

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data

THE BNMO


Dipersiapkan oleh:

Kelompok 10

Aulia Nadhirah Yasmin. B / 18221066
Rifqi Syafiq / 18221070
Hilmi Baskara Radanto / 18221072
Akmal Natha Abirama / 18221082
Farhan Algani Putra / 18221108

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Elektro Informatika ITB	Teknik dan	Nomor Dokumen		Halaman
			<i>IF2111-TB1-10</i>		38
			<i>Revisi</i>	-	02/12/2022

Daftar Isi

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	5
2.1	Fitur memilih jumlah disk pada game Tower Of Hanoi	5
2.2	Fitur menghubungkan sisi peta yang berseberangan pada game Snake On Meteor	5
2.3	Fitur penambahan obstacle pada game Snake On Meteor	6
2.4	Fitur penggunaan file listkata.txt pada game HANGMAN	6
2.5	Fitur penambahan library pada game HANGMAN	6
2.6	Game Slay The Dragon	6
3	Struktur Data (ADT)	7
3.1	Stack	7
3.2	Map	7
3.3	Tree	7
3.4	Linked List	7
4	Program Utama	7
5	Algoritma-Algoritma Menarik	8
5.1	Display yang Adaptif	8
6	Data Test	9
6.1	History	9
6.2	Reset History	10
6.3	Scoreboard	10
6.4	Reset Scoreboard	12
6.5	Tower Of Hanoi	14
6.6	Hangman	17
6.7	Snake On Meteor	23
6.8	Slay The Dragon	30
7	Test Script	31
8	Pembagian Kerja dalam Kelompok	33
9	Lampiran	34
9.1	Deskripsi Tugas Besar 2	34
9.2	Notulen Rapat	35
9.3	Log Activity Anggota Kelompok	37

1 Ringkasan

Robot video game console BNMO telah diperbaiki dan ternyata ditemukan lebih banyak bug pada sistemnya. Sehingga, BNMO perlu diprogram ulang dan perlu dilakukan pengembangan fitur lebih lanjut. Adapun daftar fitur yang ditambah adalah:

a. Scoreboard

Ketika permainan sudah selesai, program akan meminta nama pemain yang hanya berupa satu kata untuk disimpan di dalam ADT Map Scoreboard. Dalam satu permainan, tidak boleh ada nama yang sama. Scoreboard pada setiap permainan diurutkan berdasarkan skor dari besar ke kecil. Jika ada skor yang sama, maka skor yang terbaru dimasukkan ke scoreboard ditampilkan di bawah dibanding yang sebelumnya. Jika suatu permainan belum mempunyai daftar scoreboard maka akan ditampilkan “Scoreboard kosong”.

b. Reset Scoreboard

Menghapus scoreboard pada permainan yang dipilih oleh pemain. Terdapat pula pilihan untuk melakukan reset terhadap scoreboard pada semua permainan.

c. History <n>

Menampilkan daftar urutan permainan apa saja yang sudah dimainkan oleh pemain sebanyak n. History permainan diurut dari permainan yang terakhir dimainkan hingga yang paling awal dimainkan. Jika input <n> lebih dari history yang sudah ada, maka akan menampilkan seluruh history permainan yang ada.

d. Reset History

Menghapus semua history permainan yang ada.

Kemudian, dilakukan pula penambahan permainan pada console BNMO. Adapun daftar permainan yang ditambah adalah:

a. Hangman

Hangman adalah permainan tebak kata yang memiliki beberapa spesifikasi. Pada awal permainan, Program akan menentukan satu kata dan pemain memiliki 10 kesempatan untuk menebak huruf yang tidak ada di dalam kata tersebut. Jika huruf ada di dalam kata, maka huruf yang tertebak akan ditampilkan. Jika salah, kesempatan pemain berkurang satu. Pemain tidak boleh menebak huruf yang sudah ditebak sebelumnya pada kata yang sama. Jika seluruh huruf pada satu kata telah lengkap ditebak, maka akan diberikan poin dan pemain terus berlanjut hingga pemain kehabisan kesempatan.

b. Tower of Hanoi

Tower of Hanoi adalah permainan yang terdiri dari 3 tiang, yaitu tiang A, B, dan C. Pada awal permainan, pemain akan diminta untuk memasukkan banyaknya piringan untuk dimainkan. Kemudian, seluruh piringan akan tertumpuk di tiang A dari yang paling besar di bawah hingga yang paling kecil di atas. Pemain diminta untuk memindahkan seluruh piringan tersebut dengan urutan yang sama dengan tiang A awal pada tiang C. Terdapat beberapa peraturan yang ada: piringan yang berada di atas tidak boleh lebih besar dibandingkan yang ada di bawah, skor yang didapat pemain bergantung dari seberapa banyak langkah yang dibutuhkan untuk menyusun piringan.

c. Snake on Meteor

Snake on Meteor merupakan game yang seperti snake pada konsol lama tetapi ditambah dengan fitur meteor. Permainan dimulai dengan ular dengan tanda H yang

berarti kepala dan dilanjutkan dengan angka berurut dari 1 sebagai badan ular. Permainan berlanjut hingga seluruh komponen snake terkena meteor hingga panjang snake adalah 0 atau kepala snake terkena meteor atau ekor baru tidak dapat dispawn karena tidak ada area di kiri, atas, bawah, atau kanan ekor lama.

d. Slay the Dragon

Slay The Dragon merupakan game tambahan yang mengambil inspirasi dari game Pokemon milik Nintendo. Slay The Dragon merupakan game *turn-based combat* dengan pemain sebagai pemegang giliran pertama. Pemain memiliki empat opsi *move* yang dapat dipilih dengan batasan penggunaan masing-masing. Game akan terus berlanjut sampai *health bar* pemain atau lawan mencapai nol.

Laporan ini mencakup deskripsi umum dari persoalan program pada bagian 1, penjelasan tambahan mengenai spesifikasi program pada bagian 2, penjelasan mengenai struktur data (ADT) pada bagian 3, penjelasan mengenai algoritma program utama pada bagian 4, algoritma menarik yang ditemukan selama pengerjaan tugas besar pada bagian 5, hasil tes data pada bagian 6 dan skenario pada bagian 7, pembagian kerja dalam kelompok pada bagian 8, dan diakhiri lampiran pada bagian 9.

Tugas Besar 2 BNMO ini sangat membantu kami dalam memahami dan mengimplementasikan struktur data (ADT) yang lebih kompleks dibandingkan sebelumnya, yaitu Linked List, Stack, dan Map. Belajar dari pengalaman sebelumnya, kami meningkatkan koordinasi supaya tidak terjadi miskomunikasi serta kesalahpahaman dan melakukan pengerjaan program lebih awal supaya tidak terlalu terbebani saat sudah mendekati deadline.

2 Penjelasan Tambahan Spesifikasi Tugas

Ada beberapa fitur yang kami tambahkan di luar dari fitur spesifikasi wajib.

2.1 Fitur memilih jumlah disk pada game Tower Of Hanoi

Pada saat memulai game Hanoi, pengguna akan pertama kali ditanyakan jumlah disk yang akan digunakan untuk bermain game tersebut. Fitur ini implementasikan pada fungsi *isiTiang* yang terdapat pada *stack.c* dan *stack.h*, Fungsi *isiTiang* akan melakukan proses transversal dari *n* (input pengguna) sampai dengan 1. menggunakan increment variabel *i*. Lebar maksimum dari disk yang akan dibuat adalah $n^2 - 1$. Panjang disk pada setiap levelnya adalah $((2^n - 1) - (2^{i-1} - 1)) / 2$ yang disimpan dalam variabel *half*.

Loop transversal akan menentukan level dimana disk tersebut berada, didalam disk tersebut ada Loop While yang melakukan proses sebanyak lebar maksimum tower tersebut. Loop While akan membuat variabel *graph* yang merupakan bagian dari struct yang merepresentasikan setiap disk. Graph berfungsi untuk menyimpan aspek visual dari disk. Loop While akan menambahkan karakter BLANK sebelum variabel increment yaitu *j*, lebih besar daripada *half*. Setelah lebih besar daripada *half*, hingga *j* lebih besar daripada $2^n - 1 - \text{half}$. Setelah itu, while loop akan terus menambahkan karakter BLANK. Setelah proses tersebut, disk yang elemennya sudah di set dari awal akan di Push kedalam stack yang ada.

2.2 Fitur menghubungkan sisi peta yang berseberangan pada game Snake On Meteor

Pada saat Head dari Snake berada di ujung map ($x = 0$, $y = 0$, $x = 4$, atau $y = 4$), maka pemain dapat memilih untuk melanjutkan gerakan “menembus” peta dari permainan Snake yang membuat Head dari Snake berada di posisi yang baru yaitu di ujung sisi yang berseberangan dengan posisi sebelum bergerak. Apabila Head dari Snake melewati sisi atas peta, maka Head akan muncul dari sisi bawah peta dan begitu pula sebaliknya. Hal yang sama juga terjadi apabila Head dari Snake melewati sisi kiri atau kanan peta, Head akan muncul dari sisi yang berseberangan.

2.3 Fitur penambahan obstacle pada game Snake On Meteor

Pada saat memulai game, akan di-generate obstacle pada sembarang titik dengan jumlah obstacle beragam diantara 1 sampai 3. Obstacle tidak bisa dilalui oleh Snake serta tidak bisa pula ada food atau meteor yang muncul di titik obstacle. Apabila Head dari Snake mengenai obstacle, maka permainan berakhir. Obstacle tidak membuat Head dari Snake hilang.

2.4 Fitur penggunaan file listkata.txt pada game HANGMAN

Pada saat memulai game, fungsi akan load listkata.txt dan memasukkan kata-kata yang ada di dalam txt tersebut ke dalam library.

2.5 Fitur penambahan library pada game HANGMAN

Pada saat memulai game, pengguna akan ditawarkan untuk menambahkan kata untuk ditebak dalam game hangman tersebut. Kata yang dimasukkan pengguna akan ditulis kedalam listkata.txt sebagai kata tebakan yang baru.

2.6 Game Slay The Dragon

Berikut adalah spesifikasi game SLAY THE DRAGON:

- Permainan merupakan *turn-based combat* yang berlangsung antara pemain dan lawan, dimana pemain dan lawan memiliki *health bar* masing-masing sebesar 100 poin.
- Permainan dimulai dengan suatu narasi, lalu pemain akan diminta untuk memilih diantara dua opsi, yaitu ATTACK atau ACTION.
- Pemain akan diberikan empat opsi yang berbeda berdasarkan pilihan sebelumnya, apabila memilih ATTACK akan diberikan pilihan berupa SPELL atau SWORD, sedangkan apabila memilih ACTION akan diberikan pilihan berupa HEAL atau SKIP.
- Masing-masing opsi memiliki batas penggunaan, sehingga pemain hanya bisa menggunakan tiap opsi secara terbatas.
- HEAL akan meningkatkan *health bar* pemain, SKIP akan membuang giliran untuk meningkatkan batasan penggunaan opsi yang lain sebanyak satu, dan SWORD serta SPELL digunakan untuk menyerang lawan.

- Apabila masukan tidak valid, maka pemain dianggap telah menggunakan gilirannya lalu giliran akan diberikan kepada lawan.
- Lawan memiliki tiga jenis opsi yang dipilih secara random melalui *random number generator*, yaitu FIRE BREATH, CLAW, atau MISS.
- Setelah pemain dan lawan telah menggunakan gilirannya akan diberikan info seberapa besar *health bar* masing-masing, lalu permainan akan berlanjut sampai salah satu *health bar* telah habis.
- Apabila permainan telah selesai, akan ditampilkan apakah pemain menang atau kalah
- Pemain kalah ketika *health bar* sudah menyentuh angka nol sedangkan pemain menang apabila *health bar* lawan sudah menyentuh nol.
- Skor akan bernilai sama dengan besar poin *health bar* pemain apabila pemain menang sedangkan apabila kalah akan bernilai nol.

3 Struktur Data (ADT)

3.1 Stack

ADT Stack yang digunakan dalam program BNMO diimplementasikan dengan Array dengan Elemen type berupa struct yang berisi Word yang merupakan graph dan Integer yang merupakan label. ADT Stack diimplementasikan di file Stack.h dan Stack.c dan digunakan untuk permainan Tower of Hanoi. ADT Stack dibutuhkan karena Tower of Hanoi menggunakan fungsi utama dalam Stack yaitu Pop dan Push untuk memindahkan disk yang ada.

3.2 Map

ADT Map merupakan kumpulan pasangan <key,value> dengan key bersifat unik dalam kumpulan tersebut. Map kami pakai untuk implementasi fitur SCOREBOARD karena user bernilai unik (tidak boleh ada yang sama) dan setiap user menyimpan nilai score. Pada BNMO, Map kami implementasikan dengan key yang menyimpan nilai user dalam tipe ADT Word dan value yang menyimpan nilai score dalam tipe integer. ADT Map ini kami simpan dalam sebuah array yang berisi kumpulan map untuk mempermudah akses scoreboard pada setiap permainan.

3.3 Tree

ADT Tree merupakan suatu himpunan yang terdiri dari akar serta dua buah himpunan lain yang *disjoint* yang merupakan pohon biner, disebut sebagai sub-pohon kiri dan sub-pohon kanan dari pohon tersebut. ADT ini menyimpan Left dan Right yang masing-masing merupakan pointer ke akar sub-pohon kiri dan kanan. Tiap akar dan daun berisikan infotype yang berupa integer. ADT Tree diimplementasikan di file tree.h dan tree.c dan digunakan untuk permainan Slay The Dragon. ADT ini digunakan karena sesuai dengan spesifikasi game Slay The Dragon dan mempermudah pemrosesan *move* pemain dalam *turn-based combat*.

3.4 Linked List

ADT Linked List merupakan sebuah list yang diimplementasikan dengan struktur berkait dengan alternatif double pointer. ADT ini menyimpan First dan Last yang merupakan pointer ke

elemen list dengan Elemen type berupa struct yang berisi infotype yang merupakan integer, next dan previous yang merupakan pointer ke elemen list lain, dan point yang merupakan point atau koordinat dari masing-masing elemen list. ADT Linked List diimplementasikan di file listdp.h dan listdp.c dan digunakan untuk permainan Snake On Meteor. ADT Linked List ini dibutuhkan karena kesesuaiannya dengan spesifikasi yang ditentukan dan juga mempermudah pemrosesan list yaitu Snake itu sendiri di dalam permainan Snake On Meteor.

4 Program Utama

Program utama BNMO dijalankan dengan file bernama main.c. Pada tugas besar 2 kali ini ditambahkan terlebih dahulu beberapa include dari header file fitur dan ADT baru dibandingkan sebelumnya pada tugas besar 1. ADT baru yang ditambahkan adalah stack, map, dan linked list. Fitur baru yang ditambahkan adalah HISTORY, RESET HISTORY, SCOREBOARD, RESET SCOREBOARD. Permainan baru yang ditambahkan adalah Hangman, Tower of Hanoi, dan Snake on Meteor.

Program dimulai dengan tambahan deklarasi variabel dengan ADT baru, yaitu Stack untuk History dan Map untuk Scoreboard. Kemudian, program meminta pemain memasukkan command START atau LOAD <filename>. Terdapat beberapa perubahan pada fitur LOAD dimana LOAD kini membaca file .txt yang berisi tambahan data, yaitu history yang akan disimpan di dalam StackHistory SH dan scoreboard yang akan disimpan di dalam TabScore TS. Ketika program sudah berhasil membaca *file* data, program akan berlanjut ke looping menu utama selama kondisi boolean on = true.

Pada menu utama, terdapat beberapa fitur tambahan yang dapat diakses oleh pengguna dengan memasukkan command yang sesuai, yaitu:

- a. HISTORY untuk menampilkan riwayat permainan. Ketika pemain selesai memainkan suatu permainan, nama permainan tersebut akan otomatis masuk ke riwayat permainan dengan memanfaatkan fungsi PushHistory() yang digunakan di dalam file playgame.c.
- b. RESET HISTORY untuk menghapus riwayat permainan yang sudah ada dengan memanfaatkan fungsi MakeEmptyHistory().
- c. SCOREBOARD untuk menampilkan scoreboard dari setiap permainan. Ketika pemain selesai memainkan suatu permainan, program akan meminta nama pemain. Kemudian, nama pemain dan skor akan dimasukkan ke dalam scoreboard dengan memanfaatkan fungsi InsertMap() yang digunakan di dalam file playgame.c.
- d. RESET SCOREBOARD untuk menghapus scoreboard dari permainan yang dipilih oleh pemain

Kemudian, terdapat perubahan pada fitur SAVE dimana SAVE menulis data history dan scoreboard terakhir pada file .txt. Ketika fungsi QUIT dipanggil, program akan memberi pilihan bagi pengguna untuk melakukan save terlebih dahulu atau tidak. Jika pengguna tidak memilih save, maka variabel on akan langsung bernilai false sehingga looping berhenti dan program akan berhenti. Jika memilih save, maka akan dilakukan pemanggilan fungsi saveGame() terlebih dahulu sebelum keluar dari program.

5 Algoritma-Algoritma Menarik

5.1 Display yang Adaptif

Algoritma ini membuat Game Tower of Hanoi memiliki display game yang adaptif sesuai dengan jumlah disk yang dimasukkan oleh pengguna. Terdapat dua fungsi yang melaksanakan algoritma ini, yaitu `printDivider` dan `printLabel`, keduanya terdapat di `Hanoi.c` dan `Hanoi.h`. `Divider` mencetak karakter '+' pada awal dan akhir divider dan mencetak karakter '-' sebanyak $((2*n-1)*3)+20$. Angka ini didapatkan karena lebar masing masing tower adalah $2*n-1$, dan terdapat tiga buah tower, ditambahkan 20 karena antar tower terdapat spacing sebanyak 8 karakter BLANK, sehingga ditambahkan 16 karakter dengan ditambahkan 4 buah karakter BLANK lainnya untuk spacing sisi paling kiri tower.

Algoritma `print label` menuliskan label yang ada di bawah tower A, B dan C. Pertama-tama algoritma ini akan mencetak karakter '[' sebagai pembatas kiri label kemudian dilanjutkan dengan for loop selama 3 kali dimana dalam setiap loop variabel `j` sebagai counter berapa steps yang telah dia lakukan. Kemudian sebanyak setengah dari panjang tower, fungsi akan mencetak karakter BLANK. Di tengah tower, tergantung kepada variabel `i` yaitu increment dari for loop, akan di print A, B, atau C. Diikuti dengan mencetak 8 buah karakter yang menjadi spacer antar tower, kecuali jika for loop ada dalam loop ketiga, karena setelah loop ketiga `printLabel` akan mencetak suatu garis pembatas '[' berdasarkan apakah counter sudah sama dengan panjang `printDivider` atau belum. Fungsi `printDivider` dan `printLabel` akan memberikan panjang yang sama.

6 Data Test

Pada bagian ini akan dijelaskan dan ditampilkan hasil tes dari fitur yang dimiliki pada program BNMO

6.1 History

Ketika pemain telah selesai memainkan sebuah permainan, permainan tersebut akan ditambahkan ke history. Program akan menampilkan history sebanyak `<n>` teratas sesuai dengan yang diinginkan pemain dengan command berupa `HISTORY <n>`. Ketika pemain melakukan input lebih dari banyaknya history yang ada, maka akan ditampilkan seluruh history yang ada.

Gambar 6.1.1 Tiga history teratas sebelum bermain RNG

```
ENTER COMMAND : HISTORY 3
Berikut adalah daftar Game yang telah dimainkan:

1. HANGMAN
2. TOWER OF HANOI
3. SNAKE ON METEOR
```

Gambar 6.1.2 Tiga history teratas setelah bermain RNG


```

ENTER COMMAND : HISTORY 3
Berikut adalah daftar Game yang telah dimainkan:

1. RNG
2. HANGMAN
3. TOWER OF HANOI

```

Ketika pemain melakukan input lebih dari banyaknya history yang ada, maka akan ditampilkan seluruh history yang ada.

Gambar 6.1.3 Seluruh permainan pada history

```

ENTER COMMAND : HISTORY 9
Berikut adalah daftar Game yang telah dimainkan:

1. RNG
2. HANGMAN
3. TOWER OF HANOI
4. SNAKE ON METEOR

```

6.2 Reset History

Program akan menghapus history yang ada. Sebelum history akan benar-benar dihapus, program akan meminta kepastian kembali kepada pemain.

Gambar 6.2.1 Menghapus history

```

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

History berhasil dihapus!
ENTER COMMAND :

```

Gambar 6.2.2 History kosong karena sudah dihapus

```

ENTER COMMAND : HISTORY
Kamu belum punya history game! >:(

ENTER COMMAND :

```

6.3 Scoreboard

Setiap suatu permainan berakhir, pemain mendapatkan skor akhir dan program akan meminta input nama pemain untuk disimpan di scoreboard yang hanya berupa satu kata dalam bentuk Word.

Gambar 6.3.1 Permainan berakhir, ditampilkan skor akhir dan program meminta nama pemain

```

Loading GAME ASAL....

  G A M E   O V E R

Skor akhir : 100

Masukkan nama: MARSHA

Berhasil menambahkan score ke SCOREBOARD!

```

Kemudian, ketika sudah kembali ke menu awal dan pemain memasukkan command SCOREBOARD, akan ditampilkan daftar scoreboard pada semua permainan. Daftar pemain dan skor pada scoreboard diurutkan berdasarkan skor dari tertinggi ke terendah.

Gambar 6.3.2 Tampilan Scoreboard. Pemain bernama MARSHA ditambahkan pada game asal

```

ENTER COMMAND : SCOREBOARD

***** SCOREBOARD GAME RING *****
| NAMA          | SKOR          |
|-----|-----|
| BNMO          | 19            |
| Finn          | 12            |
|-----|-----|

***** SCOREBOARD GAME Diner DASH *****
| NAMA          | SKOR          |
|-----|-----|
| Jake          | 58            |
| Finn          | 31            |
| Marceline     | 30            |
|-----|-----|

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

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

```

```

***** SCOREBOARD GAME SNAKE ON METEOR *****
| NAMA          | SKOR          |
|-----|
| Marshall      | 77            |

***** SCOREBOARD GAME ROCK PAPER SCISSOR *****
| NAMA          | SKOR          |
|-----|
| Naruto        | 90            |

***** SCOREBOARD GAME GAME ASAL *****
| NAMA          | SKOR          |
|-----|
| MARSHA        | 100           |
| Sasuke        | 89            |

ENTER COMMAND :

```

Jika pemain sudah memasukkan nama yang sudah ada pada scoreboard permainan tersebut, maka skornya akan ter-*update* ke skor yang baru. Dan jika terdapat skor yang sama, maka skor yang terbaru dimasukkan ke scoreboard ditampilkan di bawah dibanding yang sebelumnya.

Gambar 6.3.3 Skor Sasuke berubah

```

***** SCOREBOARD GAME GAME ASAL *****
| NAMA          | SKOR          |
|-----|
| MARSHA        | 100           |
| Sasuke        | 100           |

```

6.4 Reset Scoreboard

Program akan meminta pilihan bagi pemain untuk menghapus scoreboard total pada semua permainan atau hanya salah satu permainan saja. Pemain diminta untuk melakukan input 0 untuk menghapus semua scoreboard permainan atau input angka urutan permainan tertentu sesuai yang ditampilkan oleh program.

Gambar 6.4.1 Tampilan scoreboard sebelum scoreboard game asal dihapus

```

***** SCOREBOARD GAME GAME ASAL *****
| NAMA          | SKOR          |
|-----|
| MARSHA        | 100           |
| Sasuke        | 100           |

```

Gambar 6.4.2 Tampilan pilihan permainan yang scoreboardnya ingin dihapus

```

ENTER COMMAND : RESET SCOREBOARD

DAFTAR SCOREBOARD:

0. ALL
1. RNG
2. Diner DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. ROCK PAPER SCISSOR
7. GAME ASAL

SCOREBOARD YANG INGIN DIHAPUS: 

```

Gambar 6.4.3 Tampilan setelah scoreboard GAME ASAL berhasil dihapus

```

SCOREBOARD YANG INGIN DIHAPUS: 7

Scoreboard berhasil direset!

```

Gambar 6.4.4 Tampilan scoreboard GAME ASAL

```

***** SCOREBOARD GAME GAME ASAL *****
| NAMA          | SKOR          |
-----
----- SCOREBOARD KOSONG -----

```

Gambar 6.4.5 Tampilan scoreboard setelah menghapus ALL scoreboard

```

***** SCOREBOARD GAME RNG *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

***** SCOREBOARD GAME Diner DASH *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

***** SCOREBOARD GAME HANGMAN *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

***** SCOREBOARD GAME TOWER OF HANOI *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

***** SCOREBOARD GAME SNAKE ON METEOR *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

***** SCOREBOARD GAME ROCK PAPER SCISSOR *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

***** SCOREBOARD GAME GAME ASAL *****
| NAMA          | SKOR          |
|-----|-----|
|----- SCOREBOARD KOSONG -----|

```

6.5 Tower Of Hanoi

Program pada pertama kali akan meminta input kepada pengguna mengenai jumlah disk yang akan dimainkan dalam Tower of Hanoi.

Gambar 6.5.1 Meminta input jumlah disk kepada Pengguna

```

Sebelum mulai, tentukan jumlah disk ^_^
Tentukan jumlah disk :

```

Bila pengguna memasukkan input selain angka maka program akan terus meminta input kepada pengguna

Gambar 6.5.2 Pesan Error jika pengguna memasukkan input non-angka

```

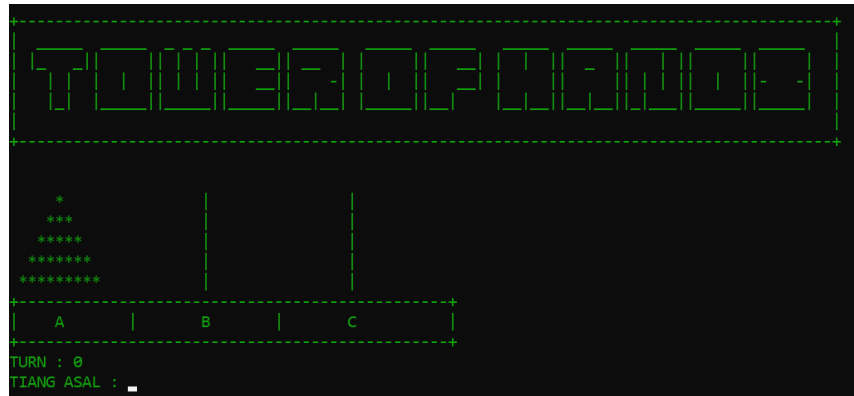
Tentukan jumlah disk : a

-- INPUT INVALID, HARUS BERUPA ANGKA --
Tentukan jumlah disk :

```

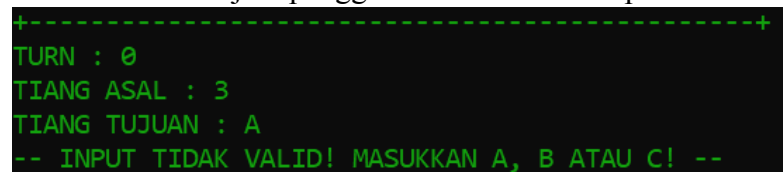
Kemudian, pengguna akan disuguhkan dengan splash screen yang berisi judul game dan tampilan disk sesuai dengan jumlah disk yang diberikan oleh pengguna.

Gambar 6.5.3 Tampilan Splash Screen



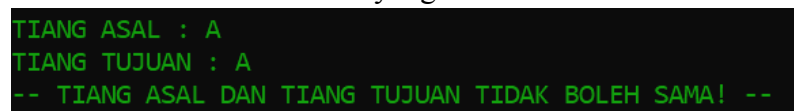
Pengguna akan diminta input Tiang asal dan Tiang Tujuan, ada beberapa Error Message yang akan dimunculkan bila pengguna tidak memasukkan input yang benar. Bila pengguna memasukkan input yang bukan merupakan A, B, C maka akan muncul pesan error berikut.

Gambar 6.5.4 Pesan Error jika pengguna memasukkan Input Selain A, B atau C



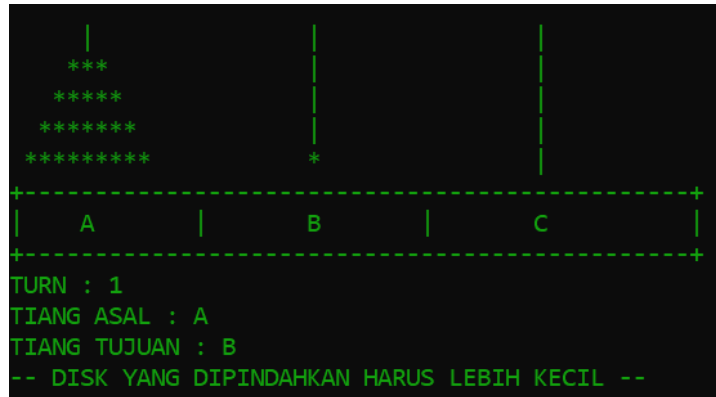
Jika pengguna memasukkan Tiang Asal dan Tiang Tujuan yang sama maka akan dimunculkan pesan error berikut.

Gambar 6.5.5 Pesan Error Jika Pengguna memasukkan Input Tiang Tujuan dan Tiang Asal yang sama



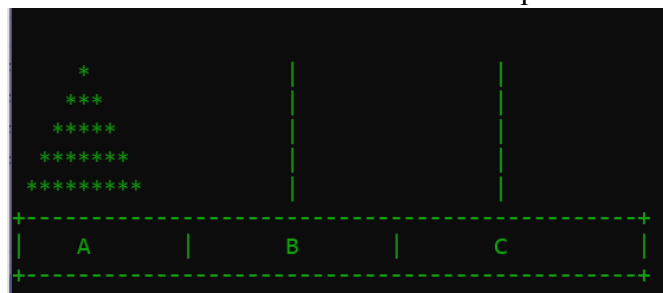
Jika pengguna memasukkan Input Tiang Asal yang memiliki elemen TOP yang lebih besar daripada Tiang Tujuan maka akan dimunculkan pesan error berikut.

Gambar 6.5.6 Pesan Error Jika Pengguna memasukkan Input Tiang Asal yang memiliki elemen TOP yang lebih besar daripada Input Tiang Tujuan

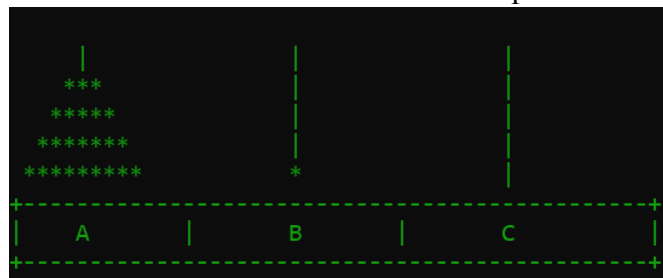


Jika pengguna telah memasukkan input yang valid maka Disk akan dipindahkan dari Tiang Asal ke Tiang Tujuan.

Gambar 6.5.7 Kondisi sebelum disk dipindahkan



Gambar 6.5.8 Kondisi setelah disk dipindahkan



Setelah itu, Pengguna akan terus dimintai input hingga pengguna berhasil menyusun kembali tower di tower C. Setelah Pengguna berhasil menyusun Tower C maka splash screen Game Over akan muncul dan menampilkan score.

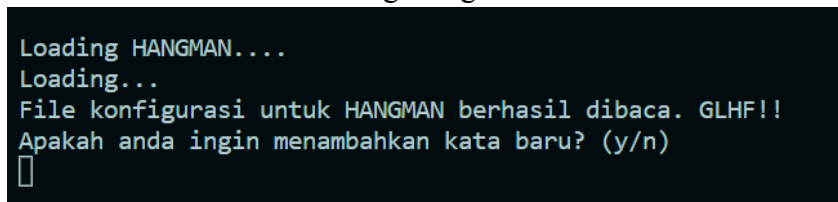
Gambar 6.5.9 Splash Screen Game Over



6.6 Hangman

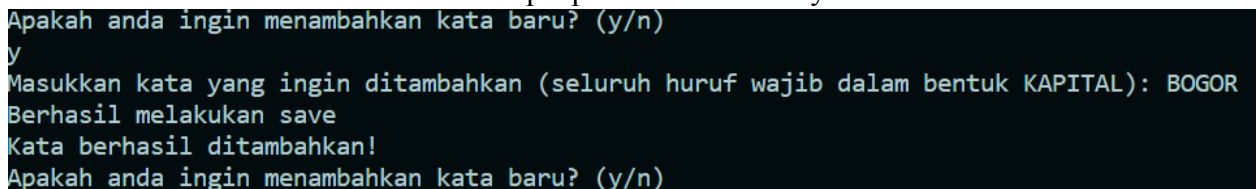
Saat program dijalankan, program hangman akan me-load file listkata.txt yang berisi seluruh kata yang bisa digunakan untuk nantinya ditebak oleh pengguna. Pengguna juga akan ditanyakan jika ia ingin menambahkan kata baru atau tidak. input yang dapat diproses hanya berupa y untuk yes/ya jika pengguna ingin menambahkan kata ke dalam permainan dan n untuk no/tidak jika pengguna ingin langsung bermain hangman.

Gambar 6.6.1 Loading Hangman serta menu awal



Jika pengguna memilih y, maka pengguna dapat memasukkan kata untuk nanti ditebak, jika kata yang dimasukkan belum ada di library list kata, maka kata akan dimasukkan ke library dan muncul pesan “kata berhasil ditambahkan!”

Gambar 6.6.2 Menu opsi penambahan library listkata.



Sedangkan jika kata yang dimasukkan sudah ada di library, maka akan muncul pesan bahwa kata sudah ada di library.

Gambar 6.6.3 Menu opsi penambahan listkata jika kata yang dimasukkan sudah ada.

```
Apakah anda ingin menambahkan kata baru? (y/n)
y
Masukkan kata yang ingin ditambahkan (seluruh huruf wajib dalam bentuk KAPITAL): TOKYO
Kata sudah ada dalam list kata! Masukkan kata lain
Apakah anda ingin menambahkan kata baru? (y/n)
```

Pengguna akan terus ditanyakan untuk menambahkan kata baru hingga pengguna menginput “n” saat ditanyakan apakah pengguna ingin menambahkan kata baru.

Gambar 6.6.4 Game utama Hangman

```
Apakah anda ingin menambahkan kata baru? (y/n)
n
Selamat bermain hangman!

Seluruh huruf yang telah dicoba:
Nyawamu masih ada 10 nih!

Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
_ _ _ _ _
Masukkan huruf tebakanmu! :
```

Jika pengguna telah selesai menambahkan kata dan menginput huruf ‘n’, pengguna akan langsung masuk ke game hangman, disini akan disediakan clue berapa banyak huruf dari kata yang pengguna coba untuk tebak dan seluruh huruf yang pengguna telah digunakan dalam menebak jawaban.

Gambar 6.6.5 Game Hangman jika tebakan yang dimasukkan salah

```
Masukkan huruf tebakanmu! :
x

Jawabanmu salah! Nyawamu berkurang satu!

_ _ _ | _ _ _

Seluruh huruf yang telah dicoba: X
Nyawamu masih ada 9 nih!

Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
- - - - -

Masukkan huruf tebakanmu! :
█
```

Jika input tebakan pengguna salah, maka nyawa pengguna akan berkurang satu dan akan mulai muncul rangkaian logo hangman menandakan bahwa nyawa pengguna semakin sedikit. Semakin lengkap logo hangman, semakin sedikit juga nyawa pengguna. Pengguna akan diberi 10 nyawa saat game dimulai sebelum untuk menebak jawaban yang benar.

Gambar 6.6.6 Game Hangman jika tebakan yang dimasukkan benar

```
Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
- - - - -

Masukkan huruf tebakanmu! :
a
Tebakan-mu Benar!

Seluruh huruf yang telah dicoba: X A
Nyawamu masih ada 9 nih!

Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
- - - - _ A

Masukkan huruf tebakanmu! :
█
```

Jika jawaban pengguna benar, maka huruf yang diinput pengguna akan muncul di bagian soal, dan membantu pengguna untuk menyusun huruf dengan benar. Seluruh huruf yang telah dimasukkan oleh pengguna akan muncul di bagian paling atas, dan pengguna dilarang untuk menggunakan huruf yang sudah masuk bagian itu / sudah pernah pengguna pakai sebelumnya.

Gambar 6.6.7 Game hangman jika tebakan yang dimasukan pernah dimasukkan sebelumnya

```
Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
_ _ _ _ _ A

Masukkan huruf tebakanmu! :
a
Yahh, kamu udah pernah nebak huruf ini, coba huruf lain!

Seluruh huruf yang telah dicoba: X A
Nyawamu masih ada 9 nih!

Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
_ _ _ _ _ A

Masukkan huruf tebakanmu! :
█
```

Jika pengguna berhasil menebak seluruh huruf yang ada di soal, maka pengguna akan mendapatkan poin sebanyak jumlah huruf yang ditebak, Soal akan berganti menjadi kata lain dan akan terus memberikan soal hingga nyawa pengguna habis. Seluruh huruf yang telah digunakan pengguna boleh digunakan lagi tetapi nyawa tidak ter-reset.

Gambar 6.6.8 Game hangman jika pengguna berhasil menebak seluruh huruf dengan benar

```
Seluruh huruf yang telah dicoba: X A V I E
Nyawamu masih ada 9 nih!

Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 6
VIE _ _ A

Masukkan huruf tebakanmu! :
n
Tebakan-mu Benar!

Yeayy! Kamu Menang :D, Kamu mendapatkan 6 poin!
Kata yang kamu tebak adalah VIENNA

Seluruh huruf yang telah dicoba:
Nyawamu masih ada 9 nih!

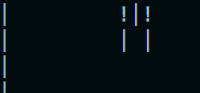
Lengkapi bagian yang kosong dibawah!:
Panjang Kata: 7
_ _ _ _ _ _ _

Masukkan huruf tebakanmu! :
█
```

Jika nyawa pengguna telah habis, maka logo hangman pun menjadi lengkap menandakan permainan telah selesai, akan ditampilkan skor akhir yang berupa akumulasi dari hasil skor pengguna setiap berhasil menebak kata.

Gambar 6.6.9 Game hangman jika nyawa telah habis / Game Over

```
Seluruh huruf yang telah dicoba: X Y U A S J K L N P Q  
Nyawamu masih ada 1 nih!  
  
Lengkapi bagian yang kosong dibawah!:  
Panjang Kata: 7  
Y _ _ _ _ AN  
  
Masukkan huruf tebakanmu! :  
g  
  
Jawabanmu salah! Nyawamu berkurang satu!
```



```
      |-----|  
      |         | O  
      |         |||  
      |         |||  
_____|_____
```

```
Kamu kalah :( Kata yang benar adalah: YEREVAN  
Skor akhir : 6
```

6.7 Snake On Meteor

Saat program dijalankan pertama kali, akan muncul tampilan awal beserta dengan posisi awal snake, obstacle, dan food.

Gambar 6.7.1 Game digenerate pertama kali

```
Loading SNAKE ON METEOR...
Selamat datang di Snake On Meteor!
Mengenerate peta, snake, obstacle, dan makanan...
Berhasil digenerate!
Berikut merupakan peta permainan:
[2 ][1 ][H ][ ][ ]
[ ][ ][ ][ ][ ][ ]
[ ][ ][X ][ ][ ]
[ ][ ][ ][o ][ ][ ]
[ ][ ][ ][ ][ ][ ]
```

Setelah itu, turn pertama dimulai dan pemain akan diminta input berupa ke arah mana snake ingin digerakkan. Apabila input adalah 'w', maka snake akan bergerak ke atas. Apabila input adalah 's', maka snake akan bergerak ke bawah. Apabila input adalah 'a', maka snake akan bergerak ke kiri. Apabila input adalah 'd', maka snake akan bergerak ke kanan.

Gambar 6.7.2 Snake bergerak ke bawah

```
TURN 1
Masukkan gerakan : s
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][2 ][1 ][ ][ ]
[ ][ ][H ][ ][ ]
[ ][ ][X ][ ][ ]
[ ][ ][ ][o ][ ][ ]
[ ][ ][ ][ ][m ]

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

Gambar 6.7.3 Snake bergerak ke kanan

```

TURN 2
Masukkan gerakan : d
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][2][ ][ ]
[ ][ ][1][H][ ]
[ ][ ][X][ ][ ]
[ ][ ][ ][o][m ]
[ ][ ][ ][ ][ ]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

```

Gambar 6.7.4 Snake bergerak ke atas

```

TURN 3
Masukkan gerakan : w
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][H][ ]
[ ][m][2][1][ ]
[ ][ ][X][ ][ ]
[ ][ ][ ][o][ ]
[ ][ ][ ][ ][ ]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

```

Gambar 6.7.5 Snake bergerak ke kiri

```

TURN 6
Masukkan gerakan : a
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][2][ ][ ]
[ ][H][1][ ][ ]
[ ][ ][ ][ ][ ]
[ ][ ][ ][m][ ]
[ ][o][ ][ ][X ]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 7
Masukkan gerakan : a
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[H][1][2][ ][ ]
[ ][ ][ ][m][ ]
[ ][ ][ ][ ][ ]
[ ][o][ ][ ][X ]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

```

Apabila snake menembus peta permainan, maka snake akan muncul di sisi seberangnya.

Gambar 6.7.6 Snake muncul di sisi yang berseberangan bila menembus peta

```

TURN 7
Masukkan gerakan : a
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[H][1][2][ ][ ]
[ ][ ][ ][m][ ]
[ ][ ][ ][ ][ ]
[ ][o][ ][ ][X]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 8
Masukkan gerakan : a
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[1][2][ ][ ][H]
[ ][ ][ ][ ][ ]
[ ][ ][ ][m][ ]
[ ][o][ ][ ][X]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

```

Apabila snake memakan food, maka badan snake akan bertambah panjang sebanyak 1

Gambar 6.7.7 Snake memakan food

```

TURN 2
Masukkan gerakan : d
Berhasil bergerak!
Berikut merupakan peta permainan:
[m][ ][ ][ ][ ]
[ ][X][ ][ ][ ]
[ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ]
[ ][2][1][H][o]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 3
Masukkan gerakan : d
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[ ][X][ ][ ][ ]
[ ][ ][o][ ][m]
[ ][ ][ ][ ][ ]
[ ][3][2][1][H]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

```

Apabila meteor mengenai badan snake, maka badan snake akan berkurang sebanyak 1 dan bagian tubuh selanjutnya akan tersambung ke bagian yang sebelumnya

Gambar 6.7.8 Meteor mengenai tubuh Snake

```

TURN 5
Masukkan gerakan : w
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[ ][X][ ][ ][ ]
[ ][ ][o][ ][H ]
[ ][ ][ ][ ][1 ]
[ ][ ][ ][3 ][m ]

Anda terkena meteor!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[ ][X][ ][ ][ ]
[ ][ ][o][ ][H ]
[ ][ ][ ][ ][1 ]
[ ][ ][ ][2 ][m ]

Silahkan lanjutkan permainan

```

Apabila snake bergerak ke tubuhnya sendiri, maka akan dimintai ulang input yang valid

Gambar 6.7.9 Snake bergerak ke tubuhnya sendiri

```

TURN 7
Masukkan gerakan : d
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][3 ][4 ][ ]
[ ][ ][2 ][m ][ ]
[ ][ ][1 ][H ][ ]
[ ][ ][ ][o ][ ]
[ ][ ][ ][ ][X ]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 8
Masukkan gerakan : a
Gerakan tidak valid! Masukkan ulang gerakan : 

```

Apabila snake bergerak ke titik yang terdapat meteor, maka akan dimintai ulang input yang valid

Gambar 6.7.10 Snake bergerak ke titik meteor


```

TURN 7
Masukkan gerakan : d
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][3][4][ ]
[ ][ ][2][m][ ]
[ ][ ][1][H][ ]
[ ][ ][ ][o][ ]
[ ][ ][ ][X][ ]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 8
Masukkan gerakan : a
Gerakan tidak valid! Masukkan ulang gerakan : w
Meteor masih panas! Anda belum dapat pergi ke titik tersebut.
Silahkan masukkan command lainnya
Masukkan gerakan : 

```

Apabila input yang dimasukkan pemain tidak valid, maka akan dimintai ulang inputnya

Gambar 6.7.11 Apabila memasukkan input yang tidak valid

```

Loading SNAKE ON METEOR....
Selamat datang di Snake On Meteor!
Mengenerate peta, snake, obstacle, dan makanan...
Berhasil digenerate!
Berikut merupakan peta permainan:
[X][ ][ ][ ][ ]
[2][X][ ][ ][ ]
[1][ ][ ][ ][ ]
[H][ ][ ][o][ ]
[ ][ ][ ][ ][ ]

TURN 1
Masukkan gerakan : HAHA
Gerakan tidak valid! Masukkan ulang gerakan : 

```

Apabila meteor mengenai Head dari snake, maka permainan selesai

Gambar 6.7.12 Meteor mengenai Head dari Snake

```

TURN 9
Masukkan gerakan : a
Berhasil bergerak!
Kepala Anda terkena meteor! Permainan berakhir!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[2][ ][ ][m][1][ ]
[ ][ ][ ][ ][ ]
[ ][ ][ ][ ][ ]
[ ][o][ ][ ][X][ ]

Silahkan lanjutkan permainan

Permainan berakhir!
Skor : 4
Skor akhir : 4

```

Apabila snake mengenai obstacle, maka permainan berakhir

Gambar 6.7.13 Snake mengenai obstacle

```
Berikut merupakan peta permainan:
[ ][ ][4][ ][ ]
[ ][ ][3][ ][ ]
[ ][ ][2][1][ ]
[o][ ][ ][m][ ][ ]
[ ][ ][ ][H][X ]

Silahkan lanjutkan permainan

TURN 10
Masukkan gerakan : d
Berhasil bergerak!
Berikut merupakan peta permainan:
[ ][ ][ ][ ][ ]
[ ][m][4][ ][ ]
[ ][ ][3][2][ ][ ]
[o][ ][ ][ ][ ][ ]
[ ][ ][ ][1][H ]

Anda menabrak obstacle! Permainan berakhir!

Permainan berakhir!
Skor : 10
Skor akhir : 10
```

Apabila snake tidak bisa bergerak kemana-mana lagi, maka permainan berakhir

Gambar 6.7.14 Snake tidak bisa bergerak lagi akibat terkunci

```
TURN 32
Masukkan gerakan : W
Berhasil bergerak!
Berikut merupakan peta permainan:
[X][ ][o][ ][ ]
[ ][4][5][6][ ][ ]
[ ][3][H][7][ ][ ]
[m][2][1][8][9][ ]
[ ][ ][ ][11][10]

Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

Anda sudah tidak bisa bergerak lagi!
Permainan berakhir!
Skor : 24
Skor akhir : 24
```

Apabila game berakhir, akan menampilkan tampilan akhir berupa skor yang diperoleh

Gambar 6.7.15 Tampilan akhir permainan

STEI- ITB	<nomor dokumen>	Halaman 26 dari 38 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.		

```
Permainan berakhir!  
Skor : 2  
Skor akhir : 2
```

6.8 Slay The Dragon

Saat program dijalankan akan muncul narasi, lalu pemain akan diminta untuk memasukkan input.

Gambar 6.8.1: Game saat di generate pertama kali

```

+++++
You slowly make your way up to snowy mountains,
the beast's lair.

You finally reached the lair's entrance.
You tried sneaking in but suddenly
a loud roar came from inside the cave.

Two yellow eyes emerge from the dark, followed by
the grand entrance of the Red Dragon.


      \\\|/
       | @__oo
    / ^\ / ^\ / (,,,|
   ) /^\) ^V )
  ) /^V )
 ) / / )
< > )/V || | )
     | (,,) )
    || / ^\ )
   | \__( ) _)_
   \___(_____) ;;; ___;

                                o
                               /\
                              /\

+++++
HP MONSTER : 100
YOUR HP : 100
THIS IS THE NUMBER OF MOVES AVAILABLE TO YOU:
ACTION:
1. HEAL: 3
2. SKIP: 5
ATTACK:
3. SWORD: 6
4. SPELL: 2

> WHAT DO YOU DO? (ATTACK/ACTION)
> 

```

Pemain dapat memilih opsi ATTACK atau ACTION. Apabila memilih ATTACK akan diberikan opsi lanjutan berupa SWORD atau SPELL sedangkan ACTION akan diberikan opsi lanjutan berupa HEAL atau SKIP

Gambar 6.8.2: Memilih ATTACK

```
> WHAT DO YOU DO? (ATTACK/ACTION)
> ATTACK
> WHAT ATTACK DO YOU WANT? (SWORD/SPELL)
> 
```

Gambar 6.8.3: Memilih ACTION

```
> WHAT DO YOU DO? (ATTACK/ACTION)
> ACTION
> WHAT ACTION DO YOU DO? (SKIP/HEAL)
> █
```

Gambar 6.8.4: Memilih SWORD

```
> WHAT ATTACK DO YOU WANT? (SWORD/SPELL)
> SWORD
You slashed the Dragon.

The Dragon swoops in... BUT IT MISSES!
+++++
HP MONSTER : 75
YOUR HP : 100
```

Gambar 6.8.5: Memilih SPELL

```
> WHAT ATTACK DO YOU WANT? (SWORD/SPELL)
> SPELL
You casted a spell and hit the Dragon.

The Dragon uses it's claws!
+++++
HP MONSTER : 35
YOUR HP : 110
```

Gambar 6.8.6: Memilih HEAL

```
You drank your healing potion and heal yourself.
HP +25

The Dragon swoops in... BUT IT MISSES!
+++++
HP MONSTER : 75
YOUR HP : 125
```

Gambar 6.8.7: Memilih SKIP

```

> WHAT ACTION DO YOU DO? (SKIP/HEAL)
> SKIP
You take a rest and prepare yourself.
Sword +1

The Dragon swoops in... BUT IT MISSES!
+++++
HP MONSTER : 20
YOUR HP : 50
THIS IS THE NUMBER OF MOVES AVAILABLE TO YOU:
ACTION:
1. HEAL: 3
2. SKIP: 4
ATTACK:
3. SWORD: 7
4. SPELL: 0

```

Apabila pemain memasukkan input yang tidak sesuai dengan yang diminta, akan dianggap telah menggunakan giliran lalu giliran akan diserahkan kepada lawan

Gambar 6.8.8: Salah memasukkan input

```

> WHAT DO YOU DO? (ATTACK/ACTION)
> SPELL
Command not recognized, the Dragon attacks you for wasting your turn!!

The Dragon uses Fire Breathing!
+++++
HP MONSTER : 35
YOUR HP : 75

```

Pemain juga memiliki batasan penggunaan tiap *move*. Setiap menggunakan sebuah *move*, batas penggunaan akan berkurang sebanyak satu. Apabila pemain mencoba menggunakan *move* yang tidak dapat digunakan lagi, maka akan dianggap sama seperti salah memasukkan input.

Gambar 6.8.9: Batas Penggunaan masing-masing *move* dan berkurang ketika digunakan

```

THIS IS THE NUMBER OF MOVES AVAILABLE TO YOU:
ACTION:
1. HEAL: 3
2. SKIP: 5
ATTACK:
3. SWORD: 6
4. SPELL: 2

> WHAT DO YOU DO? (ATTACK/ACTION)
> ATTACK
> WHAT ATTACK DO YOU WANT? (SWORD/SPELL)
> SWORD
You slashed the Dragon.

The Dragon swoops in... BUT IT MISSES!
+++++
HP MONSTER : 75
YOUR HP : 100
THIS IS THE NUMBER OF MOVES AVAILABLE TO YOU:
ACTION:
1. HEAL: 3
2. SKIP: 5
ATTACK:
3. SWORD: 5
4. SPELL: 2

```

Gambar 6.8.10: Menggunakan *move* yang sudah tidak bisa digunakan lagi

```

THIS IS THE NUMBER OF MOVES AVAILABLE TO YOU:
ACTION:
1. HEAL: 3
2. SKIP: 5
ATTACK:
3. SWORD: 6
4. SPELL: 0

> WHAT DO YOU DO? (ATTACK/ACTION)
> ATTACK
> WHAT ATTACK DO YOU WANT? (SWORD/SPELL)
> SPELL
You have no moves left, the Dragon attacks you!!

```

Permainan akan dihentikan ketika *health bar* pemain atau lawan mencapai nol. Saat pemain mencapai nol maka pemain akan kalah dengan skor nol, sedangkan apabila lawan mencapai nol maka pemain akan menang dengan skor sesuai *health bar* pemain.

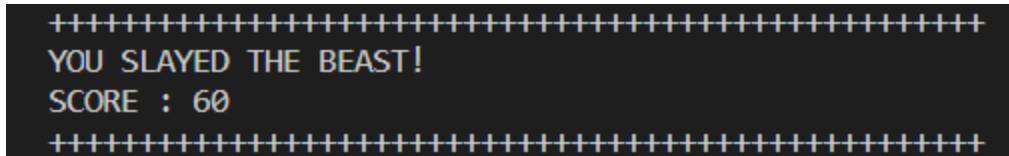
Gambar 6.8.11: Pemain kalah

```

+++++
YOU LOSE...
SCORE : 0
+++++

```

Gambar 6.8.12: Pemain menang



7 Test Script

No	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	HISTORY	Memeriksa apakah banyak History permainan yang ditampilkan sesuai dengan input pengguna	<ol style="list-style-type: none"> Memainkan game RNG Memasukkan command "HISTORY 3" 	HISTORY 3	Ditampilkan history 3 permainan terakhir yang dimainkan	Gambar 6.1.2
2	HISTORY	Memeriksa apakah ketika input dari pengguna lebih banyak dari history yang ada menampilkan seluruh history	<ol style="list-style-type: none"> Memasukkan command "HISTORY 9" 	HISTORY 9	Ditampilkan seluruh history permainan	Gambar 6.1.3
3	RESET HISTORY	Memeriksa apakah program menghapus history yang sudah ada	<ol style="list-style-type: none"> Memasukkan command "RESET HISTORY" 	RESET HISTORY HISTORY	Tidak ada history permainan Ketika memasukkan command "HISTORY"	Gambar 6.2.1 Gambar 6.2.2
4	SCOREBOARD	Memeriksa apakah nama pemain yang telah dimasukkan ke program ada di scoreboard	<ol style="list-style-type: none"> Memainkan game GAME ASAL Setelah permainan berakhir, memasukkan nama "MARSHA" Memasukkan command "SCOREBOARD" 	MARSHA SCOREBOARD	Nama Marsha dan scorenya ditampilkan pada scoreboard game asal	Gambar 6.3.2
5	SCOREBOARD	Memeriksa apakah nama pemain yang telah ada di	<ol style="list-style-type: none"> Memainkan game GAME ASAL 	Sasuke SCOREBOARD	Nama Sasuke dan skornya ditampilkan pada	Gambar 6.3.3

		scoreboard, skornya akan terupdate dengan yang baru	<ol style="list-style-type: none"> Setelah permainan berakhir, memasukkan nama "Sasuke" Memasukkan command "SCOREBOARD" 		scoreboard game asal dengan skor yang baru	
6	RESET SCOREBOARD	Memeriksa apakah berhasil menghapus scoreboard sesuai keinginan pemain	<ol style="list-style-type: none"> Menginput command "RESET SCOREBOARD" Memilih permainan yang ingin scoreboardnya dihapus 	RESET SCOREBOARD	Menampilkan scoreboard kosong pada permainan yang dipilih oleh pemain	Gambar 6.4.3 Gambar 6.4.4
7	Tower Of Hanoi	Memeriksa apakah file berhasil melakukan aksi yang sesuai dan apakah program berhasil menangani input yang salah	<ol style="list-style-type: none"> Melakukan run program Memasukkan test-case input yang tidak valid Memainkan Game hingga selesai 	<p>Input jumlah yang bukan merupakan angka</p> <p>Input selain A, B, C</p> <p>Input Tower Tujuan dan Tower Asal yang sama</p> <p>Input Tower Asal yang memiliki ukuran TOP yang lebih besar daripada TOP Tower Tujuan</p> <p>Input yang valid</p>	<ol style="list-style-type: none"> Mengeluarkan pesan error yang sesuai untuk input yang tidak valid Berhasil memainkan game sampai akhir dan mengembalikan score yang sesuai 	Gambar 6.5.1 Gambar 6.5.2 Gambar 6.5.3 Gambar 6.5.4 Gambar 6.5.5 Gambar 6.5.6 Gambar 6.5.7 Gambar 6.5.8 Gambar 6.5.9
7	HANGMAN	Memeriksa apakah program telah melakukan aksi yang sesuai dengan input	<ol style="list-style-type: none"> Melakukan run program Memainkan game hingga selesai 	<p>Input Kata yang akan ditambahkan ke library</p> <p>Input y/n berdasarkan pertanyaan</p>	Berhasil memainkan game hingga game over atau nyawa sudah habis	Gambar 6.6.1 Gambar 6.6.2 Gambar 6.6.3 Gambar 6.6.4 Gambar 6.6.5 Gambar 6.6.6 Gambar 6.6.7 Gambar 6.6.8

				Input Jawaban tiap soal		Gambar 6.6.9
8	Snake On Meteor	Memeriksa apakah program menampilkan output yang benar dan berhasil menangani input yang salah	<ol style="list-style-type: none"> 1. Melakukan run program 2. Memainkan permainan 3. Memasukkan input yang tidak valid 4. Melanjutkan permainan hingga selesai 	<p>Input gerakan yang valid (w/a/s/d)</p> <p>Input gerakan yang tidak diperbolehkan (ke badan sendiri atau ke meteor)</p> <p>Input selain yang valid</p>	<ol style="list-style-type: none"> 1. Berhasil lanjut ke proses selanjutnya 2. Meminta ulang input hingga valid 	<p>Gambar 6.7.1</p> <p>Gambar 6.7.2</p> <p>Gambar 6.7.3</p> <p>Gambar 6.7.4</p> <p>Gambar 6.7.5</p> <p>Gambar 6.7.6</p> <p>Gambar 6.7.7</p> <p>Gambar 6.7.8</p> <p>Gambar 6.7.9</p> <p>Gambar 6.7.10</p> <p>Gambar 6.7.11</p> <p>Gambar 6.7.12</p> <p>Gambar 6.7.13</p> <p>Gambar 6.7.14</p> <p>Gambar 6.7.15</p>
9	Slay The Dragon	Memeriksa apakah program menampilkan output yang benar sesuai dengan input yang diberikan	<ol style="list-style-type: none"> 1. Run Program 2. Memainkan Permainan 3. Memasukkan input yang tidak valid 4. Menyelesaikan permainan 	<p>Input valid:</p> <ol style="list-style-type: none"> 1. attack/action 1.1 apabila memilih attack: spell/sword 1.2 apabila memilih action: skip/heal <p>Input tidak valid:</p> <ol style="list-style-type: none"> 1. Input (spell/sword/heal/skip) sebelum attack/action 2. Input move yang sudah habis 3. Input move diluar enam yang sudah disebutkan 	<ol style="list-style-type: none"> 1. Berhasil memainkan game hingga win atau lose. 2. Menghasilkan output yang sesuai input. 	<p>Gambar 6.8.1</p> <p>Gambar 6.8.2</p> <p>Gambar 6.8.3</p> <p>Gambar 6.8.4</p> <p>Gambar 6.8.5</p> <p>Gambar 6.8.6</p> <p>Gambar 6.8.7</p> <p>Gambar 6.8.8</p> <p>Gambar 6.8.9</p> <p>Gambar 6.8.10</p> <p>Gambar 6.8.11</p> <p>Gambar 6.8.12</p>

8 Pembagian Kerja dalam Kelompok

NIM	Nama	Deskripsi Tugas
18221066	Aulia Nadhirah Yasmin. B	<ul style="list-style-type: none"> • Membuat game Tower of Hanoi • Memodifikasi ADT Stack • Membuat Driver ADT Tree, Stack, Point, dan Linked List • Melengkapi Laporan
18221070	Rifqi Syafiq	<ul style="list-style-type: none"> • Membuat game Hangman • Mengontak asisten dan melakukan penjadwalan asistensi 1 • Melengkapi CONSOLE • Menulis laporan
18221072	Hilmi Baskara Radanto	<ul style="list-style-type: none"> • Membuat fungsi SCOREBOARD, RESET SCOREBOARD, HISTORY, RESET HISTORY • Melakukan implementasi yang dibutuhkan untuk scoreboard dan history pada beberapa file pada program. • Mengontak asisten dan melakukan penjadwalan asistensi 2 • Membuat driver StackHistory, Map, dan ArrayOfMap
18221082	Akmal Natha Abhirama	<ul style="list-style-type: none"> • Membuat game Snake On Meteor • Membuat game bonus ADT Tree
18221108	Farhan Algani Putra	<ul style="list-style-type: none"> • Membuat game Snake On Meteor • Membuat dan memodifikasi ADT Linked List • Memodifikasi ADT Point • Melengkapi laporan

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

Asistensi III

Tanggal : 25 November 2022	Catatan Asistensi:
Tempat : Zoom	
<p>Kehadiran Anggota Kelompok:</p> <p>No</p> <p>NIM</p> <p>Tanda tangan</p> <p>1</p> <p>18221072</p>  <p>2</p> <p>18221066</p>  <p>3</p> <p>18221070</p>  <p>4</p> <p>18221082</p>  <p>5</p> <p>18221108</p> 	
	<p>Tanda Tangan Asisten: </p>

Asistensi IV

Tanggal : 1 Desember 2022	Catatan Asistensi: 1. Laporan bagian ringkasan dan program utama cukup dari tambahan spesifikasi yang baru ada di tugas besar 2 aja. 2. Bonus pada hangman jika dikerjakan salah satunya aja akan dapat nilai tapi parsial 3. Diner Dash memakai saldo sebagai score
Tempat : Zoom	
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 18221072  2 18221066  3 18221070  4 18221082  5 18221108 	
	Tanda Tangan Asisten: 

9.3 Log Activity Anggota Kelompok

Tanggal	Lokasi	NIM	Aktivitas
18/11/22	Grup LINE	18221108 18221070 18221072 18221066 18221082	Melakukan pembagian tugas
20/11/22	Kost	18221072	Commit History and Reset History to GitHub
22/11/22	Rumah	18221066	Menyelesaikan base game Tower Of Hanoi
25/11/22	Zoom	18221108 18221070 18221072 18221066 18221082	Asistensi I
26/11/22	Kost	18221070	Memulai Pengerjaan game hangman serta edit console
27/11/22	Kost	18221108	Upload Snake On Meteor, ADT Linked List, ADT Point
29/11/12	Kost	18221072	Commit Scoreboard and Reset Scoreboard to GitHub
29/11/12	Rumah	18221066	Menyelesaikan Bagian Bonus Tower Of Hanoi
30/11/22	Zoom	18221072 18221066 18221108	Zoom menyatukan beberapa file
30/11/22	Kost	18221108	Menambahkan score pada game Rock Paper Scissor
1/12/22	Rumah	18221066	Membuat prototype game bonus
1/12/22	Selasar Gaib	18221108 18221070 18221072 18221066 18221082	Menyatukan beberapa file ke main

1/12/22	Zoom	18221108 18221070 18221072 18221066 18221082	Asistensi II
1/12/22	Kost	18221082	Finalisasi game Slay The Dragon (game bonus)
2/12/22	Kost	18221108	Upload updated and finished Snake On Meteor, ADT Linked List, ADT Point
2/12/22	Kost	18221072	Implementasi history dan scoreboard pada playgame, skipgame, deletgame, dan creategame
2/12/22	Rumah	18221066	Menyelesaikan Driver Tree, Point dan ListDP.
2/12/22	Kost	18221070	Finalisasi game Hangman