigurasi, fungsi-fungsi roll, menu, dan game mode	19 November 2021
18220041	Membuat ADT Player	19 November 2021
18220002	Mengisi form asistensi 2	19 November 2021
18220012	Membuat fitur Buff	21 November 2021
18220002	Membuat fitur Skill	21 November 2021
18220046	Memperbaiki ADT Player, fungsi-fungsi <i>main</i> , membuat fungsi-fungsi “console.c”, dan variabel global	21 November 2021
18220002	Memperbaiki fitur Skill (RandomSkill), ADT List	22 November 2021
18220011	Membuat ADT State	22 November 2021
18220046	Menambahkan fungsi ADT Array, fungsi <i>teleport</i> , memperbaiki fitur roll dan fungsi-fungsi main, restrukturisasi Buff, Skill, dan ADT Player	22 November 2021
18220002	Memperbaiki fitur Skill	23 November 2021
18220041	Memperbaiki fungsi pada fitur Buff dan Skill	24 November 2021
18220012	Memperbaiki fitur Buff	24 November 2021
18220002	Membuat template laporan, mengisi laporan bagian ringkasan, deskripsi tugas besar	24 November 2021
18220011	Merapikan template laporan, lampiran	24 November 2021
18220041	Memperbaiki fitur Skill, Buff, dan ADT Player	25 November 2021
18220046	Restrukturisasi dan modularisasi program, menambahkan fitur Skill, Buff, memperbaiki ADT State	25 November 2021



18220011	Mengisi laporan bagian ADT State (Stack)	26 November 2021
18220002	Mengisi laporan bagian ADT Array of Character, Array of Integer, Mesin Kata, Mesin Karakter, List, Player	26 November 2021
18220046	Memperbaiki ADT Mesin Karakter, Mesin Kata, <i>file</i> console, fitur game round, map, skill, dan pembacaan konfigurasi	26 November 2021
18220046	Membuat fitur Undo	26 November 2021
18220011	Melengkapi laporan bagian ADT Array of Character, Array of Integer, Mesin Kata, Mesin Karakter, List, Player	27 November 2021
18220041	Menambahkan teks UI pada fitur Skill	27 November 2021
18220046	Memperbaiki kesalahan interface, fitur Undo, menyempurnakan file main dan fitur game round, menambahkan skill bonus	27 November 2021
18220046	Restrukturisasi <i>directories</i> dan menyempurnakan main menu	28 November 2021
18220046	Menyempurnakan fitur Buff, Skill, membuat batch file, melakukan <i>compile</i> program utama	28 November 2021
18220011	Mengisi laporan bagian program utama	28 November 2021
18220002	Menyempurnakan laporan, memasukkan laporan ke dalam <i>repository</i> Github	28 November 2021
18220046	Menambahkan bagian Algoritma-Algoritma Menarik pada laporan	28 November 2021