

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data


BNMO

Dipersiapkan oleh:

Kelompok 8

1. Michael Sihotang (18221054)
2. Muhammad Dastin Fauzi (18221062)
3. Imanuel Raditya (18221112)
4. Miralistya Cahya Fatimah (18221116)
5. Seren Elizabeth Siahaan (18221160)

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-08</i>		39
		<i>Revisi</i>	<i>01</i>	<i>28 Oktober 2022</i>

Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Fitur Obstacle pada Snake on Meteor	4
2.2 Fitur Menghubungkan sisi Peta yang Berseberangan	5
2.3 Fitur Opsi Jumlah Piringan	5
2.4 Fitur Penambahan Kata pada Hangman	6
2.5 Fitur Penyimpanan Kata pada File txt	7
3 Struktur Data	7
3.1 Struktur Data Stack	8
3.2 Struktur Data Array Map	9
3.3 Struktur Data Linked List	9
4 Program Utama	9
5 Algoritma-Algoritma Menarik	10
5.1 Pengkondisian Makan pada Snake on Meteor	10
6 Data Test	11
6.1 Data Test SCOREBOARD	11
6.2 Data Test RESET SCOREBOARD	14
6.3 Data Test HISTORY	16
6.4 Data Test RESET HISTORY	16
6.5 Data Test Hangman	17
6.6 Data Test Tower of Hanoi	20
6.7 Data Test Snake on Meteor	22
6.8 Data Test Load	24
6.9 Data Test Save	26
7 Test Script	29
8 Pembagian Kerja dalam Kelompok	31
9 Lampiran	31
9.1 Deskripsi Tugas Besar 2	31
9.2 Notulen Rapat	34
9.3 Log Activity Anggota Kelompok	38

1 Ringkasan

BNMO merupakan suatu program robot video *game* yang dapat menjalankan suatu permainan di dalamnya. Pada BNMO ini ada beberapa hal yang dapat dilakukan oleh pengguna yaitu memainkan *game*, menambahkan *game*, menghapus *game*, mengurutkan *game* yang akan dimainkan, menampilkan *game* yang telah dimainkan, dan menampilkan scoreboard *game*. Semua perubahan yang kita lakukan terhadap program *game* pada saat bermain seperti menambahkan dan juga menghapus *game* dapat disimpan pada sebuah *file* dengan *type* txt. Berikut adalah penjelasan mengenai cara kerja dari program BNMO ini. Pada program ini, terdapat lima *game* yang pasti bisa dimainkan oleh pengguna, yaitu RNG, DINER DASH, HANGMAN, TOWER OF HANOI, dan SNAKE ON METEOR. Namun, pada sistem ini tentunya ada fitur untuk melakukan pembuatan *game* yang baru sehingga *game* yang lain juga dapat dimainkan dalam sistem ini.

Pada saat awal memulai program BNMO, program akan menampilkan sebuah *main menu* yang berisi *welcome page*. Pada fase tersebut, pengguna diberi dua pilihan opsi mengenai cara untuk mulai menjalankan program *game*. Opsi tersebut adalah START dan LOAD. *Command* START atau LOAD adalah *command* pertama yang harus diberikan oleh pengguna. Para pengguna dapat memilih satu di antara dua *command* tersebut. Ketika pengguna memberikan *command* lain (selain START dan LOAD) sebagai perintah pertama, program akan menampilkan bahwa masukan dari pengguna salah, dan akan diminta untuk memberikan perintah kembali sampai perintah yang diberikan oleh pengguna adalah salah satu di antara START dan LOAD. Untuk penjelasan lebih lanjut mengenai START dan LOAD telah dijelaskan pada laporan Tugas Besar 1.

Saat pengguna telah memilih cara dalam memulai menjalankan program, pengguna dapat menjalankan fitur-fitur yang terdapat pada program BNMO melalui pemanggilan *command*. Daftar *command* yang dapat diberikan adalah antara lain CREATE GAME, LIST GAME, QUEUE GAME, PLAY GAME, SKIP GAME, DELETE GAME, SAVE, HELP, dan QUIT dimana *command* ini sudah dijelaskan pada laporan Tugas Besar 1. Adapun *command* tambahan yang dapat diberikan antara lain SCOREBOARD, RESET SCOREBOARD, HISTORY, RESET HISTORY. Apabila pengguna memberikan *command* selain *command-command* yang telah disebutkan, maka masukan dari pengguna tersebut akan dianggap tidak *valid* dan tidak akan diproses oleh program. Penjelasan singkat mengenai *command* fitur - fitur tambahan pada program BNMO adalah sebagai berikut.

Pertama, *Command* SCOREBOARD adalah perintah yang digunakan untuk menampilkan nama dan skor setiap pemain pada setiap *game* yang ada. Pada setiap *game over* atau menang pada *game*, maka program akan meminta untuk *input* nama pemain untuk dimasukkan kedalam *scoreboard*. Nama pemain yang *valid* adalah nama yang belum terpakai di *scoreboard game* yang sedang dimainkan. Urutan pada *scoreboard* yang ditampilkan akan mengikuti urutan pada *command* LIST GAME. Urutan nama pada *scoreboard* akan diurutkan mulai dari skor yang tertinggi. Apabila ada skor yang sama, maka skor yang lebih dahulu dimasukkan akan berada pada urutan yang lebih dahulu. Apabila pemain melakukan *command* CREATE GAME, maka setiap *game* yang dibuat akan dimasukkan kedalam *scoreboard*. Apabila pemain melakukan *command* PLAY GAME pada *game* yang dibuat sendiri, maka akan langsung dianggap *game over* dan menghasilkan skor yang random. Skor yang dihasilkan akan dimasukkan kedalam *scoreboard*. Kedua, *Command* RESET SCOREBOARD yang digunakan

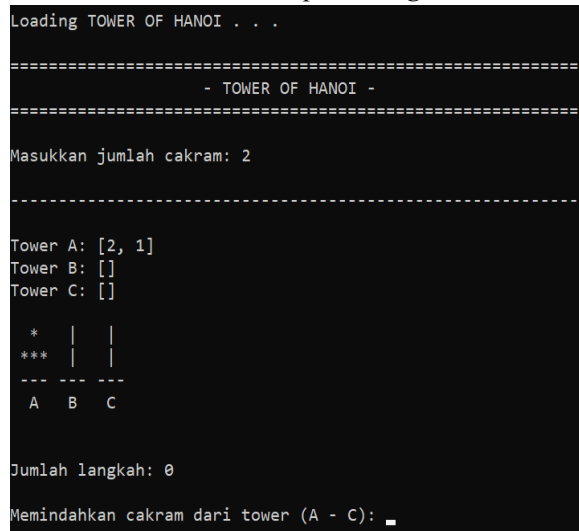
untuk melakukan reset terhadap *scoreboard* permainan. Reset dapat dilakukan untuk menghapus semua informasi pada setiap permainan maupun memilih salah satu permainan untuk direset. Ketiga, *Command HISTORY* yang digunakan untuk melihat permainan apa saja yang telah dimainkan dari data yang sudah ada dari file konfigurasi (jika LOAD) dan dari mulai *Start Game* juga, dengan <n> adalah jumlah permainan yang telah dimainkan yang ingin ditampilkan. Urutan teratas merupakan permainan terakhir yang dimainkan. Jika <n> lebih besar dari jumlah permainan yang telah dimainkan, akan menampilkan seluruh permainan yang telah dimainkan. Keempat, *Command RESET HISTORY* yang merupakan *command* yang digunakan untuk menghapus semua *history* permainan yang dimainkan. Selain menambahkan beberapa command, kami juga menambahkan beberapa permainan kedalam program BNMO ini. Permainan tersebut di antaranya adalah Tower Of Hanoi, Hangman, dan Snake On Meteor. Permainan tersebut dapat dimainkan oleh *user* secara langsung dan akan menghasilkan sebuah *score* di akhir permainan.

Program BNMO ini menggunakan bahasa pemrograman C dan akan menerapkan beberapa struktur data yang sesuai. ADT yang digunakan pada program ini adalah ADT Mesin Karakter dan Mesin Kata, ADT Queue, ADT Array, ADT Map, ADT Linked List, dan ADT Stack. Hal - hal lain mengenai ADT yang digunakan pada program ini akan dijelaskan dengan lebih *detail* pada poin nomor tiga. Pada program BNMO yang kami buat juga terdapat beberapa fitur tambahan yang akan dijelaskan pada poin nomor dua. Selain itu, terdapat program utama atau *main* yang digunakan sebagai wadah sinkronisasi antar program *command* sehingga program BNMO dapat dijalankan secara utuh dan selaras. Hal - hal mengenai program utama akan dijelaskan secara lebih lanjut pada poin nomor empat, diikuti dengan Data Test dan juga Test Script mengenai program BNMO secara keseluruhan yang akan dijabarkan pada poin nomor enam dan tujuh.

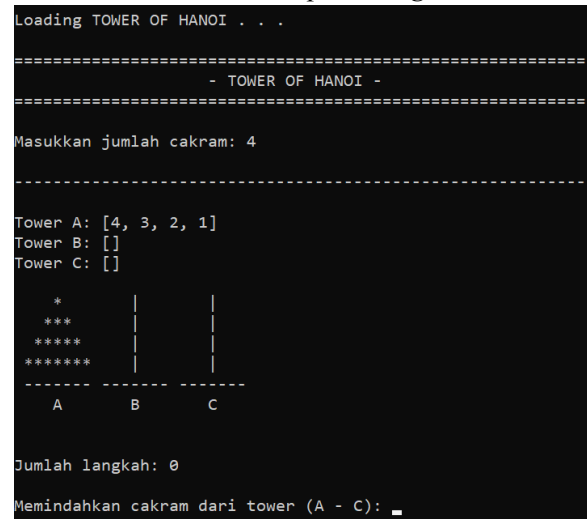
2.3 Fitur Opsi Jumlah Piringan

Fitur opsi jumlah piringan ini ada pada *game* Tower of Hanoi. Sebelum permainan dimulai, akan diminta opsi jumlah piringan yang digunakan. Kemudian, skor yang diperoleh akan disesuaikan dengan jumlah piringan pada *game*.

Gambar 2.3.1 Opsi Piringan 2



Gambar 2.3.2 Opsi Piringan 3



2.4 Fitur Penambahan Kata Pada Hangman

Fitur penambahan kata pada hangman ada pada game hangman. Pada awal permainan, maka akan diminta kepada pemain untuk memilih menambahkan kata ke daftar kata di hangman atau bisa juga langsung bermain.

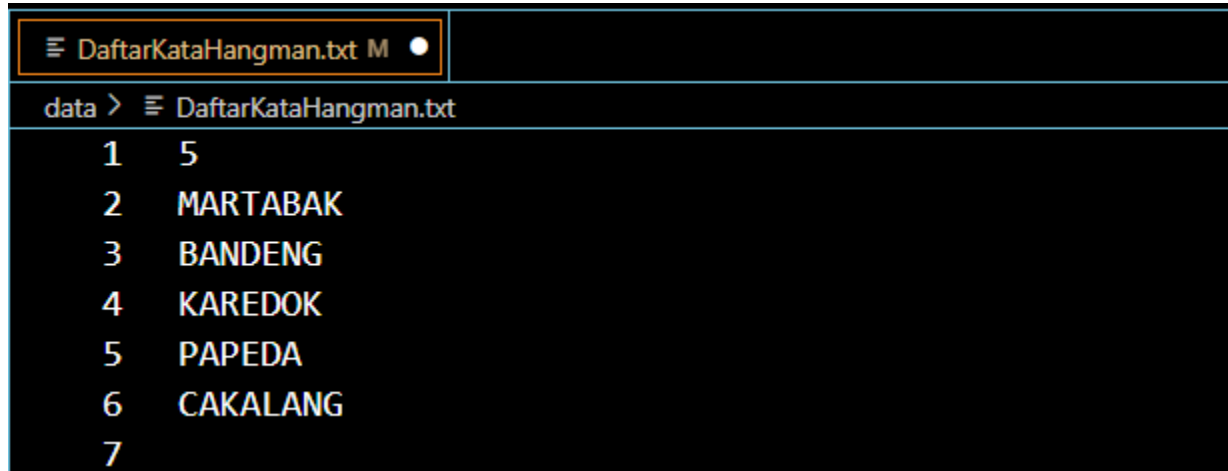
Gambar 2.4.1 Gambar penambahan kata pada hangman



2.5 Fitur Penyimpanan kata Pada File txt

Fitur penyimpanan kata pada file txt ada di game hangman sehingga setiap kata yang bisa ditebak disimpan pada sebuah file txt.

Gambar 2.4.1 Gambar fitur penyimpanan kata pada file



3 Struktur Data (ADT)

Pada program yang kami buat, kami menggunakan beberapa struktur data (ADT) untuk menyelesaikan permasalahan yang ada dan membuat program BNMO menjadi program yang dapat berjalan dengan baik dan tepat. Beberapa struktur data (ADT) yang kami gunakan adalah ADT Array, ADT Mesin Karakter, ADT Mesin Kata, ADT Queue, ADT Map, ADT Stack, dan ADT Linked List. Pada laporan Tugas Besar 2 ini, akan dijelaskan lebih lanjut penjelasan penggunaan ADT Map yang di modifikasi, ADT Stack, dan ADT Linked List.

3.1 Struktur Data Stack

Pada pengerjaan program BNMO ini, kami menggunakan struktur data *Stack*. Struktur data yang kami gunakan merupakan struktur data *stack* dengan representasi struktur berkait. Struktur data ini juga direpresentasikan secara *eksplisit* dan alokasi dinamik. Tipe data *stack* akan menyimpan *address* TOP dan count dengan *type integer*. *Element* dari *stack* akan menyimpan Info dengan *type* InfoType dan Next dengan *type address*. Info yang disimpan oleh *element* dari *stack* merupakan sebuah integer. Struktur data *stack* ini tersimpan pada *file* source/ADT/stack.c dan pada source/ADT/stack.h untuk header ADT.

Struktur data stack diterapkan pada program BNMO yang kami buat dalam pembuatan game Tower of Hanoi. Pada game ini, prinsip dari struktur data stack sangat diterapkan yaitu LIFO (Last In First Out). Pada game Tower of Hanoi, setiap cakram yang dimasukkan terakhir akan dikeluarkan terlebih dahulu agar cakram yang dibawahnya bisa diproses. Setiap pemrosesan elemen hanya bisa dilakukan pada bagian TOP yaitu elemen stack paling atas. Operasi yang digunakan ada dua yaitu Push dan Pop. Operasi Push berfungsi untuk menambahkan elemen pada stack di bagian TOP sehingga elemen yang baru menjadi TOP dari stack, sedangkan Pop berfungsi untuk menghapus elemen dari stack, menyimpan nilai elemen tersebut kedalam suatu

variabel, dan mengubah TOP menjadi (TOP-1) dari TOP sebelumnya. Pada struktur data ini juga ada prosedur *viewstack* untuk menampilkan semua info dari masing-masing elemen *stack*. Pada proses game ini, akan disediakan tiga *stack* yang dapat menjadi perpindahan dari setiap elemen. Pemain harus bisa memindahkan elemen dari *stack* pertama hingga *stack* terakhir dengan urutan elemen yang sama.

Kami memilih struktur data *stack* dengan representasi struktur berkait agar lebih mudah dalam pemindahan setiap piringannya. Konsep dari *stack* dengan representasi biasa dengan representasi berkait juga sangat mirip. Hanya saja dengan struktur berkait bisa melakukan pemindahan setiap piringan lebih mudah.

3.2 Struktur Data Array MAP

Pada tugas besar yang kedua ini, kami memanfaatkan kembali struktur data *Map* namun dengan pengaplikasian yang sedikit berbeda. Struktur data *Map* yang telah kami gunakan pada tugas besar yang sebelumnya diaplikasikan dengan lebih lanjut untuk menyelesaikan persoalan baru pada tugas besar kedua kali ini. Persoalan yang dimaksud tersebut adalah *command SCOREBOARD*. Untuk mengerjakan *SCOREBOARD*, kami membuat sebuah struktur data *Array Map* yaitu sebuah *array* yang didalamnya menyimpan data berupa tabel penyimpanan elemen *Tab*, *MapCapacity*, dan juga *MapNeff*. Elemen yang disimpan pada tabel penyimpanan *Tab* memiliki tipe *Map*, yaitu hasil bentukan dari ADT *Map*. Data *MapCapacity* pada struktur *Array Map* bertipe integer dan merupakan sebuah bilangan yang menandakan kapasitas dari *Array Map* tersebut. Sedangkan data *MapNeff* pada struktur *Array Map* memiliki tipe *integer* dan berperan sebagai penanda jumlah elemen efektif pada *Array Map*. *Array map* yang kami buat bersifat dinamis sehingga akan ada proses alokasi dan dealokasi didalamnya. Struktur data *Array Map* ini tersimpan pada *file* *source/ADT/array/arraymap.c* dan pada *source/ADT/array/arraymap.h* untuk header program.

Struktur *Array Map* digunakan pada *command SCOREBOARD* karena pada program BNMO ini, setiap permainan akan memiliki sebuah *map* yang menyimpan *score* dan juga *username* pemain. Untuk itu, semua *Map* dari masing-masing permainan perlu untuk ditampung pada sebuah wadah agar nantinya dapat diproses dan digunakan dengan baik pada *command SCOREBOARD*. Pada setiap awal dijalankan program BNMO, program akan secara otomatis menambahkan *map* sebanyak jumlah permainan yang terdapat pada *list game* pada *Array Map* ini. Ketika *command SCOREBOARD* dipanggil oleh *user*, program akan menampilkan isi dari masing-masing *Map* yaitu *key* yang berupa *username* pemain diikuti dengan *value*-nya, yaitu *score*. Pada program ini, kami tidak menggunakan ADT *Set* untuk menyimpan *username* pemain, karena pada dasarnya, *set* sendiri merupakan bagian dari *Map*, atau dengan kata lain, *Map* sudah mencakup *Set*. Pada ADT *Map*, *key* yang tersimpan haruslah unik. Maka, pada program ini, *username* yang disimpan pada *Map* haruslah unik dan akan ada prosedur pengecekan dan juga *handling* apabila *user* memasukkan *username* yang telah digunakan sebelumnya pada *Map* yang sama. Oleh karena itu, kami memutuskan untuk tidak menggunakan ADT *Set* untuk mengatasi pengecekan keunikan *username* yang disimpan.

3.3 Struktur Data Linked List

Pada program BNMO ini, terdapat permainan Snake On Meteor. Untuk mengerjakan *game* tersebut, diperlukan sebuah struktur data yaitu struktur data *linked list*. Struktur data *linked list* yang kami buat menggunakan struktur data *linked list* dengan variasi *double pointer*. Struktur ini telah kami modifikasi untuk menyesuaikan dengan permainan Snake On Meteor. Struktur *Linked List* yang kami buat dan gunakan menyimpan data berupa *First* dan juga *last* dari *list* tersebut, yang merupakan sebuah *address*. *Address* sendiri merupakan sebuah tipe bentukan yang menyimpan data berupa elemen *Infoldp*, posisi, dan juga *address* penunjuk *next* dan juga *prev*. Elemen *Infoldp* adalah sebuah elemen yang bertipe string yang berperan sebagai informasi yang nantinya akan ditampilkan pada layar. Posisi adalah sebuah tipe bentukan *location* yang merupakan sebuah *point*. Oleh karena itu, pada posisi ini terdiri dari dua *integer* yaitu *x* dan *y* yang menyatakan *point* atau letak pada koordinat kartesian. Sedangkan *next* dan juga *prev* adalah sebuah *address* yang merupakan *pointer* kepada *address* setelah dan juga *address* sebelum dari *address* saat ini. Struktur data *Linked List* ini tersimpan pada *file source/ADT/listdp/listdp.c* dan pada *source/ADT/listdp/listdp.h* untuk header program.

Pada permainan Snake On Meteor, kami menggunakan struktur *linked list* yang menyimpan *First* dan *Last* karena informasi mengenai dua hal tersebut akan sangat diperlukan, khususnya pada *linkedlist snake*. Untuk algoritma mengenai ular yang akan berbelok maupun memakan sebuah makanan, akan diperlukan informasi yang perlu diacu dengan menggunakan data *First* dari *list* yang sudah disimpan sebelumnya. Begitu pula dengan algoritma mengenai penambahan tubuh pada *snake* yang akan bertambah pada tail dari *snake*, sehingga diperlukan informasi yang diacu dengan menggunakan data *Last* dari *list* yang sudah tersimpan sebelumnya. Kami juga menggunakan tipe bentukan *point* karena pada permainan ini memerlukan pencatatan letak posisi yaitu menggunakan koordinat pada sebuah petak, yang dapat direpresentasikan dengan menggunakan tipe bentukan *point*. Selain itu, kami juga menggunakan *linked list* dengan variasi *double pointer* karena *pointer next* maupun *prev* diperlukan pada berbagai algoritma pengecekan, seperti perubahan posisi pada setiap bagian tubuh yang akan berjalan dengan efisien dengan memanfaatkan *pointer prev* dan juga pemanfaatan *pointer next* pada algoritma yang memerlukan proses traversal dari *First* hingga *Last*.

4 Program Utama

Pada program buatan kami, alur program dimulai dari *main.c*. Program ini kemudian akan menampilkan banner program dan memerintahkan untuk input START atau LOAD. Apabila melakukan input START maka secara otomatis akan menampilkan HELP yang terdiri dari command-command yang dapat digunakan dalam program dan penjelasan dari program yang kami buat. Apabila melakukan input LOAD maka harus menginput file yang akan dijalankan. Hal ini akan melanjutkan permainan pada file yang sudah tersimpan terlebih dahulu.

Pada saat menginput START maka akan menampilkan command apa saja yang bisa di input oleh pengguna. Command yang dapat diinput antara lain :

- SAVE : menyimpan state game pemain saat ini
- CREATE GAME : menambahkan game baru pada daftar game
- LIST GAME : menampilkan daftar game yang disediakan
- DELETE GAME : menghapus sebuah game dari daftar game
- QUEUE GAME : mendaftarkan game ke dalam antrian game pribadi

- PLAY GAME : memainkan sebuah game
- SKIP GAME : melewati game sebanyak yang diinginkan
- SCOREBOARD : melihat nama dan skor untuk semua game
- RESET SCOREBOARD : mereset scoreboard permainan
- HISTORY : menampilkan n permainan yang telah dimainkan
- RESET HISTORY : menghapus semua history permainan
- QUIT : keluar dari program game

Pada saat melakukan QUEUE GAME maka akan menginput game yang ingin dimainkan maka pada saat PLAY GAME akan secara otomatis memainkan game pada urutan pertama. Game akan dimainkan sesuai dengan urutan pada QUEUE GAME yang diinput oleh pemain. Untuk urutan game juga dapat di *skip* untuk memainkan game pada antrian berikutnya. Pemain juga dapat dibuat sendiri oleh pemain dan dapat dihapus juga oleh pemain. Game yang dapat dimainkan adalah game yang tersedia pada program yaitu RNG, DINER DASH, HANGMAN, TOWER OF HANOI, dan SNAKE ON METEOR lalu untuk game yang lain akan langsung dianggap *game over* dan akan menghasilkan score yang acak.

Setelah pengguna selesai menggunakan program tentu pengguna akan melakukan command QUIT untuk keluar dari program. Pada saat menginput command QUIT, maka akan ada dua pilihan yaitu untuk menyimpan file yang sudah diproses saat menggunakan program atau tidak menyimpan file.

5 Algoritma-Algoritma Menarik

5.1 Pengkondisian Makan pada Snake on Meteor

Pada bagian badan snake, kami membuat dengan type string untuk angka pada setiap badan. Oleh karena itu, untuk pengaturan badan selanjutnya akan ditambah dengan satu maka harus diubah ke integer terlebih dahulu, lalu ditambah dengan satu, dan diakhiri dengan pengubahan ke bentuk string kembali sehingga harus menggunakan algoritma seperti pada gambar berikut

Gambar 5.1.1 Potongan program yang menggunakan algoritma menarik

```
if(Searchlistdp((*snake), Posisix>Lastldp(*snake))-1, Posisiy>Lastldp(*snake))) == Nilldp && Searchlistdp(
    InsVlastlistdp(snake, wordToString(intToWord(wordToInt(stringToWord(Infoldp>Lastldp(*snake)))) + 1)),
} else if(Posisiy>Lastldp(*snake)) != 0 && Searchlistdp((*snake), Posisix>Lastldp(*snake), Posisiy>Lastld
    InsVlastlistdp(snake, wordToString(intToWord(wordToInt(stringToWord(Infoldp>Lastldp(*snake)))) + 1)),
} else if(Posisiy>Lastldp(*snake)) == 0 && Searchlistdp((*snake), Posisix>Lastldp(*snake), Posisiy>Lastld
    InsVlastlistdp(snake, wordToString(intToWord(wordToInt(stringToWord(Infoldp>Lastldp(*snake)))) + 1)),
} else if(Posisiy>Lastldp(*snake)) != 4 && Searchlistdp((*snake), Posisix>Lastldp(*snake), Posisiy>Lastld
    InsVlastlistdp(snake, wordToString(intToWord(wordToInt(stringToWord(Infoldp>Lastldp(*snake)))) + 1)),
} else if(Posisiy>Lastldp(*snake)) == 4 && Searchlistdp((*snake), Posisix>Lastldp(*snake), Posisiy>Lastld
    InsVlastlistdp(snake, wordToString(intToWord(wordToInt(stringToWord(Infoldp>Lastldp(*snake)))) + 1)),
} else if(Searchlistdp((*snake), (Posisix>Lastldp(*snake))+1)%5, Posisiy>Lastldp(*snake)) == Nilldp && Se
    InsVlastlistdp(snake, wordToString(intToWord(wordToInt(stringToWord(Infoldp>Lastldp(*snake)))) + 1)),
}
```

6 Data Test

Program BNMO ini terdiri dari beberapa fitur atau *command* yang dapat diberikan oleh pengguna. Oleh karena itu, diperlukan *testing* pada beberapa kondisi untuk memastikan apakah tiap fitur tersebut dapat berjalan dengan tepat dan benar. Berikut adalah *testing* yang kami lakukan terhadap *command* - *command* yang terdapat pada program BNMO ini, beserta dengan penjelasan mengenai hasil atau *output* yang akan diberikan oleh program pada tiap kondisinya.

6.1 Data Test SCOREBOARD

BNMO memiliki suatu *scoreboard*, yakni daftar nama pemain beserta skor yang mereka dapatkan. Setiap kali pemain telah menyelesaikan permainan dari suatu *game*, maka program akan meminta *input* nama dari pemain tersebut, dengan syarat nama yang dimasukkan harus berbeda dengan nama – nama pemain yang sudah ada lebih dahulu di *scoreboard*. Urutan nama pemain pada scoreboard diurutkan mulai dari pemain dengan skor tertinggi hingga skor paling rendah. Bila ada 2 pemain dengan skor yang sama, maka nama pemain yang lebih dahulu memainkan permainan akan ditampilkan lebih dahulu. Bila scoreboard suatu game kosong, maka akan ditampilkan pesan bahwa scoreboard kosong. Pengguna dapat menampilkan *scoreboard* melalui pemanggilan *command* SCOREBOARD.

Gambar 6.1.1 Tampilan ketika pemain berhasil menyimpan nama dan skor ke dalam scoreboard game RNG setelah menyelesaikan permainan

```
Kesempatan Anda habis. X yang tepat adalah 116
Skor Anda: 0

Masukkan username anda: Jess
Masukkan command: SCOREBOARD

**** SCOREBOARD GAME RNG ****
| NAMA      | SCORE |
|-----|
| Jess      | 0      |

**** SCOREBOARD GAME DINER DASH ****
| NAMA      | SCORE |
|-----|
| SCOREBOARD KOSONG

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

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

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

Masukkan command: _
```

Gambar 6.1.2 Tampilan ketika pemain gagal menyimpan nama yang sudah lebih dahulu tertera dalam scoreboard game RNG setelah menyelesaikan permainan

```
Ya, X adalah 50
Skor Anda: 20

Masukkan username anda: Jess
Username sudah terdaftar.

Masukkan username anda: JESS
Username sudah terdaftar.

Masukkan username anda: Jess
Username sudah terdaftar.
```

Gambar 6.1.3 Tampilan ketika pemain dengan skor yang lebih tinggi dari pemain sebelumnya berhasil menyimpan nama dan skor ke dalam scoreboard game RNG setelah menyelesaikan permainan

```
Masukkan username anda: Gary
Masukkan command: SCOREBOARD

**** SCOREBOARD GAME RING ****
| NAMA | SCORE |
|-----|
| Gary | 20 |
| Jess | 0 |

**** SCOREBOARD GAME DINER DASH ****
| NAMA | SCORE |
|-----|
| SCOREBOARD KOSONG |

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

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

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

Masukkan command: █
```

Gambar 6.1.4 Tampilan ketika pemain dengan skor yang sama dengan pemain sebelumnya berhasil menyimpan nama dan skor ke dalam scoreboard game RNG setelah menyelesaikan permainan

```
Kesempatan Anda habis. X yang tepat adalah 199
Skor Anda: 0

Masukkan username anda: Hansel

Masukkan command: SCOREBOARD

**** SCOREBOARD GAME RING ****
| NAMA      | SCORE |
|-----|
| Gary      | 20    |
| Jess      | 0     |
| Hansel    | 0     |
|-----|

**** SCOREBOARD GAME DINER DASH ****
| NAMA      | SCORE |
|-----|
|----- SCOREBOARD KOSONG -----|

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

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

**** SCOREBOARD GAME SNAKE ON METEOR ****
| NAMA      | SCORE |
|-----|
|----- SCOREBOARD KOSONG -----|
```

Gambar 6.1.5 Tampilan ketika pemain dengan nama yang sama berhasil menyimpan nama dan skor ke dalam 2 scoreboard game yang berbeda setelah menyelesaikan permainan

```
Selamat, Anda berhasil menyelesaikan Tower of Hanoi!
Skor Anda: 8

Masukkan username anda: Jess

Masukkan command: SCOREBOARD

**** SCOREBOARD GAME RING ****
| NAMA      | SCORE |
|-----|
| Gary      | 20    |
| Jess      | 0     |
| Hansel    | 0     |
|-----|

**** SCOREBOARD GAME DINER DASH ****
| NAMA      | SCORE |
|-----|
|----- SCOREBOARD KOSONG -----|

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

**** SCOREBOARD GAME TOWER OF HANOI ****
| NAMA      | SCORE |
|-----|
| Jess      | 8     |
|-----|

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

Masukkan command: _
```

6.2 Data Test RESET SCOREBOARD

Pengguna dapat menghapus seluruh informasi dalam salah satu *scoreboard game* maupun semua *scoreboard game* sekaligus melalui pemanggilan *command* RESET SCOREBOARD. Ketika *scoreboard* ditampilkan setelah dilakukan proses *reset*, maka *scoreboard game* yang telah dihapus datanya akan menampilkan pesan bahwa *scoreboard* kosong.

Gambar 6.2.1 Tampilan scoreboard awal

```
Masukkan command: SCOREBOARD

**** SCOREBOARD GAME RING ****
| NAMA      | SCORE |
|-----|-----|
| Gary      | 20    |
| Jess      | 0      |
| Hansel    | 0      |

**** SCOREBOARD GAME DINER DASH ****
| NAMA      | SCORE |
|-----|-----|
| James     | 54666 |
| John      | 42968 |
| Bobi      | 16334 |

**** SCOREBOARD GAME HANGMAN ****
| NAMA      | SCORE |
|-----|-----|
| Willy     | 14     |
| Kyla      | 8      |
| Roy       | 0      |

**** SCOREBOARD GAME TOWER OF HANOI ****
| NAMA      | SCORE |
|-----|-----|
| Vivi     | 8      |
| Wendy     | 5      |
| Gilbert   | 5      |

**** SCOREBOARD GAME SNAKE ON METEOR ****
| NAMA      | SCORE |
|-----|-----|
| Jen       | 10     |
| Steven    | 6      |
| Henry     | 4      |

Masukkan command: _
```

Gambar 6.2.2 Tampilan scoreboard ketika berhasil melakukan RESET SCOREBOARD pada scoreboard game RNG

```
Masukkan command: RESET SCOREBOARD
Daftar scoreboard:
0. ALL
1. RNG
2. DINER DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR

Scoreboard yang ingin dihapus: 1
Apakah anda yakin ingin melakukan reset scoreboard? (Y/N): Y

Scoreboard berhasil di-reset.

Masukkan command: SCOREBOARD

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

**** SCOREBOARD GAME DINER DASH ****
| NAMA | SCORE |
----- SCOREBOARD KOSONG -----
| James | 54666 |
| John | 42968 |
| Bobi | 16334 |
----- SCOREBOARD KOSONG -----

**** SCOREBOARD GAME HANGMAN ****
| NAMA | SCORE |
----- SCOREBOARD KOSONG -----
| Willy | 14 |
| Kyla | 8 |
| Roy | 0 |
----- SCOREBOARD KOSONG -----

**** SCOREBOARD GAME TOWER OF HANOI ****
| NAMA | SCORE |
----- SCOREBOARD KOSONG -----
| Vivi | 8 |
| Wendy | 5 |
| Gilbert | 5 |
----- SCOREBOARD KOSONG -----

**** SCOREBOARD GAME SNAKE ON METEOR ****
| NAMA | SCORE |
----- SCOREBOARD KOSONG -----
| Jen | 10 |
| Steven | 6 |
| Henry | 4 |
----- SCOREBOARD KOSONG -----
```

Gambar 6.2.3 Tampilan scoreboard ketika berhasil melakukan RESET SCOREBOARD pada seluruh scoreboard game

```
Masukkan command: RESET SCOREBOARD
Daftar scoreboard:
0. ALL
1. RNG
2. DINER DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR

Scoreboard yang ingin dihapus: 0
Apakah anda yakin ingin melakukan reset scoreboard? (Y/N): Y

Scoreboard berhasil di-reset.

Masukkan command: SCOREBOARD

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

**** SCOREBOARD GAME DINER DASH ****
| NAMA | SCORE |
----- SCOREBOARD KOSONG -----

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

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

**** SCOREBOARD GAME SNAKE ON METEOR ****
| NAMA | SCORE |
----- SCOREBOARD KOSONG -----
```

6.3 Data Test HISTORY

Pengguna dapat melihat data riwayat game apa saja yang telah dimainkan dengan urutan game yang muncul lebih dahulu adalah game yang terakhir kali dimainkan. Hal ini dapat dilakukan melalui pemanggilan command HISTORY <N>, dengan N sebagai jumlah riwayat game yang ingin ditampilkan. Apabila jumlah game yang ingin ditampilkan lebih banyak daripada jumlah data riwayat game yang tersimpan, maka program akan menampilkan seluruh data riwayat game.

Gambar 6.3.1 Tampilan riwayat game ketika BNMO belum pernah dimainkan

```
Masukkan command: HISTORY 1
Berikut adalah daftar Game yang telah dimainkan:
Tidak ada permainan yang pernah dimainkan
Masukkan command:
```

Gambar 6.3.2 Tampilan ketika pengguna ingin menampilkan riwayat 2 game yang terakhir dimainkan

```
Masukkan command: HISTORY 2
Berikut adalah daftar Game yang telah dimainkan:
1. SNAKE ON METEOR
2. SNAKE ON METEOR
```

Gambar 6.3.2 Tampilan ketika pengguna ingin menampilkan riwayat game dengan jumlah riwayat yang ingin ditampilkan lebih besar daripada jumlah total riwayat yang tersimpan

```
Masukkan command: HISTORY 20
Berikut adalah daftar Game yang telah dimainkan:
1. SNAKE ON METEOR
2. SNAKE ON METEOR
3. SNAKE ON METEOR
4. TOWER OF HANOI
5. TOWER OF HANOI
6. TOWER OF HANOI
7. HANGMAN
8. HANGMAN
9. HANGMAN
10. DINER DASH
11. DINER DASH
12. DINER DASH
13. RING
14. RING
15. RING
16. RING
17. RING
```

6.4 Data Test RESET HISTORY

Pengguna dapat melakukan penghapusan terhadap seluruh data riwayat game yang telah dimainkan melalui command RESET HISTORY. Apabila pengguna memanggil command HISTORY setelah dilakukan proses reset, maka daftar game yang telah dimainkan akan menjadi kosong.

Gambar 6.4.1 Tampilan seluruh data riwayat game yang telah dimainkan melalui pemanggilan command *HISTORY <N>*

```
Masukkan command: HISTORY 20
Berikut adalah daftar Game yang telah dimainkan:
1. SNAKE ON METEOR
2. SNAKE ON METEOR
3. SNAKE ON METEOR
4. TOWER OF HANOI
5. TOWER OF HANOI
6. TOWER OF HANOI
7. HANGMAN
8. HANGMAN
9. HANGMAN
10. DINER DASH
11. DINER DASH
12. DINER DASH
13. RNG
14. RNG
15. RNG
16. RNG
17. RNG
```

Gambar 6.4.2 Tampilan setelah dilakukan penghapusan data riwayat game yang telah dimainkan

```
Masukkan command: RESET HISTORY
Apakah anda yakin ingin menghapus semua history? (Y/N): Y

History berhasil di-reset.

Masukkan command: HISTORY 1
Berikut adalah daftar Game yang telah dimainkan:
Tidak ada permainan yang pernah dimainkan
```

6.5 Data Test Hangman

Pengguna program dapat memainkan game Hangman yang terdapat dalam program BNMO. Hangman merupakan sebuah permainan tebak-tebakan kata dimana list kata telah ditentukan dan dapat dibuat oleh user untuk ditambahkan ke dalam list kata. Awal permainan, pengguna diberikan 10 kesempatan untuk menebak satu per satu huruf yang terdapat dalam kata yang terpilih. Apabila tebak benar, maka huruf yang terbek akan ditampilkan pada layar. Apabila salah, maka pengguna kehilangan satu kesempatan. Pengguna juga tidak boleh menebak huruf yang sudah ditebak sebelumnya pada kata yang sama. Jika pengguna berhasil menebak suatu kata maka poin akan diberikan berdasarkan panjang kata yang berhasil ditebak. Permainan berakhir jika pengguna kehabisan kesempatan untuk menebak.

Gambar 6.5.1 Tampilan awal Hangman ketika dimainkan dengan command *PLAY GAME*

```
Masukkan command: PLAY GAME
Berikut adalah daftar antrian game-mu
1. HANGMAN

Loading HANGMAN . . .

Selamat datang di game Hangman!

Silakan pilih mode berikut:
1. Menambah dictionary
2. Bermain

Masukkan pilihan: _
```

Gambar 6.5.2 Tampilan ketika menginput yang tidak valid dalam pilihan mode Hangman

```
Silakan pilih mode berikut:
1. Menambah dictionary
2. Bermain

Masukkan pilihan: 3
Pilihan tidak valid.
```

Gambar 6.5.3 Tampilan ketika menambahkan kata ke dalam dictionary tetapi sudah ada kata tersebut di dictionary program Hangman

```
Selamat datang di game Hangman!

Silakan pilih mode berikut:
1. Menambah dictionary
2. Bermain

Masukkan pilihan: 1

Masukkan kata baru yang ingin ditambahkan: HUJAN
Kata sudah ada di dalam daftar kata.
```

Gambar 6.5.4 Tampilan ketika berhasil menambahkan kata ke dalam dictionary program Hangman

```
Silakan pilih mode berikut:
1. Menambah dictionary
2. Bermain

Masukkan pilihan: 1

Masukkan kata baru yang ingin ditambahkan: CERAH
Kata berhasil ditambahkan.
```

Gambar 6.5.5 Tampilan ketika memilih mode bermain dalam Hangman

```
Silakan pilih mode berikut:
1. Menambah dictionary
2. Bermain

Masukkan pilihan: 2

Tebakan sebelumnya : -
Kata: _ _ _ _ _
Kesempatan: 10
Masukkan tebak:
```

Gambar 6.5.6 Tampilan ketika menginput huruf yang benar dalam suatu kata

```
Tebakan sebelumnya : -
Kata: _ _ _ _ _
Kesempatan: 10
Masukkan tebak: B

Tebakan sebelumnya : b
Kata: B _ _ _ _
Kesempatan: 10
Masukkan tebak:
```

Gambar 6.5.7 Tampilan ketika menginput huruf yang salah dalam suatu kata

```
Tebakan sebelumnya : b
Kata: B _ _ _ _ _
Kesempatan: 10
Masukkan tebakan: Z

Tebakan sebelumnya : bz
Kata: B _ _ _ _ _
Kesempatan: 9
Masukkan tebakan:
```

Gambar 6.5.8 Tampilan ketika menginput huruf yang telah diinput sebelumnya pada kata yang sama

```
Tebakan sebelumnya : bz
Kata: B _ _ _ _ _
Kesempatan: 9
Masukkan tebakan: B
Tebakan sudah pernah dilakukan!
```

Gambar 6.5.9 Tampilan ketika berhasil menebak semua huruf dalam suatu kata

```
Tebakan sebelumnya : bzinO
Kata: B I N O _ O
Kesempatan: 9
Masukkan tebakan: m

Berhasil menebak kata BINOMO! Kamu mendapatkan 6 poin!
```

Gambar 6.5.10 Tampilan ketika telah berhasil menebak semua huruf dalam suatu kata

```
Berhasil menebak kata BINOMO! Kamu mendapatkan 6 poin!

Tebakan sebelumnya : -
Kata: _ _ _ _ _
Kesempatan: 9
Masukkan tebakan:
```

Gambar 6.5.11 Tampilan ketika telah habis kesempatan untuk menebak semua huruf dalam suatu kata

```
Tebakan sebelumnya : sdfghjklp
Kata: H _ _ _ _
Kesempatan: 1
Masukkan tebakan: m

Kamu gagal menebak kata HALO!

Kesempatan habis! Total poin yang kamu kumpulkan adalah 6 ~(^.^)~ !
```

6.6 Data Test Tower of Hanoi

Pengguna program dapat memainkan game Tower of Hanoi yang terdapat dalam program BNMO. Tower of Hanoi merupakan permainan yang terdiri atas piringan dan tiga buah tiang (A,B,C) dimana piringan pada awal permainan terletak pada tiang A dan disusun dengan urutan piringan dari ukuran terbesar di posisi paling bawah hingga ukuran terkecil di posisi paling atas. Tugas pengguna adalah memindahkan piringan tersebut sesuai urutannya ke dalam tiang C. Pengguna juga dapat menggunakan perantara tiang B guna menaruh cakram lainnya untuk sementara. Permainan berakhir ketika seluruh cakram telah ditaruh dengan urutan yang sesuai di tiang C. Skor pemain dihitung dengan rumus.

Gambar 6.6.1 Tampilan awal Tower of Hanoi ketika dimainkan dengan command PLAY GAME

```
Masukkan command: PLAY GAME
Berikut adalah daftar antrian game-mu
1. TOWER OF HANOI

Loading TOWER OF HANOI . . .

=====
- TOWER OF HANOI -
=====

Masukkan jumlah cakram: _
```

Gambar 6.6.2 Tampilan ketika berhasil memasukkan jumlah cakram serta menampilkan visualisasi dari posisi tiang dan cakram

```
Masukkan jumlah cakram: 3

-----
Tower A: [3, 2, 1]
Tower B: []
Tower C: []

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

Jumlah langkah: 0

Memindahkan cakram dari tower (A - C):
```

Gambar 6.6.3 Tampilan ketika berhasil memindahkan cakram dari satu tiang ke tiang lainnya

```
Memindahkan cakram dari tower (A - C): A
Ke tower (A - C): B

Memindahkan cakram ke Tower B

-----
Tower A: [3, 2]
Tower B: [1]
Tower C: []

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

Jumlah langkah: 1
```

Gambar 6.6.4 Tampilan ketika gagal memindahkan cakram yang ukurannya lebih besar daripada ukuran cakram pada tiang tujuan

```
Memindahkan cakram dari tower (A - C): A
Ke tower (A - C): B

Cakram tidak dapat dipindahkan ke Tower B karena berukuran lebih besar.

Tower A: [3, 2]
Tower B: [1]
Tower C: []

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

Jumlah langkah: 1
```

Gambar 6.6.5 Tampilan ketika memindahkan cakram dengan tower asal dan tujuan yang sama

```
Memindahkan cakram dari tower (A - C): B
Ke tower (A - C): B

Tower asal dan tujuan tidak boleh sama.

Tower A: [3, 2]
Tower B: [1]
Tower C: []

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

Jumlah langkah: 1
```

Gambar 6.6.6 Tampilan ketika menginput tower yang tidak ada pada game (tidak valid)

```
Memindahkan cakram dari tower (A - C): Y
Ke tower (A - C): Y

Tower tidak valid

Tower A: [3, 2]
Tower B: [1]
Tower C: []

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

Jumlah langkah: 1
```

Gambar 6.6.7 Tampilan ketika berhasil memenangkan game Tower of Hanoi

```
Memindahkan cakram dari tower (A - C): A
Ke tower (A - C): C

Memindahkan cakram ke Tower C
-----

Tower 1: []
Tower 2: []
Tower 3: [3, 2, 1]

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

Jumlah langkah: 7

Selamat, Anda berhasil menyelesaikan Tower of Hanoi!
Skor Anda: 8
```


Gambar 6.7.4 Tampilan ketika snake berhasil makan makanan

```
TURN 5
Silahkan masukkan command anda: s
Berhasil bergerak!
Berikut merupakan peta permainan:
.....
| | ### | | |
| | | | 2 |
| | | | 1 |
| | | H | H |
| | | | O |
| | | | |
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 6
Silahkan masukkan command anda: s
Berhasil bergerak!
Berikut merupakan peta permainan:
.....
| H ### | | | |
| O | | | |
| | | 2 | 2 |
| | | | 1 |
| | | | H |
| | | | |
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 7
Silahkan masukkan command anda: a
```

Gambar 6.7.5 Tampilan ketika snake bergerak ke tempat meteor (masih panas)

```
TURN 8
Silahkan masukkan command anda: a
Berhasil bergerak!
Berikut merupakan peta permainan:
.....
| | ### | | |
| O | | | |
| | | | |
| | | | 3 |
| H | H | 1 | 2 |
| | | | |
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 9
Silahkan masukkan command anda: a
Meteor masih panas! Anda belum dapat kembali ke titik tersebut.
Silahkan masukkan command lainnya

TURN 9
Silahkan masukkan command anda:
```

Gambar 6.7.6 Tampilan ketika snake terkena meteor

```
TURN 9
Silahkan masukkan command anda: w
Berhasil bergerak!
Berikut merupakan peta permainan:
.....
| | ### | | |
| O | | | |
| | | | |
| | H | | |
| | 1 | 2 | H |
| | | | |
Anda terkena meteor
Berikut merupakan peta permainan sekarang:
.....
| | ### | | |
| O | | | |
| | | | |
| | H | | |
| | 1 | 2 | H |
| | | | |
Silahkan lanjutkan permainan

TURN 10
Silahkan masukkan command anda: a
```

Gambar 6.7.7 Tampilan ketika berhasil game selesai saat kepala snake terkena meteor dan dapat skor

```
Berhasil bergerak!
Berikut merupakan peta permainan:
.....
| | ### | | |
| O | | | |
| H | | | |
| 1 | 2 | | |
| | | | H |
| | | | |
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 12
Silahkan masukkan command anda: w
Berhasil bergerak!
Berikut merupakan peta permainan:
.....
| H ### | | | |
| H | | | |
| 1 | | | |
| 3 | 2 | | O |
| | | | |
Kepala snake terkena meteor :c
Game berakhir!
Skor: 6
Masukkan username anda:
```

Gambar 6.7.8 Tampilan ketika snake terkena obstacle

```

TURN 7
Silahkan masukkan command anda: a
Berhasil bergerak!
Berikut merupakan peta permainan:
-----
| | | | |
| | O | | |
| | | | |
| | ### | |
| | | H | 1 |
| | | | M |
| | | | |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 8
Silahkan masukkan command anda: w
Anda menabrak obstacle :(
Game berakhir!
Skor: 6
Masukkan username anda:

```

Gambar 6.7.9 Tampilan ketika kepala snake tidak bisa bergerak kemanapun

```

TURN 100
Silahkan masukkan command anda: w
Berhasil bergerak!
Berikut merupakan peta permainan:
-----
| O | | | | |
| | ### | | |
| | | M | 5 | 4 |
| | | 6 | H | 3 |
| | | | 1 | 2 |
| | | | |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 101
Silahkan masukkan command anda: a
Kamu sudah tidak dapat bergerak ke arah manapun:(
Game berakhir!
Skor: 14
D:\0. kuliah\0. matkul 3\alstrukdat\tubesril\Tugas-Besar-Algoritma-dan-Struktur-Data-STI>_

```

Gambar 6.7.10 Tampilan ketika snake dapat menembus

```

TURN 2
Silahkan masukkan command anda: a
Berhasil bergerak!
Berikut merupakan peta permainan:
-----
| H | 1 | | |
| 2 | 2 | | |
| | | | H |
| | | ### |
| | | | O |
| | | | |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 3
Silahkan masukkan command anda: a
Berhasil bergerak!
Berikut merupakan peta permainan:
-----
| 1 | 2 | | H |
| | 3 | | |
| | | H | |
| | | ### |
| | | | O |
| | | | |
-----
Anda beruntung tidak terkena meteor! Silahkan lanjutkan permainan

TURN 4
Silahkan masukkan command anda:

```

6.8 Data Test Load

Pada awal menjalankan program, selain memulai BNMO dengan menggunakan *command* START, pengguna juga memiliki opsi untuk memulai BNMO dengan *command* LOAD. *Command* tersebut harus diikuti dengan nama file yang ingin dibuka. Ketika pengguna memilih LOAD, *game* yang dapat dimainkan pada saat itu adalah *game* yang berasal dari suatu *state* yang sudah pernah disimpan oleh pengguna, bukan hanya *game* yang berasal dari *default* program BNMO. Data lainnya seperti scoreboard berisi daftar nama pemain beserta skor yang

dimiliki dan riwayat permainan yang telah dimainkan juga akan dimuat oleh program sesuai dengan *state* terakhir program ketika menjalankan *command* SAVE yang terakhir.

Ketika *command* LOAD diberikan pengguna, diikuti nama *file* yang *valid*, maka program akan menampilkan *help menu* untuk memberikan bantuan kepada pengguna berupa *list* daftar *command* sebagai bantuan daftar fitur apa saja yang dapat dijalankan pada program tersebut. Apabila pengguna memberikan suatu nama file yang tidak *valid* yaitu tidak ada *file* dengan nama tersebut, maka program akan menampilkan pesan kesalahan dan meminta pengguna untuk kembali memasukkan *command* beserta nama *file* yang *valid*. Kasus ini juga berlaku jika pengguna memasukkan nama *file* namun dengan format selain '.txt' atau '.TXT'. Ketika pengguna hanya memberikan *command* LOAD tanpa memberikan nama *file*, maka program akan menganggapnya sebagai *command* yang *invalid* dan meminta pengguna untuk kembali memberikan *command* yang *valid*.

Gambar 6.8.1 Tampilan ketika pengguna memasukkan nama file yang invalid

```
*****
c h o o s e << START >> o r << LOAD [file.txt] >>
Masukkan command: LOAD asalnyobahehe.txt
LOADNG FILE . . . . . [FAILED XX]
Load file gagal. File tidak ditemukan.
Masukkan command: _
```

Gambar 6.8.2 Tampilan ketika pengguna memasukkan command load tanpa diikuti nama file

```
*****
c h o o s e << START >> o r << LOAD [file.txt] >>
Masukkan command: LOAD
Command tidak dikenali, silahkan masukkan command yang valid.
Masukkan command: _
```

Gambar 6.8.3 Tampilan ketika pengguna memasukkan format file yang invalid

```
*****
c h o o s e << START >> o r << LOAD [file.txt] >>
Masukkan command: LOAD aiueo.docx
Format file tidak valid.
```

Gambar 6.8.4 Isi dari file save3.txt

```

c console.c  save3.txt x
data > save3.txt
1 5
2 RING
3 DINER DASH
4 HANGMAN
5 TOWER OF HANOI
6 SNAKE ON METEOR
7 1
8 HANGMAN
9 0
10 0
11 1
12 Rara 4
13 0
14 0
15

```

Gambar 6.8.5 Tampilan ketika berhasil menjalankan program melalui LOAD sebuah file save3.txt beserta mencoba mengakses data yang tersimpan didalamnya

```

c h o o s e << START >> o r << LOAD [file.txt] >>

Masukkan command: LOAD save3.txt
LOADNG FILE . . . . . [SUCCEEDED //]

Save file berhasil dibaca. BNMO berhasil dijalankan.

=====
+:*+:*+:*+:*+:*+:*+. H E L P M E N U .+:*+:*+:*+:*+:*+:*+.
=====
      daftar dan cara kerja command:
1. LIST GAME      : menampilkan daftar game yang disediakan
2. QUEUE GAME     : mendaftarkan game ke dalam antrian game pribadi
3. PLAY GAME      : memainkan sebuah game
4. SKIP GAME <n>  : melewati game sebanyak yang diinginkan
5. CREATE GAME     : menambahkan game baru pada daftar game
6. DELETE GAME     : menghapus sebuah game dari daftar game
7. SCOREBOARD     : melihat nama dan skor untuk semua game
8. RESET SCOREBOARD : mereset scoreboard permainan
9. HISTORY <n>    : menampilkan n permainan yang telah dimainkan
10. RESET HISTORY  : menghapus semua history permainan
11. QUIT           : keluar dari program game ^^
12. SAVE <filename.txt> : menyimpan state game pemain saat ini
=====

Masukkan command: HISTORY 5
Berikut adalah daftar Game yang telah dimainkan:
1. HANGMAN

Masukkan command:

```

6.9 Data Test Save

Pengguna program dapat menggunakan command SAVE untuk menyimpan file sesuai dengan state terakhir program, agar pengguna dapat melanjutkan kembali *progress*-nya ketika ingin menggunakan program kembali. SAVE dapat dilakukan dengan dua cara yaitu langsung melakukan *input* SAVE <filename> pada command atau SAVE saat melakukan input QUIT pada command program. Data yang disimpan adalah data jumlah permainan yang dimiliki beserta nama - nama *game* tersebut, jumlah riwayat permainan yang dimainkan beserta nama - nama *game* tersebut, serta jumlah data yang ada pada *scoreboard* masing - masing *game* beserta nama pemain dan skor yang didapatkan.

Gambar 6.9.1 Tampilan ketika memulai permainan dengan command START dan memuat file konfigurasi default

```
Masukkan command: START
File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan.

=====
+:+.+:*+.:*+.:*+.:*+.  H E L P  M E N U .+:*+.:*+.:*+.:*+.:*+.

                        daftar dan cara kerja command:
1. LIST GAME           : menampilkan daftar game yang disediakan
2. QUEUE GAME          : mendaftarkan game ke dalam antrian game pribadi
3. PLAY GAME           : memainkan sebuah game
4. SKIP GAME <n>       : melewati game sebanyak yang diinginkan
5. CREATE GAME          : menambahkan game baru pada daftar game
6. DELETE GAME         : menghapus sebuah game dari daftar game
7. SCOREBOARD          : melihat nama dan skor untuk semua game
8. RESET SCOREBOARD    : mereset scoreboard permainan
9. HISTORY <n>         : menampilkan n permainan yang telah dimainkan
10. RESET HISTORY      : menghapus semua history permainan
11. QUIT               : keluar dari program game ^^
12. SAVE <filename.txt> : menyimpan state game pemain saat ini

=====

Masukkan command:
```

Gambar 6.9.2 Tampilan file konfigurasi default

```
data > ≡ config.txt
1      5
2      RNG
3      DINER DASH
4      HANGMAN
5      TOWER OF HANOI
6      SNAKE ON METEOR
```

Gambar 6.9.3 Tampilan daftar game yang dimiliki setelah digunakan oleh pengguna yang diakses melalui command LIST GAME

```
Masukkan command: LIST GAME
Berikut adalah daftar game yang tersedia
1. RNG
2. DINER DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. JIGSAW

Masukkan command: _
```

Gambar 6.9.4 Tampilan scoreboard setelah digunakan pengguna yang diakses melalui command SCOREBOARD

```
Masukkan command: SCOREBOARD

**** SCOREBOARD GAME RNG ****
| NAMA      | SCORE |
|-----|
| Gigi      | 60    |
| Vivi     | 40    |

**** SCOREBOARD GAME DINER DASH ****
| NAMA      | SCORE |
|-----|
| SCOREBOARD KOSONG ----

**** SCOREBOARD GAME HANGMAN ****
| NAMA      | SCORE |
|-----|
| Nini      | 8     |

**** SCOREBOARD GAME TOWER OF HANOI ****
| NAMA      | SCORE |
|-----|
| Dio       | 8     |

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

**** SCOREBOARD GAME JIGSAW ****
| NAMA      | SCORE |
|-----|
| Ria       | 85    |

Masukkan command:
```

Gambar 6.9.5 Tampilan riwayat permainan setelah digunakan pengguna yang diakses melalui command HISTORY

```
Masukkan command: HISTORY 10
Berikut adalah daftar Game yang telah dimainkan:
1. JIGSAW
2. TOWER OF HANOI
3. HANGMAN
4. RNG
5. RNG

Masukkan command:
```

Gambar 6.9.6 Tampilan ketika berhasil melakukan SAVE

```
Masukkan command: SAVE testsave.txt
Save file berhasil disimpan
```

Gambar 6.9.7 Tampilan file konfigurasi akhir

```
data > E testsave.txt
1 6
2 RNG
3 DINER DASH
4 HANGMAN
5 TOWER OF HANOI
6 SNAKE ON METEOR
7 JIGSAW
8 5
9 JIGSAW
10 TOWER OF HANOI
11 HANGMAN
12 RNG
13 RNG
14 2
15 Gigi 60
16 Vivi 40
17 0
18 1
19 Nini 8
20 1
21 Dio 8
22 0
23 1
24 Ria 85
25
```

7 Test Script

N o.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Scoreboard	Menampilkan scoreboard yang sudah tersimpan di dalam file txt yang digunakan	Memberikan command SCOREBOARD pada command setelah START atau LOAD	SCOREBOARD	Gambar 6.1.1 Gambar 6.1.2 Gambar 6.1.3 Gambar 6.1.4 Gambar 6.1.5	Sesuai dengan hasil yang Diharapkan
2	Reset Scoreboard	Melakukan reset terhadap scoreboard permainan	Memberikan command RESET SCOREBOARD pada command setelah START atau LOAD	RESET SCOREBOARD	Gambar 6.2.1 Gambar 6.2.2 Gambar 6.2.3	Sesuai dengan hasil yang Diharapkan
3	History	Melihat permainan apa saja yang telah dimainkan	Memberikan command HISTORY <n> pada command setelah START atau LOAD	HISTORY <n>	Gambar 6.3.1 Gambar 6.3.2 Gambar 6.3.3	Sesuai dengan hasil yang Diharapkan

4	Reset History	Menghapus semua history permainan yang dimainkan	Memberikan command RESET HISTORY pada command setelah START atau LOAD	RESET HISTORY	Gambar 6.4.1 Gambar 6.4.2	Sesuai dengan hasil yang Diharapkan
5	Hangman	Menjalankan game Hangman pada program, pada akhir game akan diberikan skor hasil permainan	Mengetik PLAY GAME setelah memasukkan game HANGMAN ke dalam QUEUE GAME dan game HANGMAN berada di urutan pertama QUEUE GAME	QUEUE GAME lalu memilih 3 dan PLAY GAME	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 Gambar 6.5.10 Gambar 6.5.11	Sesuai dengan hasil yang Diharapkan
6	Tower of Hanoi	Menjalankan game Tower of Hanoi pada program, pada akhir game akan diberikan skor hasil permainan	Mengetik PLAY GAME setelah memasukkan game TOWER OF HANOI ke dalam QUEUE GAME dan game HANGMAN berada di urutan pertama QUEUE GAME	QUEUE GAME lalu memilih 4 dan PLAY GAME	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	Sesuai dengan hasil yang Diharapkan
7	Snake on Meteor	Menjalankan game Tower of Hanoi pada program, pada akhir game akan diberikan skor hasil permainan	Mengetik PLAY GAME setelah memasukkan game SNAKE ON METEOR ke dalam QUEUE GAME dan game HANGMAN berada di urutan pertama QUEUE GAME	QUEUE GAME lalu memilih 5 dan PLAY GAME	Gambar 6.7.1 Gambar 6.7.2 Gambar 6.7.3 Gambar 6.7.4 Gambar 6.7.5 Gambar 6.7.6 Gambar 6.7.7 Gambar 6.7.8 Gambar 6.7.9 Gambar 6.7.10	Sesuai dengan hasil yang Diharapkan
8	Load	Memeriksa apakah <i>file</i> berhasil dibaca dan menampilkan tampilan ketika <i>file</i> berhasil dibuka	memberikan <i>command</i> LOAD diikuti dengan nama <i>file</i> yang <i>valid</i>	LOAD	Gambar 6.8.1 Gambar 6.8.2 Gambar 6.8.3 Gambar 6.8.4 Gambar 6.8.5	Sesuai dengan hasil yang Diharapkan
9	Save	Melakukan save file	Mengetik SAVE<filename>	SAVE <filename>	Gambar 6.9.1 Gambar 6.9.2	Sesuai dengan hasil

			pada command setelah START atau LOAD		Gambar 6.9.3 Gambar 6.9.4 Gambar 6.9.5 Gambar 6.9.6 Gambar 6.9.7	yang Diharapkan
--	--	--	--------------------------------------	--	--	-----------------

8 Pembagian Kerja dalam Kelompok

Tabel 8.1.1 Pembagian kerja dalam Kelompok

No.	Nama Anggota - NIM	Deskripsi Kontribusi
1.	Michael Sihotang – 18221054	ADT Listdp, ADT Stack, Game Snake on Meteor, Driver ADT, Laporan
2.	Muhammad Dastin Fauzi – 18221062	Game Snake on Meteor, Laporan
3.	Imanuel Raditya – 18221112	ADT ArrayMap, Main program, command tambahan, Game Hangman beserta Bonus, Game Tower of Hanoi beserta Bonus
4.	Miralistya Cahya Fatimah – 18221116	ADT Listdp, Game Snake on Meteor, Bonus Snake on Meteor, Laporan
5.	Seren Elizabeth Siahaan – 18221160	Game Hangman beserta Bonus, Laporan

9 Lampiran

9.1 Deskripsi Tugas Besar 2

Latar Belakang

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.

Spesifikasi Umum

Buatlah sebuah permainan berbasis CLI (command-line interface). Sistem ini dibuat dalam bahasa C dengan menggunakan struktur data yang sudah kalian pelajari di mata kuliah ini.

Kalian boleh menggunakan (atau memodifikasi) struktur data yang sudah kalian buat untuk praktikum pada tugas besar ini. Library yang boleh digunakan hanya `stdio.h`, `stdlib.h`, `time.h` dan `math.h`.

System Mechanics

1. About the System

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

2. Main Menu

Ketika program pertama kali dijalankan, BNMO akan memperlihatkan main menu yang berisi welcome page dan beberapa menu pilihan yaitu START dan LOAD. Setelah itu, main menu akan menerima input commands yang akan dijelaskan pada bagian berikutnya.

3. Command

Pada setiap giliran, pemain dapat memasukkan command-command berikut:

- START : Salah satu command yang dimasukkan pertama kali oleh pemain ke BNMO. Setelah menekan Enter, dibaca file konfigurasi default yang berisi list game yang dapat dimainkan.
- LOAD <filename> : Salah satu command yang dimasukkan pertama kali oleh pemain ke BNMO. Memiliki satu argumen yaitu filename yang merepresentasikan suatu *save file* yang ingin dibuka. Setelah menekan Enter, akan dibaca save file <filename> yang berisi list game yang dapat dimainkan, histori dan scoreboard game.
- SAVE <filename> : command yang digunakan untuk menyimpan state game pemain saat ini ke dalam suatu file. Command SAVE memiliki satu argumen yang merepresentasikan nama file yang akan disimpan pada disk.
- CREATE GAME : command yang digunakan untuk menambahkan game baru pada daftar game.
- LIST GAME : command yang digunakan untuk menampilkan daftar game yang disediakan oleh sistem.
- DELETE GAME : command yang digunakan untuk menghapus sebuah game dari daftar game.

- QUEUE GAME : command yang digunakan untuk mendaftarkan permainan kedalam list.
- PLAY GAME : command yang digunakan untuk memainkan sebuah permainan.
- SKIP GAME <n> : command yang digunakan untuk melewati permainan sebanyak n.
- SCOREBOARD : command yang digunakan untuk menampilkan scoreboard yang terdiri dari nama dan skor permainan.
- RESET SCOREBOARD : command yang digunakan untuk mereset scoreboard permainan yang ada.
- HISTORY <n> : command yang digunakan untuk menampilkan permainan yang telah dimainkan.
- RESET HISTORY : command yang digunakan untuk mereset scoreboard permainan yang ada.
- QUIT : Keluar dari program.
- HELP : Bantuan command-command yang disebutkan di atas. Tampilan dan kata-kata dibebaskan.
- COMMAND LAIN : Command-command lain selain yang disebutkan diatas tidak valid.

4. Spesifikasi Game

1. RNG

BNMO tidak selalu menikmati *game* yang sudah pasti *outcome*-nya. Karena itu, ia suka dengan *game* yang melibatkan RNG (*Random number generator*).

2. Diner Dash

Indra dan Doni juga suka permainan yang menegangkan. Oleh karena itu, ia ingin ada sebuah game Diner Dash dalam BNMO. Secara singkat, Diner Dash merupakan permainan mengantar makanan namun terurut berdasarkan prioritasnya.

3. Tower of Hanoi

BNMO melihat Finn dan Jake sedang bermain Tower of Hanoi secara langsung, dengan adanya 3 tiang, dapat disebutkan sebagai tiang A, tiang B, dan tiang C dan posisinya terurut dari kiri ke kanan. Secara singkat, tower of hanoi merupakan game memindahkan posisi kepingan pada tiang yang tersedia dengan menggunakan prinsip stack.

4. Hangman

BNMO suka dengan *game* yang melibatkan tebak-tebakan kata sehingga ia membuat *game* yang terinspirasi dari *game* hangman. Secara singkat, hangman merupakan *game* untuk menebak kata yang ada pada program. Jumlah huruf akan diberitahu lalu pemain menebak huruf per huruf.

5. Snake on Meteor

BNMO memiliki sebuah *game* andalannya yang menjadi daya tarik bagi para pemainnya, yaitu *snake on meteor*. Singkatnya, *game* ini mirip dengan *game* *snake* yang ada pada berbagai konsol lama, tetapi dipersulit dengan adanya kehadiran meteor yang dapat mengenai *snake* tersebut. Dampak yang didapat oleh pemain apabila *snake* terkena serangan meteor tersebut adalah panjang *snake* akan berkurang sebanyak 1 unit.







9.2 Notulen Rapat

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







No. Kelompok/Kelas : 8 / K2
Nama Kelompok : SnakeOnHanoi
Anggota Kelompok (Nama/NIM) :
1. Michael Sihotang (18221054)
2. Muhammad Dastin Fauzi (18221062)
3. Imanuel Raditya (18221112)
4. Miralistya Cahya Fatimah (18221116)
5. Seren Elizabeth Siahaan (18221160)

Asisten Pembimbing : Graciella Valeska Liander (18219075)

Asistensi I

Tanggal : 23 November 2022	Catatan Asistensi: <ul style="list-style-type: none"> - Semua game yang ada di ListGame, scorenya dimasukkan kedalam scoreboard, termasuk game game hasil dari createGame - sesuaiin sama listGame, Mapnya ditambahkan kedalam array of map - prioritas nambah sesuai spesifikasi, yaitu tambah kiri, lalu tambah atas/bawah, meskipun letak ekor bukan di koordinat x ke 4 dan ekor menghadap kiri, tetap menambahkan ekor barunya pada posisi atas/bawah - Pastiin Tower of Hanoi yang telah dibuat sebagai bonus kemarin sesuai dengan spesifikasi sekarang - Tree model apa terserah, yang penting pakai ADT tree - persentase bonus tidak sebesar itu, akan lebih besar nilainya apabila game utama sesuai spesifikasi
Tempat : Zoom meetings	
Kehadiran Anggota Kelompok: <div> <div>1</div> <div>18221054</div> <div></div> </div> <div> <div>2</div> <div>18221062</div> <div></div> </div> <div> <div>3</div> <div>18221112</div> <div></div> </div> <div> <div>4</div> <div>18221116</div> <div></div> </div> <div> <div>5</div> <div>18221160</div> <div></div> </div>	
	Tanda Tangan Asisten: <div></div> <div>graciellari</div>

Asistensi II

Tanggal : 30 November 2022	Catatan Asistensi: <ul style="list-style-type: none"> - Untuk laporan apakah harus menuliskan semua command di tubes 1 yang terupdate karena disesuaikan dengan spesifikasi tubes 2, seperti playgame, creategame, deletegame, save, dan load selain semua command dan game pada tubes 2 ini? Tidak perlu, yang perlu dituliskan save dan load saja. - Apabila menerapkan bonus pada Snake On Meteor, maka apabila ada kasus seperti ini maka penambahan ekor snake yang baru akan dilakukan pada $y = 0$ (seberangnya)
Tempat : Zoom meetings	
Kehadiran Anggota Kelompok: <div style="text-align: center;"> <p>1 18221054</p>  <p>2 18221062</p>  <p>3 18221112</p>  <p>4 18221116</p>  <p>5 18221160</p>  </div>	
	Tanda Tangan Asisten:  <p>graciellari</p>

	A	B	C	D	E
1				4	
2					
3					
4			1 H		
5			2	3	#####
6					

9.3 Log Activity Anggota Kelompok

NO	Tanggal	NIM	Nama	Aktivitas
1	18/11/2022	18221054	Michael Sihotang	Push ADT Map dan ADT LinkedList (untuk snake on meteor)
2	20/11/2022	18221112	Imanuel Raditya	Fix ADT Map dan membuat ADT ArrayMap untuk command tambahan
3	21/11/2022	18221116	Miralistya Cahya F	Push inialisasi snake dan petak untuk tempat snake bejalan
4	21/11/2022	18221054	Michael Sihotang	Push ADT Listdp (List Double Pointer) untuk snake on meteor
5	21/11/2022	18221116	Miralistya Cahya F	Push belok snake
6	21/11/2022	18221054	Michael Sihotang	Fix jalan snake dan belok snake
7	22/11/2022	18221112	Imanuel Raditya	Push history, scoreboard, dan revisi Tower of Hanoi (Bonus Tubes 01)
8	22/11/2022	18221116	Miralistya Cahya F	Push dropfood untuk Snake on Meteor
9	23/11/2022	18221062	Muhammad Dastin F	Push meteor yang ada di snake on meteor
10	23/11/2022	18221116	Miralistya Cahya F	Push Revisi food dan meteor
11	23/11/2022	18221054	Michael Sihotang	Push revisi inialisasi snake menjadi string
12	24/11/2022	18221112	Imanuel Raditya	Revisi scoreboard dan Tower of Hanoi
13	25/11/2022	18221062	Muhammad Dastin F	Push revisi meteor
14	25/11/2022	18221116	Miralistya Cahya F	Push revisi ekor
15	25/11/2022	18221054	Michael Sihotang	Push kondisional game over dan skor

16	27/11/2022	18221112	Imanuel Raditya	Revisi history dan command Load
17	28/11/2022	18221116	Miralistya Cahya F	Finalisasi snake on meteor dan push BONUS snake on meteor
18	29/11/2022	18221160	Seren Elizabeth S	Push game hangman
19	29/11/2022	18221112	Imanuel Raditya	Revisi hangman
20	01/12/2022	18221116	Miralistya Cahya F	Finalisasi