

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data


Console 'BNMO' Part 2

Dipersiapkan oleh:

Kelompok 07

18221067	Fawwaz Abrial Saffa
18221069	Gibran Fasha Ghazanfar
18221087	Willy Frans Farel Sijabat
18221101	Ilmagita Nariswari
18221163	Aufar Ramadhan

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB2-07		38
		Revisi	1	02-12-2022

Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Penambahan fitur opsi jumlah piringan pada Tower of Hanoi	4
2.2 Penambahan obstacle pada Snake on Meteor	4
2.3 Penghubungan sisi peta yang bersebrangan pada Snake on Meteor	5
2.4 Game berbasis Tree: Labtek Hunt	5
3 Struktur Data (ADT)	5
3.1 Stack	5
3.1.1 Sketsa Struktur Data	5
3.1.2 Persoalan yang Diselesaikan	6
3.1.3 Alasan Pemilihan	6
3.1.4 Implementasi	7
3.2 Set	7
3.2.1 Sketsa Struktur Data	7
3.2.2 Persoalan yang Diselesaikan	8
3.2.3 Alasan Pemilihan	8
3.2.4 Implementasi	8
3.3 Map	8
3.3.1 Sketsa Struktur Data	8
3.3.2 Persoalan yang Diselesaikan	10
3.3.3 Alasan Pemilihan	10
3.3.4 Implementasi	10
3.4 Linked List	10
3.4.1 Sketsa Struktur Data	10
3.4.2 Persoalan yang Diselesaikan	13
3.4.3 Alasan Pemilihan	13
3.4.4 Implementasi	13
4 Program Utama	14
4.1 Pemanggilan Command	14
4.1.1 SCOREBOARD	14
4.1.2 RESET SCOREBOARD	14
4.1.3 HISTORY <n>	14
4.1.4 RESET HISTORY	14

4.1.5 SAVE	14
4.1.6 LOAD	14
4.1.7 PLAY GAME	15
4.1.8 DELETE GAME	15
4.2 Game Hangman	15
4.3 Game Tower of Hanoi	15
4.4 Game Snake on Meteor	15
5 Data Test	16
5.1 SCOREBOARD	16
5.2 RESET SCOREBOARD	17
5.3 HISTORY <n>	20
5.4 RESET HISTORY	21
5.5 HANGMAN	22
5.6 TOWER OF HANOI	26
5.7 SNAKE ON METEOR	27
6 Test Script	31
7 Pembagian Kerja dalam Kelompok	33
8 Lampiran	33
8.1 Deskripsi Tugas Besar 2	33
8.2 Notulen Rapat	34
8.3 Log Activity Anggota Kelompok	34
8.4 Lampiran Form Asistensi	34

1 Ringkasan

Indra dan Doni memiliki sebuah robot video game console bernama BNMO (dibaca: Binomo). Dua bulan lalu, BNMO mengalami kerusakan dan berhasil diperbaiki. Akan tetapi, setelah perbaikan, BNMO mendapatkan lebih banyak *bug* dalam sistemnya. Oleh karena itu, Indra dan Doni sedang mencari programmer yang lebih handal untuk memprogram ulang robot video game console kesayangan mereka.

BNMO adalah sebuah video game console berbasis CLI (Command Line Interface) yang diimplementasikan menggunakan bahasa C. Fitur-fitur utama BNMO adalah memainkan game, menambahkan game, menghapus game, dan mengurutkan game-game yang akan dimainkan. Saat pertama kali dinyalakan, user akan melihat tampilan Welcome Page dan beberapa tampilan menu seperti START dan LOAD. Setelah itu, BNMO akan menerima input-input commands lainnya.

Laporan ini berisi penjelasan tentang game yang telah kami buat. Bagian pertama berisi ringkasan berisi deskripsi umum persoalan dan isinya secara singkat. Bagian kedua berisi tentang penjelasan tambahan spesifikasi tugas. Bagian ketiga menceritakan struktur data yang dipakai pada program. Bagian keempat berisikan algoritma-algoritma menarik yang ditemukan pada program. Bagian kelima berisi tentang data test. Bagian keenam berisi tentang test script. Bagian ketujuh berisikan tentang test script. Bagian kedelapan berisikan tentang pembagian kerja.

Permainan ini digunakan menggunakan bahasa C dengan struktur data yang telah dipelajari pada mata kuliah IF2111 Algoritma dan Struktur Data. ADT-ADT yang digunakan yaitu array, matriks, mesin karakter dan mesin kata, queue, map, set, tree, dan linked list.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 *Penambahan fitur opsi jumlah piringan pada Tower of Hanoi*

Pada awal game Tower of Hanoi, user akan diminta untuk memasukkan jumlah piringan yang disebutkan. Skor akan berbasis jumlah optimal pemindahan piringan.

2.2 *Penambahan obstacle pada Snake on Meteor*

Ketika Kepala dari snake mengenai obstacle, maka permainan berakhir. Obstacle muncul di awal permainan dan tidak dapat ditembus oleh snake. Selain itu, makanan juga tidak dapat muncul pada titik yang memiliki obstacle.

2.3 Penghubungan sisi peta yang bersebrangan pada Snake on Meteor

Ketika kepala snake melewati sisi atas peta, maka kepala snake akan muncul dari sisi bawah peta. Hal yang sama berlaku ketika kepala snake melewati sisi kiri/kanan peta, maka kepala snake akan muncul dari sisi yang berlawanan.

2.4 Game berbasis Tree: Labtek Hunt

Labtek Hunt adalah game berbasis tree yang berupa Command Line Interface RPG berbasis *choices*. Dalam game ini, user pertama-tama punya health sebanyak 100 dan ditempatkan di Labtek V. Tujuan dari game ini adalah untuk mengumpulkan skor sebanyak-banyaknya dengan cara memasuki ruangan-ruangan di Labtek V. Skor didapatkan dari memasuki ruangan yang memiliki treasure. Selain itu, saat memasuki ruangan yang memiliki bomb, health dan skor user akan berkurang. User maksimal bertemu bomb dua kali. User juga tidak bisa memasuki ruangan yang telah dimasuki.

3 Struktur Data (ADT)

3.1 Stack

3.1.1 Sketsa Struktur Data

```
/* File : stack.h */
/* deklarasi stack yang diimplementasi dengan tabel kontigu dan ukuran sama */
/* TOP adalah alamat elemen puncak */
/* Implementasi dalam bahasa C dengan alokasi statik */
/* ID dimulai dari 1 */
/* Elemen bertipe integer */

#ifndef stackt_H
#define stackt_H

#include "boolean.h"

#define StackNil -1
#define StackMaxEl 150
/* Nil adalah stack dengan elemen kosong . */

/* Contoh deklarasi variabel bertipe stack dengan ciri TOP : */
/* Versi I : dengan menyimpan tabel dan alamat top secara eksplisit*/
typedef struct {
    int T[StackMaxEl+1]; /* tabel penyimpan elemen */
    int TOP; /* alamat TOP: elemen puncak */
} Stack;
/* Definisi stack S kosong : S.TOP = Nil */
/* Elemen yang dipakai menyimpan nilai Stack T[1]..T[StackMaxEl] */
```

```

/* Jika S adalah Stack maka akses elemen : */
/* S.T[(S.TOP)] untuk mengakses elemen TOP */
/* S.TOP adalah alamat elemen TOP */

/* Definisi akses dengan Selektor : Set dan Get */
#define Top(S) (S).TOP
#define InfoTop(S) (S).T[(S).TOP]

/* ***** Prototype ***** */
/* *** Konstruktor/Kreator *** */
void CreateEmptyStack(Stack *S);
/* I.S. sembarang; */
/* F.S. Membuat sebuah stack S yang kosong berkapasitas MaxEl */
/* jadi indeksanya berada pada range [1,MaxEl] */
/* Ciri stack kosong : TOP bernilai StackNil */

/* ***** Predikat Untuk test keadaan KOLEKSI ***** */
boolean IsEmptyStack(Stack S);
/* Mengirim true jika Stack kosong: lihat definisi di atas */
boolean IsFullStack(Stack S);
/* Mengirim true jika tabel penampung nilai elemen stack penuh */

/* ***** Menambahkan sebuah elemen ke Stack ***** */
void Push(Stack * S, int X);
/* Menambahkan X sebagai elemen Stack S. */
/* I.S. S mungkin kosong, tabel penampung elemen stack TIDAK penuh */
/* F.S. X menjadi TOP yang baru,TOP bertambah 1 */

/* ***** Menghapus sebuah elemen Stack ***** */
void Pop(Stack * S, int* X);
/* Menghapus X dari Stack S. */
/* I.S. S tidak mungkin kosong */
/* F.S. X adalah nilai elemen TOP yang lama, TOP berkurang 1 */

#endif

```

3.1.2 Persoalan yang Diselesaikan

Persoalan yang diselesaikan dengan ADT Stack adalah Tower of Hanoi dan History.

3.1.3 Alasan Pemilihan

Stack digunakan untuk Tower of Hanoi karena dalam permainan ini, yang bisa dipindahkan dari satu tower ke tower lainnya adalah piringan TOP dari masing-masing tower. Selain itu, stack dipilih untuk history karena penyimpanan history yaitu menggunakan metode push dan TOP berupa permainan yang baru dimainkan.

3.1.4 Implementasi

Implementasi dari ADT Stack terdapat di kode stack.c dalam folder ADT (/src/ADT/stack.c).

3.2 Set

3.2.1 Sketsa Struktur Data

```
#ifndef set_H
#define set_H
#include <stdio.h>
#include "boolean.h"

/* MODUL Set
Deklarasi stack yang dengan implementasi array eksplisit-statik rata kiri
*/

#define Nil 0
#define MaxEl 100

typedef char infotype;
typedef int address;

typedef struct
{
    infotype Elements[MaxEl];
    address Count;
} Set;

/* Definisi Set S kosong : S.Count = Nil */
/* S.Count = jumlah element Set */
/* S.Elements = tempat penyimpanan element Set */

/* ***** Prototype ***** */

/* *** Konstruktor/Kreator *** */
void CreateEmptySet(Set *S);
/* I.S. Sembarang */
/* F.S. Membuat sebuah Set S kosong berkapasitas MaxEl */
/* Ciri Set kosong : count bernilai Nil */

/* ***** Predikat Untuk test keadaan KOLEKSI ***** */
boolean IsEmptySet(Set S);
/* Mengirim true jika Set S kosong*/
/* Ciri Set kosong : count bernilai Nil */

boolean IsFullSet(Set S);
/* Mengirim true jika Set S penuh */
/* Ciri Set penuh : count bernilai MaxEl */
```

```

/* ***** Operator Dasar Set ***** */
void InsertSetElmt(Set *S, int Elmt);
/* Menambahkan Elmt sebagai elemen Set S. */
/* I.S. S mungkin kosong, S tidak penuh
   S mungkin sudah beranggotakan Elmt */
/* F.S. Elmt menjadi anggota dari S. Jika Elmt sudah merupakan anggota,
   operasi tidak dilakukan */

void DeleteSetElmt(Set *S, int Elmt);
/* Menghapus Elmt dari Set S. */
/* I.S. S tidak kosong
   Elmt mungkin anggota / bukan anggota dari S */
/* F.S. Elmt bukan anggota dari S */

boolean IsMemberSet(Set S, int Elmt);
/* Mengembalikan true jika Elmt adalah member dari S */

#endif

```

3.2.2 Persoalan yang Diselesaikan

Persoalan yang diselesaikan dengan ADT Set adalah pengecekan huruf input di Hangman agar input tidak berulang.

3.2.3 Alasan Pemilihan

ADT Set dipilih karena adanya fungsi InsertSet yang memeriksa apakah sudah ada infotype Elmt. Apabila sudah ada, infotype Elmt tidak akan di-insert.

3.2.4 Implementasi

Implementasi dari ADT Set terdapat pada kode set.c dalam folder ADT (/src/ADT/set.c).

3.3 Map

3.3.1 Sketsa Struktur Data

```

#ifndef map_H
#define map_H
#include <stdio.h>
#include "boolean.h"
#include "mesinkata.h"

/* MODUL Map */

#define Nil 0

```



```

#define MaxEl 100
#define Undefined -999

// typedef int bool;
typedef Word keytype;
typedef int valuetype;
typedef int address;

typedef struct {
    keytype Key;
    valuetype Value;
} infotype;

typedef struct {
    infotype Elements[MaxEl+1];
    address Count;
} Map;

/* Definisi Map M kosong : M.Count = Nil */
/* M.Count = jumlah element Map */
/* M.Elements = tempat penyimpanan element Map */

/* ***** Prototype ***** */

/* *** Konstruktor/Kreator *** */
void CreateEmptyMap(Map *M);
/* I.S. Sembarang */
/* F.S. Membuat sebuah Map M kosong berkapasitas MaxEl */
/* Ciri Map kosong : count bernilai Nil */

/* ***** Predikat Untuk test keadaan KOLEKSI ***** */
boolean IsEmptyMap(Map M);
/* Mengirim true jika Map M kosong*/
/* Ciri Map kosong : count bernilai Nil */

boolean IsFullMap(Map M);
/* Mengirim true jika Map M penuh */
/* Ciri Map penuh : count bernilai MaxEl */

/* ***** Operator Dasar Map ***** */
valuetype ValueMap(Map M, keytype k);
/* Mengembalikan nilai value dengan key k dari M */
/* Jika tidak ada key k pada M, akan mengembalikan Undefined */

void InsertMap(Map *M, keytype k, valuetype v);
/* Menambahkan Elmt sebagai elemen Map M. */
/* I.S. M mungkin kosong, M tidak penuh
    M mungkin sudah beranggotakan v dengan key k */
/* F.S. v menjadi anggota dari M dengan key k, terurut secara menurun
    berdasarkan value, jika ada anggota M lain dengan value yang sama, elemen

```

```

yang lebih baru diletakkan belakangan.
    Jika k sudah ada, operasi tidak dilakukan */

void DeleteMap(Map *M, keytype k);
/* Menghapus Elmt dari Map M. */
/* I.S. M tidak kosong
   element dengan key k mungkin anggota / bukan anggota dari M */
/* F.S. element dengan key k bukan anggota dari M */

boolean IsMemberMap(Map M, keytype k);
/* Mengembalikan true jika k adalah member dari M */

#endif

```

3.3.2 Persoalan yang Diselesaikan

Persoalan yang diselesaikan dari ADT Map adalah scoreboard. Implementasi map menggunakan list dengan elemennya berupa struct yang berisi key dan value, dimana key berupa nama pemain dan valuenya berupa score pemain tersebut dalam suatu game.

3.3.3 Alasan Pemilihan

ADT Map dipilih karena memiliki *key* dan *value*.

3.3.4 Implementasi

Implementasi dari ADT Map terdapat pada kode map.c dalam folder ADT (/src/ADT/map.c).

3.4 *Linked List*

3.4.1 Sketsa Struktur Data

```

/* File : listlinier.h */
/* contoh ADT list berkait dengan representasi fisik pointer */
/* Representasi address dengan pointer */
/* infotype adalah integer */

#ifndef listlinier_H
#define listlinier_H

#include "boolean.h"

#define Nil NULL

typedef int infotype;

```

```

typedef struct {
    infotype X;
    infotype Y;
} Point;

typedef struct tElmtlist *address;
typedef struct tElmtlist {
    Point info;
    address next;
} ElmtList;
typedef struct {
    address First;
} List;

/* Definisi list : */
/* List kosong : First(L) = Nil */
/* Setiap elemen dengan address P dapat diacu Info(P), Next(P) */
/* Elemen terakhir list : jika addressnya Last, maka Next(Last)=Nil */
#define Absis(point) (point).X
#define Ordinat(point) (point).Y

#define Info(P) (P)->info
#define Next(P) (P)->next
#define First(L) ((L).First)

/* PROTOTYPE */
/***** TEST LIST KOSONG *****/
boolean IsListEmpty (List L);
/* Mengirim true jika list kosong */

/***** PEMBUATAN LIST KOSONG *****/
void CreateEmptyListL (List *L);
/* I.S. sembarang */
/* F.S. Terbentuk list kosong */

/***** Manajemen Memori *****/
address Alokasi (infotype X, infotype Y);
/* Mengirimkan address hasil alokasi sebuah elemen */
/* Jika alokasi berhasil, maka address tidak nil, dan misalnya */
/* menghasilkan P, maka info(P)=X, Next(P)=Nil */
/* Jika alokasi gagal, mengirimkan Nil */

void Dealokasi (address *P);
/* I.S. P terdefinisi */
/* F.S. P dikembalikan ke sistem */
/* Melakukan dealokasi/pengembalian address P */

/***** PENCARIAN SEBUAH ELEMEN LIST *****/
address Search (List L, infotype X, infotype Y);
/* Mencari apakah ada elemen list dengan info(P)= X */
/* Jika ada, mengirimkan address elemen tersebut. */

```

```

/* Jika tidak ada, mengirimkan Nil */

int indexOf(List L, infotype X, infotype Y);
/* Mencari indeks keberapa sebuah elemen dengan absis X dan ordinat Y pada
list
    Jika elemen tidak ada di dalam list, maka akan mengembalikan nilai -9999
*/

/***** PRIMITIF BERDASARKAN NILAI *****/
/**** PENAMBAHAN ELEMEN ****/
void InsVFirst (List *L, infotype X, infotype Y);
/* I.S. L mungkin kosong */
/* F.S. Melakukan alokasi sebuah elemen dan */
/* menambahkan elemen pertama dengan nilai X jika alokasi berhasil */

void InsVLast (List *L, infotype X, infotype Y);
/* I.S. L mungkin kosong */
/* F.S. Melakukan alokasi sebuah elemen dan */
/* menambahkan elemen list di akhir: elemen terakhir yang baru */
/* bernilai X jika alokasi berhasil. Jika alokasi gagal: I.S.= F.S. */

/**** PENGHAPUSAN ELEMEN ****/
void DelVFirst (List *L, infotype *X, infotype *Y);
/* I.S. List L tidak kosong */
/* F.S. Elemen pertama list dihapus: nilai info disimpan pada X */
/* dan alamat elemen pertama di-dealokasi */

void DelVLast (List *L, infotype *X, infotype *Y);
/* I.S. list tidak kosong */
/* F.S. Elemen terakhir list dihapus: nilai info disimpan pada X */
/* dan alamat elemen terakhir di-dealokasi */

/***** PRIMITIF BERDASARKAN ALAMAT *****/
/**** PENAMBAHAN ELEMEN BERDASARKAN ALAMAT ****/
void InsertFirstListL (List *L, address P);
/* I.S. Sembarang, P sudah dialokasi */
/* F.S. Menambahkan elemen ber-address P sebagai elemen pertama */

void InsertAfterListL (List *L, address P, address Prec);
/* I.S. Prec pastilah elemen list dan bukan elemen terakhir, */
/* P sudah dialokasi */
/* F.S. Insert P sebagai elemen sesudah elemen beralamat Prec */

void InsertLastListL (List *L, address P);
/* I.S. Sembarang, P sudah dialokasi */
/* F.S. P ditambahkan sebagai elemen terakhir yang baru */

/**** PENGHAPUSAN SEBUAH ELEMEN ****/
void DelFirstListL (List *L, address *P);
/* I.S. List tidak kosong */
/* F.S. P adalah alamat elemen pertama list sebelum penghapusan */

```

```

/*      Elemen list berkurang satu (mungkin menjadi kosong) */
/* First element yg baru adalah suksesor elemen pertama yang lama */

void DelP (List *L, infotype X, infotype Y);
/* I.S. Sembarang */
/* F.S. Jika ada elemen list beraddress P, dengan info(P)=X */
/* Maka P dihapus dari list dan di-dealokasi */
/* Jika tidak ada elemen list dengan info(P)=X, maka list tetap */
/* List mungkin menjadi kosong karena penghapusan */

void DellastListL (List *L, address *P);
/* I.S. List tidak kosong */
/* F.S. P adalah alamat elemen terakhir list sebelum penghapusan */
/*      Elemen list berkurang satu (mungkin menjadi kosong) */
/* Last element baru adalah predesesor elemen terakhir yg lama, */
/* jika ada */

void DelAfterListL (List *L, address *Pdel, address Prec);
/* I.S. List tidak kosong. Prec adalah anggota list */
/* F.S. Menghapus Next(Prec): */
/*      Pdel adalah alamat elemen list yang dihapus */

/***** PROSES SEMUA ELEMEN LIST *****/
void PrintInfoListL (List L);
/* I.S. List mungkin kosong */
/* F.S. Jika list tidak kosong, iai list dicetak ke kanan: [e1,e2,...,en] */
/* Contoh : jika ada tiga elemen bernilai 1, 20, 30 akan dicetak: [1,20,30] */
/* */
/* Jika list kosong : menulis [] */
/* Tidak ada tambahan karakter apa pun di awal, akhir, atau di tengah */

int NbElmtListL (List L);
/* Mengirimkan banyaknya elemen list; mengirimkan 0 jika list kosong */

#endif

```

3.4.2 Persoalan yang Diselesaikan

Persoalan yang diselesaikan dengan ADT linked list adalah pengimplementasian game Snake on Meteor. Tipe data yang ditampung pada linked list adalah point yang menyimpan nilai $\langle x, y \rangle$.

3.4.3 Alasan Pemilihan

Linked list pada snake dipilih karena akan diperlukan manipulasi HEAD dan TAIL linked list.

3.4.4 Implementasi

Implementasi dari ADT Linked List terdapat pada kode listlinier.c dalam folder ADT (/src/ADT/listlinier.c).

4 Program Utama

4.1 Pemanggilan Command

4.1.1 SCOREBOARD

Pemanggilan command SCOREBOARD bertujuan untuk menampilkan scoreboard tiap game. Apabila scoreboard kosong, maka akan ditampilkan pesan "Scoreboard kosong 😞".

4.1.2 RESET SCOREBOARD

Pemanggilan command RESET SCOREBOARD bertujuan untuk me-reset/menghapus skor-skor pada scoreboard tiap game. RESET SCOREBOARD bisa dilakukan untuk semua game atau salah satu saja.

4.1.3 HISTORY <n>

HISTORY <n> merupakan command yang digunakan untuk melihat permainan apa saja yang telah dimainkan dari data yang sudah ada dari file konfigurasi (Jika LOAD) dan dari mulai Start Game juga, dengan <n> adalah jumlah permainan yang telah dimainkan yang ingin ditampilkan. Urutan teratas merupakan permainan terakhir yang dimainkan. Jika <n> lebih besar dari jumlah permainan yang telah dimainkan, akan menampilkan seluruh permainan yang telah dimainkan.

4.1.4 RESET HISTORY

RESET HISTORY merupakan command untuk menghapus semua history di permainan.

4.1.5 SAVE

SAVE merupakan command yang digunakan untuk menyimpan *state* sekarang BNMO ke sebuah text file. Command ini sebelumnya sudah diimplementasikan pada Tugas Besar 1, tetapi pada Tubes Besar 2 terdapat perubahan pada konfigurasi text file penyimpanan. Sekarang command SAVE juga menyimpan *state* history dan scoreboard.

4.1.6 LOAD

LOAD merupakan command yang digunakan untuk membaca sebuah text file yang merupakan *save file* dan menjadikan *state* tersebut *state* sekarang BNMO. Command ini sebelumnya sudah diimplementasikan pada Tugas Besar 1, tetapi

pada Tubes Besar 2 terdapat perubahan pada konfigurasi text file penyimpanan. Sekarang command LOAD juga membaca *state* history dan scoreboard.

4.1.7 PLAY GAME

PLAY GAME merupakan command yang digunakan ketika ingin menjalankan sebuah game. Command ini sebelumnya sudah diimplementasikan pada Tugas Besar 1, tetapi pada Tubes Besar 2 terdapat perubahan pada permintaan nama pemain dan menyimpan nama dan score pemain ke scoreboard. Sekarang command PLAY GAME setelah game telah dimainkan akan meminta nama pemain dan meng-*update* scoreboard bila nama belum terpakai. Jika nama sudah terpakai, akan terus diminta nama.

4.1.8 DELETE GAME

DELETE GAME merupakan command yang digunakan ketika ingin menghapus sebuah game dari list game. Command ini sebelumnya sudah diimplementasikan pada Tugas Besar 1, tetapi pada Tubes Besar 2 terdapat perubahan yaitu ketika sebuah game dihapus, semua nama game pada history akan dihapuskan dan scoreboard untuk game tersebut juga akan dihapuskan.

4.2 Game Hangman

Hangman merupakan game tebak-tebakan kata. Pada game ini, list kata dibaca dari array. Pemain diberikan 10 kesempatan untuk menebak kata yang dimaksud. Apabila pemain salah, maka muncul gambar orang yang sedang digantung, sedikit demi sedikit. Semakin banyak kesalahan, akan semakin banyak juga potongan gambar orang yang sedang digantung. Apabila huruf tebak sesuai, maka huruf tersebut akan muncul pada layar. Pemain tidak boleh menebak kata dengan huruf yang sama.

4.3 Game Tower of Hanoi

Tower of Hanoi merupakan game puzzle. Pada game ini, seorang user dihadapkan dengan tiga *tower* dan piringan-piringan pada *tower* pertama. Obyektif game ini adalah untuk memindahkan semua piringan dari *tower* pertama ke tower yang ketiga, dengan tidak ada piringan yang lebih besar menimpa piringan yang lebih kecil. Pada game ini, pemain dipersilakan untuk mengisi sendiri jumlah piringan yang diinginkan.

4.4 Game Snake on Meteor

Game Snake mirip dengan permainan Snake yang ada pada permainan konsol jaman dahulu, tetapi yang membedakan adalah adanya meteor yang dapat mengenai *snake* tersebut. Kepala, badan, dan ekor Snake semuanya merupakan bagian dari *linked list*. Pada setiap putaran, pemain memasukkan input berupa 'w' yang akan menggerakkan *snake* ke atas, 'a' yang akan menggerakkan *snake* ke kiri, 's' yang akan menggerakkan *snake* ke bawah, atau 'd' yang akan menggerakkan *snake* ke kanan.

Selain itu, setiap putaran setelah permainan berhasil digenerate(turn >1), 1 meteor akan di-random pada titik tertentu(ditandai dengan huruf m). Apabila salah satu bagian dari snake terkena meteor, maka bagian tersebut akan dihapus dari snake dan panjang dari snake akan berkurang sebanyak 1. Apabila komponen dari snake (kepala/badan/ekor) terkena meteor (akan disebut hit untuk mempermudah pemahaman kalian), maka bagian badan sebelum hit akan tersambung dengan bagian badan setelah hit. Setelah terkena hit, ada kemungkinan badan snake berada di koordinat yang saling diagonal. Selanjutnya, makanan tidak dapat muncul di titik ini dan kepala snake juga tidak bisa mengunjungi titik ini di turn selanjutnya.

5 Data Test

5.1 SCOREBOARD

Fitur yang dites
Scoreboard
Hasil yang seharusnya diberikan
<pre>// Misal tahap ini adalah tahap game over dari game // EIFFEL TOWER Skor akhir: 12 Nama: BNMO</pre>
<pre>ENTER COMMAND: SCOREBOARD **** SCOREBOARD GAME TOWER OF HANOI **** NAMA SKOR ----- BNMO 12 Finn 9 **** SCOREBOARD GAME DINER DASH**** NAMA SKOR ----- BNMO 80 Finn 62 **** SCOREBOARD GAME SNAKE ON METEOR**** NAMA SKOR ----- BNMO 80 Finn 62 </pre>

**** SCOREBOARD GAME RNG****

NAMA	SKOR
BNMO	13
Finn	11

**** SCOREBOARD GAME HANGMAN****

NAMA	SKOR
BNMO	99
Finn	1

Data test

===== ENTER COMMAND : SCOREBOARD

*** SCOREBOARD RNG ***

(1) BRO : 65

*** SCOREBOARD Diner DASH ***

(1) ILM : 110000

*** SCOREBOARD HANGMAN ***

--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD TOWER OF HANOI ***

--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD SNAKE ON METEOR ***

--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD 2048 ***

(1) NOOB : 0

*** SCOREBOARD LABTEK HUNT ***

--- Scoreboard kosong ðŸŸ© main yuk! ---

5.2 RESET SCOREBOARD

Fitur yang dites
Reset Scoreboard
Hasil yang seharusnya diberikan
<p>ENTER COMMAND: RESET SCOREBOARD</p> <p>DAFTAR SCOREBOARD:</p> <ul style="list-style-type: none"> 0. ALL 1. RNG 2. Diner DASH 3. HANGMAN 4. TOWER OF HANOI 5. SNAKE ON METEOR <p>SCOREBOARD YANG INGIN DIHAPUS: 0</p> <p>APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK)? YA</p> <p>Scoreboard berhasil di-reset.</p>
<p>ENTER COMMAND: RESET SCOREBOARD</p> <p>DAFTAR SCOREBOARD:</p> <ul style="list-style-type: none"> 0. ALL 1. RNG 2. Diner DASH 3. HANGMAN 4. TOWER OF HANOI 5. SNAKE ON METEOR <p>SCOREBOARD YANG INGIN DIHAPUS: 4</p> <p>APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD TOWER OF HANOI (YA/TIDAK)? YA</p> <p>Scoreboard berhasil di-reset.</p>
Data test

```

===== ENTER COMMAND : RESET SCOREBOARD
DAFTAR SCOREBOARD:
0. ALL
1. RNG
2. Diner DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. 2048
7. LABTEK HUNT
SCOREBOARD YANG INGIN DIHAPUS: 6
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD 2048 (YA/TIDAK)? YA
Scoreboard berhasil direset.

===== ENTER COMMAND : SCOREBOARD

*** SCOREBOARD RNG ***
(1) BRO : 65

*** SCOREBOARD Diner DASH ***
(1) ILM : 110000

*** SCOREBOARD HANGMAN ***
--- Scoreboard kosong ðŸŸ¶ main yuk! ---

*** SCOREBOARD TOWER OF HANOI ***
--- Scoreboard kosong ðŸŸ¶ main yuk! ---

*** SCOREBOARD SNAKE ON METEOR ***
--- Scoreboard kosong ðŸŸ¶ main yuk! ---

*** SCOREBOARD 2048 ***
--- Scoreboard kosong ðŸŸ¶ main yuk! ---

*** SCOREBOARD LABTEK HUNT ***
--- Scoreboard kosong ðŸŸ¶ main yuk! ---

```

```

===== ENTER COMMAND : RESET SCOREBOARD
DAFTAR SCOREBOARD:
0. ALL
1. RNG
2. Diner DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. 2048
7. LABTEK HUNT
SCOREBOARD YANG INGIN DIHAPUS: 0
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SEMUA SCOREBOARD (YA/TIDAK)? YA
Scoreboard berhasil direset.

===== ENTER COMMAND : SCOREBOARD

*** SCOREBOARD RNG ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD Diner DASH ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD HANGMAN ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD TOWER OF HANOI ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD SNAKE ON METEOR ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD 2048 ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

*** SCOREBOARD LABTEK HUNT ***
--- Scoreboard kosong ðŸŸ© main yuk! ---

```

5.3 HISTORY <n>

Fitur yang dites
History
Hasil yang seharusnya diberikan
ENTER COMMAND: HISTORY 2 Berikut adalah daftar Game yang telah dimainkan 1. EIFFEL TOWER 2. RNG
ENTER COMMAND: HISTORY 8 Berikut adalah daftar Game yang telah dimainkan

1. EIFFEL TOWER 2. RNG 3. EIFFEL TOWER 4. RISEWOMAN 5. LUNCH SLOW
Data test
<pre>===== ENTER COMMAND : HISTORY 2 Berikut adalah daftar Game yang telah dimainkan : 1. 2048 2. RNG</pre>
<pre>===== ENTER COMMAND : HISTORY 7 Berikut adalah daftar Game yang telah dimainkann : 1. 2048 2. RNG 3. Diner DASH</pre>

5.4 RESET HISTORY

Fitur yang dites
Reset History
Hasil yang seharusnya diberikan
ENTER COMMAND: RESET HISTORY APAKAH KAMU YAKIN INGIN MELAKUKAN RESET HISTORY? YA History berhasil di-reset.
ENTER COMMAND: RESET HISTORY APAKAH KAMU YAKIN INGIN MELAKUKAN RESET HISTORY? TIDAK History tidak jadi di-reset. Berikut adalah daftar Game yang telah

dimainkan

1. EIFFEL TOWER
2. RNG
3. EIFFEL TOWER
4. RISEWOMAN
5. LUNCH SLOW

Data test

```
===== ENTER COMMAND : RESET HISTORY

APAKAH KAMU YAKIN INGIN MELAKUKAN RESET HISTORY (YA/TIDAK)? YA

History berhasil di-reset.

===== ENTER COMMAND : HISTORY 5

Tidak ada permainan yang pernah dimainkan.
```

5.5 HANGMAN

Fitur yang dites

Hangman

Hasil yang seharusnya diberikan

Tebakan sebelumnya: -

Kata: _ _ _ _

Kesempatan: 10

Masukkan tebakan: A

Tebakan sebelumnya: a

Kata: _ A _ A

Kesempatan: 10

Masukkan tebakan: b

Tebakan sebelumnya: ab

Kata: _ A _ A

Kesempatan: 9

Masukkan tebakan: M

Tebakan sebelumnya: abm

Kata: M A _ A
Kesempatan: 9
Masukkan tebakan: F

Tebakan sebelumnya: abmf
Kata: M A _ A
Kesempatan: 8
Masukkan tebakan: t

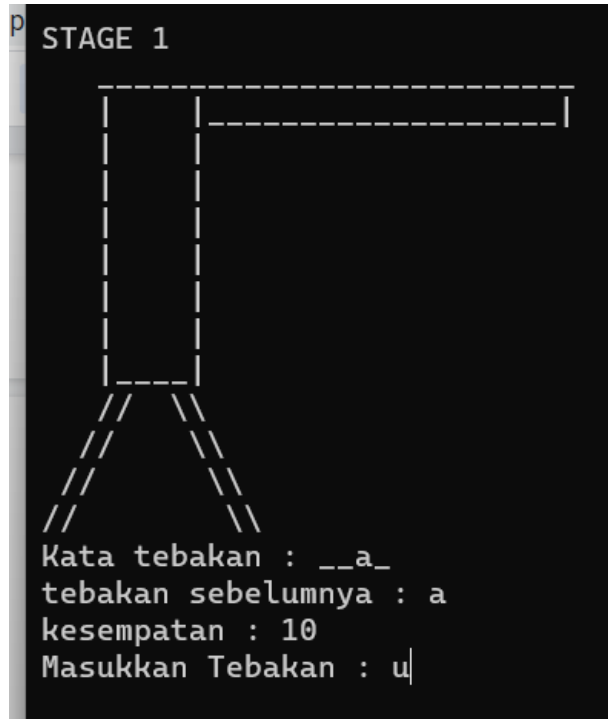
Berhasil menebak kata MATA! Kamu mendapatkan 4 poin!

Tebakan sebelumnya: -
Kata: _ _ _ _ _
Kesempatan: 8
Masukkan tebakan:

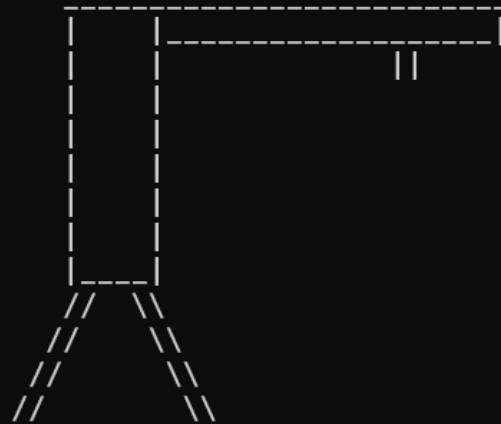
Permainan berlanjut hingga kesempatan habis

Data test



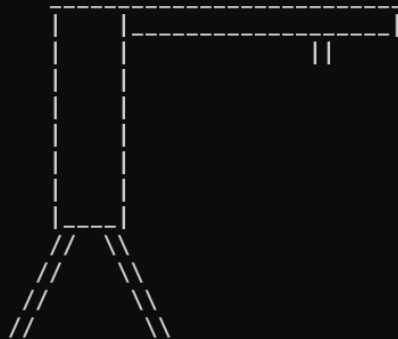


STAGE 1



Kata tebakan : ika_
tebakan sebelumnya : auik
kesempatan : 9
Masukkan Tebakan : n|

Selamat kamu berhasil menebak kata ikan
STAGE 2



Kata tebakan : _ _ _ _ _
tebakan sebelumnya : -
kesempatan : 9
Masukkan Tebakan : |



5.6 TOWER OF HANOI

Fitur yang dites					
Tower of Hanoi					
Hasil yang seharusnya diberikan					
Awal			Akhir		
<div><div><div>*</div><div>***</div><div>*****</div><div>*****</div><div>*****</div><div>-----</div><div>A</div></div><div><div></div><div></div><div></div><div></div><div></div><div>-----</div><div>B</div></div><div><div></div><div></div><div></div><div></div><div></div><div>-----</div><div>C</div></div></div> <div>TIANG ASAL: A</div> <div>TIANG TUJUAN: B</div> <div>Memindahkan piringan ke B...</div>			<div><div><div></div><div></div><div></div><div></div><div></div><div>-----</div><div>A</div></div><div><div></div><div></div><div></div><div></div><div></div><div>-----</div><div>B</div></div><div><div>*</div><div>***</div><div>*****</div><div>*****</div><div>*****</div><div>-----</div><div>C</div></div></div> <div>Kamu berhasil!</div> <div>Skor didapatkan: 10</div> <div>Nama: BNMO</div>		
Data test					
Awal:					

Penjelasan	Visualisasi																									
Kepala H: <4,2> Badan 1: <3,2> Badan 2: <2,2>	<table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>2</td><td>1</td><td>H</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>													2	1	H										
		2	1	H																						
Kepala H: <0,0> Badan 1: <0,1> Badan 2: <0,2>	<table><tr><td>H</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	H					1					2														
H																										
1																										
2																										
Kepala H: <1,1> Badan 1: <0,1> Badan 2: <0,0>	<table><tr><td>2</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>H</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	2					1	H																		
2																										
1	H																									
Data test																										

```

-----
|      |      |  2 |  1 |  H |
-----
|      |      |      |      |      |
-----
|      |  o  |      |      |      |
-----
|      |      |      |      |      |
-----
|      |      |      |||||
-----

TURN 1
Silahkan masukkan command: |

```

```

Berhasil gerak!

-----
|      |      |      |      |      |
-----
|      |      |      |      |  m |
-----
|      |  o  |  H |  1 |  2 |
-----
|      |      |      |      |      |
-----
|      |      |      |||||
-----

Selamat! Kamu gak kena meteor!

TURN 5
Silahkan masukkan command: |

```

Berhasil gerak!

| | | | | o |

| | | 4 | | |

| | 2 | 3 | | |

| | 1 | H | | |

| | | ||||| m |

Selamat! Kamu gak kena meteor!

TURN 8

Silahkan masukkan command: |



6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	SCORE-BOARD	Memeriksa fitur scoreboard	Melakukan input SCOREBOARD	SCOREBOARD	Scoreboard ditampilkan	Sesuai harapan
2	RESET SCORE-BOARD	Memeriksa fitur reset scoreboard	Melakukan input RESET SCOREBOARD untuk semua scoreboard	RESET SCOREBOARD 0	Scoreboard semua game ter-reset	Sesuai harapan
3	HISTORY <n>	Memeriksa fitur history	Melakukan input HISTORY setelah memainkan 3 game	HISTORY 2	Dua game terlama yang dimainkan muncul di history	Sesuai harapan
4	RESET HISTORY	Memeriksa fitur reset history	Melakukan reset HISTORY setelah memainkan game	RESET HISTORY	History ter-reset	Sesuai harapan

5	HANGMAN	Memeriksa game hangman	Melakukan penembakan kata	C I C K	Saat menginput huruf-huruf, huruf yang terdapat pada kata yang sedang ditebak akan muncul pada Hangman. Saat salah menginput huruf, muncul potongan gambar orang yang sedang digantung. Saat menginput huruf yang sama dua kali, kesempatan tidak berkurang.	Sesuai harapan
6	TOWER OF HANOI	Memeriksa game Tower of Hanoi	Menentukan piringan dan memindahkan piringan dari tiang	1. 4 2. A 3. A 4. A B	4 piringan muncul pada tower of Hanoi. Piringan terkecil di A pindah ke B. Piringan tidak berpindah karena tiang tujuan dan asal sama. Piringan tidak berpindah karena piringan teratas B lebih kecil daripada piringan teratas A.	Sesuai harapan
7	SNAKE ON METEOR	Memeriksa Snake on Meteor	Memindahkan snake, memakan makanan, dan menabrakkan diri ke obstacle	A D S	Snake bergerak ke kanan. Snake memakan makanan, dan badannya bertambah besar. Snake menabrak	Sesuai harapan

					obstacle, dan game over.	
8	LABTEK HUNT	Memeriksa game Labtek Hunt	Mencari treasure di ruangan-ruangan	1 2 1 3	Naik ke lantai selanjutnya, lalu masuk ke salah satu ruangan dan mendapatkan treasure. Naik ke lantai selanjutnya, lalu masuk ke salah satu ruangan dan kena bom.	Sesuai harapan

7 Pembagian Kerja dalam Kelompok

NIM - Nama	Fitur
18221067 - Fawwaz Abrial Saffa	Snake on Meteor, Reset history, Menggabungkan kode, penyesuaian fungsi lama
18221069 - Gibran Fasha Ghazanfar	Tower of Hanoi, History
18221087 - Willy Frans Farel Sijabat	Hangman, ADT set, ADT set_hangman, ADT Map
18221101 - Ilmagita Nariswari	Reset scoreboard, ADT linked list, ADT tree, bonus game, laporan
18221163 - AUFAR Ramadhan	View scoreboard, ADT stack, penyesuaian fungsi lama

8 Lampiran

8.1 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.

8.2 Notulen Rapat

20 November 2022 - LINE

Pembagian kerja

1 Desember 2022 - Labtek V ITB Ganesha

Penggabungan program dan pengecekan main

8.3 Log Activity Anggota Kelompok

Tanggal	Pengerjaan
20 November 2022	Setiap anggota melakukan rapat
1 Desember 2022	Setiap anggota menggabungkan program dan mengecek <i>main</i>

8.4 Lampiran Form Asistensi





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



No. Kelompok/Kelas : 7/K01
Nama Kelompok : FGWIA
Anggota Kelompok (Nama/NIM) :
1. Fawwaz Abrial Saffa/18221067
2. Gibran Fasha Ghazanfar/18221069
3. Willy Frans Farel Sijabat/18221087
4. Ilmagita Nariswari/18221101
5. Aufar Ramadhan/18221163

Asisten Pembimbing : Kadek Surya Mahardika (13519165)





Asistensi I



Tanggal : 25 November 2022	Catatan Asistensi: Q : Apakah meteor boleh jatuh ke makanan? A : Gak boleh. Q : Buat scoreboard harus make set gak kak? soalnya kan sebenarnya key di map itu berfungsi sebagai set juga. A : Iya harus, alesannya si biar set kepake aja. Diantara asisten juga lagi didiskusiin bakal diubah apa enggak, tapi blom ada jawaban.
Tempat : Zoom	

<p>Kehadiran Anggota Kelompok:</p> <p>No NIM Tanda tangan</p> <p>1 18221067</p>  <p>2 18221069</p>  <p>3 18221087</p>  <p>4 18221101</p>  <p>5 18221163</p>	<p>Q : Kira2 kalo bakal diubah waktu dikasihtau ke kaminya kapan ya? A : Enggak tau.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------

 <p>6</p>	
	<p>Tanda Tangan Asisten:</p>  <p>Kadek Surya Mahardika 13519165</p>

Asistensi II

Tanggal : 1 Desember 2022	Catatan Asistensi: <ul style="list-style-type: none"> - Progres laporan - Progres program
Tempat : Zoom	
Kehadiran Anggota Kelompok:	
<div style="text-align: center;"> <p>No</p> <p>NIM</p> <p>Tanda tangan</p> <p>1</p> <p>18221067</p>  <p>2</p> <p>18221069</p>  <p>3</p> <p>18221087</p>  <p>4</p> <p>18221101</p>  <p>5</p> </div>	

<p>18221163</p>  <p>6</p>	
	<p>Tanda Tangan Asisten:</p>  <p>Kadek Surya Mahardika 13519165</p>

8.5 Link Github

<https://github.com/fawwazabrians/Tugas-Besar-IF2111>