

LAPORAN TUGAS BESAR

IF2111/Algoritma dan Struktur Data

Mobitangga


Dipersiapkan oleh:

Kelompok 2

18219076	Yahya Aqrom
18220007	Joanna Margareth Nauli
18220025	Joshi Ryu Setiady
18220049	Rachmad Hidayat
18220050	Alya Apriliyanti
18220088	Ahmad Wafika Samsea

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB-02</i>		<i>15</i>
		<i>Revisi</i>	-	-

Daftar Isi

Ringkasan	2
Penjelasan Tambahan Spesifikasi Tugas	3
Skill Mesin Waktu	3
Skill Baling Baling Jambu	3
Save Game	4
Load Game	4
Dukungan 3 Sampai 4 Pemain	4
Struktur Data (ADT)	4
ADT Array	4
ADT List Terkait dengan Representasi Fisik Pointer	4
ADT Mesin Karakter dan Mesin Kata	5
ADT Queue	6
ADT Stack	6
Program Utama	6
Pembagian Kerja dalam Kelompok	8
Lampiran	8
Deskripsi Tugas Besar	8
Notulen Rapat	9
Log Activity Anggota Kelompok	13

1 Ringkasan

Program Mobitangga adalah permainan berbasis *command-line interface* (CLI) yang menggabungkan game ular tangga dengan modifikasi-modifikasi tambahan. Modifikasi dari permainan ular tangga biasa ini ditunjukkan dengan adanya fitur baru yaitu skill beserta buff yang beragam dengan keunggulannya masing-masing yang juga bervariasi. Eksekusi perintah di dalam permainan ini dijalankan dengan menggunakan *command* yang terdiri dari Skill, Map, Buff, Inspect, Roll, EndTurn, serta Undo. Bila sudah ada pemain yang mencapai petak N, inilah yang menandakan bahwa permainan akan berakhir.

Laporan Tugas Besar ini secara umum merangkum ringkasan permainan, penjelasan tambahan spesifikasi yang kelompok kami gunakan, penjelasan dari struktur data yang dipakai, penjelasan dari program utama game, beserta dengan lampiran-lampiran yang mendukung keutuhan tugas.

Tugas besar ini dirangkai menggunakan bahasa C dan berhasil dijalankan pada mesin berbasis Windows dan berbasis Unix. Program memanfaatkan struktur-struktur data yang dipelajari pada mata kuliah IF2111 Algoritma dan Struktur Data STI, meliputi ADT Array, ADT Mesin Karakter dan Mesin Kata, ADT Queue, ADT Stack, serta ADT List Terkait.

2 Penjelasan Tambahan Spesifikasi Tugas

Pada tugas besar ini, kami mengerjakan beberapa fitur bonus. Berikut ini spesifikasi dari fitur bonus yang dikerjakan.

2.1 Skill Mesin Waktu

Fitur tambahan Mesin Waktu merupakan *skill* tambahan yang dapat dimiliki *player* dengan probabilitas 10% dari semua *skill*. *Skill* dari Mesin Waktu adalah *player* yang menggunakan *skill* ini dapat memindahkan *player* lain sebanyak x langkah ke belakang (contoh : Mesin Waktu 5 akan memundurkan 1 *player* lain sebanyak 5 langkah). Bila *player* yang menjadi target dari *skill* ini pindah ke petak yang memiliki teleporter dan memiliki *buff* Imunitas Teleport, ia akan diberi pilihan untuk menggunakan *buff* tersebut untuk menolak menggunakan teleporter tersebut.

2.2 Skill Baling Baling Jambu

Fitur Baling Baling Jambu merupakan *skill* tambahan yang dapat dimiliki *player* dengan probabilitas 10% dari semua *skill*. *Skill* dari Baling Baling Jambu adalah *player* yang menggunakan *skill* ini dapat memindahkan *player* lain sebanyak x langkah ke depan (contoh : Baling Baling Jambu 5 akan memajukan 1 *player* lain sebanyak 5 langkah). Bila *player* yang menjadi target dari *skill* ini pindah ke petak yang memiliki teleporter dan memiliki *buff* Imunitas Teleport, ia akan diberi pilihan untuk menggunakan *buff* tersebut untuk menolak menggunakan teleporter tersebut.

2.3 Save Game

Fitur tambahan *Save Game* yang berada dalam *Game Loop* berfungsi untuk menyimpan *state* permainan dengan syarat dapat digunakan setelah pemain bergerak (memilih *roll*). Yang disimpan dalam *save file* yaitu panjang peta, peta, *maxroll*, banyaknya teleporter, petak teleporter, petak akhir teleporter, jumlah pemain, ronde, pemain yang melakukan *Save Game*, nama pemain, posisi pemain, jumlah *skill* yang dimiliki pemain, *id skill*, *amount skill*, dan *buff* yang sedang aktif.

2.4 Load Game

Fitur tambahan Load Game yang berada dalam Main Menu berfungsi untuk melanjutkan game yang telah dimainkan sebelumnya. Fitur ini akan membaca informasi dari file eksternal yang telah dibuat pada saat fitur Save Game.

2.5 Dukungan 3 Sampai 4 Pemain

Fitur tambahan dukungan 3 sampai 4 pemain yang berada pada saat game dimulai menggunakan *queue* untuk membuat urutan player.

3 Struktur Data (ADT)

3.1 ADT Array

Struktur dari ADT Array digunakan untuk representasi *command Buff*, *Map*, *Teleport* dan *Inspect*. Map direpresentasikan dengan ADT array, “.” di dalam array direpresentasikan dengan angka petak tersebut. Sedangkan, “#” direpresentasikan dengan angka 0. Selain itu, teleport juga menggunakan ADT Array indeks+1 dari array merepresentasikan petak awal teleporter dan elemen pada indeks tersebut berisi petak tujuan dari teleporter, sedangkan petak yang tidak berisi teleporter elemennya akan diisi dengan angka 0.

Alasan ADT Array dipilih sebagai memori tempat menyimpan elemen karena dengan array akan memudahkan dalam mengakses elemen yang ada di dalamnya. Selain itu, cocok untuk Buff karena array setiap buff memiliki banyak elemen yang tetap.

3.2 ADT List Terkait dengan Representasi Fisik Pointer

Struktur dari ADT List Terkait yang dipakai untuk *command skill* pada program ini berbentuk *array* berisi 4 elemen dengan setiap elemen berisi *address* dari *skill* pertama yang dimiliki tiap *player*. *Address* pada elemen tersebut merupakan *pointer* ke Element List Skill yang berisi *skill_id*, *amount*, dan *next*. *Skill_id* dan *amount* merupakan *integer* yang merepresentasikan jenis *skill* dan jumlah langkah (bila ada).

Berikut merupakan pembagian *skill_id* dan jenis skillnya:

- 1 → Pintu Ga Ke Mana Mana
- 2 → Mesin Waktu
- 3 → Baling Baling Jambu
- 4 → Cermin Pengganda
- 5 → Senter Pembesar Hoki
- 6 → Senter Pengecil Hoki
- 7 → Mesin Penukar Posisi

Untuk *amount*, bila *skill* tersebut tidak memiliki jumlah langkah maka akan diisi dengan 0. *Skill* yang memiliki jumlah langkah ($\neq 0$) sebagai atribut tambahan adalah Mesin Waktu dan Baling Baling Jambu. *Next* merupakan *address* dari skill berikutnya yang dimiliki *player* tersebut. Bila *player* tersebut tidak memiliki *skill* lagi, maka *next* akan berisi Nil.

Digunakannya ADT List Terkait dengan Representasi Fisik Pointer adalah karena beberapa alasan. Alasan pertama adalah masing-masing *player* memiliki jumlah *skill* yang berbeda-beda dan selalu dinamis, oleh karena itu kami menggunakan list terkait agar dapat menambahkan atau mengurangi *skill player* dengan bebas dan fungsi untuk menunjukkan list dari skill yang dimiliki seorang *player* dapat disesuaikan dengan mudah. Alasan lainnya adalah untuk akses pada *skill* tertentu juga lebih mudah dibanding menggunakan array biasa (bila menggunakan array biasa, setelah *skill* digunakan akan meninggalkan spot kosong sehingga akan menyulitkan dalam menunjukkan *list skill* yang dimiliki *player* tersebut).

ADT List Terkait diimplementasikan pada file *skill.c* dan *SaveLoad.c*.

3.3 ADT Mesin Karakter dan Mesin Kata

Struktur dari ADT Mesin Karakter dan Mesin Kata yang dipakai untuk fitur SaveLoad (terdiri dari prosedur *readFile*, *saveFile*, dan *loadFile*) sama dengan ADT yang digunakan saat pembelajaran dan praktikum. Namun, ada sedikit perubahan penulisan pada beberapa fungsi dan prosedur, deklarasi panjang *TabKata*, serta penambahan fungsi *StrToInt*. Pada mesin karakter, terdapat prosedur *START()* yang digunakan untuk membaca konfigurasi file eksternal dan prosedur *ADV()* yang digunakan untuk berpindah ke karakter selanjutnya. Pada mesin kata, terdapat prosedur *STARTKATA()*, *ADVKATA()*, *SalinKata()*, *IgnoreBlank()*, dan fungsi *StrToInt*. ADT ini diimplementasikan sebagai *mesin_kar.h*, *mesin_kar.c*, *mesin_kata.h* dan *mesin_kata.c*.

Alasan ADT Mesin Karakter dan Mesin Kata digunakan pada beberapa implementasi dalam tugas besar ini yaitu terdapat beberapa fitur yang membutuhkan input file konfigurasi eksternal saat memulai game baru (*readFile*) maupun melanjutkan game (*loadFile*). Begitu pun, saat pemain ingin menyimpan state game pada *command save* akan dihasilkan output berupa file konfigurasi eksternal. Proses-proses ini membutuhkan pembacaan dan penyimpanan informasi yang dapat dipenuhi oleh ADT Mesin Karakter dan Mesin Kata.

3.4 ADT Queue

Struktur data ADT Queue digunakan untuk membuat giliran player atau *player turn* di “gameplay.c”. ADT Queue adalah ADT yang mengikuti aturan *First In, First Out* (FIFO), sehingga ADT akan menambah elemen di urutan terakhir dan mengurangi elemen dari urutan pertama. ADT ini diimplementasikan pada gameplay.h dan gameplay.c.

Cara penggunaan dari ADT ini untuk permainan Mobitangga adalah dengan melakukan initializePlayerQueue yaitu menciptakan queue dan memberikan nilai kepada queue sejumlah pemain. Nilai untuk tiap pemain adalah 1 untuk pemain pertama, 2 untuk pemain kedua, 3 untuk pemain ketiga, dan 4 untuk pemain keempat. Sehingga untuk permainan dengan dua pemain akan berisi playerQueue = {1,2} atau permainan dengan tiga pemain akan berisi playerQueue = {1,2,3}.

Contoh: Permainan dimulai dengan dua pemain, maka playerQueue {1,2}. Ketika pemain pertama selesai dan melakukan command End Turn, maka queue akan di delete dan di add dengan bilangan pemain tersebut, sehingga menjadi {1,2} -> {2} -> {2,1}. Sistem tersebut berlangsung sepanjang permainan berlangsung. Ronde akan berganti bila playerQueue kembali ke {1,2}.

3.5 ADT Stack

Struktur data dari ADT stack: infotype dari elemen stack bertipe integer. Struktur stack terdiri atas (infotype) dan TOP (addressAttack). Prototype manajemen memori dari ADT ini memuat fungsi dan prosedur untuk menyalin data player ke dalam list baru. Selanjutnya terdapat prototype representasi logik stack dari ADT ini yang memuat fungsi/prosedur untuk pengecekan stack kosong, membuat stack kosong, menambahkan elemen stack, serta menghapus elemen stack.

ADT Stack adalah list linear yang bekerja dengan akses elemennya mengikuti aturan *Last In, First Out* (LIFO). Persoalan yang diselesaikan adalah fitur undo. Alasan pemilihan adalah ADT stack dapat melakukan push setiap kali ada perubahan game states dan pop ketika dilakukan Undo. Diimplementasikan sebagai ADT stack dengan nama file header “stack.h”.

4 Program Utama

Program utama dimulai dengan file utama main.c yang berfungsi sebagai *driver* program lain dengan memasukkan fungsi-fungsi yang telah dibuat sebelumnya dalam folder terpisah, yaitu gameplay.h, buff.h, inspect.h, map.h, roll.h, skill.h, undo.h, dan SaveLoad.h. Program utama akan menampilkan *string* selamat datang dan *Main Menu* yang berisi pilihan *New Game*, *Load Game*, atau *Exit*. Pemain dapat memilih untuk bermain game dengan memulai game baru, melanjutkan game jika pemain telah bermain game sebelumnya, atau keluar dari game.

Jika pemain memilih angka 1 yaitu *New Game*, permainan akan dimulai dan pemain diminta untuk memasukkan jumlah *player* dengan batasan 2 hingga 4 pemain, nama *player* sebanyak jumlah player, serta nama file konfigurasi dalam format .txt file. Jika pemain memilih

angka 2 yaitu *Load Game*, pemain akan diminta untuk memasukkan nama *save file* dalam format .txt file yang akan digunakan untuk melanjutkan permainan. Jika pemain memilih angka 3 yaitu *Exit*, program permainan akan ditutup dan pemain keluar dari game. Setelah input pada pilihan 1 dan 2 dimasukkan, pemain akan masuk ke dalam *Game Loop*.

Dalam *Game Loop*, program akan menampilkan ronde game, posisi tiap pemain, giliran pemain yang akan mulai bermain game, dan *skill* didapatkan secara *random* yang dimiliki pemain tersebut. Jika pemain memulai game baru, posisi semua pemain berada pada petak 1 dan game akan berada pada ronde 1. Jika pemain melanjutkan game, posisi pemain dan ronde game akan sama dengan posisi pemain dan ronde game yang disimpan dari game yang telah dimainkan sebelumnya. Kemudian, masih dalam *Game Loop*, program menampilkan beberapa *command* yang dapat dipilih pemain. Pemilihan *command* dilakukan dengan memasukkan angka dari 1 hingga 8. Berikut ini beberapa pilihan *command*:

1. *Skill*

Fitur *skill* akan menampilkan beberapa *skill* yang dimiliki pemain sebanyak 1 hingga 10 *skill* dan pemain akan diminta untuk memasukkan angka antara 0, angka positif (1 - 10 sesuai dengan jumlah *skill*), atau angka negatif (-1 - -10 sesuai dengan jumlah *skill*). *Input* 0 berarti pemain keluar dari menu *skill*. *Input* angka positif berarti pemain memilih untuk menggunakan *skill* yang sesuai dengan nomor pilihannya. Sedangkan, *input* angka negatif berarti pemain membuang *skill* yang ada pada nomor tersebut.

2. *Map*

Fitur *map* akan menampilkan peta game dan posisi masing-masing pemain yang dilambangkan menggunakan “*” (bintang). Nomor petak pemain ditampilkan di akhir setiap baris peta.

3. *Buff*

Fitur *buff* akan menampilkan daftar dari *buff* yang dimiliki pemain.

4. *Inspect*

Fitur *inspect* akan menampilkan apakah dalam suatu petak yang dipilih terdapat teleporter atau tidak. Pemain akan diminta untuk memasukkan nomor petak yang dipilih. Kemudian, fitur ini akan menampilkan 3 kemungkinan keadaan petak. Pertama, petak tersebut memiliki teleporter menuju petak lain. Kedua, petak tersebut kosong. Ketiga, petak tersebut merupakan petak terlarang.

5. *Roll*

Fitur *roll* akan digunakan untuk memindahkan posisi pemain dalam gilirannya. Pemain akan mendapatkan nomor dadu secara *random*, baik melibatkan *skill* (jika mempunyai *skill* senter pembesar atau pengecil hoki) maupun tidak. Kemudian, fitur ini akan menampilkan nomor dadu yang didapatkan. Jika nomor dadu yang dipilih merupakan petak terlarang, pemain tidak dapat bergerak. Jika nomor dadu yang dipilih merupakan petak kosong, pemain dapat bergerak maju atau dapat memilih bergerak maju atau mundur (sesuai dengan keadaan petak).

6. *Save*

Fitur *save* akan meminta pemain untuk memasukkan nama *file* yang ingin disimpan. Setelah pemain memasukkan nama *file*, ditampilkan pernyataan bahwa *file* berhasil

disimpan. Fitur ini digunakan pemain untuk menyimpan state permainan dengan syarat dapat digunakan setelah pemain bergerak (memilih roll).

7. *End Turn*

Fitur *end turn* dapat digunakan pemain untuk mengakhiri gilirannya dengan syarat dapat digunakan setelah pemain bergerak (memilih roll).

8. *Undo*

Jika minimal ada 1 pemain yang menyelesaikan ronde X, fitur undo ini dapat digunakan untuk mengulang ronde X dengan mengembalikan state permainan sama seperti akhir ronde X-1.

Setiap kali pemain telah selesai memilih dan menjalankan pilihan nomor pada menu *command*, program terus melakukan proses *looping* yang artinya tampilan pada *Game Loop* akan terus-menerus diulang hingga ada pemain memilih nomor 6 yaitu *Save* atau ada pemain yang telah memenangkan game (mencapai petak maksimum). Kemudian, program akan menampilkan peringkat pemain sesuai dengan lokasi pemain.

5 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1.	Program Main	18219076, 18220049	18219076
2.	ADT Array	18220049	18220049
3.	ADT List Terkait dengan Representasi Fisik Pointer (listlinier skill)	18220025	18220025
4.	ADT Mesin Karakter dan Mesin Kata	18220049	18220049
5.	ADT Queue	18219076	18219076
6.	ADT Stack	18220088	18220088
7.	Fitur Gameplay	18219076, 18220049	18220049, 18219076
8.	Fitur Buff	18220007, 18220025	18220007, 18220025
9.	Fitur End turn	18219076	18219076, 18220007
10.	Fitur Inspect	18220050	18220050
11.	Fitur Map	18220049	18220049
12.	Fitur Roll	18219076	18219076
13.	Fitur SaveLoad	18220049, 18220050	18220049, 18220050
14.	Fitur Skill	18220025	18220025
15.	Fitur Undo	18220088	18220088

6 Lampiran

6.1 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.

Buatlah sebuah permainan berbasis CLI (*command-line interface*). Permainan ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah kalian pelajari di mata kuliah ini. Kalian boleh menggunakan (atau memodifikasi) struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini.

6.2 Notulen Rapat

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






No. Kelompok/Kelas : 2
Nama Kelompok : -
Anggota Kelompok (Nama/NIM) :
1. Ahmad Wafika Samsea/18220088
2. Alya Apriliyanti/18220050
3. Joanna Margareth Nauli/18220007
4. Joshi Ryu Setiady/18220025
5. Yahya Aqrom/18219076
6. Rachmad Hidayat/18220049



Asisten Pembimbing : Jonathan Yudi Gunawan

Asistensi I


Tanggal : 7 November 2021	Catatan Asistensi: Asistensi 3 kali itu harus atau sebutuhnya? Tanggalnya dibebasin. Nanti seharusnya dikasih tau dan diingetin buat asistensi selanjutnya kapan.
Tempat : Google Meet	






STEI- ITB	IF2111 TB 02	Halaman 9 dari 15 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.		


<p>Kehadiran Anggota Kelompok: No NIM Tanda tangan</p> <p>1. Ahmad Wafika Samsea / 18220088</p>  <p>2. Joshi Ryu Setiady / 18220025</p>  <p>3. Joanna Margareth Nauli / 18220007</p>  <p>4. Alya Apriliyanti / 18220050</p>  <p>5. Yahya Aqrom / 18219076</p>  <p>YAHYA AQROM</p>	<p>Kalo mau nanya tentang tubes harus lewat asistensi atau boleh lewat chat? Boleh lewat chat juga.</p> <p>Untuk panjang peta (N), MaxRoll, dan banyak teleporter (M) ditentukan oleh pemain saat NEW GAME (bersamaan jumlah pemain dan nama pemain) ? Atau telah ditentukan? Itu nanti bikin lewat file konfigurasi ada di halaman 12 spesifikasi.</p> <p>Kita lihat yang ngerjain lewat commit? commit tidak terlalu dilihat, tapi lewat peer assessment.</p> <p>Presentasi melalui demonstrasi program dengan asisten</p> <p>Skill mesin waktu dan baling-baling bambu, kalo kita cuma buat salah satu dari bonus tersebut terhitung full bonus gak kak? Terhitung dan paling enak dikerjain save sama load.</p> <p>Save itu berarti text file nya tersimpan ya kak? iya sama file state-nya.</p> <p>“Jika setelah teleport pemain menemukan teleporter lain, maka pemain tidak akan mengalami teleport untuk kedua kalinya.” ini maksudnya kalo ada teleport petak keluar dan teleport petak awal dalam satu tempat gitu kak? Iya betul. Jadi misal ada teleporter 3 ke 5, terus 5 ke 7, berarti kalo dari 3 bakal ke 5, gak akan ke 7.</p> <p>Cara mendapatkan buff itu bagaimana kak? Itu aktif-nya kalo dari skill doang. Skill nggak auto aktif tapi bakal kerja di buff.</p> <p>ADT list menyimpan list pemain? iya. Terus misal ADT ada properti. Kalo ada properti gitu boleh diubah ADT-nya.</p> <p>Mau nanya buff senter pembesar sama pengecil, itu cara dapetin angkanya terserah kita kah atau gimana? cara dapetin angkanya pake roll. Mau nanya tentang INSPECT, bisa digunakan berkali-kali kak? bisa, teleport-nya gak ditunjukkan di map.</p>
---	--

<p>6. Rachmad Hidayat / 18220049</p> 	<p>Bisa beberapa pemain dalam satu petak?</p> <p>“Skill dapat digunakan berulang kali selama pemain belum melakukan roll dadu” ini maksudnya gimana ya kak?</p> <p>Nanti file konfigurasinya pake punya siapa? pake punya kita.</p> <p>Apakah boleh dalam bentuk lain buat tampilannya? Boleh kalo ada ide lain.</p> <p>Kalau permainan sudah berakhir, aksi selanjutnya apa ya kak?</p> <p>Di spesifikasi gak ada sih, jadi dibebasin aja mau keluar atau melihat yang lain.</p>
	<p>Tanda Tangan Asisten:</p>  <p>Jonathan Yudi Gunawan</p>

Asistensi II

Tanggal : 20 November 2021	<p>Catatan Asistensi:</p> <p>Progress: Udah bikin file main, gameplay, sama beberapa command yaitu roll dan endturn. Sisanya masih dalam tahap proses.</p> <p>Pertanyaan: Nanti programnya bisa dipake buat konfigurasi, atau dari konfigurasi yang kita buat ajaa? Bisa dari beberapa konfigurasi asal nanti programnya bisa jalan.</p> <p>Yang masih bingung itu, map dibikin beberapa map atau bagaimana kak? Aku belum tau nanti teknis nya gimana, tapi harus bikin program yang menyesuaikan file konfigurasinya gitu.</p>
Tempat : Google Meet	
<p>Kehadiran Anggota Kelompok:</p> <p>No NIM Tanda tangan</p> <p>1. Ahmad Wafika Samsea / 18220088</p>  <p>2. Joshi Ryu Setiady / 18220025</p>	

	<p>ADT harus dibuat driver nya ya kak? kalo di spesifikasi yang dibaca sih diwajibin kak. Jadi ada 4 ADT yang diwajibin ada, terus dibikin ADT-nya.</p>
<p>3. Joanna Margareth Nauli / 18220007</p>	<p>Driver nya itu sesuai sama per fungsi ADT-nya kak? Iya, per-fungsi aja skenarionya. Gak perlu panjang tapi buat nge-tes fungsi-nya aja jalan atau nggak..</p>
	<p>Case buat command kan sama dengan angka. Bisa dibandingin ga yaa sama di atasnya? Pake if itu switch nya.</p>
<p>4. Alya Apriliyanti / 18220050</p>	<p>Buat yang skill, di awal turn ada gatcha skill. Itu pake apa yaa? Pake fungsi random, nanti hasilnya dari 0-1.</p>
	<p>Mesin karakter sama kata itu buat baca file konfigurasi yaa kak? iya, betul.</p>
<p>5. Yahya Aqrom / 18219076</p>	<p>Kalo save sama load, kan nge-save state dari game. Berarti kaya nge-save terus menghasilkan file. Nanti file itu ada di dalam konfigurasi yang sama. Atau beda lagi? Dibuat di file yang beda.</p>
 <p>YAHYA AQROM</p>	<p>Buat undo, kira-kira pake ADT apa? pake ADT stack sih kak.</p> <p>Itu menghubungkan bagian undo harusnya kaya gimana kak? Betul pake stack, terus dimodif pake integer ganti pake struct, terus pake kondisinya giliran siapa sekarang (state). Undo lumayan juga perlu ubah-ubah ADT.</p>
<p>6. Rachmad Hidayat / 18220049</p> 	<p>Rencana yang map pake array of tuple, satunya pake * #, satu nya lagi pake teleportasi, itu bisa kak? Bisa. Tapi tujuan teleporter nya kalo pake tuple ga bisa.</p> <p>Teleport itu gimana? Jadi nanti diisi bilangan yang jadi teleportasinya.</p> <p>Cara pengerjaannya kan tergantung tiap-tiap kelompok yaa. Kalo misal pake cara lain gapapa yaa kak? gapapaa.</p>

	<p>Kalo menurut kakak, buat buff itu dibanding list, ada cara yang lebih simpel ga yaa? Kalo mau simpel sih pake boolean.</p> <p>Aku kepikiran array of boolean gitu karena sama player. gimana kalo kaya gitu kak? iyaa gitu boleh, yang penting bisa pake boolean.</p> <p>Buat command inspect gimana yaa kak? Kalo mau membaca file konfigurasinya dari ADT array. Kalo mau mengecek ada inspect ngga ambil dari map. Map juga dari ADT array.</p> <p>Saya sekarang pisahin foldernya jadi ADT sama FUNCTION boleh ga yaa? Beda file aja kaya praktikum. Terus boleh kalo mau beda folder.</p>
	<p>Tanda Tangan Asisten:</p>  <p>Jonathan Yudi Gunawan</p>

6.3 Log Activity Anggota Kelompok

NIM	Nama	Log Act
18219076	Yahya Aqrom	<p>06/11/2021 : Diskusi kelompok I (Meet)</p> <p>07/11/2021 : Asistensi I</p> <p>16/11/2021 : Membuat gameplay, Membuat linuxPlay dan winPlay.exe, Membuat fungsi Roll dan menambahkan ADT renamedqueue.</p> <p>20/11/2021 : Membuat panduan gameplay, Asistensi II</p> <p>22/11/2021 : Memperbaiki roll, fix error</p> <p>25/11/2021 : Memperbaiki welcome game dan teleporter roll</p>

		27/11/2021 : Memperbaiki bug roll dan gameplay, Diskusi kelompok II (Meet) 28/11/2021 : Memperbaiki bug, menambahkan driver, mengerjakan laporan, dan meet untuk finalisasi
18220007	Joanna Margareth Nauli	06/11/2021 : Diskusi kelompok I (Meet) 07/11/2021 : Asistensi I 20/11/2021 : Asistensi II 25/11/2021 : Mengerjakan fitur buff 27/11/2021 : Memperbaiki dan menambahkan fitur buff, diskusi kelompok II (Meet) 28/11/2021 : Memperbaiki dan finalisasi fitur buff, merge ke branch skill, mengerjakan laporan, dan meet untuk finalisasi
18220025	Joshi Ryu Setiady	05/11/2021 : Membuat repository github 06/11/2021 : Diskusi kelompok I (Meet) 07/11/2021 : Asistensi I 20/11/2021 : Asistensi II 23/11/2021 : Mengerjakan fitur skill 25/11/2021 : Memperbaiki error pada fitur skill 27/11/2021 : Mengerjakan implementasi fitur skill, Diskusi kelompok II (Meet) 28/11/2021 : Memperbaiki implementasi fitur skill, memperbaiki skill dan buff, mengerjakan driver list linier, mengerjakan laporan, meet untuk finalisasi
18220049	Rachmad Hidayat	06/11/2021 : Diskusi kelompok I (Meet) 07/11/2021 : Asistensi I 20/11/2021 : Asistensi II 22/11/2021 : Menambahkan ADT Array dan fungsi Map 23/11/2021 : Memperbaiki Map dan fungsi readFile 25/11/2021 : Memperbaiki bug Map dan mengupdate readFile, Menambahkan main menu di gameplay 26/11/2021 : Menambahkan fitur save 27/11/2021 : Diskusi kelompok II (Meet), membuat rank player, memperbaiki gameplay 28/11/2021 : fix gameplay, fungsi save load, mengerjakan laporan, dan meet untuk finalisasi
18220050	Alya Apriliyanti	06/11/2021 : Diskusi kelompok I (Meet) 07/11/2021 : Asistensi I, menulis form asistensi

		19/11/2021 : Mengerjakan fitur inspect 20/11/2021 : Asistensi II, menulis form asistensi 25/11/2021 : Menggabungkan fitur inspect ke main 27/11/2021 : Diskusi kelompok II (Meet), membuat load file 28/11/2021 : memperbaiki bug load file, mengerjakan laporan, meet untuk finalisasi
18220088	Ahmad Wafika Samsea	06/11/2021 : Diskusi kelompok I (Meet) 07/11/2021 : Asistensi I 18/11/2021 : Mengerjakan fitur undo 20/11/2021 : Asistensi II 23/11/2021 : Memperbaiki fitur undo 27/11/2021 : Diskusi kelompok II (Meet) 28/11/2021 : Meet untuk finalisasi, mengerjakan laporan