

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data

BNMO

Dipersiapkan oleh:

Kelompok 13

Muhammad Naufal - 18221089

Ibnu Khairy Algifari - 18221091


Christopher Febrian Nugraha - 18221115

Natasya Vercelly Harijadi - 18221119

Kevin Prayoga Abdinegara - 18221141

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-13</i>		<i>34</i>
		<i>Revisi</i>	<i>1</i>	<i>2-12-2022</i>

Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Bonus Permainan Alstrokedungeon	4
2.2 Bonus Permainan Snake on Meteor	4
2.3 Bonus permainan Hangman	5
2.4 Bonus permainan Tower of Hanoi	5
3 Struktur Data (ADT)	6
3.1 ADT stack	6
3.2 ADT stackTOH	6
3.3 ADT listofmap	6
3.4 ADT listlinier	6
3.5 ADT point	7
3.6 ADT set	7
3.7 ADT dungeon_tree	7
4 Program Utama	8
4.1 Pembacaan dan Inisialisasi File Dari Data	8
4.2 Main Menu	8
4.3 Command CREATE GAME	8
4.4 Command DELETE GAME	8
4.5 Command PLAY GAME	8
4.6 Command SCOREBOARD	9
4.7 Command RESET SCOREBOARD	9
4.8 Command HISTORY <n>	9
4.9 Command RESET HISTORY	9
4.10 Command SAVE <filename>	9
5 Algoritma-Algoritma Menarik	9
5.1 ADT listofmap dengan keunikan InsertMap	9
5.2 ADT Tree dengan keunikan	9
6 Data Test	10
6.1 Data Test 1 : SCOREBOARD	10
6.2 Data Test 2 : RESET SCOREBOARD	11
6.3 Data Test 3 : HISTORY	11

6.4 Data Test 4 : RESET HISTORY	12
6.5 Data Test 5 : Hangman	12
6.6 Data Test 6 : Tower of Hanoi	14
6.7 Data Test 7 : Snake on Meteor	15
7 Test Script	19
8 Pembagian Kerja dalam Kelompok	24
9 Lampiran	26
9.1 Deskripsi Tugas Besar 2	26
9.2 Notulen Rapat	26
a. Rapat 18 November 2022	26
b. Rapat 25 November 2022	27
c. Rapat 1 Desember 2022	27
9.3 Log Activity Anggota Kelompok	27
9.4 Form Asistensi Tugas Besar	31

1 Ringkasan

Setelah robot BNMO berhasil diperbaiki, Indra dan Doni ingin mengembangkan lagi BNMO dengan menambahkan fitur-fitur dan permainan baru sehingga dibutuhkan *programmer* andal untuk melakukannya.

Laporan ini berisi tentang penjelasan program-program yang telah dibuat. Bagian pertama mencakup ringkasan yang di dalamnya terdapat deskripsi umum persoalan, isi umum laporan secara singkat, dan kesimpulan tentang hasil tugas besar secara umum. Bagian kedua berisi penjelasan tambahan spesifikasi fitur-fitur baru yang belum rinci dari deskripsi tugas besar. Bagian ketiga menjelaskan struktur data yang digunakan untuk mengatasi persoalan pada BNMO. Bagian keempat berisi penjelasan program utama, fitur-fitur baru yang ada pada BNMO. Bagian keenam dan ketujuh adalah bagian testing, diisi dengan data dan script yang digunakan untuk menguji program yang baru. Bagian kedelapan berisi pembagian tugas tiap anggota. Bagian terakhir berisi lampiran-lampiran.

Pengembangan permainan BNMO ini dibuat dengan menggunakan bahasa C dan ADT yang sudah dipelajari dalam mata kuliah IF2111 Algoritma dan Struktur Data STI. ADT yang digunakan pada program BNMO kali ini yaitu stack, stackTOH, listofmap, set, listlinier, point, dungeontree, dan mesinkar_dungeon. Kami menggunakan ADT tersebut karena merasa sudah cukup untuk menjalankan program serta spesifikasi yang diberikan. Selain itu, program yang dimiliki pun mampu berjalan di linux.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Bonus Permainan Alstrokedungeon

Permainan Alstrokedungeon dimulai dengan pembuatan map dungeon dengan kedalaman yang ditentukan pemain. Game ini menggunakan ADT tree untuk membuat keseluruhan peta dari dungeon. Program akan menampilkan cerita dan gameplay berupa *RPG Roguelike*. Terdapat pilihan pendamping yang akan membantu player dalam permainan ini. Dalam dungeon, terdapat beberapa monster yang akan menghalangi pemain dari menjelajah lebih lanjut, sehingga harus dilawan. Skor dihitung dari banyaknya jumlah *gold* dijumlahkan dengan poin yang didapat ketika mengalahkan monster. Player bisa menggunakan serangan biasa ataupun skill. Skill bisa berupa skill defensif atau ofensif, tergantung pendamping yang dipilih player. Jika player atau pendamping terbunuh ($HP \leq 0$) maka permainan berakhir. Player dinyatakan memenangkan permainan apabila bisa mengalahkan boss di level terakhir.

2.2 Bonus Permainan Snake on Meteor

Pada awal permainan, selain terdapat snake dan food, juga terdapat 2 buah obstacle. Letak obstacle diacak oleh program. Obstacle ini tidak akan berubah posisinya dari awal hingga akhir permainan. Program dapat menerima input yang membuat snake menabrak obstacle. Jika kepala snake menabrak obstacle, maka snake akan mati. Skor dihitung dari panjang tubuh snake dikali 2 (termasuk kepala). Meteor, food, dan snake yang di-generate di awal tidak mungkin berada pada koordinat yang sama dengan koordinat obstacle. Ekor baru yang muncul setelah snake memakan food juga tidak mungkin di-generate di koordinat yang sama dengan koordinat obstacle. Jika suatu saat snake tidak dapat menambahkan ekor atau menggerakkan kepala sama sekali karena terhalang meteor/dirinya sendiri/obstacle, maka permainan berakhir.

Snake dapat menembus dinding field, misalnya jika kepala snake berada pada titik (0,0) lalu pemain memasukkan arah “w”, kepala snake akan berada pada (0,4). Hal yang sama berlaku pada sisi kiri dan kanan field, jika ditembus, kepala snake akan muncul pada sisi yang berseberangan. Fitur ini diatur pada prosedur teleport.

2.3 Bonus permainan Hangman

Program bonus Hangman adalah menambahkan kata baru ke permainan. Setiap kata yang ditambahkan adalah kata yang berbeda dari yang ada di kamus. Kata yang dimasukkan ke kamus menggunakan huruf kapital. Jika user memasukan kata dengan huruf kecil, maka program otomatis meng-*convert* semua huruf pada kata menjadi kapital. Kata yang ditambahkan ke kamus masuke ke file utama program BNMO, artinya setiap kata yang ditambahkan User 1 ke kamus akan otomatis digunakan oleh User lainnya.

2.4 Bonus permainan Tower of Hanoi

Program bonus Hangman adalah bisa menerima input piringan. Maksimum piringan yang bisa dimasukkan adalah 100. Semakin banyak piringan, maka skor maksimum akan lebih besar dari pada skor maksimum untuk piringan yang lebih sedikit. Skor dihitung berdasarkan berapa banyak langkah user untuk menyelesaikan perpindahan piringan dibandingkan dengan langkah efektif dengan perhitungan yang sudah ada di program. Jika user melakukan perpindahan piringan hingga game berakhir sebanyak langkah efektif–yang dihitung berdasarkan perhitungan program tergantung pada jumlah piringan– memindahkan piringan, maka user mendapatkan skor maksimum. Jika lebih dari langkah efektif, maka skor makin berkurang dari skor maksimum.

3 Struktur Data (ADT)

3.1 ADT stack

ADT Stack berupa program yang menampung elemen-elemen berupa string dalam bentuk tumpukan. ADT Stack mampu untuk mengeksekusi fungsi dan prosedur primitif, diantaranya merupakan Pop dan Push. Persoalan yang diselesaikan adalah persoalan History Game dan dari program yang akan memanfaatkan ADT ini yang digunakan untuk merepresentasikan urutan dari permainan yang telah dimainkan, dengan TOP adalah permainan yang baru saja dimainkan. Alasan lain ADT ini dipilih adalah karena input berupa elemen-elemen yang berada pada urutan tumpukan dan memerlukan proses masuk-keluar dengan mekanisme LIFO (Last in First out). ADT ini dideklarasikan pada stack.h, diimplementasikan pada stack.c, dan diuji di driverstack.c

3.2 ADT stackTOH

ADT Stack TOH berupa program yang menampung elemen-elemen berupa integer dalam bentuk tumpukan. ADT Stack TOH mampu untuk mengeksekusi fungsi dan prosedur primitif, diantaranya merupakan PopTower dan PushTower. ADT ini dibuat karena kebutuhan untuk menggunakan infotype integer pada sebuah Stack. Persoalan yang diselesaikan adalah persoalan game Tower of Hanoi. Alasan ADT ini dipilih adalah karena input berupa elemen-elemen berada pada urutan tumpukan dan memerlukan proses masuk-keluar dengan mekanisme LIFO (Last in First out) dan kebutuhan atas game Tower of Hanoi yang menggunakan mekanisme Stack. ADT ini dideklarasikan pada stackTOH.h, diimplementasikan pada stackTOH.c, dan diuji di driverstackTOH.c

3.3 ADT listofmap

ADT listofmap digunakan untuk menyimpan semua data berupa nama dan skor dari setiap game. Setiap game hanya memiliki satu kumpulan nama dan skor yang disebut map. Dalam satu map berisi nama yang tidak boleh sama satu sama lain. ADT listofmap dimanfaatkan untuk menyimpan nilai nama berupa char *keytype dan skor berupa int valuetype ke dalam map dan terdiri dari map yang jumlahnya sama dengan jumlah game sehingga dibutuhkan list untuk mengumpulkan semua map. Prosedur yang sering dipakai yaitu InsertMap dengan skor yang dimasukkan akan diurutkan berdasarkan jumlah tertinggi. Jika skor sama, maka skor yang lebih dulu dimasukkan ke scoreboard ditampilkan lebih dulu.

3.4 ADT listlinier

ADT list linier berupa program yang digunakan untuk menampung elemen dengan sistem list berkait, hal ini dilakukan dengan tujuan supaya elemen yang disimpan tidak harus kontigu dan bisa dilakukan dealokasi tanpa mengganggu dan mengubah lokasi elemen lain. Persoalan yang diselesaikan oleh ADT ini adalah penyimpanan alamat kepala dari snake dalam game snake on meteor supaya mudah mengakses kepala snake karena banyak operasi yang membutuhkan kepala snake (contoh: moveSnake). Selain itu, ADT ini dimodifikasi untuk menyimpan infotype dalam bentuk point (titik/koordinat), menjadi list of point. Oleh karena itu, ADT ini

menyelesaikan persoalan pemodelan snake. ADT ini dideklarasikan pada `listlinier.h`, diimplementasikan pada `listlinier.c`, dan diuji di `driverlistlinier.c`.

3.5 ADT point

ADT Point merupakan bentuk gabungan antara 2 integer sebagai absis dan ordinat. Persoalan yang diselesaikan ADT ini adalah implementasi koordinat pada C, sehingga segala item pada game Snake on Meteor bisa ditampilkan pada *field* kotak-kotak dengan ukuran 5x5 unit. ADT ini dideklarasikan pada `point.h`, diimplementasikan pada `point.c`, dan diuji di `driverpoint.c`.

3.6 ADT set

ADT Set berupa program yang digunakan untuk menampung elemen dengan nilai yang berbeda dengan tipe yang sama dalam sebuah list. Ada dua tipe struktur data pada Set, yang pertama Set untuk menyimpan `char` dan Set untuk menyimpan `char*`. Persoalan yang diselesaikan oleh ADT Set adalah menyimpan kata dan karakter yang unik dalam permainan Hangman. Alasan kedua ADT ini dipilih karena kebutuhan untuk menyimpan nilai-nilai yang unik dalam sebuah list sehingga tidak ada duplikasi. ADT ini dideklarasikan pada `set.h`, diimplementasikan pada `set.c`, dan diuji di `driverset.c`.

3.7 ADT dungeon_tree

ADT tree berupa program yang digunakan untuk menampung elemen berupa integer dengan sistem tree, yang terdiri atas root(node paling atas), leaf(node paling bawah), parent dan children. Persoalan yang diselesaikan oleh ADT ini adalah game AlstrokeDungeon, yang mana ADT ini digunakan untuk men-*generate* map dungeon yang akan dimainkan pemain, ADT ini digunakan karena game membutuhkan masukan “kedalaman” dan “tingkat kesulitan” untuk map-nya. Selain itu, terdapat pula program untuk membuat sebuah List of Node yang dipakai untuk mengubah elemen dari node pada tree tanpa mengakses tree secara rekursif. ADT ini dideklarasikan pada `dungeon_tree.h`, diimplementasikan pada `dungeon_tree.c`, dan diuji pada `driverdungeon_tree.c`.

3.8 ADT mesinkar_dungeon

ADT mesinkar_dungeon ini berupa program yang merupakan modifikasi dari ADT mesinkar yang biasanya digunakan untuk menerima input dari stdin. ADT ini berfungsi untuk menampilkan ASCII art untuk game bonus AlstrokeDungeon. Konsiderasi penambahan modifikasi ADT ini adalah tidak semua program menggunakan ADT ini, dan fungsi ADT ini spesifik sehingga tidak bisa digantikan dengan ADT mesinkar biasa. ADT ini dideklarasikan pada `mesinkar_dungeon.h`, diimplementasikan pada `mesinkar_dungeon.c`, dan diuji pada `drivermesinkardungeon.c`.

4 Program Utama

Program utama dari BNMO sebagian besar sama sudah terlaksanakan pada bagian Tugas Besar 1, akan tetapi terdapat beberapa perubahan dikarenakan terdapat command SCOREBOARD, RESET SCOREBOARD, HISTORY, dan RESET HISTORY. Selain itu, program telah ditambahkan dengan game HANGMAN, TOWER OF HANOI, SNAKE ON METEOR, dan ALSTROKEDUNGEON, serta menghilangkan game default yang maintenance sehingga sekarang semua game bisa dimainkan. Pada game buatan pemain, game akan langsung mengeluarkan output skor acak. Program hanya membaca file console saja karena semua file telah disatukan ke dalam console.

4.1 Pembacaan dan Inisialisasi File Dari Data

Program akan meminta input berupa START atau LOAD. Jika START, maka program akan membaca dari jumlah spasi yang terdapat di command. Ketika nol, maka akan masuk ke START lalu membaca file config.txt yaitu file konfigurasi dari sistem yang memiliki tujuh game *default*. Jika memilih LOAD <filename>, maka program mendeteksi jumlah spasinya satu dan akan membaca file yang diketik. Namun, terdapat perubahan pada command LOAD karena LOAD akan membaca file .txt dengan urutan list game dengan tipe arraydin, history game bertipe stack, dan scoreboard game bertipe listofmap dari game yang ada di list game. Jika isi file telah terbaca semua, maka akan menghasilkan output file telah terbaca. Akan tetapi, file tidak akan terbaca jika file bukan berada di data. Jika input command salah atau file tidak terdapat di folder data, maka program akan meminta input command kembali hingga benar. Setelah benar, maka akan lanjut ke tahap main menu.

4.2 Main Menu

Setelah melakukan pembacaan file, program akan masuk ke main menu dan meminta input command kembali dengan memiliki pilihan command berupa LIST GAME, QUEUE GAME, CREATE GAME, DELETE GAME, SKIP GAME, PLAY GAME, SCOREBOARD, RESET SCOREBOARD, HISTORY, RESET HISTORY, SAVE, HELP, dan QUIT. Program akan membaca dari jumlah spasinya. Jika melebihi dari dua maka program akan meminta input command kembali. Jika kurang dari sama dengan dua dan input command benar, maka akan masuk ke dalam fungsi tersebut. Jika fungsi yang dijalankan berupa QUIT, maka program akan keluar dari main menu dan program telah berakhir.

4.3 Command CREATE GAME

Perbedaan ketika tugas besar 1 dengan tugas besar 2, command ini secara otomatis membuat map setiap game buatan pemain masuk ke dalam list game.

4.4 Command DELETE GAME

Perbedaan ketika tugas besar 1 dengan tugas besar 2, command ini akan menghapus history serta scoreboard sebelum menghapus dari list game ketika game yang dihapus memenuhi syarat.

4.5 Command PLAY GAME

Perbedaan ketika tugas besar 1 dengan tugas besar 2, command ini akan melakukan push ke dalam history setelah dihapus dari antrian game. Kemudian, game yang akan dimainkan tidak memiliki game maintenance, semua game dapat dimainkan. Ketika game dimainkan dan mendapatkan skor, maka akan secara otomatis meminta nama pemain maksimal yaitu 13 huruf.

Ketika melebihi jumlah maksimal, maka akan meminta input nama kembali hingga benar dan otomatis masuk ke dalam scoreboard game tersebut.

4.6 Command SCOREBOARD

Command ini akan menampilkan scoreboard dari setiap game yang ada, dengan nama pemain maksimal yaitu 13 huruf dan urutan yang ditampilkan di scoreboard sesuai dengan jumlah skor. Ketika skor sama, maka yang ditampilkan duluan adalah pemain yang lebih dulu memainkan game tersebut.

4.7 Command RESET SCOREBOARD

Program akan meminta input urutan game yang akan dihapus scoreboardnya dan diubah menjadi nol, dengan urutan nol berupa *ALL* yaitu menghapus semua scoreboard game dan diubah menjadi nol. Ketika sudah memilih urutan, maka program akan memastikan ulang apakah ingin melakukan reset atau tidak. Jika iya, maka program akan menghapus scoreboard pada urutan yang telah dipilih. Jika tidak, maka reset dibatalkan.

4.8 Command HISTORY <n>

Program akan menampilkan game sebanyak jumlah yang kita input dan yang telah dimainkan dengan urutan paling atas yaitu game yang dimainkan terakhir kali. Jika input jumlah game yang ditampilkan melebihi history yang ada, maka game tetap menampilkan semua history yang ada.

4.9 Command RESET HISTORY

Program akan mengecek apakah history ada atau tidak. Jika ada, maka program akan mengkonfirmasi apakah ingin melakukan reset atau tidak. Jika iya, maka program akan menghapus semua history hingga kosong. Jika tidak, maka program akan menampilkan semua history yang masih ada. Ketika melakukan pengecekan di awal apakah history ada atau tidak, dan jika tidak, maka program akan menampilkan bahwa tidak ada daftar game yang telah dimainkan.

4.10 Command SAVE <filename>

Ketika melakukan save, perbedaan tugas besar 1 dengan tugas besar 2 yaitu save file akan membaca list game, history game, dan list of map ke dalam suatu file .txt yang diketik oleh *user*.

5 Algoritma-Algoritma Menarik

5.1 ADT listofmap dengan keunikan InsertMap

Algoritma dari ADT listofmap cukup menarik untuk dibahas. Hal ini karena Map yang dibuat akan dikumpulkan dalam suatu list. Menariknya itu map merupakan ADT yang memiliki key dan value per elemennya, lalu dikumpulkan dalam suatu kumpulan yang bernama list. Selain itu, ketika ingin memasukan suatu key atau value ke dalam listofmap, maka value akan diurutkan menurun. Hal lain yang menarik disini, key berupa char* atau string, sehingga jika ingin validasi apakah sudah ada key yang tercatat dalam listofmap, maka harus melakukan perbandingan per huruf apakah sama atau tidak.

5.2 ADT Tree dengan keunikan

Selain itu, algoritma dari ADT tree menarik karena Node dari sebuah Ternary Tree dikumpulkan dalam suatu list untuk mempermudah akses. Node memiliki tipe bentukan berupa pointer ke tree left, mid, dan right, serta info berupa integer. Ketika dimasukkan kedalam list, daun (node) akan terurut secara pre-order. Untuk mengakses sebuah node spesifik, kita bisa membandingkan level dari node tersebut dan mencari nilainya di level tersebut tanpa menggunakan search tree secara rekursif.

6 Data Test

6.1 Data Test 1 : SCOREBOARD

Test ini digunakan untuk menguji program Scoreboard

```
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 Minesweeper ****
| NAMA          | SKOR          |
|-----|-----|
| SCOREBOARD KOSONG -----|

**** SCOREBOARD GAME Alstrokedungeon ****
| NAMA          | SKOR          |
|-----|-----|
| SCOREBOARD KOSONG -----|
```

Secara default program akan menampilkan scoreboard kosong dari game yang sudah tersedia.

```
ENTER COMMAND: SCOREBOARD

**** SCOREBOARD GAME RNG ****
| NAMA          | SKOR          |
|-----|-----|
| KEVIN          | 25            |
|-----|-----|
```

Setiap user sudah menyelesaikan permainan suatu game, skor permainan dari user akan otomatis masuk ke scoreboard.

```
**** SCOREBOARD GAME INI GAME BARU ****
| NAMA          | SKOR          |
|-----|-----|
| SCOREBOARD KOSONG -----|
```

Apabila user menjalankan CREATE GAME, maka game yang dibuat otomatis memiliki scoreboard kosong di program.

6.2 Data Test 2 : RESET SCOREBOARD

Test ini digunakan untuk menguji Reset Scoreboard

```
DAFTAR SCOREBOARD:
0. ALL
1. RNG
2. Diner DASH
3. HANGMAN
4. TOWER OF HANOI
5. SNAKE ON METEOR
6. Minesweeper
7. Alstrokedungeon

SCOREBOARD YANG INGIN DIHAPUS: 1

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

Scoreboard berhasil di-reset.
```

User akan diminta untuk memasukkan nomor game yang ingin dihapus scoreboardnya. Jika nomor game yang dimasukkan tidak ada di daftar scoreboard maka akan menampilkan output tidak valid.

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

User menghapus scoreboard dari game RNG yang sudah ada di data test 1. Setelah melakukan reset scoreboard, scoreboard pada game RNG menjadi kosong.

6.3 Data Test 3 : HISTORY

Test ini digunakan untuk menguji History

```
ENTER COMMAND: HISTORY 4

Tidak ada daftar game yang telah dimainkan.

Press ENTER key to back...
```

Secara default history akan menampilkan pesan tidak ada daftar game yang telah dimainkan

```
ENTER COMMAND: HISTORY 2

Berikut adalah daftar game yang telah dimainkan:
1. HANGMAN
2. RNG
```

Apabila user telah memainkan beberapa game yang berbeda, maka akan ditampilkan di dalam history sesuai dengan urutan yang paling terakhir dimainkan, history juga hanya akan menampilkan jumlah game yang dimainkan sebanyak yang diminta (dalam hal ini diminta 2 game saja).

```
ENTER COMMAND: HISTORY 5

Berikut adalah daftar game yang telah dimainkan:
1. HANGMAN
2. RNG
```

Apabila user menginput angka yang lebih daripada jumlah game yang pernah dimainkan user, daftar history yang ditampilkan hanya akan ditampilkan sebanyak yang pernah dimainkan user.

6.4 Data Test 4 : RESET HISTORY

Test ini digunakan untuk menguji Reset History

```

ENTER COMMAND: RESET HISTORY

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

History tidak jadi di-reset. Berikut adalah daftar Game yang telah dimainkan:
1. HANGMAN
2. RNG

Press ENTER key to back...

```

Apabila user tidak jadi melakukan reset history, maka akan muncul pesan diatas disertai dengan daftar history yang tidak jadi dihapus.

```

ENTER COMMAND: RESET HISTORY

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

History berhasil di-reset.

Press ENTER key to back...

```

Apabila user jadi melakukan reset history, maka akan muncul pesan history berhasil di-reset dengan kondisi history yang sekarang sudah kosong

```

ENTER COMMAND: HISTORY 2

Tidak ada daftar game yang telah dimainkan.

Press ENTER key to back...

```

History yang telah berhasil direset akan kembali kosong seperti gambar diatas.

6.5 Data Test 5 : Hangman

Test ini digunakan untuk menguji program Hangman.

```

=====
+ 1. Mulai Permainan :
+
=====
+ 2. Tambahkan kata baru :
+
=====
+ 3. Mulai
+
=====
Masukkan perintah :

```

```

Masukkan perintah : 4
Masukkan tidak valid!

Masukkan perintah :

```

Program diawali dengan meminta masukan angka dari 3 pilihan yaitu: 1 untuk memulai permainan, 2 untuk menambahkan kata baru, 3 untuk keluar/quit, selain itu inputan dianggap tidak valid.

```
Masukkan perintah : 1
Tebakan sebelumnya: -
- - - - -
Kesempatan: 10
Masukkan tebakan: K
Tebakan sebelumnya: K
- - - - -
Kesempatan: 9
```

Program akan memulai permainan Hangman apabila menerima input 1. Secara otomatis program akan *generate* sebuah kata untuk ditebak, setiap tebakan yang salah akan mengurangi kesempatan menebak.

```
Tebakan sebelumnya: K
- - - - -
Kesempatan: 9
Masukkan tebakan: k
Karakter tidak valid!

Masukkan tebakan: /
Karakter tidak valid!
```

Program hanya menerima input berupa character yang *case-insensitive* ketika bermain, input selain itu dianggap tidak valid dan tidak dianggap sebagai tebakan yang tidak valid (tidak ada pengurangan kesempatan). Kata yang sebelumnya telah dimasukkan dianggap sebagai tebakan yang tidak valid.

```
Tebakan sebelumnya: KZXCVMQWERT
_ E _ _ _ N _
Kesempatan: 1
Masukkan tebakan: y

Kamu gagal menebak kata. Kata tebakan adalah BELALANG.

Game Over!

Skor: 0
```

Setiap berhasil menebak kata, poin akan bertambah sebanyak panjang kata yang ditebak. Ketika kesempatan menebak berakhir, game akan berakhir.

```
Masukkan kata baru : TIKUS
Kata TIKUS berhasil dimasukkan.
```

```
Masukkan perintah : 2
Ketik 'q' jika ingin membatalkan.
Masukkan kata baru : q
Masukkan perintah :
```

Penambahan kata pada Hangman menerima semua kata inputan secara bebas, apabila program menerima input 'q', program akan otomatis membatalkan command untuk memasukkan kata baru pada Hangman.

```
Ketik 'q' jika ingin membatalkan.  
Masukkan kata baru : KUDA  
Kata sudah tersedia!
```

User tidak bisa memasukkan kata yang sudah ada di kamus.

[illegible]

Program akan langsung keluar apabila menerima masukan 3.

6.6 Data Test 6 : Tower of Hanoi

Test ini digunakan untuk menguji program Tower of Hanoi

[illegible]

Program diawal akan meminta input banyaknya disk yang akan dimainkan, masukkan hanya berupa integer kurang dari sama dengan 100, input selain itu akan dianggap tidak valid.

```

|   |   |
*** *   |
A   B   C
---
TIANG ASAL : A
TIANG TUJUAN : B

Tiang tujuan tidak valid!

```

Piringan dengan ukuran yang lebih besar tidak bisa menimpa piringan yang lebih kecil, jika kondisi demikian terjadi, maka program akan menunjukkan perintah yang tidak valid.

```

|   |   *
|   |   ***
A   B   C
---
Kamu berhasil!

Skor didapatkan = 4.00

```

Permainan akan berakhir apabila semua piringan telah berpindah ke tower C.

6.7 Data Test 7 : Snake on Meteor

Test ini digunakan untuk menguji game Snake on Meteor.

Program diawali dengan generate 2 buah obstacle, snake (head dan 2 unit tubuh), dan food.

```

Selamat datang di Snek on Meteor!
Mengenerate peta, snek, obstacle, dan makanan...
Berhasil di-generate!

+---+---+---+---+
| | | | | | |
+---+---+---+---+
| # | | | | |
+---+---+---+---+
| | | | # | |
+---+---+---+---+
| 2 | 1 | H | | O |
+---+---+---+---+
| | | | | |
+---+---+---+---+
Silakan masukkan command Anda: _

```

Selanjutnya pemain menggerakkan snake dengan cara memasukkan “w/a/s/d/W/A/S/D”. Input selain itu tidak akan diterima dan program akan meminta input ulang.


```

Silakan masukkan command Anda: asdad
Input tidak valid!
Silakan masukkan command yang lain.

+---+---+---+---+
| # |   |   | M |   |
+---+---+---+---+
| O |   |   |   |   |
+---+---+---+---+
|   |   |   | H |   |
+---+---+---+---+
|   |   | 2 | 1 | # |
+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
Silakan masukkan command Anda:

```

```

Silakan masukkan command Anda: w
Berhasil bergerak!

+---+---+---+---+
| # |   |   |   |   |
+---+---+---+---+
| O |   |   | H |   |
+---+---+---+---+
|   |   |   | 1 |   |
+---+---+---+---+
| M |   |   | 2 | # |
+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+

```

Jika command menyebabkan snake menabrak tubuhnya sendiri atau meteor panas, program akan meminta input ulang dan tidak menggerakkan snake.

```

+---+---+---+---+
| # | # |   |   |
+---+---+---+---+
|   | O |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   | M |   |   |
+---+---+---+---+
| 2 | 1 | H |   |
+---+---+---+---+
Silakan masukkan command Anda: w
Meteor masih panas, Anda belum dapat kembali ke titik tersebut.
Input tidak valid!
Silakan masukkan command yang lain.

+---+---+---+---+
| # | # |   |   |
+---+---+---+---+
|   | O |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   | M |   |   |
+---+---+---+---+
| 2 | 1 | H |   |
+---+---+---+---+
Silakan masukkan command Anda:

```

```

+---+---+---+---+
| # | # |   |   |
+---+---+---+---+
|   | O |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   | M |   |   |
+---+---+---+---+
| 2 | 1 | H |   |
+---+---+---+---+
Silakan masukkan command Anda: a
Anda tidak dapat bergerak ke tubuh Anda sendiri!
Input tidak valid!
Silakan masukkan command yang lain.

+---+---+---+---+
| # | # |   |   |
+---+---+---+---+
|   | O |   |   |
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
|   | M |   |   |
+---+---+---+---+
| 2 | 1 | H |   |
+---+---+---+---+
Silakan masukkan command Anda: _

```

Jika snake bergerak menabrak dinding field, snake akan muncul pada sisi yang berseberangan.

```

+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
|   |   | 2 | 1 | H |
+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
|   | # | O |   |   |
+---+---+---+---+
| # |   |   |   |   |
+---+---+---+---+
Silakan masukkan command Anda: d
Berhasil bergerak!

+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
| H |   |   | 2 | 1 |
+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
| M | # | O |   |   |
+---+---+---+---+
| # |   |   |   |   |
+---+---+---+---+

```

Jika snake menabrak obstacle, permainan berakhir. Skor dihitung dari panjang tubuh snake (beserta kepala) dikali 2.

```

Silakan masukkan command Anda: a
Berhasil bergerak!
Mati nabrak obstacle:(

+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
| O |   |   |   | # |
+---+---+---+---+
| M |   |   |   |   |
+---+---+---+---+
|   |   |   |   | 2 |
+---+---+---+---+
|   |   | # | 1 |   |
+---+---+---+---+
Game over! Score Anda: 6

```

Jika kepala snake kena meteor, permainan berakhir. Kepala snake hilang. Skor dihitung dari panjang tubuh snake (tanpa kepala) dikali 2.

```

Silakan masukkan command Anda: w
Berhasil bergerak!
Kepala snek terkena meteor!:(

+---+---+---+---+
|   |   |   |   |   |
+---+---+---+---+
|   |   |   | O |   |
+---+---+---+---+
|   |   | 4 |   |   |
+---+---+---+---+
|   | # | 3 | M |   |
+---+---+---+---+
| # |   | 2 | 1 |   |
+---+---+---+---+
Game over! Score Anda: 8

```

Jika snake tidak dapat bergerak lagi, permainan berakhir. Skor dihitung dari panjang tubuh snake (beserta kepala) dikali 2.

```

+---+---+---+---+
| O | M |   | 2 |   |
+---+---+---+---+
| 4 | 5 |   | 3 |   |
+---+---+---+---+
|   |   |   | # |   |
+---+---+---+---+
|   |   | # |   |   |
+---+---+---+---+
| H |   |   | 1 |   |
+---+---+---+---+
Silakan masukkan command Anda: s
Berhasil bergerak!
Snek makan
Snek tidak dapat bergerak!:(

+---+---+---+---+
| H | M |   | 3 |   |
+---+---+---+---+
| 5 |   |   | 4 |   |
+---+---+---+---+
| 6 |   |   | # |   |
+---+---+---+---+
|   |   | # |   |   |
+---+---+---+---+
| 1 | O |   | 2 |   |
+---+---+---+---+
Game over! Score Anda: 14

```

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Scoreboard	Mengecek apakah score tersimpan di dalam scoreboard dan bekerja dengan baik	Di main menu: Masukkan command SCOREBOARD Enter	Dapat dilihat pada bagian 6.1	Scoreboard ditampilkan dengan urutan yang sesuai dan scoreboard terupdate setiap pemain selesai bermain	Scoreboard terlihat dengan urutan yang sesuai dan sudah terupdate
2	Reset Scoreboard	Mengecek apakah scoreboard dapat direset	Di main menu: Masukkan command RESET SCOREBOARD Masukkan input angka urutan game yang akan dihapus menjadi nol Masukkan input kata YA/TIDAK sesuai keinginan Enter	Dapat dilihat pada bagian 6.2	Scoreboard game yang dimaksud akan terhapus dan menjadi 0	Scoreboard game berhasil terhapus
3	History <n>	Mengecek apakah history permainan pemain dapat disimpan, dan apakah urutan history sudah benar, serta apakah history yang ditampilkan sudah sesuai dengan angka masukan	DI main menu: Masukkan command HISTORY diiringi dengan angka masukan jumlah history game yang ingin ditampilkan Enter	Dapat dilihat pada bagian 6.3	History ditampilkan sebanyak n, kemudian game yang paling <i>recent</i> dimainkan akan berada pada urutan paling atas history	History yang ditampilkan sudah sesuai keinginan
4	Reset History	Mengecek apakah history game ada, dan mengecek	Di main menu: Masukkan command RESET HISTORY	Dapat dilihat pada bagian 6.4	Jika input kata YA, program akan menghapus seluruh history yang ada, lalu	History terhapus semua saat user memasukkan kata YA

		apakah history dapat dihapus	Masukkan input kata YA/TIDAK sesuai keinginan Enter		akan menampilkan pesan berhasil. Kemudian jika input kata TIDAK, maka program akan menampilkan pesan reset tidak jadi dilakukan	History batal dihapus saat user memasukkan kata TIDAK
5	Game Hangman	Menguji apakah game Hangman bisa dimainkan	<p>Di main menu ada 2 pilihan:</p> <p>> Menggunakan PLAY GAME</p> <p>Masukkan command QUEUE GAME lalu enter</p> <p>Input nomor yang menunjukkan urutan game HANGMAN lalu enter</p> <p>Enter</p> <p>Masukkan command PLAY GAME lalu enter</p> <p>> Menggunakan SKIP GAME dengan asumsi game Hangman berada di lokasi n pada queue</p> <p>Masukkan command SKIP GAME</p> <p>Diikuti dengan angka senilai n-1</p> <p>Enter</p> <p>Pada laman game Hangman: program akan meminta masukan angka</p> <p>Apabila pemain memilih 1, permainan akan dimulai</p> <p>Apabila pemain memilih 2, pemain akan diminta memasukkan kata baru dan harus valid</p>	Dapat dilihat pada bagian 6.5	Program akan otomatis memulai game hangman dan menampilkan laman utama game. Semua input yang dimasukan memberikan output yang sesuai dengan ketentuan yang sudah dijelaskan.	Game dapat dimainkan dan menampilkan skor akhir permainan jika berhasil. serta mengeluarkan skor akhir pemain. Setiap pemain melakukan kesalahan input program juga sudah dapat menampilkan pesan kesalahan Pada saat pemain gagal menebak kata yang diminta, program juga berhasil melakukan terminate loop game, dan menampilkan skor akhir pemain.

			<p>Apabila pemain memilih 3, pemain akan keluar dari game</p> <p>Apabila pemain menginput selain angka 1/2/3, maka program akan memunculkan pesan kesalahan.</p> <p>saat pilihan pemain adalah 1</p> <p>Program memulai permainan dengan meminta pemain memasukkan sebuah huruf <i>case-insensitive</i>,</p> <p>program akan mengecek huruf tersebut bagian dari kata tebakan atau tidak, inilah yang menjadi ukuran pengurangan kesempatan pemain</p> <p>lakukan terus hingga kesempatan berakhir</p> <p>Game selesai</p>			
6	Game Tower of Hanoi	Menguji apakah game Tower of Hanoi bisa dimainkan	<p>Di main menu ada 2 pilihan:</p> <p>> Menggunakan PLAY GAME</p> <p>Masukkan command QUEUE GAME lalu enter</p> <p>Input nomor yang menunjukkan urutan game Tower of Hanoi lalu enter</p> <p>Enter</p> <p>Masukkan command PLAY GAME lalu enter</p> <p>> Menggunakan SKIP GAME dengan asumsi game Tower of Hanoi</p>	Dapat dilihat pada bagian 6.6	Program akan memainkan game TOWER OF HANOI dan menampilkan laman utama game. Semua input yang dimasukan memberikan output yang sesuai dengan ketentuan yang sudah dijelaskan.	Game dapat dimainkan dan berhasil dijalankan. serta mengeluarkan skor akhir pemain. Pada saat pemain melakukan kesalahan input program juga sudah dapat menampilkan pesan kesalahan Pada saat pemain berhasil memindahkan

			<p>berada di lokasi n pada queue</p> <p>Masukkan command SKIP GAME</p> <p>Diikuti dengan angka senilai n-1</p> <p>Enter</p> <p>Pada laman game Tower of Hanoi: program akan meminta masukan angka jumlah piringan</p> <p>Berikan masukan yang benar yaitu integer atau masukan salah yaitu float atau char</p> <p>Setelah masukan piringan benar, program akan meminta masukan tiang awal dan tiang tujuan</p> <p>Berikan masukan yang benar yaitu semua tiang asalkan piringan yang lebih besar tidak “menindih” piringan yang lebih kecil</p> <p>Lakukan terus hingga semua piringan telah berpindah tiang</p> <p>Game selesai</p>			semua piringan , program berhasil menampilkan skor dan melakukan terminate game.
7	Game Snake on Meteor	Menguji apakah game Snake on Meteor bisa dimainkan	<p>Di main menu ada 2 pilihan:</p> <p>> Menggunakan PLAY GAME</p> <p>Masukkan command QUEUE GAME lalu enter</p> <p>Input nomor yang menunjukkan urutan game Snake on Meteor lalu enter</p> <p>Enter</p>	Dapat dilihat pada bagian 6.7	Program akan memainkan game SNAKE ON METEOR dan menampilkan laman utama game. Semua input yang dimasukan memberikan output yang sesuai dengan ketentuan	Game dapat dimainkan, sudah dapat mengidentifikasi kesalahan dan kasus yang ada dan menampilkannya ke layar.

			<p>Masukkan command PLAY GAME lalu enter</p> <p>> Menggunakan SKIP GAME dengan asumsi game Snake on Meteor berada di lokasi n pada queue</p> <p>Masukkan command SKIP GAME</p> <p>Diikuti dengan angka senilai n-1</p> <p>Enter</p> <p>Pada laman game Snake on Meteor: program akan men-generate 2 obstacle, 1 food, dan seekor snake berukuran 3 unit. Program selalu meminta input user selama belum game over. Setelah input user, program akan selalu menampilkan current field dan pesan terkait input user (misalnya input valid sehingga snake dapat bergerak, input invalid, dll.) dan keadaan snake (snake makan, snake kena meteor).</p> <p>Masukan yang benar adalah w/a/s/d/W/A/S/D yang tidak membuat snake menabrak meteor panas, atau dirinya sendiri. Snake dapat menembus dinding. Snake dapat bergerak ke arah obstacle. Hati-hati! Menabrak obstacle = kalah.</p> <p>Jika masukan user tidak valid, program akan meminta ulang input.</p> <p>Lakukan input arah hingga game over. Kondisi game</p>		yang sudah dijelaskan.	
--	--	--	---	--	---------------------------	--

			<p>over: snake tidak dapat bergerak, ekor snake tidak dapat tumbuh, kepala snake kena meteor/obstacle. Skor dihitung berdasarkan panjang tubuh snake dikali 2. Jika kepala snake kena meteor, kepala snake hilang, maka skor = panjang tubuhnya saja.</p> <p>Game selesai</p>			
--	--	--	---	--	--	--

8 Pembagian Kerja dalam Kelompok

No.	Fitur	NIM Coder	NIM Tester
1.	ADT Set (primitif dan driver)	18221089, 18221091	18221089, 18221091, 18221115, 18221119, 1822114
2.	ADT List of Map (primitif dan driver)	18221141	18221089, 18221091, 18221115, 18221119, 1822114
3.	ADT Stack (primitif dan driver)	18221141	18221089, 18221091, 18221115, 18221119, 1822114
4.	ADT Stack TOH (primitif dan driver)	18221089	18221089, 18221091, 18221115, 18221119, 1822114
5.	ADT List Linier (primitif dan driver)	18221119	18221089, 18221091, 18221115, 18221119, 1822114
6.	ADT Point (primitif dan driver)	18221119	18221089, 18221091, 18221115, 18221119, 1822114
7.	ADT Dungeon Tree (primitif dan driver)	18221115	18221089, 18221091, 18221115, 18221119, 1822114
8.	ADT Mesinkar_dungeon	18221115	18221089, 18221091, 18221115, 18221119, 1822114
9.	Scoreboard	18221141	18221089, 18221091,

			18221115, 18221119, 18221141
10..	Reset Scoreboard	18221141	18221089, 18221091, 18221115, 18221119, 18221141
11.	History	18221141	18221089, 18221091, 18221115, 18221119, 18221141
12.	Reset History	18221141	18221089, 18221091, 18221115, 18221119, 18221141
13.	Game Hangman	18221089, 18221091	18221089, 18221091, 18221115, 18221119, 18221141
14.	Game Tower of Hanoi	18221089	18221089, 18221091, 18221115, 18221119, 18221141
15.	Game Snake on Meteor	18221119	18221089, 18221091, 18221115, 18221119, 18221141
16.	ADT Tree (primitif dan driver)	18221115	18221115
17.	Game Alstrokedungeon (Game bonus)	18221115	18221115
18.	Update command Start, Load, Create Game, Delete Game, Play Game, dan Save	18221141	18221141
19.	Update Main Menu (Program Utama)	18221141	18221141

9 Lampiran

9.1 Deskripsi Tugas Besar 2

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`.

Dalam tugas besar ini dicantumkan beberapa tambahan fitur pada main menu, yaitu: SCOREBOARD yang menampilkan score pemain dalam sebuah game, RESET SCOREBOARD merupakan command yang digunakan untuk melakukan reset terhadap scoreboard permainan. HISTORY <n> merupakan command yang digunakan untuk melihat permainan apa saja yang telah dimainkan dari data yang sudah ada dari file konfigurasi (**Jika LOAD**) dan dari mulai Start Game juga, dengan <n> adalah jumlah permainan yang telah dimainkan yang ingin ditampilkan. Urutan teratas merupakan permainan terakhir yang dimainkan. Jika <n> lebih besar dari jumlah permainan yang telah dimainkan, akan menampilkan seluruh permainan yang telah dimainkan. RESET HISTORY merupakan command yang digunakan untuk menghapus semua history permainan yang dimainkan.

Selain itu juga ditambahkan beberapa game default yang dapat dimainkan, yaitu: HANGMAN (game tebak kata), SNAKE ON METEOR (game ular jaman dahulu dengan twist meteor dan obstacle), TOWER OF HANOI (numpuk-numpuk piringan dari yang paling besar ke paling kecil), dan GAME BONUS yaitu ALSTROKEDUNGEON.

9.2 Notulen Rapat

a. Rapat 18 November 2022

Rapat ini dilaksanakan di ITB. Pada rapat ini, kami membahas tentang pembagian tugas untuk masing masing bagian dari Tugas besar dengan hasil:

- Pembetulan command dasar: Kevin
- Pembuatan Scoreboard, reset Scoreboard, history dan reset history: Kevin
- Game Hangman: Ibnu
- Game Tower of Hanoi: Naufal
- Game Snake on Meteor: Natasya
- Game Bonus: Christoper
- PJ Laporan: Ibnu

Rapat ini dilaksanakan langsung setelah demo tugas besar 1, dengan tujuan agar tugas untuk masing masing bagian dapat langsung dikerjakan.

b. Rapat 25 November 2022

Rapat ini dilaksanakan di ITB setelah kelas, dengan tujuan review progress, dan perbaikan beberapa bug, serta diskusi mengenai beberapa hal yang perlu ditambahkan atas beberapa hasil revisi spesifikasi

c. Rapat 1 Desember 2022

Rapat ini dilaksanakan melalui zoom setelah asistensi ke-2. Pada rapat ini, kami membahas tentang progress lanjutan dari masing masing tugas. Rapat ini menghasilkan:

- Perbaikan pada beberapa program yang mengalami bug
- Revisi lanjutan laporan tugas besar

9.3 Log Activity Anggota Kelompok

Waktu	NIM	Keterangan
18/11/2022	18221089, 18221091, 18221115, 18221119, 18221141	Meet perdana, bagi tugas
22/11/2022	18221119	Membuat ADT point dan listlinier
23/11/2022	18221115	Membuat prototipe ADT dungeon_tree
24/11/2022	18221119	Revisi ADT point dan listlinier, membuat Snake on Meteor
24/11/2022	18221141	Membuat ADT stack, driver stack, command history, command reset history, dan update command play game serta config.txt
25/11/2022	18221119	Membuat dan revisi Snake on Meteor: generateFood, generateMeteor, generateSnake, generateObstacle
25/11/2022	18221089	Menambahkan Tower of Hanoi berserta ADT Stack TOH
25/11/2022	18221089, 18221091, 18221115, 18221119, 18221141	Rapat kedua membahas kendala dari masing-masing bagian
25/11/2022	18221141	Update command history
26/11/2022	18221089	Merevisi Diner Dash dan Tower of Hanoi
26/11/2022	18221115	Mulai membuat fungsi inti alstrokedungeon.c serta memperbaiki ADT dungeon_tree
26/11/2022	18221119	Membuat dan revisi Snake on Meteor: generateFood, generateMeteor, generateHead, growSnake, isInputValid, moveSnake, isCollide, invalidMove,

		displayField
27/11/2022	18221091	Membuat game hangman dasar yang belum disesuaikan
27/11/2022	18221119	Membuat dan revisi Snake on Meteor: generateFood, generateMeteor, generateHead, growSnake, isValidInput, moveSnake, isCollide, invalidMove, displayField
28/11/2022	18221089	Mengerjakan laporan
28/11/2022	18221119	Revisi Snake on Meteor: generateFood, generateMeteor, generateHead, growSnake, moveCmd, isCollide, invalidMove
29/11/2022	18221119	Revisi dan menambahkan prosedur pada Snake on Meteor: generateFood, generateMeteor, generateHead, growSnake, isValidInput, moveSnake, isCollideMeteor, collideMeteor, teleport, isCollideObstacle, collideObstacle, inputCmd, displayField, main.
29/11/2022	18221141	Membuat ADT listofmap dan savefile.txt, serta memperbarui ADT arraydin, ADT mesinfile, command start dan load, command help, command history, command reset history, command save, command play game, command create game, command skip game, dan command quit game
29/11/2022	18221115	Mengerjakan laporan
30/11/2022	18221089	Mengerjakan laporan, merevisi Hangman dan menambahkan ADT Set
30/11/2022	18221141	Update ADT listofmap, command reset history, command play game, game RNG, game diner dash, history, dan menambahkan command scoreboard dan command reset scoreboard
30/11/2022	18221119	Major debug snek on meteor. Menambahkan status dead/alive pada beberapa parameter prosedur/fungsi.
01		

01/12/2022	18221141	Update command help, command history, command queue game, command reset history, command reset scoreboard, command scoreboard, command skip game, command start dan load, game tower of hanoi, ADT listofmap, dan menggabungkan semua file kecuali game ke dalam sebuah file bernama console, serta membuat folder khusus untuk semua game ke dalam folder bernama GAME
01/12/2022	18221091	Mengerjakan laporan bagian test script, revisi beberapa typo, revisi game hangman
01/12/2022	18221089, 18221091, 18221115, 18221119, 18221141	Rapat ketiga membahas perbaikan pada program yang mengalami bug dan merevisi laporan
01/12/2022	18221119	Debug snake. Ubah display snake.
01/12/2022	18221089	Mengerjakan Laporan, membuat driver untuk ADT dan merevisi Hangman
02/12/2022	18221089, 18221091, 18221115, 18221119, 18221141	Mengerjakan laporan
02/12/2022	18221141	Mengerjakan laporan bagian Struktur Data ADT stack dan listofmap, membuat penjelasan bagian program utama dan mencantumkan algoritma menarik, kemudian melakukan update console, ADT listlinier, dan game diner dash, melakukan perbaikan pada kode agar terbaca di linux
02/12/2022	18221091	Melanjutkan pengerjaan laporan bagian test script, notulensi rapat kelompok, revisi minor bagian lainnya. revisi minor game hangman.
02/12/2022	18221119	Mengubah algoritma generate (generateFood, obstacle, meteor, head) pada snake on meteor supaya konsisten. Memperbaiki sistem crater (harusnya meteor panas) pada snake on meteor. Membuat driver listlinier dan point. Revisi ADT point. Mengerjakan laporan bagian ADT listlinier, point, data test snake on meteor, test script snake on meteor,

		penjelasan tambahan (fitur bonus permainan snake on meteor)
--	--	---



9.4 Form Asistensi Tugas Besar

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

No. Kelompok/Kelas : 13/K01
Nama Kelompok : Alstrokedead
Anggota Kelompok (Nama/NIM) :
1. Muhammad Naufal/18221089
2. Christopher Febrian Nugraha/18221115
3. Kevin Prayoga Abdinegara/18221141
4. Natasya Vercelly Harijadi/18221119
5. Ibnu Khairy Algifari/18221091

Asisten Pembimbing : Aditya Bimawan/13519064


Asistensi I

Tanggal : 24 November 2022	Catatan Asistensi: <ul style="list-style-type: none">- Bagian snake kalau misalnya pas mau numbuin ekor dia gak bisa ke kiri, atas, kanan, bawah karena ketutupan badannya apa yang harus dilakukan? → Tanya di QnA- Untuk pengerjaan kalau bisa spek wajibnya diselesaikan dulu dan fitur-fitur tambahan tingkat kesulitannya tidak sama. Kalau fitur ada yang udah selesai bantuin yang lain karena bobotnya sesuai dengan kesulitan fiturnya.- Untuk Tower of Hanoi kan bonusnya user masukin input berapa jumlah piringannya, apakah ada batasan?<ul style="list-style-type: none">- Tidak ada batasan karena ini fitur bonus- Apakah boleh membuat batasan sendiri? Karena kalau terlalu banyak misalnya 1000 itu tampilannya rusak<ul style="list-style-type: none">- Boleh buat batasan sendiri tapi jangan terlalu kecil, contohnya 6
Tempat : Zoom meeting	
Kehadiran Anggota Kelompok: 1. Muhammad Naufal/18221089 Tanda tangan 	
2. Christopher Febrian Nugraha/18221115 Tanda tangan 	

3. Kevin Prayoga Abdinegara/18221141 Tanda tangan
--



4. Natasya Vercelly Harijadi/18221119
Tanda Tangan








5. Ibnu Khairy Algifari/18221091
Tanda tangan




Tanda Tangan Asisten:



Asistensi II

Tanggal : 1 Desember 2022	Catatan Asistensi: <ul style="list-style-type: none"> - Ada masalah bagian scoreboard, kalau scorenya ≤ 0 itu gak masuk ke list of map. → Ternyata karena define Nilz 0, fixed dan sudah bisa masuk ke map → Ternyata masuk tp tetep gak ke-print/display pas di fungsi scoreboard :) sadge → Jawaban akhir: karena tidak ada kondisional yg ngecek apakah mapnya kosong atau tidak - Snek suka tiba2 berhenti di tengah → ternyata permasalahannya karena arah yang dituju = tempat meteor di-generate → sudah solved :) yey - Snek kalau kepala kena meteor/obstacle yang di-display kepalanya atau meteor/obstaclenya? → Meteor/obstaclenya
Tempat : Zoom meeting	
Kehadiran Anggota Kelompok:	
1. Muhammad Naufal/18221089 Tanda tangan 	
2. Christopher Febrian Nugraha/18221115 Tanda tangan 	
3. Kevin Prayoga Abdinegara/18221141 Tanda tangan 	
4. Natasya Vercelly Harijadi/18221119 Tanda Tangan 	
5. Ibnu Khairy Algifari/18221091 Tanda tangan 	

	Tanda Tangan Asisten: 
--	--