

# **LAPORAN TUGAS BESAR**


## **IF2111 Algoritma dan Struktur Data**

### **BNMO**

**Dipersiapkan oleh:**  
**Kelompok 9 K02**

18221044 Kinanti Wening Asih  
18221058 Marvel Subekti  
18221094 Raka Admiharfan Fatihah  
18221120 Carissa Tabina Rianda  
18221170 Amjad Adhie Prasetyo

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-TB2-09</i>		<i>37</i>
		<i>Revisi</i>	<i>0</i>	<i>2 Desember 2022</i>

## Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1    Spesifikasi Fitur <i>Undertale</i>	4
2.2    Spesifikasi Fitur <i>IsNabrakObstacle</i>	4
2.3    Spesifikasi Fitur <i>TambahKata</i>	4
2.4    Spesifikasi Fitur <i>AturPiringan</i>	4
2.5    Spesifikasi Fitur <i>TembusMap</i>	4
3 Struktur Data	4
3.1    ADT Stack	4
3.2    ADT Map	5
3.3    ADT Tree	5
3.4    ADT Snake on Meteor	6
3.5    Kesinambungan Antar Modul	7
4 Program Utama	9
5 Algoritma-Algoritma Menarik	10
5.1    Algoritma IsInteger	10
5.2    Algoritma PopSS	11
5.3    Algoritma IsUsernameEqual	12
6 Data Test	12
6.1    Data Test 1	12
6.2    Data Test 2	14
6.3    Data Test 3	15
6.4    Data Test 4	18
6.5    Data Test 5	18
6.6    Data Test 6	19
6.7    Data Test 7	19
6.8    Data Test 8	20
7 Test Script	28
8 Pembagian Kerja dalam Kelompok	30
9 Lampiran	31
9.1    Deskripsi Tugas Besar 2	31
9.2    Notulen Rapat	31
9.3    Log Activity Anggota Kelompok	32

## 1 Ringkasan

BNMO merupakan sebuah robot video *game console* yang rusak dan berhasil diperbaiki. Setelah diperbaiki, BNMO mendapat banyak bug sehingga dicari *programmer* yang dapat mengulang program robot *video console* tersebut. Setelah BNMO diperbaiki, kini masalah baru muncul, yaitu *user* ingin menambahkan beberapa *game* dan fitur baru untuk BNMO. *Game* dan fitur yang ingin ditambahkan antara lain adalah :

1. *Hangman*
2. *Tower of Hanoi*
3. *Snake on Meteor*
4. *Scoreboard*
5. *History*
6. *Undertale (Bonus Game)*

Berdasarkan deskripsi tersebut, kami melanjutkan pembuatan program BNMO yang telah kami kerjakan dengan menambahkan beberapa *command* dasar, seperti *scoreboard*, *reset scoreboard*, *history*, dan *reset history*. Kami juga menambahkan beberapa *game* yang dapat dimainkan oleh *user*.

Laporan mencakup deskripsi umum dari persoalan game, penjelasan tambahan spesifikasi tugas, penjelasan tentang struktur data (ADT), program utama, algoritma menarik yang ditemukan selama mengerjakan tugas besar, penjelasan untuk menguji keberjalanan *game*, skenario *test* yang memungkinkan untuk semua fitur yang ada, pembagian tugas, dan lampiran tambahan seperti deskripsi tugas besar, notulensi rapat, *log activity*, dan lain-lain.

Pada tugas besar 2 ini, kami melanjutkan pembuatan program yang telah kami buat pada tugas besar 1. Kami juga menambahkan beberapa ADT baru yang telah kami pelajari juga pada mata kuliah IF2210 – Algoritma dan Struktur Data. Lalu fitur-fitur baru yang dapat digunakan pada program BNMO ini adalah *Scoreboard* program yang akan menampilkan skor dan nama pemain pada semua *game* yang telah disimpan program, *Reset Scoreboard* program yang akan menghapus data *scoreboard* yang telah disimpan, *history* program yang digunakan untuk menampilkan *game* yang pernah dimainkan *user*, banyaknya *game* yang bisa ditampilkan sesuai inputan *user*, *reset history* untuk menghapus *history* game. Lalu beberapa game tambahan adalah *Hangman* yaitu game menebak sebuah kata dengan setiap kesempatan *user* memilih sebuah huruf hingga berhasil menyusun kata yang ingin ditebak, jika dari 10 kesempatan menebak *user* tidak berhasil menebak kata maka *user* akan kalah, lalu *game Tower Of Hanoi* yaitu game yang terdapat 3 buah tiang A, B, dan C, lalu *user* akan memindahkan 5 buah piringan yang terletak pada tiang A yang piringannya disusun dari urutan paling bawah adalah piringan terbesar dan semakin ke atas semakin kecil dipindahkan ke tiang C dan harus sesuai urutan. Lalu *game* terakhir adalah *Snake On Meteor* yaitu *game* dimana kita berperan sebagai ular yang harus menghindari *obstacle* dan *meteor* dan memakan makanan untuk menambah skor.

•

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Spesifikasi Game Undertale

Game ini merupakan game tambahan/bonus. Game ini adalah adaptasi dari game “Undertale” yang dikeluarkan oleh Toby Fox pada tahun 2015. Undertale merupakan suatu game interaktif berbasis cerita yang alur ceritanya ditentukan oleh pilihan-pilihan yang dibuat oleh sang pemain ketika memainkan game tersebut. Dengan menggunakan ADT Tree, percabangan jalur cerita dapat dilakukan. Secara total, ada delapan *ending* berbeda dan sangat banyak detail-detail berbeda yang akan ditampilkan di game untuk masing-masing variasi pilihan yang dibuat oleh sang pemain. Di setiap rute, akan ada hal-hal yang unik seperti *treasure*, cerita yang berbeda, dan *mini games*. Selain memanfaatkan ADT Tree, game ini juga menggunakan ADT lain seperti Array of Map, Map, dan Snake on Meteor pada *mini games* yang disuguhkan di masing-masing “Chapter”. Game akan berakhir ketika sudah mengalahkan seluruh musuh dan mencapai puncak cerita atau di game akan disebut dengan “Chapter 4 : The End”.

### 2.2 Spesifikasi Fungsi IsNabrakObstacle

Fungsi ini digunakan di game Snake On Meteor. Fungsi ini akan bernilai *true* ketika pemain menabrakkan ularnya ke *obstacle* yang ada. Ketika kepala ular menabrak *obstacle* tersebut maka game akan berakhir dan panjang *snake* akan berkurang 1.

### 2.3 Spesifikasi Fungsi TambahKata

Fungsi ini digunakan di game *hangman*. Fungsi ini sendiri digunakan untuk menambahkan kata pada daftar kata kota dan negara, sehingga pilihan kata lebih bervariasi.

### 2.4 Spesifikasi Fungsi AturPiringan

Fungsi ini digunakan pada game *tower of hanoi*. Fungsi ini digunakan untuk mengatur jumlah piringan ketika kita ingin memainkan *game*-nya. Jumlah piringan dapat diatur supaya dapat mempermudah user ketika ingin bermain dengan tingkat kesulitan yang dapat di-*custom*.

### 2.5 Spesifikasi Fungsi TembusMap

Fungsi ini digunakan di game Snake On Meteor. Fungsi ini akan membuat *snake* menembus *map* pada dimensi 5x5. Ketika sudah di berada di pinggir kiri lalu *snake* bergerak ke kiri maka *snake* akan tembus lewat kanan dan sebaliknya. Hal ini pun berlaku untuk di sebelah sisi atas dan bawah.

•

### 3 Struktur Data (ADT)

#### 3.1 ADT Stack

- Sketsa struktur data : struktur data terdiri atas INFOTYPE bertipe pointer to char dan ADDRESS bertipe integer. Pada ADT ini didefinisikan panjang maksimum elemen MaxEl dengan nilai 100. Prototype yang terdapat pada ADT ini terdiri dari konstruktor CreateEmptyStack untuk membuat stack kosong dengan kapasitas MaxEl, predikat IsEmptyStack yang akan mengirim true jika stack kosong dan IsFullStack yang akan mengirim true jika tabel penampung nilai elemen stack penuh. Juga terdapat prosedur PushStack untuk menambah elemen pada stack, PopStack untuk menghapus elemen dari stack, DisplayStack untuk menampilkan isi stack, dan ReverseStack untuk membalik posisi stack.
- Persoalan yang diselesaikan : *Tower Of Hanoi, History, Reset History*
- Alasan pemilihan : Karena permainan *Tower Of Hanoi* yang dimana piringan yang bisa digerakkan adalah piringan paling atas atau elemen TOP yang sesuai dengan stack
- Diimplementasikan dengan nama file header “stackofdisks.h”

#### 3.2 ADT Map

- Sketsa struktur data : struktur data terdiri atas *infotype* yang terdiri juga atas *keytype* ber tipe *pointer to char* dan *valuetype* bertipe integer. Pada ADT Map ini panjang maksimum elemen didefinisikan oleh MaxEl dengan nilai 100. Prototype yang terdapat pada ADT ini terdiri dari konstruktor *CreateEmptyMap* untuk Membuat sebuah Map M kosong berkapasitas MaxEl. Lalu ada predikat *IsEmptyMap* untuk mengirim nilai *true* jika Map M kosong dan *IsFull* untuk mengirim nilai *true* jika Map M penuh. Kemudian ada operator dasar pada ADT Map yang terdiri atas *Value* untuk mengembalikan nilai value dengan key k dari M, *Insert* untuk menambahkan Elmt sebagai elemen Map M, *Delete* untuk menghapus Elmt dari Map M, *IsMemberMap* untuk mengembalikan true jika k adalah member dari map M, *PrintMap* untuk meng-output isi dari map, dan terakhir *SortMap* untuk mengurutkan map.
- Persoalan yang diselesaikan : *Scoreboard, Reset Scoreboard , Bonus Game*
- Alasan pemilihan : Karena scoreboard menyimpan dua elemen utama yaitu username sebagai key dan score dari game yang bersangkutan sebagai value . Selain itu username dari scoreboard haruslah unik , oleh sebab itu dibutuhkan ADT map yang sesuai dengan spesifikasi yang diminta . Untuk game bonus, ADT map dipakai dalam minigamesnya , dimana adt map ini digunakan dalam mini games pada level boss . Hal ini nantinya akan
-

berfungsi sebagai penanda value dari suatu kotak yang isinya bisa 3 macam , yaitu kosong , bom , dan juga treasure .

- Diimplementasikan dengan nama file header “map.h”

### 3.3 ADT Tree

- Sketsa struktur data :struktur data terdiri atas *infotype* bertipe *char*, *addressTree (Left)* dan *addressTree (right)* bertipe *pointer* tNode. Pada ADT *tree* terdapat konstruktor, seperti *BinTree Tree* menghasilkan sebuah pohon biner dari A, L, dan R, jika alokasi berhasil dan menghasilkan pohon kosong (NilBuatTree) jika alokasi gagal dan *void MakeTree* menghasilkan sebuah pohon biner P dari A, L, dan R, jika alokasi berhasil dan menghasilkan pohon P yang kosong (NilBuatTree) jika alokasi gagal. Lalu terdapat Predikat-Predikat Penting, seperti *IsTreeEmpty* mengirimkan true jika P adalah pohon biner kosong, *IsOneElmt* mengirimkan true jika P adalah pohon biner tidak kosong dan hanya memiliki 1 elemen, boolean *IsUnerLeft* mengirimkan true jika pohon biner tidak kosong P adalah pohon *unerleft*: hanya mempunyai subpohon kiri, boolean *IsUnerRight* mengirimkan true jika pohon biner tidak kosong P adalah pohon *unerright*: hanya mempunyai subpohon kanan boolean *IsBiner* mengirimkan true jika pohon biner tidak kosong P adalah pohon biner: mempunyai subpohon kiri dan subpohon kanan. Fungsi *searching* pada ADT *Tree* boolean: *SearchTree* mengirimkan true jika ada node dari P yang berNilBuatTreeai X. Fungsi-fungsi lain, seperti int *NbElmtOfTree* mengirimkan banyaknya elemen (node) pohon biner P, int *NbLeaf* mengirimkan banyaknya daun (node) pohon biner P. Operasi lain pada ADT *tree*, seperti void *AddDaunLeft* P bertambah simpulnya, dengan X sebagai simpul daun terkiri, void *AddDaun* X adalah salah satu daun Pohon Biner P dan P bertambah simpulnya, dengan Y sebagai anak kiri X (jika Kiri = true), atau sebagai anak kanan X (jika Kiri = false), void *DelDaunLeft* P dihapus daun terkirinya, dan didealokasi, dengan X adalah info yang semula disimpan pada daun terkiri yang dihapus, void *DelDaun* Semua daun bernilai X dihapus dari P. Prosedur terakhir adalah prosedur *printTree* yang menampilkan seluruh konten pohon secara PreOrder dengan pemisah berupa \n.
- Persoalan yang diselesaikan : *Undertale*.
- Alasan pemilihan : Program *undertale* memiliki banyak cabang cerita, sehingga ADT *tree* sangat cocok untuk program yang bercabang dan dengan *tree* mempermudah percabangan cerita tersebut.
- Diimplementasikan dengan nama file header “tree.h”
-

### 3.4 ADT Snake on Meteor

- Sketsa struktur data : Struktur data terdiri dari *Point* yang akan mengisi elemen list, address alamat sebelum node (*prev*), address alamat sesudah node (*next*), dan List yang berisikan alamat head dan tail. Pada ADT ini, digunakan Nil bernilai NULL untuk menandakan bahwa node kosong. Adapun fungsi-fungsi yang membersamai ADT ini layaknya ADT pada umumnya seperti *IsEmpty*, *CreateEmpty*, *CreatePoint*, dan lain-lain. Elemen list yang berupa *point* ini akan menunjukkan lokasi absis dan ordinat dari sebuah node, yang kemudian akan di-*print* pada sebuah *map* yang dijadikan sebagai output kepada user.
- Persoalan yang diselesaikan : *Snake on Meteor*
- Alasan pemilihan : Karena program *Snake on Meteor* membutuhkan adt struktur berkait yang menunjukkan lokasi daripada ular, obstacle, food, dan meteor.
- Diimplementasikan dengan nama file header “snakeonmeteor.h”

### 3.5 Kesenambungan antar modul

MODUL	ADT DIGUNAKAN
<i>Scoreboard</i>	Menggunakan ADT <i>Map</i> , <i>ArrayOfMap</i> , <i>mesinkata2</i> , dan <i>mesinkar2</i>
<i>Reset Scoreboard</i>	Menggunakan ADT <i>Map</i> , <i>ArrayOfMap</i> , <i>mesinkata2</i> , dan <i>mesinkar2</i>
<i>History</i>	Menggunakan ADT <i>Stack</i> , <i>mesinkata2</i> , dan <i>mesinkar2</i>
<i>Reset History</i>	Menggunakan ADT <i>Stack</i> , <i>mesinkata2</i> , dan <i>mesinkar2</i>
<i>Hangman</i>	Menggunakan ADT <i>Array</i> , <i>ArrayOfMap</i> , <i>mesinkata2</i> , dan <i>mesinkar2</i>
<i>Tower of Hanoi</i>	Menggunakan ADT

•





#### 4 Program Utama

Program ini dijalankan menggunakan *file* bernama *main.c*. Pada *file* ini, dilakukan include *file* header “*console.h*” yang meng-include seluruh ADT yang diperlukan dan mendeklarasi prosedur yang diperlukan pada *main.c*. Ketika program dijalankan, akan ditampilkan opening message. Lalu program mendeklarasikan beberapa variable yang dibutuhkan seperti *Queue* *QueueGame* dan *ArrayDin ListGames*. Lalu program akan meminta *user* untuk memilih *command* start atau load diikuti dengan nama *file*. Program akan melakukan *looping* untuk memasukkan *command* kedua. *Command* akan dicek menggunakan fungsi yang membandingkan antara *command* dengan input dari *user*. Jika *user* memasukkan *command* “START” maka prosedur STARTGAME dipanggil dan mengeluarkan prosedur HELP. Jika *user* memasukkan *command* LOAD, program akan menggeser pembacaan *command* lalu membaca nama *file* konfigurasi. Program akan mengambil data dari *file* yang telah dibaca dan mengeluarkan prosedur HELP. Jika bukan keduanya, program akan meminta *command* masukan kembali. program akan melakukan *looping* kembali.

Program mendeklarasi pointer to *char* bernama *GAME*. Lalu program akan meminta masukkan *command* dari *user*. Jika *command* masukan adalah SAVE, maka program akan menggeser pembacaan *command* dan membaca nama *file* konfigurasi. Program akan mengubah nama *file* masukan dengan fungsi *wordToString* dan menyimpan data dengan prosedur SAVEBNMO. Jika *command* masukan adalah CREATE, program akan menggeser pembacaan *command* dan membaca nama game yang dimasukkan. Program akan mengubah nama game dengan fungsi *wordToString* dan menggunakan prosedur CREATEGAME dan memasukkannya pada array *ListGames*. Jika *user* memasukkan *command* LIST, program akan menggeser pembacaan *command* dan menggunakan prosedur LISTGAME dan memanggil array *ListGames*. Jika *user* memasukkan *command* DELETE, program akan menggeser pembacaan *command*. Program akan membandingkan *string* nama game yang dimasukkan dan kata “GAME” lalu program akan menghapus game dari array *ListGames* dan *queue* *QueueGame*. Jika *user* memasukkan *command* QUEUE, program akan mengeluarkan prosedur QUEUEGAME dengan memanggil *queue* *QueueGame* dan array *ListGames*. Jika *user* memasukkan *command* PLAY maka program akan mengeluarkan prosedur PLAYGAME.

Jika *user* memasukkan *command* SKIPGAME, program akan menggeser pembacaan *command* dan mengecek jika *command* lanjutan merupakan *integer* dan bukan *EndWord*. Jika semua syarat terpenuhi, program akan mendefinisikan *command* lanjutan sebagai *integer* bernama *skip* dan memanggil prosedur SKIPGAME dengan jumlah skip yang sudah didefinisikan dari *command* lanjutan. Jika *user* memasukkan *command* SCOREBOARD, program akan menggeser pembacaan *command*. Jika *EndWord*, program akan memanggil fungsi *Scoreboard*. Jika *user* memasukkan *command* RESET, program akan menggeser pembacaan *command* dan mendefinisikan array bertipe *char* bernama game dengan kapasitas 50 dan jika

•

*user* memasukkan *command* lanjutan SCOREBOARD, program akan memanggil fungsi ResetScoreboard. Jika *user* memasukkan *command* HISTORY, program akan memanggil fungsi ResetHistory. Lalu jika *user* memasukkan *command* HISTORY, program akan menggeser pembacaan *command* dan mengecek jika *command* lanjutan bertipe *integer* dan bukan merupakan EndWord. Jika memenuhi syarat, program akan memanggil fungsi History dengan *command* lanjutan digunakan untuk menampilkan jumlah permainan yang telah dimainkan.

Jika *user* memasukkan *command* HELP maka program akan mengeluarkan prosedur HELP. Jika *user* memasukkan *command* selain *command* yang tersedia, maka program akan mengeluarkan output “Command tidak dikenali. Silahkan masukkan *command* yang valid”. Program akan melakukan *looping* hingga *user* memasukkan *command* QUIT. Jika *user* mengeluarkan *command* QUIT maka program akan mengeluarkan prosedur QUIT yang menandakan program telah selesai dimainkan.

## 5 Algoritma-Algoritma Menarik

Kami menemukan sebuah algoritma menarik dari program yang telah kami buat. Algoritma ini dinilai menarik karena berbeda dari algoritma lainnya yang pernah dipelajari

### 5.1 Algoritma IsInteger

Algoritma ini didefinisikan sebagai *isInteger* pada ADT mesin kata boolean *isInteger*(Word Kata);

```
boolean isInteger(Word Kata){
    //printf("Masuk\n");
    boolean isInt = true;
    int panjang = Kata.Length;
    //printf("%d", panjang);
    int i = 0;
    if(Kata.TabWord[i] == '\n'){
        return false;
    }

    while (i < panjang && Kata.TabWord[i] != '\n'){
        if(Kata.TabWord[i] == '1' || Kata.TabWord[i] == '2' || Kata.TabWord[i] == '3' || Kata.TabWord[i]
        == '5' || Kata.TabWord[i] == '4' ||
        Kata.TabWord[i] == '6' || Kata.TabWord[i] == '7' || Kata.TabWord[i] == '8' || Kata.TabWord[i] ==
        '9' || Kata.TabWord[i] == '0'){
            isInt = true;
            //printf("True = %c\n", Kata.TabWord[i]);
        }

        else{
            //printf("False = %c\n", Kata.TabWord[i]);
            //printf("Salah\n");
            return false;
        }
        i++;
    }
    return isInt;
}
```

•

Algoritma ini digunakan untuk mengecek apakah input dari user merupakan integer atau bukan. Kemudian fungsi ini akan berpengaruh pada fungsi-fungsi yang lain seperti mengubah string ke integer atau sebaliknya.

## 5.2 Algoritma PopSS

Algoritma ini didefinisikan sebagai *PopSS* pada ADT Stack of Disks sebagai

```
void PopSS(Stack * S, Disk * X, int jml_pir, char * temp);
```

```
/* ***** Menghapus sebuah elemen Stack ***** */
void PopSS(Stack * S, Disk * X, int jml_pir, char * temp)
/* Menghapus X dari Stack S. */
/* I.S. S tidak mungkin kosong */
/* F.S. X adalah nilai elemen TOP yang lama, TOP berkurang 1 */
{
    X->sym = InfoTop(*S).sym;
    X->size = InfoTop(*S).size;
    int width = jml_pir*2-1;
    int ctr;
    for (ctr=0;ctr<width;ctr++)
    {
        if (ctr==jml_pir-1)
        {
            temp[ctr] = '|';
        } else {
            temp[ctr] = ' ';
        }
    }
    InfoTop(*S).sym = temp;
    InfoTop(*S).size = 6;
    Top(*S)--;
}
```

Algoritma ini merupakan modifikasi dari ADT Stack yang dibuat khusus untuk memenuhi poin bonus dari game Tower of Hanoi. Setelah dilakukan proses “Pop” yaitu mengeluarkan “Piringan” dari suatu tiang, akan dilakukan “Insert” potongan tiang yang besarnya sesuai dengan banyak piringan, agar tampak seolah piringan diambil dari tiang tersebut.

•

### 5.3 Algoritma IsUsernameEqual

Algoritma ini didefinisikan sebagai *IsUsernameEqual* pada ADT Mesin Kata 2 sebagai boolean *IsUsernameEqual* (char \* string1, char \* string2);

```
boolean IsUsernameEqual(char*string1 , char*string2)
//Membandingkan String Dengan Memperhatikan Uppercase dan Lowercase string 1 dan 2 dan string yang
dimasukkan harus berupa character*/
{
    int i = 0;
    boolean equal = true;
    if (LengthKalimat(string1) != LengthKalimat(string2))
    {
        equal = false;
        return equal;
    }
    while (*(string1 + i) != '\0' && *(string2 + i) != '\0')
    {
        //printf("Character ke - %d\n", i+1);
        int j = *(string2 + i);
        if ((j >= 65) && (j <= 90))
        {
            if (*(string1 + i) != *(string2 + i) && (*(string1 + i) != *(string2 + i) + 32))
            {
                //printf("Gak Sama\n");
                equal = false;
                return equal;
            }
        }
        else if ((j >= 97) && (j <= 122))
        {
            if (*(string1 + i) != *(string2 + i) && (*(string1 + i) != *(string2 + i) - 32))
            {
                //printf("Gak Sama\n");
                equal = false;
                return equal;
            }
        }
        else
        {
            if (*(string1 + i) != *(string2 + i))
            {
                //printf("Gak Sama\n");
                equal = false;
                return equal;
            }
        }
        i++;
    }
    return equal;
}
```

Algoritma ini berfungsi untuk melakukan validasi terhadap 2 buah string dimana 2 buah string akan dinyatakan sama apabila character dan panjang dari string tersebut sama persis tanpa memperhatikan bentuk *uppercase* maupun *lowercase* dari character tersebut.

## 6 Data Test

Fitur-fitur yang akan dicoba adalah game-game yang dibuat, yaitu Snake on Meteor, Hangman, Tower of Hanoi, dan Undertale (bonus), Scoreboard, Reset Scoreboard, History, dan Reset History. Hasil yang diharapkan adalah sesuai dengan spesifikasi yang diberikan dan jika ditemukan beberapa fitur yang dibebaskan kepada mahasiswa, maka hasil akan kami sesuaikan

•

sebagaimana mestinya agar berkorelasi dengan output-output yang ada. Berikut adalah beberapa Data Test yang kami lakukan terhadap Tugas Besar ini.

## 6.1 Data Test 1

Test ini dilakukan untuk menguji game Snake On Meteor

```
Loading SNAKE ON METEOR ....
Selamat datang di Snake on Meteor!

Peraturan permainan Snake on Meteor:
1. Snake dapat bergerak ke atas, kiri, bawah, atau kanan dengan meng-input w/a/s/d TANPA SPASI!
2. Tujuan permainan : Dapatkan skor tertinggi dengan menghindari obstacle dan meteor yang jatuh dari langit!
3. Snake akan mati jika terkena meteor, obstacle, atau badan sendiri.

Selamat bermain!

Mengenerate peta, snake, dan makanan . . .
Berhasil digenerate!

Berikut merupakan peta permainan
-----
| | | | |
| | 2 | 1 | H |
| | | | |
| | | | |
| | | | |
| | + | | |
| | | | |
-----

TURN 1
Silahkan masukkan command anda:
```

Gambar 6.1.1. Game berhasil dijalankan

```
TURN 1
Silahkan masukkan command anda:S

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| | | | |
| | 2 | 1 | m |
| | | | |
| | | | H |
| | | | |
| | | | |
| | + | | |
| | | | |
-----

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 2
Silahkan masukkan command anda:
```

Gambar 6.1.2. Tidak kena meteor

```
TURN 3
Silahkan masukkan command anda:S

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| | | | | |
| | | | |
| | | | 2 |
| m | | | 1 |
| | + | | H | o |
| | | | |
| | | | |
-----

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 4
Silahkan masukkan command anda:D

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| | | | | |
| | | m | | |
| | | o | 3 | |
| | | | 2 | |
| | + | | 1 | H |
| | | | |
| | | | |
-----

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan
```

Gambar 6.1.3. Berhasil makan

```
TURN 4
Silahkan masukkan command anda:D

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| | | | | |
| | | m | | |
| | | o | 3 | |
| | | | 2 | |
| | + | | 1 | H |
| | | | |
| | | | |
-----

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 5
Silahkan masukkan command anda:A

Input tidak valid. Dilarang bunuh diri!
TURN 5
Silahkan masukkan command anda:
```

Gambar 6.1.4. Contoh input ke arah badan sendiri.

•

```

Berikut merupakan peta permainan
-----
| | | | | |
| | | | |
| | o | |
| | m | | 3 |
| H | + | | 2 | 1 |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 6
Silahkan masukkan command anda:D

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| | | | m | |
| | | | |
| | | o |
| | | | |
| 1 | H | | 3 | 2 |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

Ada obstacle itu lho, yo jangan mbok tabrak to mas

Game berakhir. Skor: 6

```

Gambar 6.1.5. Contoh bergerak ke obstacle

```

TURN 12
Silahkan masukkan command anda:D

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| 3 | 4 | | | |
| 2 | | | | |
| 1 | H | m | | |
| | | | + | |
| | | o | | |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 13
Silahkan masukkan command anda:D

Meteor masih panas! Anda belum dapat kembali ke titik tersebut.
Silahkan masukkan command lainnya

```

Gambar 6.1.6. Contoh bergerak ke meteor

```

TURN 18
Silahkan masukkan command anda:W

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| | | | | |
| | | H | | |
| | | 1 | | |
| o | 5 | m | + |
| | 4 | 3 | | |
-----
Anda terkena meteor!
Berikut merupakan peta permainan
-----
| | | | | |
| | | H | | |
| | | 1 | | |
| o | 4 | m | + |
| | 3 | 2 | | |
-----
Silahkan lanjutkan permainan

```

Gambar 6.1.7. Kena meteor

```

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| 4 | 5 | | | |
| 3 | | | | |
| 2 | | m | | o |
| 1 | H | | + | |
| | | | | |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 26
Silahkan masukkan command anda:D

Berhasil bergerak!
Berikut merupakan peta permainan
-----
| 5 | | | | |
| 4 | | | | |
| 3 | | | | o |
| 2 | 1 | m | + | |
| | | | | |
-----
Kepala snake terkena meteor!

Game berakhir. Skor: 10

```

Gambar 6.1.8. Head kena meteor

•

## 6.2 Data Test 2

Test ini dilakukan untuk menguji Score Board

```
ENTER COMMAND: SCOREBOARD
**** SCOREBOARD GAME RING ****
| NAMA      | SKOR  |
|-----|-----|
| Marvel_1  | 50    |

**** SCOREBOARD GAME Diner DASH ****
| NAMA      | SKOR  |
|-----|-----|
| Marvel_2  | 12947 |

**** SCOREBOARD GAME HANGMAN ****
| NAMA      | SKOR  |
|-----|-----|
| Marvel_3  | 0     |

**** SCOREBOARD GAME TOWER OF HANOI ****
| NAMA      | SKOR  |
|-----|-----|
| Marvel_4  | 9     |

**** SCOREBOARD GAME SNAKE ON METEOR ****
| NAMA      | SKOR  |
|-----|-----|
| Marvel_5  | 16    |

**** SCOREBOARD GAME MATH QUIZ ****
| NAMA      | SKOR  |
|-----|-----|
| Marvel_6  | 70    |

ENTER COMMAND: 
```

## 6.3 Data Test 3

Test ini dilakukan untuk menguji game Hangman

```
SELAMAT DATANG DI PERMAINAN HANGMAN!

1. PLAY
2. HELP
3. TAMBAHKATA
4. QUIT

Silakan pilih menu di atas: PLAY

1. KOTA
2. NEGARA

Silakan pilih tema di atas: NEGARA
```

Gambar 6.3.1. Main Menu dan Memilih PLAY

●

```

Tebakan Sebelumnya:
Kata: _ _ _ _ _
Kesempatan: 10
Masukkan tebakan: A
Tebakan kamu salah!

_ _ _ _ _
Tebakan Sebelumnya: A
Kata: _ _ _ _ _
Kesempatan: 9
Masukkan tebakan: 

```

Gambar 6. 3. 2. Menebak kata yang belum pernah ditebak

```

Kata: I N _ _ R I _
Kesempatan: 8
Masukkan tebakan: S
Selamat, tebakan kamu benar!
|
|
|
|
_ _ _ _ _
Tebakan Sebelumnya: A I N D R S
Kata: I N _ _ R I S
Kesempatan: 8
Masukkan tebakan: G
Selamat, tebakan kamu benar!
|
|
|
|
_ _ _ _ _
Berhasil menebak INGRIS! Kamu mendapatkan 7 poin!

```

Gambar 6.3.3. Tampilan jika tebakan benar

●



```
Selamat, tebakan kamu benar!
```

```
|
```

```
|
```

```
|
```

```
|
```

```
|
```

```
-----
```

```
Tebakan Sebelumnya: A I E P R N C
```

```
Kata: P E R A N C I _
```

```
Kesempatan: 8
```

```
Masukkan tebakan: S
```

```
Selamat, tebakan kamu benar!
```

```
|
```

```
|
```

```
|
```

```
|
```

```
|
```

```
-----
```

```
Berhasil menebak PERANCIS! Kamu mendapatkan 8 poin!
```

Gambar 6.3.4. Tampilan jika tebakan benar kedua kalinya

```
Tebakan Sebelumnya: A S D F G H J
```

```
Kata: _ _ A _ _ A  
Kesempatan: 2  
Masukkan tebakkan: K  
  
Tebakkan kamu salah!
```

```
-- --  
| |  
| O |  
| / \ |  
| |  
| / |  
|-- --
```

```
Tebakan Sebelumnya: A S D F G H J K
```

```
Kata: _ _ A _ _ A  
Kesempatan: 1  
Masukkan tebakan: D  
  
Kamu sudah menebak huruf DI!
```

```
-- --  
| |  
| O |  
| / \ |  
| |  
| / |  
|-- --
```

```
Kesempatan habis! Total poin yang kamu dapatkan adalah  
GAME OVER.
```

```
Skor kamu adalah 15
```

Gambar 6.3.5. Tebakan salah dan kesempatan habis

```
SELAMAT DATANG DI PERMAINAN HANGMAN!

1. PLAY
2. HELP
3. TAMBAHKATA
4. QUIT

Silakan pilih menu di atas: HELP
Cara bermain:
1. Masukkan huruf tebakan dari kata yang sudah ditetapkan
2. Input hanya berupa satu huruf kapital
3. Input huruf kecil akan dinyatakan salah
4. Input lebih dari satu huruf akan dinyatakan salah
5. Setiap tebakan kata yang berhasil, pemain akan mendapatkan 25 poin
6. Jika kesempatan habis, permainan berhenti

Press any key to continue..
K
SELAMAT DATANG DI PERMAINAN HANGMAN!
```

Gambar 6.3.6. Fitur HELP

```
SELAMAT DATANG DI PERMAINAN HANGMAN!

1. PLAY
2. HELP
3. TAMBAHKATA
4. QUIT

Silakan pilih menu di atas: TAMBAHKATA
1. KOTA
2. NEGARA

Silakan pilih tema di atas: NEGARA

Silakan memasukkan NEGARA yang ingin kamu tambahkan
(DALAM HURUF KAPITAL) :LAOS
Kata berhasil ditambahkan!
SELAMAT DATANG DI PERMAINAN HANGMAN!

1. PLAY
2. HELP
3. TAMBAHKATA
4. QUIT

Silakan pilih menu di atas: QUIT

Sad to say goodbye to you! :(((
```

Gambar 6.3.7. Fitur TAMBAHKATA dan QUIT

## 6.4 Data Test 4

Test ini dilakukan untuk menguji History

```
ENTER COMMAND: HISTORY 10
Berikut adalah daftar Game yang telah dimainkan
1. MATH QUIZ
2. SNAKE ON METEOR
3. TOWER OF HANOI
4. HANGMAN
5. Diner DASH
6. RNG

ENTER COMMAND: 
```

•

## 6.5 Data Test 5

Test ini dilakukan untuk menguji Reset Score Board

```
ENTER COMMAND: RESET SCOREBOARD
DAFTAR SCOREBOARD:
0.ALL
1. RING
2. Diner DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. MATH QUIZ
7. UNDERTALE
8. GAME NGASAL
SCOREBOARD YANG INGIN DIHAPUS: 0
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK)?
YA
Scoreboard berhasil di-reset.
```

Gambar 6.5.1 Mereset seluruh isi scoreboard

```
ENTER COMMAND: RESET SCOREBOARD
DAFTAR SCOREBOARD:
0.ALL
1. RING
2. Diner DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. MATH QUIZ
SCOREBOARD YANG INGIN DIHAPUS: 0
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK)?
TIDAK
Masukan tidak valid, silahkan masukan kembali
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK)?
TIDAK
Scoreboard gagal di-reset.

ENTER COMMAND: SCOREBOARD
**** SCOREBOARD GAME RING ****
| NAMA      | SKOR  |
|-----|
| Marvel_1  | 50    |

**** SCOREBOARD GAME Diner DASH ****
| NAMA      | SKOR  |
|-----|
| Marvel_2  | 12947 |

**** SCOREBOARD GAME HANGMAN ****
| NAMA      | SKOR  |
|-----|
| Marvel_3  | 0     |

**** SCOREBOARD GAME TOWER OF HANOI ****
| NAMA      | SKOR  |
```

Gambar 6.5.2 Tidak jadi mereset scoreboard

## 6.6 Data Test 6

Test ini dilakukan untuk menguji Reset History

```
ENTER COMMAND: RESET HISTORY
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET HISTORY? (YA/TIDAK)
YA
History berhasil direset

ENTER COMMAND: HISTORY 10
Belum ada game yang dimainkan

ENTER COMMAND: 
```

●

## 6.7 Data Test 7

Test ini dilakukan untuk menguji Tower Of Hanoi

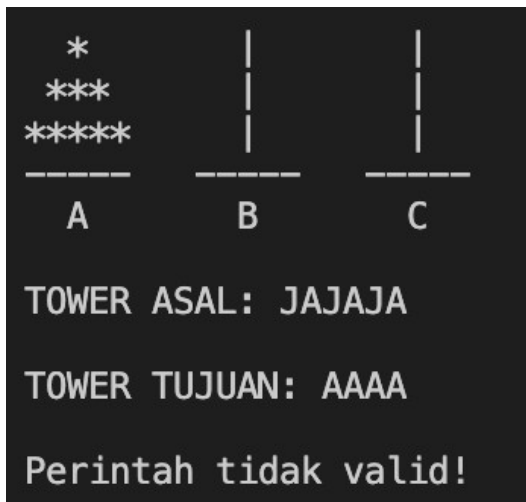
```
***** KELOMPOK 9 PUNYA *****
TOWER OF HANOI
*****
Ketik 'START' dengan huruf kapital untuk bermain!
START

Hore! Mari kita bermain tower of hanoi 🎮
Sebelumnya, apakah kamu pernah bermain tower of hanoi? (Y/N)~
Ketik (Y/N)
N

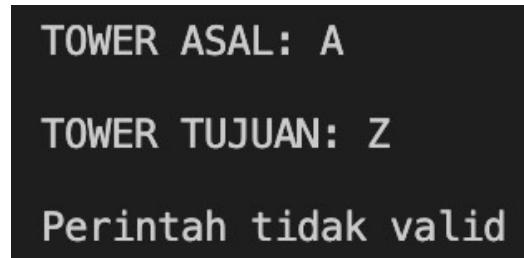
Baiklah, berikut adalah HOW TO PLAY tower of hanoi! 🎮
1. Akan terdapat 3 tower, ABC. Tower A akan terisi penuh dengan piringan.
2. Tower B dan C kosong. Tujuanmu adalah memindahkan seluruh piringan dari Tower A ke C.
3. Kamu hanya bisa memindahkan piringan ke tower kosong atau yang piringan teratasnya ukurannya
   lebih besar dari piringan yang mau dipindahkan.
4. Input untuk tower harus berupa huruf kapital A,B, atau C.

Selamat bermain!
Masukkan jumlah piringan yang diinginkan <-.->: 3
```

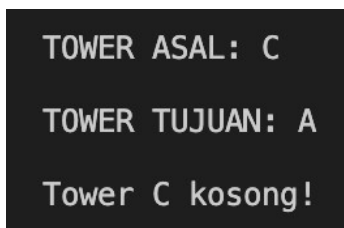
Gambar 6.7.1 Welcome Screen



Gambar 6.7.2 Invalid input



Gambar 6.7.3 Invalid input single character



Gambar 6.7.4 Pemindahan dari tower kosong

```
TOWER ASAL: A
TOWER TUJUAN: A
Tower asal tidak boleh sama dengan tower tujuan!
```

Gambar 6.7.5 Pemindahan dari tower satu ke tower yang sama

```

TOWER ASAL: A
TOWER TUJUAN: B

Memindahkan Disk ke tower B..

  |      |      |
***      |      |
*****  *      |
-----  -      -
  A      B      C

```

```

      |
    ***
  ***
  -----
    A

      |
      |
      |
    *
  -----
    B

      |
      |
      |
      |
  -----
    C

```

TOWER ASAL: A

TOWER TUJUAN: B

Ukuran disk harus lebih kecil dari disk teratas tower B!

Gambar 6.7.6 Pemindahan validGambar

### 6.7.7 Pemindahan ke tower dengan top disk lebih kecil

```
Memindahkan Disk ke tower C..

      |      |      *
      |      |     ***
      |      |    *****
-----
      A      B      C

Kamu berhasil!
Skor didapatkan: 6
```

[illegible]

### Gambar 6.7.8 Winning screen

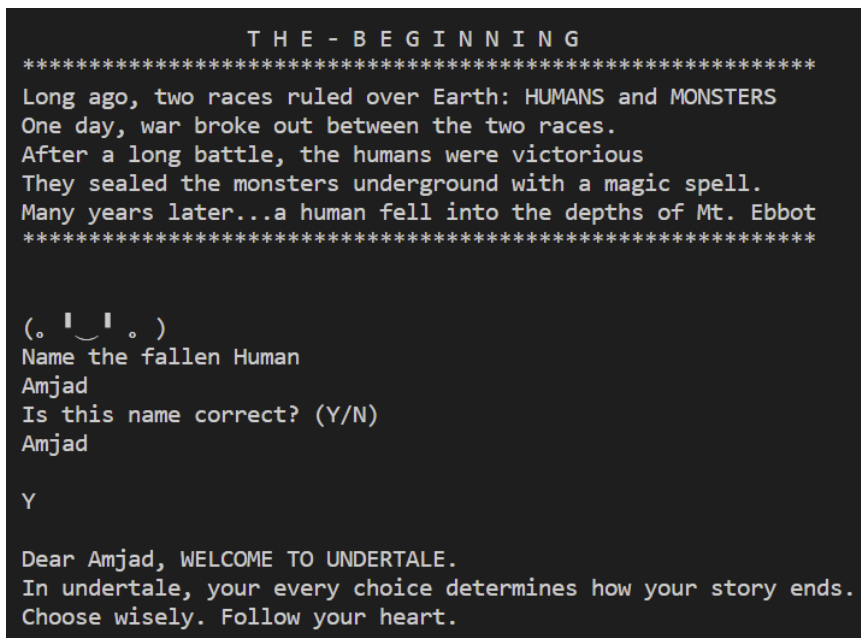
Gambar 6.7.9 20 Disk

## 6.8 Data Test 8

Test ini dilakukan untuk menguji game Undertale.

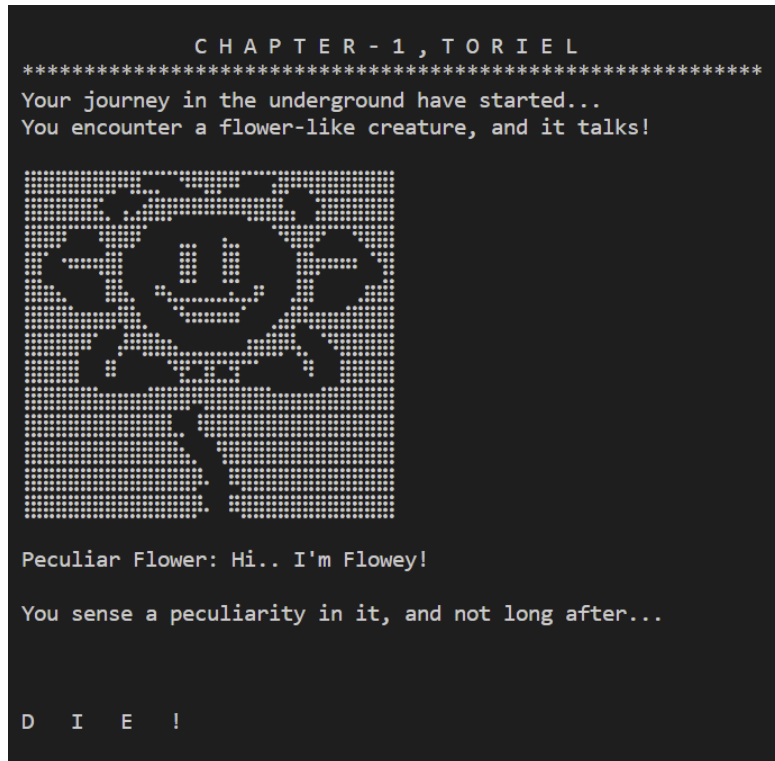


Gambar 6.8.1. Game berhasil dijalankan.



Gambar 6.8.2. Input nama pemain.

•



Gambar 6.8.3.1. Chapter 1.

```
Oh no, the flower is actually evil! (°_°)
Luckily, not long after... a goat creature came to your rescue!
Toriel: Oh.. Hello!
Toriel: My name is Toriel, keeper of the Undergrounds!
Toriel: Please be careful, my child, as this place is not safe

-----
Toriel then brings you to her home and cares for you. It seems like
She had lost her own child before, and so she is very protective of you...
But you must return to your world. You cannot remain.
To do so, one day, you went to the gate she explicitly told you to avoid...

Toriel arrives shortly afterwards with a disappointed look on her face.
Toriel: My dear child.. I cannot let you through these gates...
However... must you insist, you'll need to face me first...
Prove to me, child, that you are strong enough to proceed.

*****
      Y O U - H A V E - E N T E R E D - A - F I G H T
*****

RPSOpponet = scissors
rock / paper / scissors
Enter your choice:rock
You win
skor RPS 100
You have succeeded in defeating Toriel. Would you like to Kill or Spare her? (K/S)
K
Toriel: Why.. would.. you do this..?
Toriel perished. You open the gate in front of you to proceed in your journey.
```

●

Gambar 6.8.3.2. Lanjutan Chapter 1 dengan pilihan Kill (K).

[illegible]

Gambar 6.8.4.1. Chapter 2

```
*****
YOU - HAVE - ENTERED - A - FIGHT
*****

|-----|
|12345678|
|~~~~~|
|[M|#|C|[-]|
|[7|8|9|[+]|
|[4|5|6|[x]|
|[1|2|3|[ ]|
|[.|0|:]|=]|
|-----|

:::~::~:
:: !! WARNING !! ::
:::~::~:

Prepare Your Calculator
To win this game, you must maintain your health so that it is not below 70
Every Mistake will reduce your health by 10 , so make sure there are no mistakes

Your Health = 100

STAGE 1
GET READY ...

12 + 15 = 27
Your Answer Is Correct

15 + 8 = 23
Your Answer Is Correct
```

Gambar 6.8.4.2. Lanjutan Chapter 2.



```

Your Health = 100
Do You Want To Repeat Stage 1 ? (yes/no)
no

STAGE 2
GET READY ...

9 x 5 = 45
Your Answer Is Correct

5 x 10 = 50
Your Answer Is Correct

10 x 7 = 70
Your Answer Is Correct

Your Health = 100
Do You Want To Repeat Stage 2 ? (yes/no)
no

STAGE 3
GET READY ...

44 - 41 = 3
Your Answer Is Correct

41 - 14 = 27
Your Answer Is Correct

14 - 10 = 4
Your Answer Is Correct

Your Health = 100
Do You Want To Repeat Stage 3 ? (yes/no)
no

```

Gambar 6.8.4.3. Mini Game di Chapter 3.

```

41 - 14 = 27
Your Answer Is Correct

14 - 10 = 4
Your Answer Is Correct

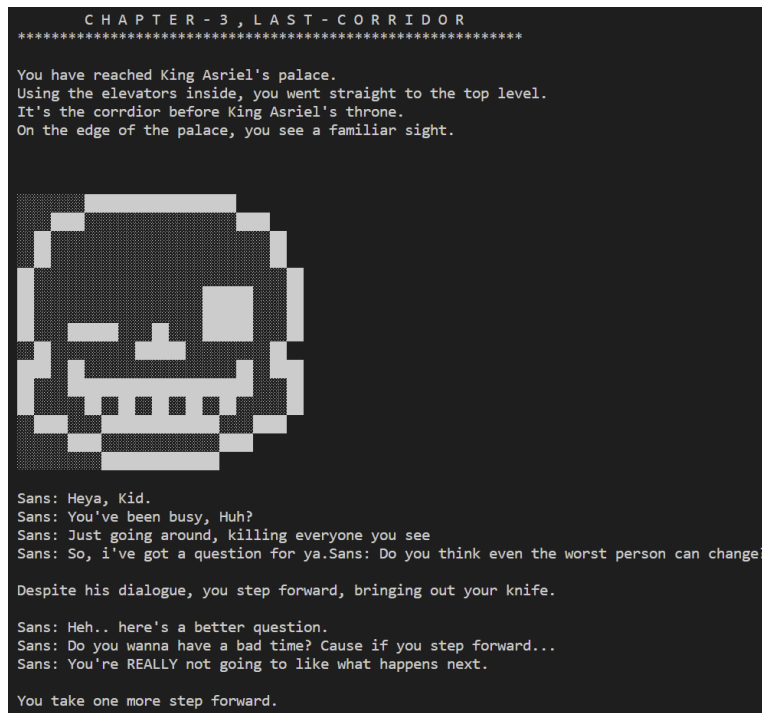
Your Health = 100
Do You Want To Repeat Stage 3 ? (yes/no)
no
Your Current Health = 100

You have succeeded in defeating Papyrus. Would you like to Kill or Spare him? (K/S)
K
Papyrus: I knew it.. humans are nothing but trash...
Papyrus perished. You aim for King Asriel's palace, the only way out from the Undergrounds..

```

Gambar 6.8.4.4. Chapter 3 Dengan pilihan Kill (K).

●



Gambar 6.8.5.1. Chapter 3



Gambar 6.8.5.2. Minigame di Chapter 3 dengan pilihan Kill (K).

●



```

Here is the game map
-----
| A | B | C | D | E |
-----
| F | G | H | I | J |
-----
| K | L | M | N | O |
-----
| P | Q | R | S | T |
-----
| U | V | W | X | Y |
-----
-999 1 -999 1 1
-999 -999 1 1 1
1 -999 0 1 1
1 1 1 1 1
1 1 1 1 1
Select a box by selecting the available letters :

```

Gambar 6.8.6.3. Minigame chapter 4

```

Here is the game map
-----
| A |   | C | D | E |
-----
| F | G | H | I | J |
-----
| K | L | WIN | N | O |
-----
| P | Q | R | S | T |
-----
| U | V | W | X | Y |
-----
-999 -1 -999 1 1
-999 -999 1 1 1
1 -999 100 1 1
1 1 1 1 1
1 1 1 1 1
*****
YOU - HAVE - ESCAPED - THE - UNDERGROUND
*****
Thank you for playing
-----
Remember that every choice you make in life Matters.

```

Gambar 6.8.6. The end.

•

## 7 Test Script

**Table 1. Test Script**

No	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur <i>scoreboard</i>	Memeriksa apakah program dapat menampilkan <i>scoreboard</i>	Memasukkan input SCOREBOARD	Data Test 2	Program menampilkan tabel yang berisi nama dan skor pemain pada semua <i>game</i>	Sesuai yang diharapkan
2	Fitur <i>reset scoreboard</i>	Memeriksa apakah program dapat menghapus data pada <i>scoreboard</i> yang telah disimpan	Memasukkan input RESET SCOREBOARD	Data Test 5	Program berhasil menghapus data <i>scoreboard</i> pada sebuah <i>game</i> ataupun seluruh <i>game</i>	Sesuai yang diharapkan
3	Fitur <i>history</i>	Memeriksa apakah program dapat menampilkan <i>game-game</i> yang telah dimainkan <i>user</i>	Memasukkan input HISTORY	Data Test 4	Program berhasil menampilkan <i>game-game</i> yang pernah dimainkan <i>user</i> sesuai input banyaknya <i>game</i> yang ingin ditampilkan	Sesuai yang diharapkan

•

4	Fitur <i>reset history</i>	Memeriksa apakah program dapat menghapus data <i>history</i>	Memasukkan input RESET HISTORY	Data Test 6	Program berhasil menghapus seluruh data <i>game</i> yang pernah dimainkan <i>user</i>	Sesuai yang diharapkan
5	<i>Game Hangman</i>	Memeriksa apakah <i>game hangman</i> dapat dijalankan	Memasukkan <i>game hangman</i> dalam <i>queue</i> dan memainkannya menggunakan PLAYGAME	Data Test 3	Permainan <i>hangman</i> dapat dijalankan	Sesuai yang diharapkan
6	<i>Game Tower Of Hanoi</i>	Memeriksa apakah <i>game Tower Of Hanoi</i> dapat dijalankan	Memasukkan <i>game Tower Of Hanoi</i> dalam <i>queue</i> dan memainkannya menggunakan PLAYGAME	Data Test 7	Permainan <i>Tower Of Hanoi</i> dapat dijalankan	Sesuai yang diharapkan
7	<i>Game Snake On Meteor</i>	Memeriksa apakah <i>game Snake On Meteor</i> dapat dijalankan	Memasukkan <i>game Snake On Meteor</i> dalam <i>queue</i> dan memainkannya menggunakan PLAYGAME	Data Test 1	Permainan <i>Snake On Meteor</i> dapat dijalankan	Sesuai yang diharapkan

## 8 Pembagian Kerja dalam Kelompok

NIM	Nama Lengkap	Deskripsi Tugas
18221044	Kinanti Wening Asih	<ul style="list-style-type: none"> <li>• Pengerjaan <i>game Tower of Hanoi</i></li> <li>• Pengerjaan bonus <i>game Tower of Hanoi</i></li> <li>• Pengerjaan <i>game</i> bonus Undertale</li> </ul>

•

		<ul style="list-style-type: none"> <li>• Pengerjaan laporan</li> <li>• Melakukan <i>debugging</i> , <i>maintaining</i> , dan <i>testing</i> keseluruhan program</li> <li>• Pembuatan ADT Tree, Stack of Disks, Queue Hybrid</li> </ul>
18221058	Marvel Subekti	<ul style="list-style-type: none"> <li>• Pengerjaan fitur <i>scoreboard</i>, <i>reset scoreboard</i>, <i>history</i>, <i>reset history</i> , <i>username</i> , dan <i>bonus game</i></li> <li>• Menyusun semua <i>console</i> dan <i>main</i> program</li> <li>• Melakukan <i>debugging</i> , <i>maintaining</i> , dan <i>testing</i> keseluruhan program</li> <li>• Revisi Fitur pada Spesifikasi 1 meliputi <i>delete game</i> , <i>play game</i> , <i>save</i> , <i>help</i> , dan <i>load</i></li> <li>• Pembuatan ADT <i>array of map</i> , <i>map</i> , dan <i>stack</i></li> <li>• <i>penulisan laporan</i></li> </ul>
18221094	Raka Admiharfan Fatihah	<ul style="list-style-type: none"> <li>• Pengerjaan <i>game Snake on Meteor</i></li> <li>• Penulisan laporan</li> <li>• Merapikan <i>repository</i></li> </ul>
18221120	Carissa Tabina Rianda	<ul style="list-style-type: none"> <li>• Pengerjaan <i>game Hangman</i></li> <li>• Pengerjaan <i>bonus game Hangman</i></li> <li>• Penulisan laporan</li> </ul>
18221170	Amjad Adhie Prasetyo	<ul style="list-style-type: none"> <li>• Pengerjaan <i>game Snake on Meteor</i></li> <li>• pembuatan ADT struktur berkait</li> <li>• Pengerjaan <i>bonus game Snake on Meteor</i></li> <li>• Penulisan laporan</li> </ul>

## 9 Lampiran

### 9.1 Deskripsi Tugas Besar 2

BNMO (dibaca: Binomo) adalah sebuah robot video game console yang dimiliki oleh Indra dan Doni. Dua bulan yang lalu, ia mengalami kerusakan dan telah berhasil diperbaiki. Sayangnya, setelah diperbaiki ia justru mendapatkan lebih banyak bug dalam sistemnya. Oleh

•

karena itu, Indra dan Doni mencari programmer lain yang lebih handal untuk ulang memprogram robot video game console kesayangannya. Pada tugas sebelumnya, kalian telah berhasil membuat Indra dan Doni bahagia dengan mengimplementasikan fitur-fitur dasar. Kini, Indra dan Doni ingin melakukan pengembangan lebih lanjut dengan menambahkan fitur serta permainan pada BNMO.

## 9.2 Notulen Rapat

### Sabtu, 19 November 2022:

- 1) Pembagian tugas berdasarkan spesifikasi
  - Asih : Tower of Hanoi
  - Carissa : Hangman
  - Raka dan Amjad : Snake on Meteor
  - Marvel : Scoreboard, Reset Scoreboard, History, Reset History, dan menggabungkan semua game dalam file utama

### Senin, 28 November 2022:

- 1) Kerjaan tambahan selain pembagian tugas awal :
  - Kinanti Wening Asih dan Marvel Subekti : Bonus Game
  - Raka, Amjad, Carissa : Mengerjakan laporan
- 2) Perubahan *console* dikabarkan dengan teman sekelompok.
- 3) Fungsi / ADT tambahan, agar diubah semua nama variabel dan fungsinya agar tidak bertabrakan dengan fungsi dan variabel lain sehingga bisa meminimalisir debugging.
- 4) Membuat copy dari folder tubes agar tidak terjadi kehilangan code saat *push* maupun *pull*.

## 9.3 Log Activity Anggota Kelompok

Waktu	NIM		Keterangan
19 November 2022	18221044, 18221170, 18221120	18221058, 18221094,	Rapat perdana pembagian tugas kelompok
25 November 2022	18221044, 18221170, 18221120	18221058, 18221094,	Asistensi 1
28 November 2022	18221044, 18221170, 18221120	18221058, 18221094,	Rapat kedua untuk pembagian tugas tambahan

●



29 November 2022	18221044, 18221170, 18221120	18221058, 18221094,	Recheck , Test case , debug semua game agar sesuai dengan spesifikasi.
2 Desember 2022	18221044, 18221170, 18221120	18221058, 18221094,	Asistensi 2
2 Desember 2022	18221044, 18221170, 18221120	18221058, 18221094,	Pengumpulan

#### 9.4 Form Asistensi

**Form Asistensi Tugas Besar**  
**IF2110/Algoritma dan Struktur Data**  
**Sem. 1 2022/2023**

No. Kelompok/Kelas : 9 / K02

Nama Kelompok : Sembilan

Anggota Kelompok (Nama/NIM) :

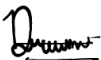

- 1.Kinanti Wening Asih/18221044
- 2.Marvel Subekti/18221058
- 3.Amjad Adhie Prasetyo/18221170
- 4.Raka Admiharfan Fatihah/18221094

•




Asisten Pembimbing


: Afif Fahreza

Asistensi I



<b>Tanggal : Jumat, 25 November 2022</b>	<b>Catatan Asistensi:</b>  <b>1. Apakah kami masih perlu membuat ADT set dan Array Of Set ? soalnya dengan menggunakan adt map dan array of map saja semua username sudah dipastikan unik dan tidak redundant.</b>  <b>Jawaban:</b> ikuti sesuai yang ada di spesifikasi saja  <b>2. Apakah untuk tower of hanoi dibolehkan untuk melakukan push terhadap spasi yg diperlukan (bukan hanya “piringan”), supaya</b>
<b>Tempat : Zoom</b>	
<b>Kehadiran Anggota Kelompok:</b>	
No NIM Tanda tangan  1 18221058   2 18221120 	

•

<p>3</p> <p>18221170</p> 	<p>mempermudah display masing2 tiang ketika game dimainkan?</p> <p><b>Jawaban:</b></p> <p>Ya, boleh.</p> <p>3. Untuk sistem penilaian tower of hanoi, apakah batasan nilai maksimal berbeda untuk tiap tinggi tiang? Apakah penentuan nilai maksimal harus mengikuti suatu aturan tertentu atau dibebaskan pada kelompok?</p> <p><b>Jawaban:</b></p> <p>Ya, batasan berbeda untuk masing-masing tinggi tiang. Penentuan batasan maksimal skor dibebaskan.</p>
<p>4</p> <p>18221094</p> 	<p><b>Tanda Tangan Asisten:</b></p>
<p>5</p> <p>18221044</p> 	

	
--	--

## Asistensi II

<b>Tanggal : Jumat, 2 Desember 2022</b>	<b>Catatan Asistensi:</b>  <b>1. Boleh tidak untuk mengrun sebuah game di laptop yang berbeda karena permasalahan versi dari compiler?</b>  <b>Jawaban:</b> <b>Boleh, dan dijelaskan saja pada demo</b>  <b>2. Permasalahan drive belum dibuat</b>  <b>Jawaban:</b> <b>Gapapa kalau tidak buat driver karena kakaknya akan run langsung di main</b>  <b>3. Menambahkan table kesinambungan antara modul</b>
<b>Tempat : Zoom</b>	
<b>Kehadiran Anggota Kelompok:</b>	
No NIM Tanda tangan  1 18221058    2 18221120  	

•

3

18221170



4

18221094




5

18221044

A handwritten signature in black ink, appearing to be 'B. H.' with a horizontal line underneath.

•

	<b>Tanda Tangan Asisten:</b>   (3)
--	---