

# LAPORAN TUGAS BESAR

## IF2111/Algoritma dan Struktur Data

### Mobitangga


Dipersiapkan oleh:

#### Kelompok 5

18220060	Laksamana Vixell Tanjaya H.
18220092	Thariq Zhafran Satyagraha
18220090	Rofif Fairuz Hawary
18220086	Aldi Fadlian Sunan
18220091	Rachita Caronica Jonur
18217035	Joe Putera

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-05</i>		<i>21</i>
		<i>Revisi</i>	-	<i>28/11/2021</i>

# Daftar Isi

Ringkasan	2
<b>Penjelasan Tambahan Spesifikasi Tugas</b>	<b>4</b>
Mekanisme Save dan Load	4
Mekanisme Jumlah Pemain	4
Mekanisme Skill Tambahan	4
Mekanisme Leaderboard	4
<b>Struktur Data (ADT)</b>	<b>5</b>
ADT Map	5
ADT Mesin Karakter dan Mesin Kata	5
ADT Player	6
ADT List	6
ADT Round	7
<b>Program Utama</b>	<b>8</b>
<b>Algoritma-Algoritma Menarik</b>	<b>9</b>
Mekanisme Penyimpanan Informasi pada Player	9
Mekanisme Round	10
<b>Pembagian Kerja dalam Kelompok</b>	<b>11</b>
<b>Lampiran</b>	<b>11</b>
Deskripsi Tugas Besar	11
Notulen Rapat	12
Rapat 1 3 November 2021 - Zoom	12
Rapat 2 5 November 2021 - Discord	12
Rapat 3 17 November 2021 - Discord	12
Log Activity Anggota Kelompok	14
Form Asistensi	18

# 1 Ringkasan

Pada tugas besar kali ini, kami diberi tantangan untuk mampu membuat sebuah permainan yang dapat dimainkan oleh dua hingga empat orang pemain dengan mengambil jalan cerita yang telah diberikan pada spesifikasi tugas besar. Selain itu, terdapat spesifikasi serta batasan-batasan tentang hal-hal yang tidak boleh dilakukan atau harus dilakukan sehingga membuat tugas ini semakin menantang. Permainan ini dimulai dengan gerakan oleh pemain pertama dengan menggunakan perintah-perintah diberikan dalam game. Setelah pemain pertama selesai menggunakan gilirannya, permainan dilanjutkan dengan gerakan dari pemain kedua. Permainan akan berakhir apabila salah satu pemain berhasil sampai ke garis finish yang terletak pada peta. Pada permainan ini, ada beberapa ADT yang digunakan, yaitu:

1. ADT Map
2. ADT Mesin Karakter
3. ADT Mesin Kata
4. ADT Player
5. ADT List
6. ADT Round

Penjelasan mengenai seluruh ADT yang kami pakai ini terdapat pada BAB III Struktur Data.

Laporan ini garis besarnya menjelaskan mengenai program yang telah kami rancang sedemikian sehingga telah menjadi permainan yang bisa membuat hati pemainnya senang. Bagian pertama laporan berisi tentang ringkasan dari laporan ini serta deskripsi dan penjelasan singkat mengenai program yang telah kami buat. Untuk bagian kedua terdiri dari berbagai struktur data yang dibuat dan digunakan. Bagian ketiga berisi tentang penjelasan mengenai program utama dan algoritma menarik yang digunakan dalam program tersebut. Bagian keempat terdiri dari *data test* dan *test script* dari program yang telah kami buat. Sedangkan bagian terakhir berisi tentang pembagian kerja dalam kelompok beserta lampiran - lampiran.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Mekanisme Save dan Load

Kami membuat sebuah fitur yang dapat digunakan untuk menyimpan atau pun memuat data permainan yang telah disimpan sebelumnya. Informasi yang akan disimpan dalam bentuk format .txt. Lalu jika player ingin melanjutkan permainan, player dapat melakukan load. Pada tahap ini, informasi akan dibaca dari file.txt yang disimpan dengan menggunakan mesin kata dan mesin karakter. Load game hanya akan mengambil data config player (skill, buff, condition) dan map. Stack tidak akan di-load dan save. Seperti halnya mengedit file di Word jika file masih terbuka, bisa undo terus menerus, akan tetapi jika sudah save and exit file, lalu load kembali, program tidak akan bisa undo sebelum pengguna melakukan minimal satu perubahan.

### 2.2 Mekanisme Jumlah Pemain

Pada tugas besar ini, jumlah maksimal pemain yang dapat bermain adalah empat orang. Fitur tersebut diimplementasikan melalui prosedur summonPlayer pada ADT Player. Nama serta informasi lainnya terkait player akan disimpan dalam bentuk array yang juga terhubung dengan ADT lain seperti skill dalam bentuk list untuk setiap player.

### 2.3 Mekanisme Skill Tambahan

Skill yang tambahan yang berhasil kami buat pada tugas besar ini, yaitu Baling Baling Jambu dan Mesin Waktu. Untuk Baling Baling, jika hasil efek Baling Baling Jambu menghasilkan posisi melebihi panjang Map, pemain tidak akan bisa berpindah tempat. Sedangkan untuk Mesin Waktu, Jika hasil efek Mesin Waktu menghasilkan posisi kurang dari 1, pemain yang dipindahkan akan mendapatkan posisi akhir 1. Skill-skill aktif tersebut hanya bisa memberikan efek tujuan ke player lain (tidak bisa menginput username diri sendiri). Selain itu, penjelasan tambahan untuk skill lain yang tersedia, yaitu Cermin Pengganda, jika setelah mengaktifkan Skill Cermin Ganda pemain masih memiliki skill diatas delapan buah, pemain akan diminta untuk menghapus skill hingga list skill menjadi delapan buah, kemudian player akan diberikan dua buah random skill. Lalu, ketika Round game berganti, setiap player yang akan melakukan turn, akan diberikan 1 skill random sesuai dengan chance yang ada di spesifikasi.

### 2.4 Mekanisme Leaderboard

Fitur Leaderboard dibuat untuk menunjukkan urutan pemenang pada permainan. Urutan tersebut diawali dengan player dari yang mencapai petak terakhir terlebih dahulu, dilanjutkan dengan player lain terurut menurun berdasarkan posisinya.

### 3 Struktur Data (ADT)

Pada program ini, terdapat beberapa struktur data (ADT) yang kami gunakan sejumlah 6 buah ADT. Berikut adalah daftar dari ADT yang kami gunakan dalam program ini.

#### 3.1 ADT Map

ADT Map merupakan implementasi dari ADT Array yang digunakan dalam program ini. ADT ini berfungsi untuk implementasi map pada game, serta menentukan lokasi-lokasi yang ada pada map, seperti lokasi yang dapat dikunjungi atau lokasi terlarang dengan petak bertandakan #. ADT ini juga diimplementasikan sebagai penanda adanya teleport pada beberapa petak di map yang ditandai dengan indeks.

Sketsa pada ADT ini terdiri dari

- void inputConfig()  
Digunakan untuk menerima input berupa file konfigurasi yang digunakan
- void readConfig()  
Digunakan untuk membaca masukan file konfigurasi
- void assignConfig()  
Digunakan untuk memasukkan input bacaan dari file konfigurasi ke dalam variabel pointer yang digunakan selama permainan
- void printConfig()  
Digunakan untuk menampilkan isi file konfigurasi, berupa detail map atau peta, pada layar
- void showMap(char array[], int pPos)  
Digunakan untuk menampilkan, tampilan map pada setiap user ke layar
- void charToInt()  
Digunakan Mengubah char ke int untuk input yang merupakan integer seharusnya.
- int rollDice(int max)  
Digunakan untuk menghasilkan value yang digunakan untuk keberjalanan permainan
- void movePlayer(int roll, int idxCurrentPlayer, Map CurrentMap)  
Digunakan untuk memindahkan posisi player berdasarkan value dari roll dadu
- void inspectMap(Map CurrentMap)  
Digunakan untuk menampilkan detail tiap petak pada map
- void teleport (pPosition pP, pUserName pU, char \*uname)  
Digunakan untuk berpindah jika player berada pada petak dengan teleport

#### 3.2 ADT Mesin Karakter dan Mesin Kata

ADT Mesin Karakter serta ADT Mesin Kata merupakan ADT yang digunakan untuk membaca teks ataupun masukan yang diberikan oleh user dalam menjalankan program. ADT ini digunakan untuk membaca file konfigurasi pada eksternal program dan membaca informasi yang digunakan saat proses save dan load.

Sketsa pada ADT ini terdiri dari

- void START()  
Digunakan untuk memulai pembacaan pita karakter
- void ADV()  
Digunakan untuk melanjutkan pembacaan pita karakter hingga menemukan MARK

- void IgnoreBlank()  
Digunakan untuk mengabaikan satu atau beberapa BLANK
- void SalinKata()  
Digunakan untuk mengakuisisi kata, menyimpan dalam CKata
- void STARTKATA()  
Digunakan untuk memulai pembacaan kata
- void ADVKATA()  
Digunakan untuk melanjutkan pembacaan kata hingga menemukan MARK

### 3.3 ADT Player

ADT Player merupakan fungsi utama yang menyimpan semua informasi terkait player, mulai dari posisi, nama player, buff, dan skill yang dimiliki oleh masing-masing player. Skill yang dimiliki oleh tiap player disimpan pada ADT ini merupakan implementasi dari ADT List. Selain itu, ADT ini juga menyimpan informasi terkait posisi player yang terhubung dengan map dalam bentuk array. Posisi dari tiap player disimpan dalam bentuk index selama permainan berlangsung.

Sketsa pada ADT ini terdiri dari

- void createEmptyPlayerList (pUserName \*pU);  
Digunakan untuk membuat sebuah array kosong untuk player
- void summonPlayer(pUserName \*pU, pIsTeleported \*pT, pPosition \*pP, pIsImune \*pI, pIsCermin \*pC, pIsSenterBesar \*pSB, pIsSenterKecil \*pSK, int n);  
Digunakan untuk memasukkan kondisi pemain sebanyak n pemain
- int getIdxOfPlayer (pUserName pU, char \*name);  
Digunakan untuk mendapatkan informasi indeks dari player.
- boolean getTeleportedConditionOfPlayer (pIsTeleported pT, int idx);  
Digunakan untuk mendapatkan informasi apakah player pernah terkena teleport
- boolean getImmunityConditionOfPlayer (pIsImune pI, int idx);  
Digunakan untuk mendapatkan informasi apakah player memiliki buff imunitas anti teleport (pintu ga kemana-mana)
- boolean getCerminConditionOfPlayer (pIsCermin pC, int idx);  
Digunakan untuk mendapatkan informasi apakah player memiliki buff cermin ganda
- boolean getSenterBesarConditionOfPlayer (pIsSenterBesar pSB, int idx);  
Digunakan untuk mendapatkan informasi apakah player memiliki buff senter pembesar
- boolean getSenterKecilConditionOfPlayer (pIsSenterKecil pSK, int idx);  
Digunakan untuk mendapatkan informasi apakah player memiliki buff senter pengecil
- int getPositionOfPlayer (pPosition pP, int idx);  
Digunakan untuk mendapatkan informasi posisi player

### 3.4 ADT List

Pada program, ADT List digunakan untuk menyimpan skill pemain. Pada implementasi pada program, ADT List skill sudah dimasukkan menjadi satu dengan ADT Player untuk menghubungkan masing-masing skill yang diterima tiap player. Hal ini dirasa efisien dalam

penggunaan tiap skill yang berbeda antar player pada game. Masing - masing player bisa menyimpan hingga maksimal 10 skill. Bila player sudah memiliki 10 skill maka player tidak menerima skill baru, kecuali ada skill lama yang ingin dibuang.

Sketsa pada ADT ini terdiri dari

- `void createEmptyPlayerSkillsList (Skill *S);`  
Digunakan untuk membuat linked list kosong yang dapat diakses oleh array
- `address newSkillNode ();`  
Digunakan untuk mereturn sebuah address dari lsNode (list skill node)
- `void preparationSkillList (Skill *pS1, Skill *pS2, Skill *pS3, Skill *pS4, int n);`  
Digunakan untuk membuat semua list player kosong
- `void printSkill (Skill pS1, Skill pS2, Skill pS3, Skill pS4, int idxP, int *nSkill);`  
Digunakan untuk menampilkan skill player pada layar
- `int banyaknyaSkill (Skill pS1, Skill pS2, Skill pS3, Skill pS4, int idxP);`  
Digunakan untuk menghitung skill yang dimiliki player)
- `int returnIdxSkill (Skill pS1, Skill pS2, Skill pS3, Skill pS4, int idxP, char *skName);`  
Digunakan untuk menampilkan nilai indeks dari skill pada player
- `boolean isSkillExists (Skill pS1, Skill pS2, Skill pS3, Skill pS4, int idxP, char *skName);`  
Digunakan untuk mendapat informasi apakah player memiliki skill tertentu
- `void deleteSkill (Skill *pS1, Skill *pS2, Skill *pS3, Skill *pS4, int idxP, int idxS);`  
Digunakan untuk menghapus skill dari list
- `void insertVSkill (Skill *pS, char *skname);`  
Digunakan untuk menambah skill pada list
- `void randomSkillGenerator(char *skname);`  
Digunakan untuk mendapatkan skill secara acak
- `int randomInt99();`  
Digunakan untuk mendapatkan angka random yang digunakan untuk mendapat skill secara acak
- `void copySkillName (Skill pS1, Skill pS2, Skill pS3, Skill pS4, int idxP, int idxSkill, char *skname);`  
Digunakan untuk menyalin informasi skill player

### 3.5 ADT Round

Pada program, ADT Round merupakan implementasi dari ADT Stack yang digunakan untuk menyimpan dalam bentuk stack suatu keadaan/round dari permainan. Ketika pengguna memasukkan perintah Undo, program akan mengambil round yang tersimpan paling atas pada Stack lalu mengubah semua kondisi permainan menjadi kondisi round sebelumnya.

Sketsa pada ADT ini terdiri dari

- `void createEmptyRound (Round *R);`  
Digunakan untuk membuat sebuah stack kosong
- `addressRound newRoundNode ();`

- Digunakan untuk me-return sebuah address yang digunakan di dalam stack
- `void pushRound (Round *R, pPosition pP, pIsTeleported pT, pIsImune pI, pIsCermin pC, pIsSenterBesar pSB, pIsSenterKecil pSK, Skill pS1, Skill pS2, Skill pS3, Skill pS4, int nPlayer);`  
Digunakan untuk menambah elemen stack yang disimpan untuk setiap round
- `int lenArrSkill (addressRound P, int idxP);`  
Digunakan untuk mendapatkan informasi banyaknya skill pada player yang akan disimpan dalam stack
- `void undoRound (Round *R, pPosition *pP, pIsTeleported *pT, pIsImune *pI, pIsCermin *pC, pIsSenterBesar *pSB, pIsSenterKecil *pSK, Skill *pS1, Skill *pS2, Skill *pS3, Skill *pS4, int nPlayer);`  
Digunakan untuk menghapus stack atau round terakhir dan mengembalikan ke round sebelumnya

## 4 Program Utama

Program utama memiliki alur yang diawali dengan memasuki loop pertama yang berisi main menu dan akan meminta input dari user apakah ingin *new game* / *load game* / *exit* / *help*. Jika user memasukkan *new game*, program akan memulai dari awal menentukan jumlah *player* dan memilih file konfigurasi yang digunakan sebagai map baru serta menentukan banyaknya *player* yang akan bermain baru kemudian masuk ke loop yang kedua. Sedangkan jika user memasukkan *load game* maka banyak pemainnya diambil dari config baru setelahnya masuk ke loop yang kedua.

Loop yang kedua merupakan *flow* dari pada *game* Mobitangga. Program akan mengeluarkan *output default* yang digunakan untuk meminta *input command* daripada user *command* apa saja yang valid bisa diakses dengan menginput *help*. Kemudian program akan menyesuaikan sesuai dengan *command* yang diberikan oleh *user*. Selain itu setiap roundnya pemain akan mendapatkan skill yang bisa digunakan secara langsung atau bisa disimpan terlebih dahulu. Jika salah satu *player* sudah mencapai garis *finish*, game akan berhenti dan akan mengeluarkan pesan kemenangan beserta leaderboard atau urutan pemenang pada game.



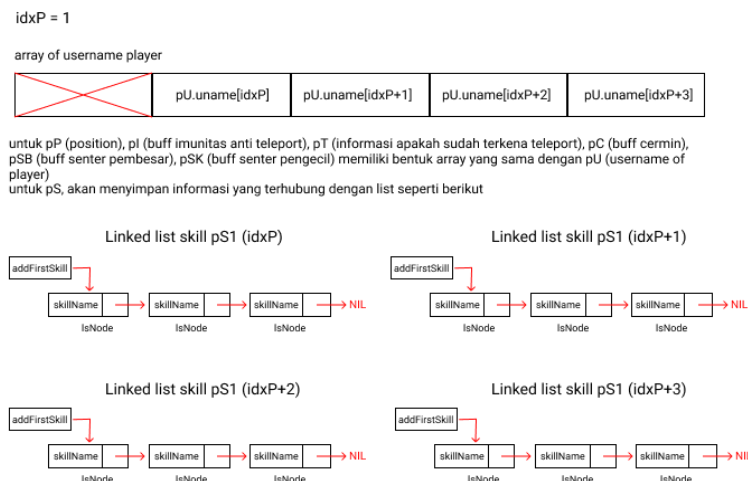
## 5 Algoritma-Algoritma Menarik

### 5.1 Mekanisme Penyimpanan Informasi pada Player

Algoritma ini akan menghubungkan player dengan beberapa kondisi yang digunakan dalam permainan. Berikut merupakan algoritma yang dimaksud

```
void summonPlayer (pUserName *pU, pIsTeleported *pT, pPosition *pP,  
pIsImune *pI, pIsCermin *pC, pIsSenterBesar *pSB, pIsSenterKecil  
*pSK, int n) {  
    for (int i = 1 ; i <= n ; i++) {  
        printf("Masukkan nama pemain ke-%d: ", i);  
        scanf("%s", ((*pU).uname[i]));  
        (*pT).isTele[i] = FALSE;  
        (*pP).pos[i] = 1;  
        (*pI).isImun[i] = FALSE;  
        (*pC).isCermin[i] = FALSE;  
        (*pSB).isSenterBesar[i] = FALSE;  
        (*pSK).isSenterKecil[i] = FALSE;  
    }  
    (*pU).Neff = n;  
}
```

Algoritma yang digunakan pada player ini menarik karena dengan adanya alokasi pada ADT player, sehingga dengan kondisi-kondisi yang terjadi pada player akan terhubung ke dalam index player dan saling berkesinambungan, mulai dari skill, buff, hingga posisi. Sedangkan untuk skill dibuat dalam ADT List yang informasi list tersebut juga ada pada array. Hal tersebut memudahkan dalam pembuatan algoritma yang lainnya. Sebagai gambaran dapat dilihat pada gambar berikut:



Gambar 5.1 Ilustrasi pada ADT Player

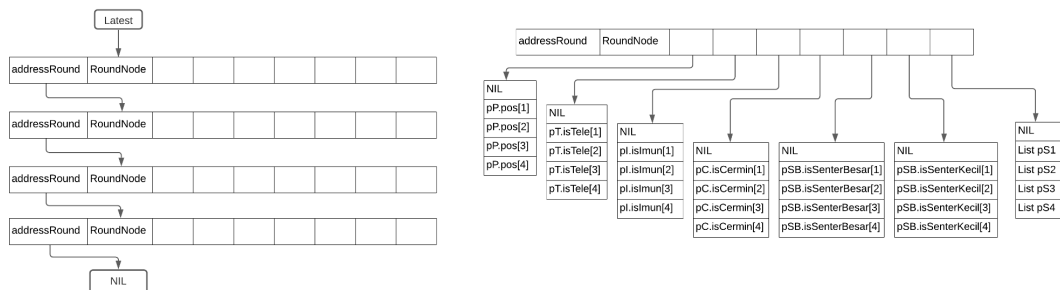
## 5.2 Mekanisme Round

Algoritma ini akan memasukkan state permainan (kondisi player) ke dalam stack setelah sebuah round selesai. Dengan adanya prosedur ini, implementasi dari UNDO dapat diterapkan.

```
void pushRound (Round *R, pPosition pP, pIsTeleported pT, pIsImune pI,
pIsCermin pC, pIsSenterBesar pSB, pIsSenterKecil pSK, Skill pS1, Skill
pS2, Skill pS3, Skill pS4, int nPlayer) {

    int nSkill; char skname[30]; addressRound P = newRoundNode();
    if (LATESTROUND(*R) != Nil) {
        NEXTROUND(P) = LATESTROUND(*R);
        LATESTROUND(*R) = P;
    } else {
        LATESTROUND(*R) = P;
    }
    for (int i = 1; i <= nPlayer; i++) {
        (POSITIONROUND(P))[i] = getPositionOfPlayer(pP, i);
        (TELEPORTEDROUND(P))[i] = getTeleportedConditionOfPlayer(pT, i);
        (IMUNITYROUND(P))[i] = getImmunityConditionOfPlayer(pI, i);
        (CERMINROUND(P))[i] = getCerminConditionOfPlayer(pC, i);
        (SENDERBROUND(P))[i] = getSenderBesarConditionOfPlayer(pSB, i);
        (SETERKROUND(P))[i] = getSenderKecilConditionOfPlayer(pSK, i);
        nSkill = banyaknyaSkill(pS1, pS2, pS3, pS4, i);
        for (int j = 1; j <= nSkill; j++) {
            strcpy(skname, "");
            copySkillName(pS1,pS2,pS3,pS4,i,j,skname);
            strcpy((SKILLROUND(P))[i][j], skname);
        }
    }
}
```

Algoritma ini menarik karena menerapkan konsep multilist. Informasi mengenai posisi dan buff player disimpan ke dalam array, sedangkan informasi mengenai skill tiap player dimasukkan ke dalam ADT Linked List. Kemudian, semua informasi tersebut disimpan ke dalam stack yang menggunakan multilist. Berikut gambaran dari ADT ini, dengan Latest sebagai Top dari stack:



**Gambar 5.2 Ilustrasi pada ADT Round**

## 6 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1	Main Program	18220060, 18220086, 18220090, 18220091, 18220092	18220060, 18220086, 18220090, 18220091, 18220092, 18217035
2	ADT Player & List (Info Player, Skill, Buff)	18220090	18220090
3	ADT Round	18220090	18220090, 18220086
4	ADT Map	18220092	18220086
5	ADT Mesin Kata dan Mesin Karakter	18220092, 18220091	18220092, 18220091
6	Fitur Skill	18220090	18220090, 18220060, 18220086
7	Fitur Map	18220092	18220092
8	Fitur Inspect Map	18220092	18220092, 18220090
9	Fitur Buff	18220090	18220090, 18220086
10	Fitur Undo	18220090	18220090, 18220086
11	Fitur Roll dengan efek Buff player	18220090, 18220092	18220090, 18220092, 18220086
12	Fitur New Game	18220090	18220090, 18220060, 18220086, 18220092
13	Navigation UI	18220060	18220092, 18220086, 18220060, 18220090
14	Fitur Teleport	18220086, 18220092	18220060, 18220086, 18220090, 18220092
15	Fitur Leaderboard	18220086	18220092, 18220086, 18220060, 18220090
16	Fitur Save	18220091	18220091, 18220092, 18220090
17	Fitur Load	18220091	18220091, 18220090
18	Error Handling Main Program	18220092	18220092
19	Char to Integer	18220092	18220092
20	Notulen	18220086, 18220091	
21	Laporan	18220086, 18220060, 18217035, 18220091	

## 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.

## 7.2 Notulen Rapat

### 7.2.1 Rapat 1 3 November 2021 - Zoom

Diskusi dan review mengenai spesifikasi tugas besar secara keseluruhan

### 7.2.2 Rapat 2 5 November 2021 - Discord

1. Rencana mau dibuat ADT tambahan yaitu ADT player, untuk menyimpan beberapa informasi pada tiap player, jadi dapat terhubung. Idenya antara menggunakan array atau linked list
2. Pembuatan buff bisa menggunakan boolean sebagai penanda kondisi untuk player tersebut
3. Mulai eksplorasi dan pembagian tugas untuk tiap anggota

### 7.2.3 Rapat 3 17 November 2021 - Discord

Penyampaian progress report dari tiap anggota

1. Thoriq
  - ADT array rencana mirip seperti praktikum, belum dipake karena lupa pakai dan belum dibutuhkan, char to int, itu karakter ke integer karena menginput dari map.txt berupa karakter jadi kita butuh mengubah karakter yang angka jadi angka beneran. Karena kalo pake mesin kata berubah jadi karakter, nah karakter itu harus diubah ke integer dengan cara ngedetect. Selain input line 2, karakternya diubah jadi integer, (pake ASCII → ketika suatu angka dikurang 0 dia bakal jadi integer)
  - Mesin karakter sama kayak di praktikum, sebenarnya tidak tau isi laporannya gimana, tapi kalo bingung bisa mengingat dari praktikum atau nonton materi praktikum kalo buat ngejelasinnya.
  - Mesin kata baru buat map. Intinya, kalo ketemu sama new line, kalo gaada apa-apa jadi new line. Kalo ada spasi, tetap ada spasi di arraynya. Trus di char to int, kalo ada spasi tuh kedetect ada angka baru. Spasi sebagai indikasi dan.
  - Map, butuh lima data (panjang peta, layout, maksimal role, jumlah teleport, teleportnya). Hasil dari map dibagi 4. Tiap dua angka dijadikan teleport dari dan ke

## 2. Rofif

- Di player lagi dibuat aci save n load ke txt, di player ada 4 txt, dipisah gitu tiap player, walaupun yg main < 4, itu ga masalah.
- Bikin komen-komen dan skemanya, biar kebayang jalan mainnya bakal kemana aja. Karena kan bakal banyak bagian di main yang diisi, karena nanti yang mau ngerjain di main, tinggal dikasih komentar biar ga tabrakan. Sampe tanggal 21 ditargetin udah ada gambarannya, jadi kalo mau ngerjain sub bagiannya, uda bisa dikerjain.
- Di adt player, intinya semua informasi tentang player disimpan di sini. Jadi position, username, kondisi buat teleport imun skill dll. Pake list buat nyimpen skill-skillnya, jadi pas awal main, program nyiapin 4 list buat player. Tp kalopun ga nyampe 4 pemain gapapa. Sisanya pake array, yang ada Neff cuma di username tp sebenarnya gausah karena nanti bisa diapus. Jadi setiap array dan setiap list jumlahnya 4. Jumlah yang nanti terpakai tuh bergantung sama jumlah pemain. Karena kalo mau ngeakses pemain kedua, tinggal akses indeksnya aja.
- Ada beberapa fungsi juga yang bisa diliat.
- Kepikiran buat confignya tuh, misalnya txt player 1 nanti bakal nyimpen username t1, trus kondisi position terakhir lagi dimana, trus line ketiga lagi nunjukkin imun atau engga, trus line selanjutnya buat nunjukkinnya skillnya gitu.

## 3. Rachita

- Codingannya masih on progress.
- Nyimpen semua info playernya di file txt, trus kayanya bakal mau bikin adt list karena kayanya butuh terpisah, biar lebih enak terpisah antara save and load
- Baru bikin yang buat ngebaca skillnya, nanti kayanya dikasih ke rofif

## 4. Aldi

- Teleport sudah jadi tapi masih ada bug
- Mekanisme si teleportnya sudah sesuai di spesifikasi kalo misalnya kena teleport nanti akan keluar di petak lain
- Player hanya bisa terkena teleport sekali
- Misalnya udah kena teleport, nanti bakal di cek punya imun atau ngga, nah ada bug di posisi ini
- Belum disamain sama thoriq (bagian petak mapnya)
- Ada bug lagi ldxP ga muncul-muncul, entah kayanya karena belum disamain di ke map

## 5. Vixell

- Awalnya udah dicoba buat main nya sama interfacenya, kayak selamat datangnya gitu.
- Udah dibikin design main di word, baru beberapa yang udah di code
- Rancangan beberapa commandnya juga uda dibuat di word
- Bingung juga kalo ngide, takut buat bingung yg lain buat masukin-masukinnya

6. Joe

- Template laporan udah ada di grup
- Bagian 3 sama 4 masih aman kayanya bisa dikelarin, yg lainnya bisa agak nantian

### 7.3 Log Activity Anggota Kelompok

No	Tanggal	NIM	Nama	Aktivitas
1	31/10/2021	18220060	Laksamana Vixell	Membuat Group Line
2	02/11/2021	18220092	Thariq Zhafran	Membuat repository dan mengumpulkan username anggota di github
3	03/11/2021	18220060 18220086 18220090 18220091 18220092 18217035	Laksamana Vixell Aldi Fadlian Sunan Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran Joe Putera	Rapat untuk mereview spesifikasi tugas
4	03/11/2021	18220090	Rofif Fairuz Hawary	Membuat prototype konsep player
5	05/11/2021	18220090	Rofif Fairuz Hawary	Merombak ulang konsep ADT player (condition player, skill, buff, dan lainnya)
6	05/11/2021	18220060 18220086 18220090 18220091 18220092 18217035	Laksamana Vixell Aldi Fadlian Sunan Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran Joe Putera	Rapat untuk membagi tugas
7	05/11/2021	18220060	Laksamana Vixell	Membuat Channel discord
8	06/11/2021	18220060 18220086 18220090 18220091 18220092 18217035	Laksamana Vixell Aldi Fadlian Sunan Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran Joe Putera	Asistensi 1
9	06/11/2021	18220090	Rofif Fairuz Hawary	Mengimplementasi fungsi dan prosedur beberapa spesifikasi ADT player yang ada di player.h di player.c
10	06/11/2021	18220086	Aldi Fadlian Sunan	Pembuatan Google Drive untuk dokumen
11	07/11/2021	18220092	Thariq Zhafran	Membuat read config file
12	08/11/2021	18220092	Thariq Zhafran	Bikin ADT Map

13	10/11/2021	18220092	Thariq Zhafran	Mengimplentasi header ADT Map di map.c
14	11/11/2021	18220090	Rofif Fairuz Hawary	Membuat beberapa fungsi prosedur baru yang diperlukan di ADT player
15	12/11/2021	18220090 18220092	Rofif Fairuz Hawary Thariq Zhafran	Rapat membahas hubungan ADT map dengan ADT player
16	12/11/2021	18220092	Thariq Zhafran	Membuat drivertest untuk ADT Map
17	13/11/2021	18220086	Aldi Fadlian Sunan	Pembuatan Fitur Teleport
18	13/11/2021	18220092	Thariq Zhafran	Membuat ADT mesin kata, mesin karakter
19	14/11/2021	18220060	Laksamana Vixell	Pembuatan Navigation UI
20	14/11/2021	18220090	Rofif Fairuz Hawary	Menyempurnakan Linked List Player yang dihubungkan dengan array-array yang berisi kondisi pemain. Mencoba membuat driver dari ADT playernya
21	14/11/2021	18220092	Thariq Zhafran	Membuat fitur ROLL beserta efeknya seperti movePlayer
22	15/11/2021	18220092	Thariq Zhafran	Menyatukan UI hasil Vixell agar dapat digunakan di main program
23	15/11/2021	18220090	Rofif Fairuz Hawary	Membuat beberapa implementasi dari skill yang sifatnya aktif (tidak berhubungan dengan buff)
24	17/11/2021	18220091	Rachita Caronica J	Pembuatan Fitur Save
25	17/11/2021	18220086	Aldi Fadlian Sunan	Revisi Fitur Teleport
26	17/11/2021	18220060 18220086 18220090 18220091 18220092 18217035	Laksamana Vixell Aldi Fadlian Sunan Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran Joe Putera	Progress report tiap anggota
27	18/11/2021	18220060 18220086 18217035	Laksamana Vixell Aldi Fadlian Sunan Joe Putera	Pembagian pengerjaan laporan
28	19/11/2021	18220090	Rofif Fairuz Hawary	Finalisasi ADT player
29	19/11/2021	18220092	Thariq Zhafran	Melengkapi dan memperbaiki fungsi prosedur yang dirasa perlu dibuat dan berguna di main program pada ADT Map
30	19/11/2021	18220090 18220091 18220092	Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran	Rapat membahas pengintegrasian semua ADT

31	19/11/2021	18220060	Laksamana Vixell	Membuat Laporan Bab 1
32	20/11/2021	18220090	Rofif Fairuz Hawary	Membuat template main (alur program dalam bentuk comment dan logic program)
33	20/11/2021	18220090 18220091 18220092	Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran	Rapat pengimplementasian skill di main program, membahas prosedur dan fungsi baru yang dibutuhkan di masing-masing ADT agar mudah untuk mengintegrasikannya di main program
34	20/11/2021	18220092	Thariq Zhafran	Mengisi template main program yang berhubungan dengan ADT yang telah saya buat seperti input config map, roll, dan lainnya.
35	21/11/2021	18220090	Rofif Fairuz Hawary	Mulai mengisi template main dengan bahasa c, membuat fitur NEWGAME (memberi skill ke setiap pemain yang baru daftar, membuat logic loop masuk ke dalam game)
36	21/11/2021	18220092	Thariq Zhafran	Membuat fitur Inspect Map, menambah efek move bisa maju dan mundur, dan mengimplentasi prosedur traversalUI di main program
37	21/11/2021	18220086	Aldi Fadlian Sunan	Mengisi Laporan bab 3 dan 7
38	21/11/2021	18220060 18220086 18220090 18220091 18220092 18217035	Laksamana Vixell Aldi Fadlian Sunan Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran Joe Putera	Asistensi 2
39	22/11/2021	18220090	Rofif Fairuz Hawary	Membuat fitur SKILL dan semua efek dari skill-skill tersebut baik skill aktif maupun skill pasif.
40	22/11/2021	18220060 18217035	Laksamana Vixell Joe Putera	Mengisi lampiran 7.1 dan laporan bab 3
41	23/11/2021	18220090 18220092	Rofif Fairuz Hawary Thariq Zhafran	Rapat bahas forceMove dan movePlayer yang berhubungan dengan Baling Baling Jambu dan Mesin Waktu
42	23/11/2021	18217035	Joe Putera	Mengerjakan laporan bab 3
43	24/11/2021	18220091	Rachita Caronica J	Pembuatan fitur Load, menambahkan prosedur/fungsi di mesin kata dan mesin karakter
44	24/11/2021	18220092	Thariq Zhafran	Debugging main program



45	25/11/2021	18220090	Rofif Fairuz Hawary	Mulai membuat ADT Round untuk pengimplementasian UNDO
46	25/11/2021	18220060 18217034	Laksamana Vixell Joe Putera	Mengerjakan Laporan bab 2 dan 4
47	25/11/2021	18220091	Rachita Caronica J	Revisi fitur Save
48	25/11/2021	18220086	Aldi Fadlian Sunan	Test dan mencoba debug fitur roll dan map
49	26/11/2021	18220086	Aldi Fadlian Sunan	Debug Fitur Teleport Pembuatan Fitur Leaderboard
50	26/11/2021	18220092	Thariq Zhafran	Melengkapi Fitur yang masih kurang di main, dan yang masih missing connection antara satu fitur dengan fitur lainnya. Seperti endturn dan roll. Membetulkan beberapa prosedur di ADT Map.
51	26/11/2021	18220090	Rofif Fairuz Hawary	Finalisasi ADT Round, mengupdate main program. membetulkan random skill generator, membuat buff effect (berpengaruh terhadap angka roll),
52	27/11/2021	18220090	Rofif Fairuz Hawary	Membuat fitur UNDO (implementasi ADT Round di main program). Update fitur save yang bisa save round.
53	27/11/2021	18220092	Thariq Zhafran	Ngerapihin main program, implementasi leaderboard buatan aldi di main program
54	27/11/2021	18220086	Aldi Fadlian Sunan	Debug Leaderboard dan main Mengisi Laporan Bab 2 dan 5
55	27/11/2021	18220060	Laksamana Vixell	Debug Main
56	27/11/2021	18220060	Laksamana Vixell	Revisi laporan bab 4
57	27/11/2021	18220091	Rachita Caronica J	Debug Save, revisi Load, debug main. Menambahkan prosedur pada map
58	27/11/2021	18220090 18220091	Rofif Fairuz Hawary Rachita Caronica J	Rapat debugging fitur Load
59	28/11/2021	18220090 18220091 18220092	Rofif Fairuz Hawary Rachita Caronica Thariq Zhafran	Rapat debugging fitur Load
60	28/11/2021	18220090	Rofif Fairuz Hawary	Mengupdate laporan, membuat penjelasan tambahan spesifikasi tugas secara garis kasar
61	28/11/2021	18220091	Rachita Caronica J	Debug Load, membuat laporan bab 5
62	28/11/2021	18220086	Aldi Fadlian Sunan	Debug map
63	28/11/2021	18220092	Thariq Zhafran	Validasi Load jumlah pemain

64	28/11/2021	18220086	Aldi Fadlian Sunan	Revisi Read Me
65	28/11/2021	18220060 18220086 18220091 18217035	Laksamana Vixell Aldi Fadlian Sunan Rachita Caronica Joe Putera	Revisi dan Finalisasi Laporan


## 7.4 Form Asistensi


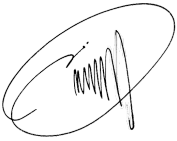




### Form Asistensi Tugas Besar IF2110/Algoritma dan Struktur Data Sem. 1 2021/2022

No. Kelompok/Kelas : 05  
 Nama Kelompok : Squidward U Tentacle's Fans Club  
 Anggota Kelompok (Nama/NIM) :  
 1. Laksamana Vixell Tanjung H./18220060  
 2. Thariq Zhafran Satyagrahan/18220092  
 3. Rofif Fairuz Hawary/18220090  
 4. Aldi Fadlian Sunan/18220086  
 5. Rachita Caronica Jonur/18220091  
 6. Joe Putera/18217035

Asisten Pembimbing : Hardy Valenthio Amansyah/18218004

#### Asistensi I



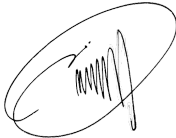



<b>Tanggal : 06 November 2021</b>	<b>Catatan Asistensi:</b>  <ol style="list-style-type: none"> <li>1. Ada pilihan dalam memilih roll, kalau ada hambatan di depan atau belakang, diam di tempat.</li> <li>2. Skill bersifat consumable, jadi akan hilang bila sudah dipakai.</li> <li>3. Ronde itu seperti fase buat ngambil skill dari turn awal, lalu bebas mau ngapain aja (ngegunain skill buat aktifin buff dsb sebelum melakukan roll) trus kalau mau diimplementasikan save buat save rondonya. Nah trus undo itu buat mengulang rundo, bukan mengulang command.</li> <li>4. Stack itu buat nge-undo ke state sebelumnya.</li> <li>5. Buff disimpan di ADT pemain. Buff bisa lebih dari satu jadi tidak mungkin single node (harus multi slot). Jangan pakai stack</li> </ol>
<b>Tempat : Google Meet</b>	
<b>Kehadiran Anggota Kelompok:</b> No NIM Tanda tangan  1 18220060   2 18220092	


 3 18220090  4 18220086  5 1820091  6 18217035 	<ol style="list-style-type: none"> <li>6. Compare (if else) string dipake untuk membuat mesin tau aksi yang akan dilakukan.</li> <li>7. Biar ga ditimpa (stringnya), dikosongin dulu variabel dalam command-nya. Atau disiapkan array of character, setiap kali selesai ditimpanya ke situ. Harus menggunakan mesin karakter ga boleh pake angka yang menunjukkan string tersebut.</li> <li>8. Teleporter (masuk dan keluar) tidak bisa ditaruh diawal dan diakhir.</li> <li>9. Jumlah petak sama roll dibebaskan, asalkan petak <math>\geq 2</math> dan roll <math>\geq 1</math>.</li> <li>10. Saat finish, bebas mau exit program atau balik ke main menu.</li> </ol>
	<b>Tanda Tangan Asisten:</b> 

## Asistensi II

<b>Tanggal : 21/11/2021</b>	<b>Catatan Asistensi:</b>
<b>Tempat : Google Meet</b>	

1. Cara bikin file config.h-nya akhirnya ga tanda "!", kalo pake "!" dikasih keterangan

<p><b>Kehadiran Anggota Kelompok:</b></p> <p>No NIM Tanda tangan</p> <p>1 18220060 </p> <p>2 18220092 </p> <p>3 18220090 </p> <p>4 18220086 </p> <p>5 1820091 </p> <p>6 18217035 </p>	<p>kenapa dibuat mark dengan “!” di adt mesin karakter. Sama asisten juga sudah diperbolehkan boleh memakai mark “!”</p> <ol style="list-style-type: none"> <li>2. Elemen numerik sudah dibaca sebagai integer, di current position sudah diset jadi posisi 4. Karena ini masih testing, jd baru di dummy position 4, tapi nanti jangan lupa diperbaiki aja biar ga nabrak.</li> <li>3. Urutan mainnya, pas awal ada summonplayer trus nanti dimasukin ke index, jadi kalo input pertama otomatis jadi player 1, input kedua jadi player 2 gitu. Jadi berurutan gitu ga dirandom.</li> <li>4. Belum ada counter yang menulis ini turn atau ronde ke berapa, jadi lebih baik dibuat sebagai penanda.</li> <li>5. strcmp buat command boleh, tapi buat baca config jangan pake strcmp.</li> <li>6. Demonya nanti setelah deadline, karena deadline buat laporan dan repo aja.</li> <li>7. Buat map bisa langsung dibuat map1.txt map2.txt dsb gapapa jadi ga harus autoload.</li> <li>8. Pas ketrigger mau dapet skill baru nanti ditawarkan mau nyimpen skill yang udah ada atau ga dapet skill sama sekali.</li> <li>9. Imunitas cuma bisa dipake kalo ada teleport, kalo ada player yg terpengaruh maju atau mundur dari skill player lain, ga bisa dicek imunitas teleport.</li> <li>10. Asisten membebaskan untuk meresolve kalo ada yang memakai skill untuk memajukan/memundurkan player lain tapi bertemu petak dengan “#”</li> </ol>
--	---

	<p><b>Tanda Tangan Asisten:</b></p>  <p>Minggu, 21 November 2021</p>