

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-05		<jml hlm>
		Revisi	-	28/11/2021

Daftar Isi

Update daftar isi di bawah ini.

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

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 Stack
6. ADT List

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

Isi dengan penjelasan tambahan mengenai spesifikasi-spesifikasi fitur yang belum rinci dari Deskripsi Tugas besar.

Ingat: Tidak semua fitur perlu dimasukkan. Hanya yang spesifikasinya belum rinci dan perlu penjelasan tambahan. Jika Anda mengerjakan bonus, maka untuk tiap fitur bonus yang Anda kerjakan, harus Anda buat spesifikasinya.

2.1 Mekanisme Save dan Load

Pada tugas besar ini, 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.

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

2.4 Mekanisme Leaderboard

Pada tugas besar ini, 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 command dari user, termasuk untuk membaca perintah undo 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 mereturn 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 penentuan berapa pemain yang akan bermain yang digunakan sebagai variabel yang dibutuhkan oleh program. Kemudian program akan memasuki loop pertama yang akan menampilkan main menu dan akan meminta input dari

user apakah ingin *new game* atau *load game*. Jika user memasukkan *new game*, program akan memulai dari awal menentukan jumlah *player* dan memilih file konfigurasi yang digunakan sebagai map baru masuk ke loop yang kedua. Sedangkan jika user memasukkan *load game* maka program akan masuk ke loop yang kedua dengan konfigurasi sesuai dengan data yang telah disimpan sebelumnya.

Loop yang kedua merupakan *flow* dari pada *game Mobitangga*. Program akan mengeluarkan *output default* yang digunakan untuk meminta *input command* daripada user. Program akan menyesuaikan sesuai dengan *command* yang diberikan oleh *user*. 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;
}
```

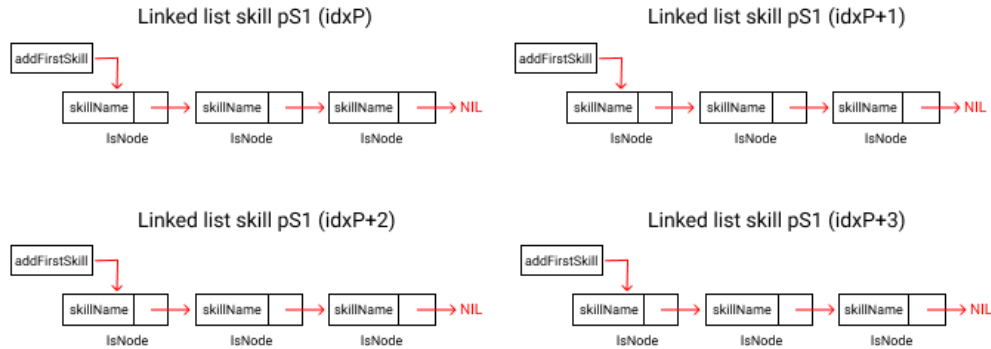
Algoritma yang digunakan pada player ini menarik karena dengan adanya alokasi pada ADT player, sehingga dengan kondisi-kondisi yang terjadi pada player akan terhubung ke dalam index player dan saling berkesinambungan, mulai dari skill, buff, hingga posisi. Selain itu algoritma ini menarik karena setiap informasi disimpan dalam bentuk array dan akan mengakses pointer ke player. 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:

idxP = 1

array of username player



untuk pP (position), pI (buff imunitas anti teleport), pT (informasi apakah sudah terkena teleport), pC (buff cermin), pSB (buff senter pembesar), pSK (buff senter pengecil) memiliki bentuk array yang sama dengan pU (username of player)
untuk pS, akan menyimpan informasi yang terhubung dengan list seperti berikut



Gambar 5.1 Ilustrasi pada ADT Player

5.2 <Algoritma 2>

Isi dengan algoritma yang dimaksud, di mana algoritma tersebut digunakan, dan mengapa menarik.

6 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1	ADT Player		
2	ADT Map		18220086
3	Fitur Teleport	18220086	18220086
4	Fitur Leaderboard	18220086	18220086
5	Laporan	18220086	
6	Manajemen dokumen dan google drive	18220086	

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 13 November 2021 - Zoom

Diskusi dan review mengenai spesifikasi tugas besar secara keseluruhan

7.2.2 Rapat 25 November 2021 - Discord

Dibuat ADT tambahan

Mulai eksplorasi dan pembagian tugas untuk tiap anggota

7.2.3 Rapat 30 November 2021 - Discord

Penyampaian progress report dari tiap anggota

1. Toriq
 - 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. Achi

- 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. AI

- 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 disain sama thoriq (bagian petak mapnya)
- Ada bug lagi ldxP ga muncul-muncul, entah kayanya karena belum disain 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 nantikan

7.3 Log Activity Anggota Kelompok

Tanggal	Kegiatan
31 October 2021	Rilis Tugas Besar
2 November 2021	Pengumpulan username dan pembuatan repository di github
3 November 2021	Rapat untuk mereview spesifikasi tugas
5 November 2021	Rapat untuk membagi tugas
5 - 17 November 2021	Eksplorasi ADT dan pengerjaan sesuai pembagian tugas
6 November 2021	Asistensi 1
17 November 2021	Progress report tiap anggota
17 - 25 November 2021	Melanjutkan pengerjaan ADT
18 November 2021	Pengerjaan laporan
21 November 2021	Asistensi 2
21 - 27 November 2021	Melanjutkan pengerjaan ADT
	Debugging
27 November 2021	Laporan selesai
28 November 2021	Finalisasi dan pengecekan ulang
28 November 2021	Pengumpulan Tugas besar

7.4 Form Asistensi

Form Asistensi Tugas Besar



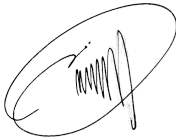
STEI- ITB	IF2111-TB-05	Halaman 12 dari 15 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		




IF2110/Algoritma dan Struktur Data
Sem. 1 2021/2022

No. Kelompok/Kelas : 05
Nama Kelompok :
Anggota Kelompok (Nama/NIM) :
1. Laksmana Vixell Tanjaya 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 Valentio Amansyah/18218004

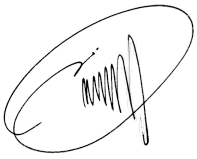



Asistensi I

Tanggal : 06 November 2021	Catatan Asistensi: 1. Ada pilihan dalam memilih roll, kalau ada hambatan di depan atau belakang, diam di tempat. 2. Skill bersifat consumable, jadi akan hilang bila sudah dipakai. 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. 4. Stack itu buat nge-undo ke state sebelumnya. 5. Buff disimpan di ADT pemain. Buff bisa lebih dari satu jadi tidak mungkin single node (harus multi slot). Jangan pakai stack 6. Compare (if else) string dipake untuk membuat mesin tau aksi yang akan dilakukan. 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.
Tempat : Google Meet	
Kehadiran Anggota Kelompok:	
No	
NIM	
Tanda tangan	
1	
18220060	
	
2	
18220092	
	
3	
18220090	
	
4	
18220086	

 5 1820091  6 18217035	8. Teleporter (masuk dan keluar) tidak bisa ditaruh diawal dan diakhir. 9. Jumlah petak sama roll dibebaskan, asalkan petak ≥ 2 dan roll ≥ 1 . 10. Saat finish, bebas mau exit program atau balik ke main menu.
	Tanda Tangan Asisten: 

Asistensi II

Tanggal : 21/11/2021	Catatan Asistensi: <ol style="list-style-type: none"> 1. Cara bikin file config.h-nya akhirnya ga tanda "!", kalo pake "!" dikasih keterangan kenapa dibuat mark dengan "!" di adt mesin karakter. Sama asisten juga sudah diperbolehkan boleh memakai mark "!" 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. 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. 4. Belum ada counter yang menulis ini turn atau ronde ke berapa, jadi lebih baik dibuat sebagai penanda. 5. strcmp buat command boleh, tapi buat baca config jangan pake strcmp. 6. Demya nanti setelah deadline, karena deadline buat laporan dan repo aja.
Tempat : Google Meet	
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 18220060  2 18220092  3	

<p>18220090</p>  <p>4 18220086</p>  <p>5 1820091</p>  <p>6 18217035</p>	<ol style="list-style-type: none"> 7. Buat map bisa langsung dibuat map1.txt map2.txt dsb gapapa jadi ga harus autoloadd. 8. Pas ketrigger mau dapet skill baru nanti ditawarkan mau nyimpen skill yang udah ada atau ga dapet skill sama sekali. 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. 10. Asisten membebaskan untuk meresolve kalo ada yang memakai skill untuk memajukan/memundurkan player lain tapi bertemu petak dengan “#”
	<p>Tanda Tangan Asisten:</p>  <p>Minggu, 21 November 2021</p>