

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data


BNMO

Dipersiapkan oleh:

Kelompok 09 / K01

Silvester Kresna W. P. P.	18221049
Jessica	18221073
Nalendro N. U. A. R. F.	18221085
Muhammad Aliefnaufal Permana	18221105
Rayhan Nugraha Putra	18221149

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB2-09</i>		<i><jml hlm></i>
		<i>Revisi</i>	<i>0</i>	<i>2 Desember 2022</i>

Daftar Isi

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	5
2.1	LOAD	5
2.2	Scan	5
2.3	INSERTSCOREBOARD	5
2.4	Add Word	6
2.5	Hangman	6
2.6	Tower Of Hanoi	6
2.7	Snake On Meteor	6
3	Struktur Data (ADT)	6
3.1	ADT Mesin Karakter	6
3.2	ADT Mesin Kata	7
3.3	ADT Array	7
3.4	ADT Stack of Word	7
3.5	ADT Stack of Integer	7
4	Program Utama	9
5	Data Test	10
5.1	Scoreboard	10
5.2	Reset Scoreboard	13
5.3	History	16
5.4	Reset History	20
5.5	Tower of Hanoi	20
5.6	Snake on Meteor	24
5.7	Hangman	27
6	Test Script	32
7	Pembagian Kerja dalam Kelompok	36
8	Lampiran	37
8.1	Deskripsi Tugas Besar 2	37
8.2	Notulen Rapat	37
8.3	Log Activity Anggota Kelompok	38

1 Ringkasan

BNMO adalah sebuah robot video game konsol milik Indra dan Doni. BNMO berbasis command-line interface (CLI) sehingga user dapat memainkan BNMO dengan memasukkan command yang tersedia pada BNMO. BNMO tersebut pernah mengalami kerusakan sehingga memerlukan perbaikan. Namun, setelah diperbaiki, BNMO memiliki lebih banyak bug dalam programnya sehingga masih memerlukan perbaikan lagi dan pemrograman ulang untuk dapat digunakan kembali. Perbaikan kedua berhasil membuat BNMO sudah dapat digunakan kembali sehingga Indra dan Doni merasa senang. Meskipun demikian, Indra dan Doni masih ingin melakukan pengembangan terhadap BNMO agar BNMO tidak hanya dapat menjalankan fitur-fitur dasar saja. Oleh karena itu, dilakukan penambahan fitur-fitur dan permainan ke dalam program BNMO. Beberapa fitur yang ditambahkan adalah fitur menampilkan scoreboard setiap game yang terdapat dalam BNMO, menghapus scoreboard suatu game, menampilkan daftar history game, dan menghapus seluruh history game. Sementara itu, game yang ditambahkan adalah Hangman, Tower Of Hanoi, dan Snake On Meteor.

Saat ini, BNMO memiliki beberapa fitur. Interface awal BNMO akan menampilkan menu utama, yaitu start dan load. User harus memasukkan command start atau load terlebih dahulu untuk dapat memanggil command-command lainnya. Start akan membaca file konfigurasi default BNMO yang berbasis txt yang berisi daftar game. Sementara itu, load akan membaca file berbasis txt sesuai dengan masukan user. BNMO menyediakan command help sebagai layanan bantuan untuk menampilkan daftar command yang dapat dipanggil dalam BNMO.

BNMO telah menyediakan lima game, yaitu RNG, Diner Dash, Hangman, Tower Of Hanoi, dan Snake On Meteor. RNG merupakan permainan menebak angka integer random, Diner Dash merupakan permainan bertemakan restoran, Hangman merupakan permainan menebak kata, Tower Of Hanoi adalah permainan strategi memindahkan piringan pada suatu tiang ke tiang lain, sementara Snake On Meteor adalah game andalan BNMO yang mirip dengan game snake pada konsol lama, namun memiliki meteor dan obstacle.

Selain dapat memainkan game yang telah tersedia pada BNMO, user juga dapat menambah game yang dibuat sendiri dengan memanggil command create game. Game buatan user yang telah ditambahkan ke dalam BNMO dapat dihapus dengan memanggil command delete game. Apabila suatu game dihapus, maka scoreboard dan history untuk game tersebut juga akan dihapus. Daftar game yang ada pada BNMO, baik game yang tersedia secara default maupun game buatan user yang telah ditambahkan ke dalam BNMO, dapat dilihat dengan memanggil command list game.

Untuk dapat memainkan game yang terdapat dalam BNMO, user perlu menjalankan command queue game terlebih dahulu. Command queue game akan meminta user untuk memasukkan game yang telah tersedia di BNMO. Game tersebut kemudian akan disimpan ke dalam daftar antrian game. Selanjutnya, command play game digunakan untuk memainkan game pada urutan pertama dalam daftar antrian game. Setelah berhasil menyelesaikan suatu game, game tersebut akan ditambahkan ke dalam history game dan skor akhir akan ditambahkan ke dalam daftar scoreboard. Apabila user tidak ingin memainkan game pada urutan teratas dalam daftar antrian game, maka user dapat memanggil command skip game dan memasukkan jumlah game yang ingin dilewati pada daftar antrian game.

BNMO menyediakan fitur scoreboard untuk menampilkan daftar scoreboard setiap game yang terdapat pada daftar game. Scoreboard setiap game ditampilkan secara terurut mengecil berdasarkan skornya. Untuk menghapus daftar scoreboard, tersedia command reset scoreboard. User dapat memilih untuk menghapus scoreboard seluruh game atau hanya game tertentu saja. Command history digunakan untuk menampilkan daftar game yang telah dimainkan. Untuk menghapus seluruh daftar history game, user dapat memanggil command reset history.

Dengan memanggil command save, user dapat menyimpan daftar game yang telah ditambahkan, daftar history game, dan daftar scoreboard ke dalam file berbasis txt sesuai masukan user. Setelah selesai, user dapat menjalankan command quit untuk keluar dari program BNMO. Command-command selain yang telah didefinisikan dalam BNMO akan dianggap sebagai command yang tidak valid.

Laporan mencakup ringkasan laporan, penjelasan tambahan mengenai spesifikasi tugas besar, struktur ADT yang digunakan, penjelasan mengenai algoritma program utama, algoritma-algoritma menarik, data test dan test script yang dilakukan terhadap program yang telah dibuat, pembagian tugas dalam kelompok, serta lampiran yang terdiri dari deskripsi tugas besar 2, notulen rapat, log activity anggota kelompok, dan form asistensi tugas besar.

Program dalam tugas besar ini dibuat menggunakan bahasa C dengan memanfaatkan ADT yang telah dipelajari pada perkuliahan IF2111 Algoritma dan Struktur Data STI yang membantu mempermudah pembuatan program. Secara umum, tugas besar ini melatih kami untuk menggunakan dan menyesuaikan struktur ADT yang cocok dan sesuai dengan permasalahan yang ada sehingga ketika dihadapkan pada suatu persoalan dapat menentukan penggunaan struktur ADT yang tepat. Hasil dari tugas besar yang telah kami buat sudah memenuhi spesifikasi yang diminta dan berhasil dijalankan dengan baik.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 LOAD

Program akan meminta ulang input apabila filename yang dimasukkan tidak ditemukan atau filename yang dimasukkan adalah “listofwords.txt” yang merupakan daftar kata untuk game Hangman.

2.2 Scan

Prosedur Scan dengan parameter output berupa Word command1 dan Word command2, digunakan untuk membaca command program dan melakukan upper word terhadap semua command dari user, kecuali nama file.txt. Scan kemudian akan memisahkan kata pertama dengan kata berikutnya pada masukan command dan mengembalikan kata pertama dan kata berikutnya.

2.3 INSERTSCOREBOARD

Prosedur INSERTSCOREBOARD digunakan untuk memasukkan skor user setelah berhasil menyelesaikan suatu game ke dalam daftar scoreboard game tersebut. Prosedur ini akan meminta input nama sampai valid, yaitu nama tidak kosong, nama hanya terdiri dari satu kata saja, dan nama belum terpakai di daftar scoreboard game tersebut.

2.4 Add Word

Command add word dapat dipanggil ketika menjalankan game Hangman, yang digunakan untuk menambahkan kata ke dalam “listofwords.txt” yang menyimpan daftar kata untuk game Hangman. User akan diminta untuk memasukkan kata yang ingin ditambahkan sampai masukan kata tersebut valid, yaitu belum terdapat dalam “listofwords.txt”, bukan hanya merupakan sebuah huruf, hanya merupakan sebuah kata, dan tidak mengandung angka. Program akan meminta masukan kata lagi sampai user tidak ingin menambah kata lagi. Setelah menambah kata, user dapat memilih untuk bermain game Hangman atau tidak. Apabila tidak bermain, maka user tidak akan diminta memasukkan nama untuk scoreboard.

2.5 Hangman

Dalam game Hangman, apabila pemain berhasil menebak kata pertama dan kesempatan pemain masih tersisa, maka program akan melakukan random untuk kata berikutnya yang harus ditebak. Kata berikutnya ini pasti berbeda dengan kata-kata sebelumnya yang telah berhasil ditebak oleh pemain, kecuali sudah tidak terdapat kata lain lagi dalam daftar kata.

2.6 Tower Of Hanoi

Pemain dapat memasukkan jumlah piringan yang diinginkan. Skor akhir yang diperoleh pada game Tower Of Hanoi disesuaikan dengan jumlah piringan pada game, jumlah piringan yang lebih sedikit akan memperoleh skor maksimal lebih rendah daripada jumlah piringan yang lebih banyak. Untuk jumlah piringan 5, maka skor maksimal adalah 10. Skor maksimal akan bertambah 1 untuk setiap penambahan piring dan berkurang 1 untuk setiap pengurangan piring. Skor maksimal tersebut akan diperoleh apabila pemain menggunakan paling optimal. Skor akan berkurang sebanyak selisih jumlah langkah pemain dengan langkah paling optimal. Skor minimum adalah 1.

2.7 Snake On Meteor

Obstacle muncul di awal permainan dengan posisi obstacle random dan jumlah obstacle random antara 1 atau 2 obstacle. Makanan dan meteor tidak dapat muncul pada titik yang memiliki obstacle. Ketika kepala snake mengenai obstacle, maka permainan berakhir. Sisi peta yang berseberangan pada game snake ini terhubung.

3 Struktur Data (ADT)

Berikut ADT yang digunakan beserta penjelasan mengenai struktur data, persoalan yang diselesaikan, alasan pemilihan dan implementasinya.

3.1 ADT Mesin Karakter

ADT mesin karakter menggunakan CC bertipe char dan EOP bertipe boolean yang menunjukkan end of process. ADT ini digunakan untuk membaca pita karakter yang digunakan dalam pembacaan file dan parsing command program. Prosedur START dan ADV digunakan untuk pembacaan file, sementara STARTC dan ADVC untuk pembacaan command. ADT ini

memudahkan dalam membaca karakter satu per satu, yang diimplementasikan dalam file `mesinkarakter.c`.

3.2 ADT Mesin Kata

ADT mesin kata menggunakan tipe data bentukan `Word` terdiri atas `TabWord` berkapasitas 100 elemen untuk menyimpan kata dan `Length` untuk menyimpan panjang kata. Untuk membaca file eksternal, terdapat prosedur `IgnoreBlank`, `CopyWord`, `STARTWORD`, dan `ADVWORD`, sementara untuk membaca command, terdapat prosedur `IgnoreBlanks`, `STARTCOMMAND`, `ADVCOMMAND`, `CopyCommand`, `STARTCOMMAND2`, `ADVCOMMAND2`, `CopyCommand2`. Selain itu, terdapat beberapa fungsi dan prosedur tambahan, seperti `PrintWord`, `WordToInt`, `StringToWord`, `IsWordSame`, `Scan`, `SplitWord`, dan sebagainya. ADT ini digunakan untuk menyelesaikan persoalan pembacaan kata pada file eksternal dan command, diimplementasikan dalam file `mesinkata.c`.

3.3 ADT Array

ADT array merupakan tipe data bentukan `Tab` terdiri atas `TI` bertipe `Word` sebagai memori tempat penyimpanan elemen yang berkapasitas 100 elemen dan `Neff` bertipe integer sebagai jumlah elemen efektif. ADT ini digunakan untuk menyimpan daftar game pada saat menjalankan command load (`load.c`), sehingga digunakan juga dalam `creategame.c`, `detelegame.c`, `listgame.c`, `playgame.c`, `scoreboard.c`, dan `queuegame.c`, serta untuk menyimpan daftar kata untuk game hangman (`hangman.c`). ADT ini memiliki primitif `MakeEmptyArray`, `NbElmtArray`, `MaxNbElArray`, `GetFirstIdxArray`, `GetLastIdxArray`, `GetElmtArray`, `SetTabArray`, `SetElArray`, `SetNeffArray`, `IsIdxValidArray`, `IsIdxEffArray`, `IsEmptyArray`, dan `IsFullArray`. ADT ini diimplementasikan dalam file `array.c` dan berfungsi untuk mempermudah menyimpan daftar pada saat melakukan load yang akan digunakan untuk fitur-fitur lainnya.

3.4 ADT Stack of Word

ADT stack of word menggunakan tipe data bentukan `Stack` terdiri atas `T` bertipe `Word` yang berkapasitas 100 elemen untuk penyimpanan elemen dan `TOP` bertipe integer untuk indeks elemen puncak. Akses elemen pada ADT ini hanya dapat dilakukan pada elemen `TOP`. ADT ini digunakan untuk menyimpan daftar history game pada saat melakukan load (`load.c`), yang mempermudah fitur history dan reset history (`history.c`), menambahkan game yang telah dimainkan ke dalam history pada `playgame.c`, dan menghapus history untuk game yang dihapus pada `deletgame.c`. Terdapat fungsi dan prosedur `CreateEmptyStack`, `IsStackEmpty`, `IsStackFull`, `Push`, `Pop`, dan `InverseStack` pada ADT ini. ADT ini diimplementasikan dalam file `stack.c`.

3.5 ADT Stack of Integer

ADT stack of integer menggunakan tipe data bentukan `StackToH` terdiri atas `bufferS` bertipe integer yang berkapasitas 100 elemen dan `TOPS` bertipe integer untuk indeks elemen puncak. Akses elemen pada ADT ini hanya dapat dilakukan pada element `TOP`. ADT ini digunakan untuk menyimpan piringan pada game Tower Of Hanoi sehingga digunakan pada file `towerofhanoi.c`. Terdapat fungsi dan prosedur `CreateSEmpty`, `IsSEmpty`, `PushS`, `PopS`, dan `LengthS` pada ADT ini. ADT ini diimplementasikan dalam file `stackToH.c`.

3.6 ADT Map

ADT map menggunakan tipe data bentukan Map yang terdiri atas Elements bertipe infotype untuk tempat penyimpanan elemen Map yang berkapasitas 50 elemen dan Count bertipe integer untuk jumlah elemen Map. Infotype sendiri merupakan tipe data bentukan yang terdiri atas Key yang selalu unik bertipe Word dan Value bertipe integer. ADT ini digunakan untuk menyimpan scoreboard game dengan key berisi nama pemain dan value berisi skor akhir pemain, sehingga dipakai untuk fitur scoreboard pada file scoreboard.c. Terdapat fungsi dan prosedur CreateEmptyMap, IsMapEmpty, IsMapFull, Value, Insert, Delete, IsMember, InsertSort, dan MaxKeyLength. ADT ini diimplementasikan dalam file map.c.

3.7 ADT Array Of Map

ADT array of map menggunakan tipe data bentukan TabMap yang terdiri dari Elmt bertipe Map untuk tempat penyimpanan elemen yang berkapasitas 100 elemen dan Nelmt bertipe integer untuk jumlah elemen efektif TabMap. ADT ini digunakan untuk menyimpan daftar scoreboard seluruh game sehingga Nelmt sama dengan jumlah game yang terdapat pada BNMO. ADT ini digunakan untuk menyimpan daftar scoreboard yang terdapat pada file eksternal yang dibaca (load.c), digunakan untuk menambahkan scoreboard dan melakukan reset scoreboard pada file scoreboard.c, yang berguna untuk fitur-fitur lainnya. Terdapat fungsi dan prosedur MakeEmptyArrMap, NbElmtArrMap, MaxNbElArrMap, GetFirstIdxArrMap, GetLastIdxArrMap, GetElmtArrMap, SetTabArrMap, SetElArrMap, SetNeffArrMap, IsIndxValid, IsIndxEff, IsEmptyArrMap, IsFullArrMap, InsertArrMap, dan DeleteArrMap pada ADT ini. ADT ini diimplementasikan dalam file arrayofmap.c.

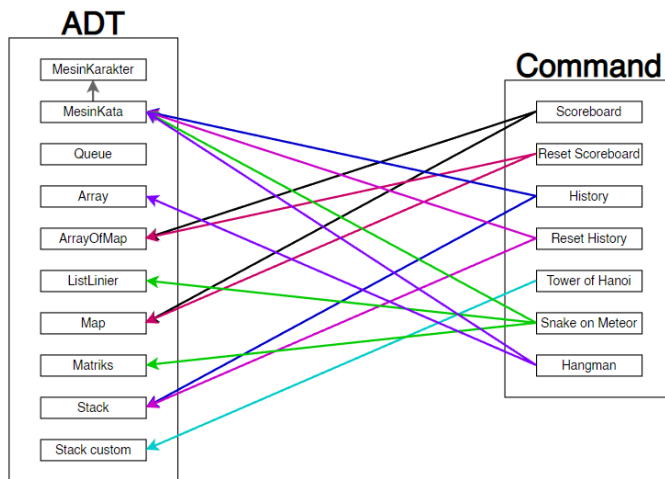
3.8 Matriks

ADT matriks menggunakan tipe data bentukan matriks yang terdiri dari TM bertipe integer berkapasitas 5x5 untuk tempat penyimpanan elemen dan Ceff bertipe integer untuk jumlah elemen efektif matriks. ADT ini digunakan untuk game Snake On Meteor untuk menampilkan peta dari game snake yang memerlukan array dua dimensi, sehingga dipakai pada file snakeonmeteor.c. Terdapat prosedur MakeEmptyMatriks dan Display pada ADT ini. ADT ini diimplementasikan dalam file matriks.c.

3.9 ADT Linked List

ADT linked list menggunakan tipe data bentukan List yang terdiri dari First dan Last yang bertipe addresslist. Address adalah pointer merujuk pada ElmtList yang terdiri dari x dan y bertipe integer untuk menyimpan posisi pada peta, serta next dan prev yang bertipe addresslist. Terdapat fungsi dan prosedur IsListEmpty, CreateEmptyList, Alokasi, Dealokasi, SearchList, InsVLast, DelVLast, InsertLast, DelLast, DelP, LengthList, serta tambahan yaitu random1, random2, random3, random4, dan random5. ADT ini digunakan pada game Snake On Meteor karena mempermudah dalam menambah dan mengurangi panjang dari snake dengan melakukan alokasi dan dealokasi, sehingga ADT ini digunakan pada file snakeonmeteor.c. ADT ini diimplementasikan dalam file listdp.c.

Relasi ADT-Command



4 Program Utama

Program utama

Pada program utama, pertama-tama, program akan menampilkan *welcome page* yang berisikan tulisan welcome, serta menampilkan gambar BNMO yang digenerate melalui Setelah menampilkan welcome page, program akan membawa pengguna ke *starting menu*. Starting Menu berisikan informasi command awal, yakni START serta LOAD. Starting Menu bertujuan untuk mengarahkan pengguna agar mengakses file yang akan digunakan pada program ini. Jika pengguna memasukkan command START, maka program akan membaca file default yang dibuat untuk menginisialisasi program template. Jika pengguna memasukkan command LOAD, maka pengguna harus menyertakan file tujuan yang akan dibaca dengan format. Kemudian program akan membaca dan mengakses file tersebut.

Setelah starting menu, program siap menerima command-command lain. Command yang dapat digunakan oleh pengguna, yakni: CREATE GAME, untuk menambahkan game custom kepada daftar game yang tersedia; LIST GAME, untuk menampilkan daftar game yang tersedia; DELETE GAME, untuk menghapus game dari daftar game; QUEUE GAME, untuk mendaftarkan permainan ke dalam daftar antrian game; PLAY GAME, untuk memainkan game pertama pada antrian game; SKIPGAME, untuk melewati permainan sebanyak n; QUIT, untuk keluar dari program; dan SAVE, untuk menyimpan state game pemain saat ini ke suatu file. SCOREBOARD, untuk melihat nama dan skor untuk semua game. RESET SCOREBOARD, untuk melakukan reset terhadap scoreboard permainan. HISTORY, 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. RESET HISTORY, untuk menghapus semua history permainan yang dimainkan.

5 Data Test

Untuk memastikan program dapat berjalan dengan lancar dan sesuai spesifikasi, dibutuhkan suatu testing. Berikut data test yang digunakan pada testing terhadap program kami beserta rincian fitur yang dites, output apa saja yang seharusnya diberikan, serta data testnya sendiri.

5.1 Scoreboard

Fungsi ini dibagi menjadi 2, “INSERTSCOREBOARD” dan “DISPLAYSCOREBOARD”.

Fungsi pertama akan selalu dijalankan setiap kali selesai bermain game apa saja yang mendapatkan skor akhir lebih dari atau sama dengan nol. Fungsi kemudian meminta masukan nama pemain yang belum pernah ada di scoreboard permainan yang dimainkan sebanyak satu kata dengan jumlah karakter tak terbatas. Masukan ini akan dimasukkan ke dalam scoreboard secara terurut menurun berdasarkan skor yang didapat para pengguna.

```
Loading GRAND THEFT AUTO VI ...

=====
||  G A M E  O V E R  ||
=====

Skor akhir = 13
Nama >> DATATEST

Skor berhasil dimasukkan ke dalam scoreboard!
```

Apabila nama yang dimasukkan sudah ada di dalam scoreboard, maka akan diminta masukan nama baru hingga kondisi sudah tercapai. Pada kasus di bawah ini, di dalam scoreboard sudah ada seseorang dengan nama “DATATEST”.

```
Loading GRAND THEFT AUTO VI ...

=====
||  G A M E  O V E R  ||
=====

Skor akhir = 93
Nama >> DATATEST

Masukan nama tidak valid. Nama sudah pernah terpakai. Mohon masukkan nama lain.
Nama >> _
```

Apabila pengguna memasukkan nama yang terdiri atas dua kata atau lebih, maka nama tersebut akan dianggap tidak valid dan diminta kembali nama yang sesuai format.

```
Nama >> DATATEST 2KATA

Masukan nama tidak valid. Nama hanya boleh terdiri dari satu kata. Mohon masukkan nama lain.
Nama >> _
```

Fungsi kedua akan menampilkan scoreboard dari seluruh permainan apabila pengguna memasukkan *command* “SCOREBOARD”. Ukuran scoreboard tiap permainan akan mengikuti nama terpanjang yang terdapat pada scoreboard masing-masing permainan.

```

ENTER COMMAND >> SCOREBOARD

***** SCOREBOARD GAME RING *****
| 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 GRAND THEFT AUTO VI *****
| NAMA | SKOR |
-----
| DATATESTTRALALALALALALALALALALALALAL | 93 |
| DATATEST | 13 |

```

Ketika pengguna membuat permainan baru menggunakan fungsi “CREATE GAME” maka scoreboard untuk permainan tersebut akan dibuat secara otomatis. Pada kasus ini, dimasukkan sebuah permainan baru bernama “CARD WARS”.

```

ENTER COMMAND >> SCOREBOARD

***** SCOREBOARD GAME RING *****
| 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 GRAND THEFT AUTO VI *****
| NAMA | SKOR |
----- SCOREBOARD KOSONG -----
| DATATESTTRALALALALALALALALALALALAL | 93 |
| DATATEST | 13 |

***** SCOREBOARD GAME CARD WARS *****
| NAMA | SKOR |
----- SCOREBOARD KOSONG -----

```

Sedangkan apabila pengguna menghapus permainan menggunakan fungsi “DELETE GAME”, maka scoreboard untuk permainan tersebut juga akan otomatis terhapus. Pada kasus ini, permainan “GRAND THEFT AUTO VI” dihapus.

```
ENTER COMMAND >> 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 -----

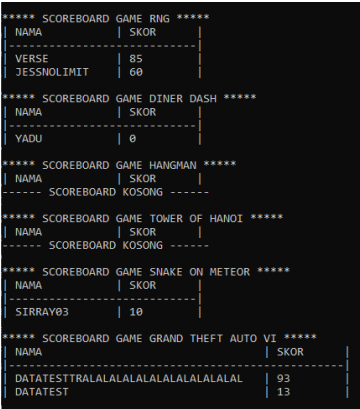
***** SCOREBOARD GAME SNAKE ON METEOR *****
| NAMA | SKOR |
----- SCOREBOARD KOSONG -----

***** SCOREBOARD GAME CARD WARS *****
| NAMA | SKOR |
----- SCOREBOARD KOSONG -----
```

Semua masukan dalam scoreboard akan tersimpan apabila pengguna memanggil fungsi “SAVE <n>.txt” dan akan muncul kembali apabila pengguna melakukan “LOAD” menggunakan file .txt tersebut.

5.2 Reset Scoreboard

RESET SCOREBOARD berfungsi untuk menghapus masukan scoreboard yang sudah ada sebelumnya. Terdapat dua kasus pada fungsi ini. Pertama, pengguna hanya menghapus scoreboard dari salah satu permainan saja.

Kondisi	Gambar
Sebelum scoreboard game “RNG” dihapus	 <pre> ***** SCOREBOARD GAME RNG ***** NAMA SKOR ----- ----- VERSE 85 JESSNOLIMIT 60 ----- ----- ***** SCOREBOARD GAME DNER DASH ***** NAMA SKOR ----- ----- YADU 0 ----- ----- ***** SCOREBOARD GAME HANGMAN ***** NAMA SKOR ----- ----- ----- SCOREBOARD KOSONG ----- ***** SCOREBOARD GAME TOWER OF HANOI ***** NAMA SKOR ----- ----- ----- SCOREBOARD KOSONG ----- ***** SCOREBOARD GAME SNAKE ON METEOR ***** NAMA SKOR ----- ----- SIRRAY03 10 ----- ----- ***** SCOREBOARD GAME GRAND THEFT AUTO VI ***** NAMA SKOR ----- ----- DATATESTTRALALALALALALALALALALALAL 93 DATATEST 13 ----- ----- </pre>

Proses penghapusan	<pre> ENTER COMMAND >> reset scoreboard DAFTAR SCOREBOARD : 0. ALL 1. RNG 2. DINER DASH 3. HANGMAN 4. TOWER OF HANOI 5. SNAKE ON METEOR 6. GRAND THEFT AUTO VI SCOREBOARD YANG INGIN DIHAPUS >> 1 APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD RNG (YA/TIDAK) >> ya Scoreboard RNG berhasil di-reset. </pre>
Setelah scoreboard game “RNG” dihapus	<pre> ENTER COMMAND >> scoreboard ***** SCOREBOARD GAME RNG ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ***** SCOREBOARD GAME DINER DASH ***** NAMA SKOR ----- ----- YADU 0 ***** SCOREBOARD GAME HANGMAN ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ***** SCOREBOARD GAME TOWER OF HANOI ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ***** SCOREBOARD GAME SNAKE ON METEOR ***** NAMA SKOR ----- ----- SIRRAY03 10 ***** SCOREBOARD GAME GRAND THEFT AUTO VI ***** NAMA SKOR ----- ----- DATATESTTRALALALALALALALALALALALAL 93 DATATEST 13 </pre>

Fungsi dapat juga menghapus semua scoreboard apabila pengguna memilih opsi “ALL”.

Kondisi	Gambar
Sebelum penghapusan semua scoreboard	<pre> ENTER COMMAND >> scoreboard ***** SCOREBOARD GAME RNG ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ***** SCOREBOARD GAME DINER DASH ***** NAMA SKOR ----- ----- YADU 0 ***** SCOREBOARD GAME HANGMAN ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ***** SCOREBOARD GAME TOWER OF HANOI ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ***** SCOREBOARD GAME SNAKE ON METEOR ***** NAMA SKOR ----- ----- SIRRAY03 10 ***** SCOREBOARD GAME GRAND THEFT AUTO VI ***** NAMA SKOR ----- ----- DATATESTTRALALALALALALALALALALALAL 93 DATATEST 13 </pre>

Proses penghapusan	<pre> ENTER COMMAND >> reset scoreboard DAFTAR SCOREBOARD : 0. ALL 1. RNG 2. DINER DASH 3. HANGMAN 4. TOWER OF HANOI 5. SNAKE ON METEOR 6. GRAND THEFT AUTO VI SCOREBOARD YANG INGIN DIHAPUS >> 0 APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK) >> ya Scoreboard ALL berhasil di-reset. </pre>
Setelah penghapusan semua scoreboard	<pre> SCOREBOARD YANG INGIN DIHAPUS >> 0 APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK) >> ya Scoreboard ALL berhasil di-reset. ENTER COMMAND >> 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 GRAND THEFT AUTO VI ***** NAMA SKOR ----- ----- SCOREBOARD KOSONG ----- ----- </pre>

Apabila pengguna memasukkan kata “tidak” pada saat konfirmasi penghapusan maka scoreboard batal 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. GRAND THEFT AUTO VI

SCOREBOARD YANG INGIN DIHAPUS >> 0

APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK) >> tidak

Reset scoreboard ALL dibatalkan.

```

Sedangkan apabila pengguna memberi masukan sembarang/kosong maka konfirmasi akan diteruskan hingga ada masukan “ya” atau “tidak”.

```

ENTER COMMAND >> reset scoreboard

DAFTAR SCOREBOARD :
0. ALL
1. RNG
2. DINER DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. CARD WARS

SCOREBOARD YANG INGIN DIHAPUS >>

Mohon masukkan nomor scoreboard yang ingin dihapus.

SCOREBOARD YANG INGIN DIHAPUS >> 1

APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD RNG (YA/TIDAK) >>
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD RNG (YA/TIDAK) >>

ENTER COMMAND >> reset scoreboard

DAFTAR SCOREBOARD :
0. ALL
1. RNG
2. DINER DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. GRAND THEFT AUTO VI

SCOREBOARD YANG INGIN DIHAPUS >> 0

APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK) >> terserah
APAKAH KAMU YAKIN INGIN MELAKUKAN RESET SCOREBOARD ALL (YA/TIDAK) >> tidak

Reset scoreboard ALL dibatalkan.

```

Hal ini juga ditangani pada kasus dimana pengguna tidak memberikan masukan atau masukan bukan sebuah bilangan atau bilangan yang dimasukkan berada di luar rentang daftar scoreboard yang dapat 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. CARD WARS

SCOREBOARD YANG INGIN DIHAPUS >>

Mohon masukkan nomor scoreboard yang ingin dihapus.

SCOREBOARD YANG INGIN DIHAPUS >> sembarang

Masukan nomor scoreboard tidak valid. Mohon masukkan nomor scoreboard pada daftar scoreboard.

SCOREBOARD YANG INGIN DIHAPUS >> 69

Masukan nomor scoreboard tidak valid. Mohon masukkan nomor scoreboard pada daftar scoreboard.

```

5.3 History

History merupakan prosedur yang terbagi menjadi 3, yaitu inserthistory, displayhistory, dan deletehistory (deletehistory berbeda dengan reset history). Masing-masing fungsi bekerja pada bagian yang berbeda.

Insert history dijalankan ketika user melakukan “PLAY GAME”. Insert history memasukkan nama game yang di play dalam bentuk tipe Word ke dalam Stack.

Display history dijalankan ketika user memberikan command “HISTORY <n>”, dengan menampilkan history game terakhir yang dimainkan sebanyak n, jika n melebihi maksimal history (isi Stack), maka yang ditampilkan hanya sebanyak isi Stack saja. Jika inputan n adalah 0, maka akan ditampilkan “Masukan tidak valid. Masukan angka harus lebih besar dari 0”. Jika history masih kosong maka akan ditampilkan “History game kosong. Tidak ada game yang telah dimainkan”. Jika inputan benar maka akan menampilkan list history dari game yang paling baru dimainkan ke yang lebih lama.

Delete history dijalankan ketika user melakukan command “DELETE GAME”, jika suatu game dihapus, maka seluruh namanya (jika lebih dari 1) akan dihapus dari history.

Berikut adalah tampilan ketika pemain melakukan command “HISTORY <n>”.

```
ENTER COMMAND >> history 2  
  
Berikut adalah daftar game yang telah dimainkan  
1. GAME 01  
2. GAME 02
```

Berikut adalah tampilan ketika pemain memberi masukkan n lebih dari game yang ada di history (pada contoh ini isi stack history ada 5).

```
ENTER COMMAND >> history 9000  
  
Berikut adalah daftar game yang telah dimainkan  
1. GAME 01  
2. GAME 02  
3. GAME 03  
4. GAME 04  
5. GAME 05
```

Berikut adalah tampilan ketika belum ada game yang dimainkan (Stack history kosong).

```
ENTER COMMAND >> history 9000  
  
History game kosong. Tidak ada game yang telah dimainkan.
```

Berikut adalah tampilan ketika user memberi masukan n sebesar 0.


```
ENTER COMMAND >> history 0
Masukan tidak valid. Masukan angka harus lebih besar dari 0.
```

Berikut adalah tampilan ketika user salah memberi masukan n (kosong/bukan integer diatas 0).

```
ENTER COMMAND >> history
Command tidak dikenali. Silahkan masukan command yang valid.

ENTER COMMAND >> history bukan_angka
Command tidak dikenali. Silahkan masukan command yang valid.
```

Berikut adalah tampilan sebelum dan sesudah play game (uji coba inserthistory)

Kondisi	Gambar
History sebelum ujicoba inserthistory	<pre>ENTER COMMAND >> history 9000 Berikut adalah daftar game yang telah dimainkan 1. GAME 01 2. GAME 02 3. GAME 03 4. GAME 04 5. GAME 05</pre>
Proses “PLAY GAME”	<pre>ENTER COMMAND >> play game Berikut adalah daftar antrian game-mu 1. GAME 06 Loading GAME 06 ... ===== G A M E O V E R ===== Skor akhir = 59 Nama >> verse6 Skor berhasil dimasukkan ke dalam scoreboard!</pre>

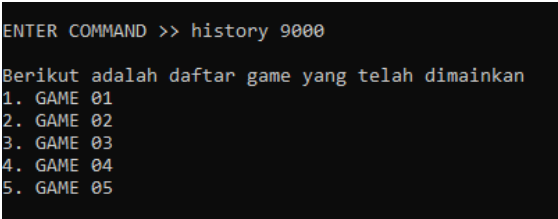
History setelah memainkan GAME 06	<pre> ENTER COMMAND >> history 9000 Berikut adalah daftar game yang telah dimainkan 1. GAME 06 2. GAME 01 3. GAME 02 4. GAME 03 5. GAME 04 6. GAME 05 </pre>
-----------------------------------	---

Berikut adalah tampilan sebelum dan sesudah delete game yang sudah ada di stack history.

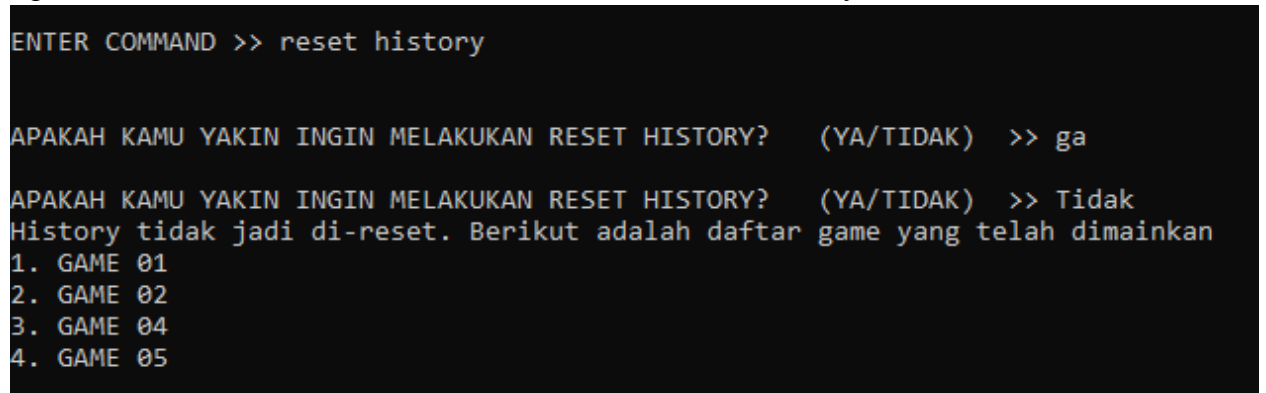
Kondisi	Gambar
History sebelum ujicoba deletehistory	<pre> ENTER COMMAND >> history 9000 Berikut adalah daftar game yang telah dimainkan 1. GAME 01 2. GAME 02 3. GAME 03 4. GAME 04 5. GAME 05 </pre>
Proses “DELETE GAME”	<pre> ENTER COMMAND >> delete game Berikut adalah daftar game yang tersedia 1. RNG 2. DINER DASH 3. HANGMAN 4. TOWER OF HANOI 5. SNAKE ON METEOR 6. GAME1 7. GAME 01 8. GAME 02 9. GAME 03 10. GAME 04 11. GAME 05 Masukkan nomor game yang akan dihapus >> 9 Game berhasil dihapus. </pre>
History setelah menghapus GAME 03	<pre> ENTER COMMAND >> history 9000 Berikut adalah daftar game yang telah dimainkan 1. GAME 01 2. GAME 02 3. GAME 04 4. GAME 05 </pre>

5.4 Reset History

RESET HISTORY berfungsi untuk me-”reset” history yang sudah ada sebelumnya. Berikut contoh tampilan sebelum dan sesudah melakukan

Kondisi	Gambar
History sebelum ujicoba “RESET HISTORY”	 <pre>ENTER COMMAND >> history 9000 Berikut adalah daftar game yang telah dimainkan 1. GAME 01 2. GAME 02 3. GAME 03 4. GAME 04 5. GAME 05</pre>
Proses “RESET HISTORY”	 <pre>ENTER COMMAND >> reset history APAKAH KAMU YAKIN INGIN MELAKUKAN RESET HISTORY? (YA/TIDAK) >> ya History berhasil di-reset.</pre>
History setelah di-”reset”	 <pre>ENTER COMMAND >> history 9000 History game kosong. Tidak ada game yang telah dimainkan.</pre>

Berikut tampilan ketika user salah melakukan input saat diminta masukkan YA/TIDAK, dan tampilan ketika user memilih untuk TIDAK melakukan reset history



```
ENTER COMMAND >> reset history

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

APAKAH KAMU YAKIN INGIN MELAKUKAN RESET HISTORY?  (YA/TIDAK) >> Tidak
History tidak jadi di-reset. Berikut adalah daftar game yang telah dimainkan
1. GAME 01
2. GAME 02
3. GAME 04
4. GAME 05
```

5.5 Tower of Hanoi

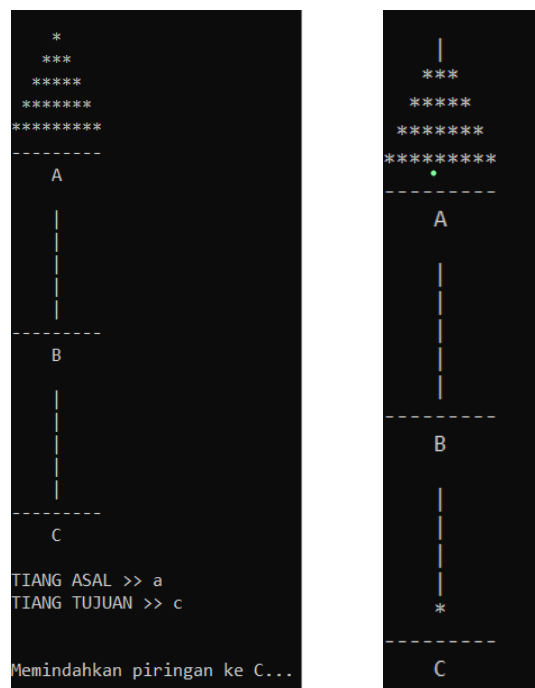
Pada Tower of Hanoi, terdapat tiga tiang. Tiga tiang tersebut adalah tiang A, tiang B, dan tiang C dan posisinya terurut dari kiri ke kanan. Pada tiang A, terdapat n piringan, dengan

piringan paling bawah merupakan piringan yang paling besar dan piringan paling atas merupakan piringan yang paling kecil.

Cara bermainnya mudah, yaitu piringan-piringan tersebut harus dipindahkan ke tiang C dengan posisi yang sama (piringan paling bawah merupakan piringan yang paling besar dan piringan paling atas merupakan piringan yang paling kecil), dengan peraturannya adalah piringan yang di bawah tidak boleh lebih besar daripada piringan yang ada di atasnya. Selain itu, pemain juga tidak dapat memindahkan piringan dari tiang yang sama. Terakhir, pemain tidak dapat memindahkan piringan dari tiang yang kosong. Apabila pemain berkehendak melanggar batasan tersebut, program akan mengeluarkan pesan kesalahan

Terdapat beberapa aturan dan mekanisme tambahan dalam permainan ini. Pemain dapat menentukan jumlah piringan yang akan ia mainkan. Sistem skoring dilakukan dengan menyesuaikan score acuan dengan jumlah piringan. Piringan yang lebih banyak akan mendapatkan skor yang lebih banyak dibandingkan dengan piringan yang lebih sedikit.

Berikut adalah tampilan ketika pemain memindahkan piringan dari tiang satu ke tiang lainnya dengan benar.



Pemain tidak dapat memindahkan piringan dari tiang kosong. Jika pemain melanggar batasan ini, program akan mengembalikan pesan kesalahan sampai pemain memasukkan input yang valid.

tersedia, program akan mengembalikan pesan kesalahan dan pemain harus memasukkan kembali input tiang sampai valid.

```

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

      |
      |
      |
      |
****
-----
B
          .

      |
      |
      |
    *
*****
*****
*****
-----
C

TIANG ASAL >> g
TIANG TUJUAN >> k

Masukan tiang asal tidak valid.
Masukan tiang tujuan tidak valid.
```

Berikut adalah tampilan ketika pemain berhasil menyelesaikan permainan.

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

Selamat, Anda berhasil!

=====
||      G A M E   O V E R      ||
=====

Skor akhir = 9


Nama >> BNM0

Skor berhasil dimasukkan ke dalam scoreboard!
```

5.6 Snake on Meteor

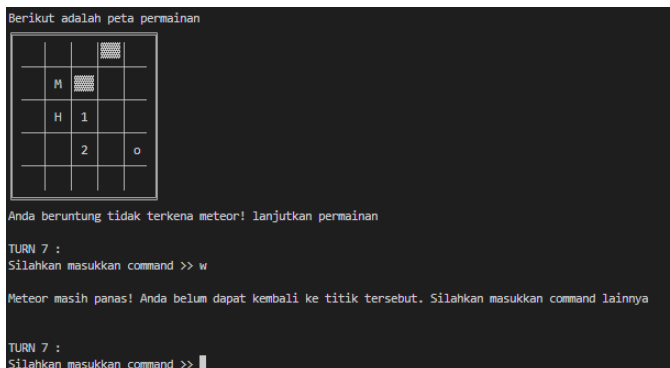
game snake on meteor merupakan game snake yang biasa kita mainkan tetapi ada sedikit perubahan yang dilakukan seperti menambahkan meteor yang dapat membunuh ular tersebut. snake on meteor memiliki banyak kasus yang harus ditangani.

kasus yang pertama adalah ketika kita memasukan input selain 'w', 'a', 's', dan 'd' maka akan menampilkan tampilan sebagai berikut.



```
TURN 1 :  
Silahkan masukkan command >> h  
  
Command tidak valid! Silahkan input command menggunakan huruf w/a/s/d  
  
TURN 1 :  
Silahkan masukkan command >> |
```

data tes kasus selanjutnya adalah ketika kita ingin bergerak ke meteor yang baru saja jatuh, maka kita akan diminta masukan input kembali karena kita tidak bisa bergerak ke wilayah tersebut.



```
Berikut adalah peta permainan  
  
Anda beruntung tidak terkena meteor! lanjutkan permainan  
  
TURN 7 :  
Silahkan masukkan command >> w  
  
Meteor masih panas! Anda belum dapat kembali ke titik tersebut. Silahkan masukkan command lainnya  
  
TURN 7 :  
Silahkan masukkan command >> |
```

kasus selanjutnya adalah ketika kita ingin bergerak ke badan sendiri maka kita akan diminta masukan input kembali.

```
Berikut adalah peta permainan
+---+
|   |   |   |   |
+---+
|   |   |   |   |
+---+
| o | M |   |   |
+---+
|   | H | 1 |   |
+---+
| 3 | 2 |   |   |
+---+

Anda beruntung tidak terkena meteor! lanjutkan permainan

TURN 4 :
Silahkan masukkan command >> s

Anda tidak dapat bergerak ke tubuh anda sendiri!
Silahkan input command yang lain

TURN 4 :
Silahkan masukkan command >> |
```

kasus selanjutnya adalah ketika kita bergerak ke obstacle maka ular akan mati dan permainan berakhir.

```
Berikut adalah peta permainan
+---+
| 2 | M |   |   |
+---+
| 1 | H |   |   |
+---+
|   |   |   | o |
+---+
|   |   |   |   |
+---+

Anda beruntung tidak terkena meteor! lanjutkan permainan

TURN 2 :
Silahkan masukkan command >> s

Berhasil bergerak!

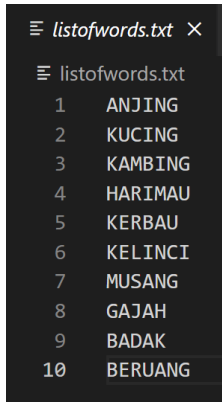
Kepala menabrak obstacel

=====
||  G A M E   O V E R  ||
=====

Skor akhir = 4
```

kasus selanjutnya adalah ketika meteor jatuh di badan ular maka panjang ular akan berkurang satu.

Pada awal permainan Hangman, akan muncul dua menu, yaitu play dan add word. User dapat memilih salah satu dari menu tersebut. Apabila user memasukkan selain play dan add word, maka program akan meminta ulang input menu hingga valid, yaitu play atau add word. Berikut adalah isi dari daftar kata yang digunakan dalam permainan Hangman pada file listofwords.txt.



Apabila user ingin menambahkan kata terlebih dahulu ke dalam daftar kata, maka user dapat melakukan input add word. Selanjutnya, program akan meminta untuk memasukkan kata yang ingin ditambahkan ke dalam daftar kata.



Program akan meminta ulang input kata sampai valid. Kata yang valid adalah hanya terdiri dari sebuah kata, bukan merupakan sebuah huruf tidak mengandung angka, dan kata tersebut belum terdapat dalam daftar kata.

```

Masukkan sebuah kata yang ingin ditambahkan ke dalam daftar kata >> burung hantu
Masukkan kata tidak valid. Anda hanya dapat memasukkan sebuah kata.
Masukkan sebuah kata yang ingin ditambahkan ke dalam daftar kata >> b
Masukkan kata tidak valid. Mohon masukkan sebuah kata, bukan huruf.
Masukkan sebuah kata yang ingin ditambahkan ke dalam daftar kata >> b4b1
Masukkan kata tidak valid. Kata hanya boleh mengandung huruf.
Masukkan sebuah kata yang ingin ditambahkan ke dalam daftar kata >> babi
Kata BABI berhasil ditambahkan ke dalam daftar kata.

```

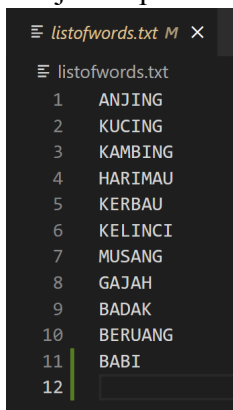
Selanjutnya, program akan menanyakan apakah user masih ingin menambahkan kata ke dalam daftar kata. Program akan meminta ulang input sampai valid, yaitu ya atau tidak.

```

Apakah Anda masih ingin menambahkan kata ke dalam daftar kata? (Ya/Tidak) >> gatau
Apakah Anda masih ingin menambahkan kata ke dalam daftar kata? (Ya/Tidak) >> tidak

```

Setelah memilih untuk tidak lagi menambahkan kata ke dalam daftar kata, maka program akan menyimpan kata yang telah ditambahkan ke dalam listofwords.txt, sehingga isi file tersebut menjadi seperti berikut.



```

listofwords.txt
1  ANJING
2  KUCING
3  KAMBING
4  HARIMAU
5  KERBAU
6  KELINCI
7  MUSANG
8  GAJAH
9  BADAK
10 BERUANG
11 BABI
12

```

Kemudian, program akan menanyakan apakah user ingin bermain Hangman. Program akan meminta ulang input sampai valid, yaitu ya atau tidak.

```

Apakah Anda ingin bermain Hangman? (Ya/Tidak) >> gatau
Apakah Anda ingin bermain Hangman? (Ya/Tidak) >> ya

```

Setelah memilih untuk bermain Hangman, maka program akan menampilkan tebakan sebelumnya yang masih kosong karena belum ada huruf yang ditebak, menampilkan banyak huruf pada kata yang harus ditebak dalam bentuk underscore, menampilkan kesempatan awal yang masih berjumlah 10, dan gambar hangman yang masih kosong. Program akan meminta user untuk memasukkan tebakan sampai valid, yaitu hanya merupakan sebuah huruf, tidak merupakan angka maupun kata, dan huruf tersebut belum pernah ditebak sebelumnya untuk kata yang sama.

[illegible][illegible]

[illegible]

```
Tebakan sebelumnya : A, S, D, P, K
Kata : _ A D A K
Kesempatan : 3
```

```
|_|
| |0
|---|---
| |
| / 
|
|_
---
```

```
Masukkan tebakan >> p

Anda sudah menebak huruf P sebelumnya. Mohon masukkan huruf lain.
```

```
Masukkan tebakan >> j
Tebakan Anda salah. Huruf J tidak terdapat pada kata.
```

[illegible]

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Scoreboard	Memeriksa apakah skor permainan tersimpan ke dalam scoreboard	Memainkan sebuah permainan dan memasukkan nama yang valid ke dalam scoreboard.	Data Test 1	Nama dan nilai akan dimasukkan ke dalam scoreboard permainan secara terurut menurun.	Sesuai hasil yang diharapkan
		Memeriksa error handling 1 (nama yang dimasukkan	Memainkan sebuah permainan dan memasukkan nama yang sudah pernah dima-		Fungsi menolak masukan tersebut dan meminta masukan baru yang valid	

		pengguna sudah ada di dalam scoreboard)	sukkan sebelum nya ke dalam scoreboard permainan tersebut			
		Memeriksa error handling 2 (nama terdiri atas 2 kata)	Memasukkan nama yang terdiri atas 2 kata atau lebih setelah memainkan sebuah permainan		Fungsi menolak masukan tersebut dan meminta masukan baru yang valid.	
		Memeriksa apakah scoreboard sudah terbentuk untuk permainan yang baru dibuat/ditambahkan	Menambahkan permainan baru menggunakan fungsi "CREATE GAME" dan membuka scoreboard seluruh permainan.		Scoreboard akan bertambah satu sesuai urutan pada daftar permainan.	
		Memeriksa apakah scoreboard permainan yang dihapus juga dihilangkan.	Menghapus permainan menggunakan fungsi "DELETE GAME" dan membuka scoreboard seluruh permainan.		Scoreboard untuk permainan tersebut dihapus dan scoreboard yang tersisa ditampilkan berdasarkan urutan pada daftar permainan.	
2	Reset Scoreboard	Memeriksa apakah scoreboard untuk permainan yang dipilih dihapus (menjadi kosong).	Memilih salah satu game yang memiliki scoreboard, mengonfirmasi dengan menjawab "Ya" dan memeriksa apakah scoreboard sudah dihapus dengan fungsi "SCOREBOARD".	Data Test 2	Scoreboard yang dipilih menjadi kosong.	Sesuai hasil yang diharapkan
		Memeriksa apakah semua scoreboard dihapus (menjadi kosong) ketika memilih opsi "0. ALL"	Memilih opsi "0", mengonfirmasi dengan menjawab "Ya" dan memeriksa apakah scoreboard sudah dihapus dengan fungsi "SCOREBOARD".		Seluruh scoreboard menjadi kosong.	
		Memeriksa error handling 1 (pilihan scoreboard yang ingin dihapus kosong).	Tidak memberi masukan ketika diminta memilih scoreboard game yang ingin dihapus.		Program akan menolak masukan dan akan meminta masukan yang valid.	
		Memeriksa error handling 2 (pilihan scoreboard yang ingin dihapus bukan integer).	Memberi masukan non integer ketika diminta memilih scoreboard game yang ingin dihapus.		Program akan menolak masukan dan akan meminta masukan yang valid.	
		Memeriksa error handling 3 (pilihan scoreboard yang	Mengisi angka sembarang yang bukan merupakan		Program akan menolak masukan	

		ingin dihapus berada di luar rentang daftar).	opsi yang tersedia ketika diminta memilih scoreboard game yang ingin dihapus.		dan akan meminta masukan yang valid.	
		Memeriksa apakah scoreboard dihapus apabila pengguna mengonfirmasi dengan menjawab “Ya”.	Memilih salah satu/ semua game dan mengonfirmasi penghapusan dengan menjawab “Ya”.		Scoreboard yang dipilih dihapus apabila pengguna menjawab “Ya” ketika diminta mengonfirmasi penghapusan.	
		Memeriksa apakah scoreboard batal dihapus apabila pengguna menjawab “Tidak”.	Membatalkan penghapusan dengan menjawab “Tidak” setelah memilih permainan yang ingin dihapus.		Scoreboard yang dipilih batal dihapus apabila pengguna menjawab “Tidak” ketika diminta mengonfirmasi penghapusan.	
		Memeriksa error handling 4 (masukan pada saat konfirmasi bukan “Ya” atau “Tidak”).	Mengonfirmasi penghapusan dengan respon sembarang selain “Ya” atau “Tidak”.		Program akan terus meminta konfirmasi hingga mendapat masukan yang sesuai.	
3	History	Memeriksa apakah ketika game yang sudah dimainkan masuk ke stack history	Melakukan play game dan menggunakan command history 1	Data Test 3	Muncul nama game yang baru saja dimainkan	Sesuai hasil yang diharapkan
		Memeriksa kasus ketika n yang dimasukkan sebesar 0	Melakukan command history 0		Program akan menolak masukkan dan meminta masukan yang valid	
		Memeriksa kasus ketika n yang diberikan melebihi history game yang sudah dimainkan	Melakukan command history namun dengan n yang tinggi (contoh: HISTORY 9000)		Muncul seluruh game yang sudah dimainkan. (banyaknya sesuai dengan banyak melakukan playgame)	
		Memeriksa kasus ketika n bukan merupakan integer	Melakukan command history namun dengan masukkan n salah (contoh: HISTORY blabla)		Program akan menolak masukkan dan meminta masukan yang valid	
		Memeriksa ketika tidak memberikan input n yang diminta (command hanya History tanpa <n>)	Melakukan command history namun tanpa memberi masukkan n (contoh: HISTORY)		Program akan menolak masukkan dan meminta masukan yang valid	
		Memeriksa kasus ketika melakukan	Melakukan DELETE GAME terhadap		Muncul list history game tanpa game	

		DELETE GAME terhadap game yang sudah ada di Stack history	game yang sudah ada di history, lalu menggunakan command history untuk mengecek kembali		yang sudah di-"delete"	
4	Reset History	Memeriksa apakah scoreboard untuk permainan yang dipilih dihapus (menjadi kosong).	Melakukan command history, lalu reset history, lalu kembali melakukan command history	Data Test 4	List history yang awalnya ada, menjadi kosong	Sesuai hasil yang diharapkan
		Memeriksa apakah history dihapus apabila pengguna mengonfirmasi dengan menjawab "Ya".	Melakukan command "RESET HISTORY" lalu mengonfirmasi "Ya"		History terhapus dan kosong ketika melakukan command history	
		Memeriksa apakah history batal dihapus apabila pengguna menjawab "Tidak".	Melakukan command "RESET HISTORY" lalu mengonfirmasi "Tidak"		History batal dihapus apabila pengguna menjawab "Tidak" ketika diminta mengonfirmasi penghapusan.	
		Memeriksa error handling (masukkan pada saat konfirmasi bukan "Ya" atau "Tidak").	Melakukan command "RESET HISTORY" lalu mengonfirmasi dengan masukkan yang salah		Program akan terus meminta konfirmasi hingga mendapat masukan yang sesuai.	
5	Tower of Hanoi	Memeriksa apakah piringan dapat dipindahkan.	Di tahap awal, menginput tiang awal: A, tiang akhir: B.	Data Test 5	Piringan berhasil dipindahkan.	sesuai hasil yang diharapkan
		Memeriksa error handling 1 (memindahkan piringan yang ukurannya lebih besar ke atas piringan lebih kecil).	Memindahkan piringan lebih besar ke atas piringan yang lebih kecil.		Program mengembalikan pesan kesalahan. Meminta melakukan input kembali.	
		Memeriksa error handling 2 (memindahkan piringan dari tiang kosong).	Memindahkan piringan dari tiang kosong ke tiang lain.		Program mengembalikan pesan kesalahan. Meminta melakukan input kembali.	
		Memeriksa error handling 3 (memindahkan piringan dari tiang yang tidak tersedia).	Melakukan input tiang selain A, B, dan C		Program mengembalikan pesan kesalahan. Meminta melakukan input kembali.	

		Memeriksa kondisi akhir permainan.	Menyelesaikan seluruh permainan. Final state nya adalah tiang C terisi penuh.		Ditampilkan pesan bahwa permainan berakhir dan skor akhir pemain.	
6	Snake On Meteor	memeriksa apakah ular dapat bergerak saat kita memasukkan input selain comand untuk bergerak	memasukan command input 'h'	Data Test 6	program akan meminta input kembali	
		memeriksa apakah ular dapat bergerak ke wilayah yang baru dijatuhi meteor	bergerak ke arah meteor yang baru saja jatuh		program akan meminta input kembali	
		memeriksa apa yang terjadi saat kita ingin bergerak ke badan sendiri	menggerakan ular ke badannya sendiri		program aka meminta input kembali. apabila tidak dapat bergerak kemana mana lagi, maka permainan akan berakhir	
		memeriksa ketika ular menabrak obstacle	menggerakan ular ke arah obstacle		ular mati dan permainan akan berakhir	
		memeriksa ketika badan ular terkena meteor	menunggu sampai meteor jatuh tepat ke badan ular		panjang ular akan berkurang satu	
		memeriksa ketika kepala ular terkena meteor	menunggu sampai meteor jatuh tepat ke kepala ular		ular mati dan permainan akan berakhir	
7	Hangman	Memeriksa apakah bisa memasukkan menu selain play dan add word	Memasukkan input menu selain play dan add word, misal "menu lain"	Data Test 7	Program akan meminta ulang input menu	Sesuai hasil yang diharapkan
		Memeriksa apakah bisa menambahkan kata yang telah terdapat dalam daftar kata pada "listofwords.txt"	Memasukkan input "anjing" yang sudah terdapat pada daftar kata		Program akan meminta ulang input kata	
		memeriksa apakah bisa menambahkan kata hanya 1 huruf	Memasukkan input "b"		Program meminta ulang input kata	
		memeriksa apakah bisa menambahkan kata yang terdiri 2 kata	Memasukkan input "burung hantu"		Program akan meminta ulang input kata	
		memeriksa apakah bisa menambahkan kata yang mengandung angka	Memasukkan input "b4b1"		Program akan meminta ulang input kata	

	memeriksa apakah bisa menambahkan sebuah kata yang belum terdapat pada daftar kata	Memasukkan input “babi”		Program menambahkan kata ke dalam daftar kata dan tersimpan dalam “listofwords.txt”	
	memeriksa apakah pemain dapat menebak angka	Memasukkan input “1”		program meminta ulang input tebakan	
	memeriksa apakah pemain dapat menebak kata	memasukkan input “saya”		program meminta ulang input tebakan	
	memeriksa apakah pemain dapat menebak huruf yang telah ditebak sebelumnya	memasukkan input “a” yang merupakan huruf yang pernah ditebak sebelumnya		program meminta ulang input tebakan	
	memeriksa apakah tebakan salah akan membuat kesempatan berkurang	memasukkan input “z” yang tidak terdapat pada kata yang harus ditebak		kesempatan berkurang 1	

7 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1	Program utama	18221049, 18221073, 18221085, 18221105, 18221149	18221049, 18221073, 18221085, 18221105, 18221149
2	ADT Stack	18221073	18221073
3	ADT Map	18221073, 18221049	18221073, 18221049
4	ADT Array Of Map	18221073	18221073
5	ADT Linked List	18221085	18221085
6	ADT Matriks	18221085, 18221073	18221085
7	Scoreboard	18221149, 18221073	18221149, 18221073
8	Reset Scoreboard	18221149, 18221073	18221149, 18221073
9	History	18221049	18221049

10	Reset History	18221049	18221049
11	Hangman	18221073	18221073
12	Tower of Hanoi	18221105	18221105, 18221073
13	Snake on Meteor	18221085	18221085, 18221073

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.

BNMO merupakan suatu robot game console yang dapat menjalankan permainan. BNMO memiliki beberapa fitur utama, yaitu:

1. Memainkan game
2. Menambahkan game
3. Menghapus game
4. Mengurutkan game yang akan dimainkan
5. Menampilkan game yang telah dimainkan
6. Menampilkan scoreboard game

8.2 Notulen Rapat

Pembagian tugas tanggal 22 November :

- scoreboard & reset scoreboard - rayhan
- history <n> & reset history - silvester
- hangman - jessica
- tower of hanoi - alief
- snake on meteor - nalen

8.3 Log Activity Anggota Kelompok

NIM	Nama	Log Activity
18221049	Silvester Kresna W.P.P.	22/11/2022 : bagi-bagi tugas 30/11/2022 : history dan reset history 01/12/2022 : menggabungkan history dan reset history dengan main, testing dan debug sampai tuntas 02/12/2022 : laporan
18221073	Jessica	27/11/2022 : Membuat dan menyelesaikan Hangman 28/11/2022 : Menambahkan parameter output pada RNG dan skorgame, membuat ADT Stack, melakukan update konfigurasi.txt, memperbaiki Tower Of Hanoi, mengubah nama-nama prosedur dan fungsi yang sama 29/11/2022 : Membuat ADT Map dan Array Of Map, memperbaiki Load 30/11/2022 : Memperbaiki Save, mengubah Display snake, melakukan update Array Of Map 01/12/2022 : Melakukan update ADT Map, memperbaiki Scoreboard, memeriksa Snake On Meteor 02/12/2022 : Merapikan file, menulis laporan
18221085	Nalendro N.U.A.R.F.	29/11/2022 : menambahkan game snake on meteor. 1/12/2022 : update snake on meteor 2/12/2022 : update bonus snake on meteor 2/12/2022 : menulis laporan
18221105	Muhammad Aliefnaufal Permana	23/11/2022 : Selesai Tower of Hanoi Mark1 24/11/2022 : Selesai Tower of Hanoi Mark2 26/11/2022 : Selesai ToH fix 2/12/2022 : Menyelesaikan laporan
18221149	Rayhan Nugraha Putra	23/11/2022 : Pembagian tugas 29/11/2022 : Menambahkan fungsi scoreboard dan reset scoreboard dan fungsi tambahan pada ADT Map dan Array of Map. 30/11/2022 : Memasukkan scoreboard ke

		dalam fungsi creategame dan deletegame 01/12/2022 : Testing semua fungsi yang berhubungan dengan scoreboard 02/12/2022 : Menulis laporan
--	--	--






8.4 Form Asistensi Tugas Besar

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

No. Kelompok/Kelas : 09 / K01
 Nama Kelompok : Allstrokedead
 Anggota Kelompok (Nama/NIM) :
 1. Silvester Kresna W. P. P. / 18221049
 2. Jessica / 18221073
 3. Nalendro N. U. A. R. F. / 18221085
 4. Muhammad Aliefnaufal P. / 18221105
 5. Rayhan Nugraha Putra / 18221149

Asisten Pembimbing : Afif Fahreza






Asistensi I

Tanggal : 25 November 2022	Catatan Asistensi:
Tempat : Zoom	
<p>Kehadiran Anggota Kelompok:</p> <p>1 18221049</p>  <p>2 18221073</p>  <p>3 18221085</p>  <p>4 18221105</p>  <p>5 18221149</p> 	
	Tanda Tangan Asisten:

- Scoreboard sama history bagusnya menggunakan ADT apa? Kalau scoreboard menggunakan ADT map. Kalau history menggunakan ADT stack
- Setiap set dan map dibuat berbeda di setiap game, berarti harus pakai array of set, array of map, atau map of map ya? Bisa.
- Score di snake itu sesuai dengan panjang snake. Score merepresentasikan panjang snake. Di contoh, score 8 itu dari mana ya? Nanti coba di confirm lagi. Tapi seharusnya 4, kepalanya hilang, jadi sisanya 4.
- Stack gaboleh akses indeks. Untuk stack di tower of hanoi gaboleh akses indeks.
- Skor untuk setiap penambahan move dari jumlah move optimalnya di tower of hanoi berkurang 1.
- Di hangman, yang bonus, ada 2 pilihan : main langsung atau tambah kata, apakah bisa tambah kata saja tanpa main? Bisa cuma tambah kata aja.
- Di hangman, kalau tambah kata, cuma boleh 1 kata atau boleh 2 kata? 1 kata aja, kalau 2 kata dianggap tidak valid.
- ADT tree dibuat hanya kalau buat game custom tree.
- Laporan tubes 2 buat baru lagi, tidak melanjutkan dari tubes 1. ADT yang sudah dimasukkan di laporan tubes 1 tidak perlu dimasukkan lagi di tubes 2.

	
--	--

Asistensi II

Tanggal : 2 Desember 2022	Catatan Asistensi: <ul style="list-style-type: none"> - Nama untuk scoreboard harus dipastikan terisi, tidak boleh kosong - Laporan milestone 1 kurang keterhubungan antar modul, misalkan command history, itu terhubung dengan ADT apa saja, masukkan ADT yang digunakan di setiap command, lebih baik lagi apabila dibuat dengan diagram digaris-garisin - Driver ADT tidak wajib - Readme wajib, isi dengan bagaimana menjalankan program
Tempat : Zoom	
Kehadiran Anggota Kelompok: <div style="text-align: center;"> <p>1 18221049</p>  <p>2 18221073</p>  <p>3 18221085</p>  <p>4 18221105</p>  <p>5 18221149</p>  </div>	
	Tanda Tangan Asisten: 