

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data


BNMO

Dipersiapkan oleh:

Kelompok 09 / K01

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

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-01-09</i>		28
		<i>Revisi</i>	-	-

Daftar Isi

Ringkasan	2
1 Penjelasan Tambahan Spesifikasi Tugas	3
1.1 <Word Upper>	3
2 Struktur Data (ADT)	3
2.1 ADT Array	4
2.2 ADT Mesin Karakter	5
2.3 ADT Mesin Kata	5
2.4 ADT Queue (Queue Game)	6
3 Program Utama	9
4 Algoritma-Algoritma Menarik	9
4.1 Initial Scan	9
5 Data Test	10
5.1 Data Test Start	10
5.2 Data Test Load	10
5.3 Data Test Save	11
5.4 Data Test Create Game	12
5.5 Data Test List Game	12
5.6 Data Test Delete Game	13
5.7 Data Test Queue Game	15
5.8 Data Test Play Game	16
5.9 Data Test Skip Game	17
5.10 Data Test Quit	18
5.11 Data Test Help	18
5.12 Data Test Command-command lain	19
6 Test Script	19
7 Pembagian Kerja dalam Kelompok	22
8 Lampiran	23
8.1 Deskripsi Tugas Besar	23
8.2 Notulen Rapat	24
8.3 Log Activity Anggota Kelompok	27

Ringkasan

BNMO adalah sebuah robot video game konsol permainan berbasis *command-line interface* (CLI) milik Indra dan Doni yang memiliki banyak bug dalam sistemnya sehingga membutuhkan perbaikan. BNMO ini memiliki beberapa game, seperti RNG (tebak angka acak) dan Diner DASH (permainan strategi bertema restoran). BNMO memiliki beberapa fitur, antara lain membaca file konfigurasi sistem (start), membaca file penyimpanan berbasis txt berdasarkan masukan pemain (load), menambahkan permainan baru pada daftar game (create game), menghapus permainan pada daftar game yang tersedia (delete game), pembuatan antrian permainan (queue game) berdasarkan daftar permainan yang tersedia (list game), kemampuan untuk melewati beberapa permainan di antrian permainan (skip game), memainkan permainan yang berada di antrian game paling awal (play game), bantuan untuk menampilkan daftar command yang dapat dipanggil (help), menyimpan state game pemain ke dalam suatu file (save), dan keluar dari program (quit).

Laporan mencakup ringkasan laporan, penjelasan tambahan mengenai spesifikasi tugas besar, struktur ADT yang digunakan, yaitu ADT Array, ADT Mesin Karakter, ADT Mesin Kata, dan ADT Queue, penjelasan mengenai algoritma program utama, algoritma-algoritma menarik, data test, test script, pembagian kerja dalam kelompok, serta lampiran yang terdiri dari deskripsi tugas besar 2, notulen rapat, dan log activity anggota kelompok.

Kesimpulan dari hasil tugas besar secara umum adalah melatih kami untuk memilih dan menggunakan struktur data yang cocok dan sesuai dengan permasalahan yang ada sehingga ketika dihadapkan pada suatu persoalan, kami dapat menentukan penggunaan struktur data yang tepat. Hasil tugas besar ini baik karena fitur-fitur yang ada pada BNMO sudah dapat berjalan dengan baik sesuai fungsinya.

1 Penjelasan Tambahan Spesifikasi Tugas

Pada tugas besar ini, kami membuat spesifikasi tambahan yang tidak dirincikan pada spesifikasi dasar tugas besar yang antara lain sebagai berikut.

1.1 <Word Upper>

Semua command dari pemain akan dilakukan upper word, kecuali nama file txt.

1.2 <LOAD>

Saat load dijalankan dan tidak menemukan file txt, maka bisa akan langsung memasukan command.

2 Struktur Data (ADT)

Kami menggunakan beberapa model ADT untuk menyelesaikan persoalan-persoalan yang spesifik untuk fungsi yang dispesifikasikan. Berikut ADT yang digunakan, primitif-primitif yang digunakan, dan alasan penggunaan disertai fungsi yang mengimplementasikan ADT tersebut.

2.1 ADT Array

- Definisi struktur data:

```
typedef struct
{
    ElType TI [IdxMax-IdxMin+1]; /* memori tempat penyimpan elemen (container)
    */
    int Neff; /* banyaknya elemen efektif */
} Tab;
```

- Primitif ADT Array:

1. Konstruktor

```
void MakeEmptyArray (Tab *T);
```

2. Selektor

```
int NbElmtArray (Tab T);
```

```
int MaxNbElArray (Tab T);
```

```
IdxType GetFirstIdxArray (Tab T);
```

```
IdxType GetLastIdxArray (Tab T);
```

```
ElType GetElmtArray (Tab T, IdxType i);
```

3. Edit/Add/Delete

```
void SetTabArray (Tab Tin, Tab *Tout);
```

```
void SetElArray (Tab *T, IdxType i, ElType v);
```

```
void SetNeffArray (Tab *T, IdxType N);
```

4. Prototype

```
boolean IsIdxValidArray (Tab T, IdxType i);
```

```
boolean IsIdxEffArray (Tab T, IdxType i);
```

```
boolean IsEmptyArray (Tab T);
```

```
boolean IsFullArray (Tab T);
```

- Persoalan yang diselesaikan
Dalam program terdapat penggunaan list array untuk menyimpan data game yang tersedia.
- Alasan memilih
ADT array mudah digunakan dan bisa mengakses index dan nilai dengan mudah
- Implementasi ADT

2.2 ADT Mesin Karakter

Definisi struktur data:

```
char CC;
boolean EOP;
boolean MBR;
```

```
static FILE *pita;
static FILE *command;
static int retval;
```

Primitif ADT Mesin Karakter:

```
void START(char * filename);
void ADV();
void STARTC();
void ADVC();
```

Alasan pemilihan

ADT ini digunakan untuk melakukan parsing command program dan pembacaan file konfigurasi ke dalam program.

2.3 ADT Mesin Kata

Definisi struktur data:

```
typedef struct {
    char TabWord[NMax]; /* container penyimpan kata, indeks yang dipakai */
    int Length;
} Word;
```

Primitif ADT Mesin Kata:

```
void IgnoreBlank()
```

```
void STARTWORD()
void ADVWORD()
void IgnoreBlanks()
```

Operasi terhadap Kata

```
void CopyCommand()
void CopyCommand2()
int LengthOfString(char *string)
void WordToString(Word kata, char *string)
Word StringToWord(char *string)
boolean IsWordSame(Word kata1, Word kata2)
Word WordUpper(Word kata)
boolean ValidTxt(Word txt)
```

Operasi Pembacaan Input dan Penulisan Output

```
void STARTCOMMAND()
void STARTCOMMAND2()
void PrintWord(Word kata)
void Scan(Word *c1, Word c*2)
void InitialScan(Word *c1, Word *c2)
```

Persoalan yang diselesaikan:

untuk mengambil inputan dari pengguna.

Alasan memilih :

Karena sudah ketentuan yang ada mengambil masukkan pengguna menggunakan stdin yang mana diajarkan dalam kuliah di materi mesin kata.

2.4 ADT Queue (Queue Game)

Definisi struktur data:

```
typedef Word ElType;
typedef struct {
    ElType buffer[CAPACITY];
    int idxHead;
    int idxTail;
} Queue;
```

Primitif ADT Mesin Queue:

1. Selektor

```
IDX__HEAD(q)
```

IDX_TAIL(q)
HEAD(q)
TAIL(q)

2. Konstruktor

void CreateQueue(Queue *q)

3. Prototype

boolean isEmpty(Queue q)
boolean isFull(Queue q)
int length(Queue q)

4. Add/delete

void enqueue(Queue *q, Eltype val)
void dequeue(Queue *q, Eltype *val)

5. Operasi lain

displayQueue(Queue q)

Persoalan yang diselesaikan:

ADT ini digunakan dalam permainan diner dash. Digunakan pada command command dasar untuk bermain seperti Enqueue makanan yang akan diproses, Dequeue makanan yang akan disajikan, serta mekanisme-mekanisma lain yang membutuhkan sifat queue pada permainan ini.

Alasan pemilihan :

ADT queue digunakan karena sifat-sifat queue, seperti Enqueue pada Tail dan Dequeue pada Head. ADT queue sangat cocok untuk merepresentasikan aturan antrian dan pemrosesan pesanan pada makanan di game ini.

Implementasi :

queue.c, queue.h

2.5 ADT Queue (Diner Dash)

Definisi struktur data:

```
typedef struct {  
    char Makanan[4];  
    int DurasiMasak;  
    int Ketahanan;  
    int Harga;  
} Pesanan;  
  
typedef struct {  
    Pesanan buffer[15];  
    int idxHead;
```

```

        int idxTail;
        int Count;
    } Queue2;

```

Primitif ADT Mesin Queue:

1. Selektor

```

int GetId(char str[])
int GetIdx(Queue2 q, char foodId[])

```

2. Konstruktor

```

void CreateQueue2(Queue2 *q)
void CreateRanPesanan(int noAntrian)

```

3. Predikat

```

boolean IsEmpty(Queue q)
boolean IsCook(char command[])
boolean IsServe(char command[])
boolean IsSkip(char command[])
boolean IsCommandValid(char command[], char foodId[], Queue2 qPesanan,
                        Queue2 qDiproses);
boolean IsDuaDigit(char str[]);
boolean IsFinish(Queue2 qPesanan, Queue2

```

4. Add/delete

```

void Enqueue(Queue2 *q, Pesanan p)
void Dequeue(Queue2 *q, Pesanan p)
void Eliminate(Queue2 *q, char foodId[])

```

5. Operasi lain

```

void Inisialisasi(Queue2 *qPesanan)
int GenerateHarga()
int GenerateDurasi()
void TampilkanPesanan()
void TampilkanDimasak()
void TampilkanSajian()
void Cook(Queue2 qPesanan, Queue2 *qDiproses, char foodId[])
void Serve(Queue *qPesanan, Queue2 *qDiproses, Queue2 *qSelesai,
            char foodId)
void Decrement(Queue2 *qDiproses)
void SettingCommand(Word kata, char command[], char foodId[])
void ProsesCommand(char command[], char foodId[], Queue2 *qPesanan,
                    Queue2 *qDiproses, Queue2 *qSelesai)

```

Persoalan yang diselesaikan:

ADT ini digunakan dalam permainan diner dash. Digunakan pada command command dasar untuk bermain seperti Enqueue makanan yang akan diproses, Dequeue makanan yang akan disajikan, serta mekanisme-mekanisma lain yang membutuhkan sifat queue pada permainan ini.

Alasan pemilihan :

ADT queue digunakan karena sifat-sifat queue, seperti Enqueue pada Tail dan Dequeue pada Head. ADT queue sangat cocok untuk merepresentasikan aturan antrian dan pemrosesan pesanan pada makanan di game ini.

Implementasi ADT:

Nama file: DD2.c

3 Program Utama

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

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

4 Algoritma-Algoritma Menarik

4.1 Initial Scan

```
void InitialScan(Word *c1, Word *c2) {  
    STARTCOMMAND();  
    if (!EndWord) {
```

```

    *c1 = WordUpper(CommandCC);
    ADVCOMMAND();
    if (!EndWord) {
        *c2 = CommandCC;
    }
}
}

```

digunakan untuk menerima input dari pengguna untuk menjalankan fungsi start dan load, dimana akan mengubah kata pertama sebagai huruf besar dan huruf kedua sebagai huruf biasa sebagai masukkan file txt.

5 Data Test

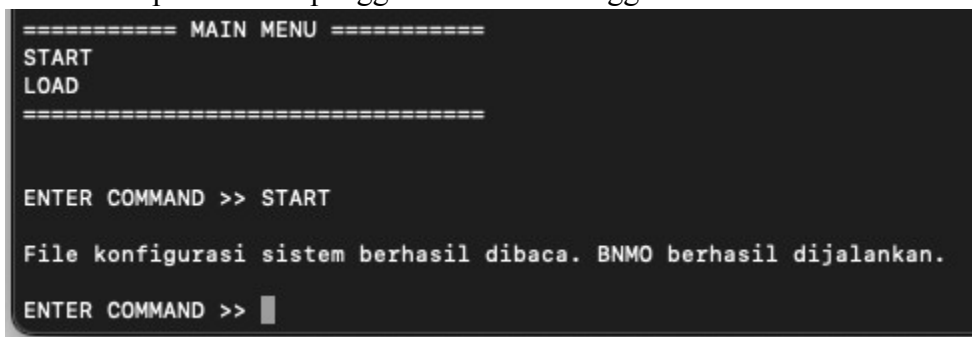
Untuk memastikan program dapat berjalan dengan lancar dan sesuai spesifikasi, dibutuhkan suatu testing. Berikut data test yang digunakan pada testing terhadap program kami beserta rincian fitur yang dites, output apa saja yang seharusnya diberikan, serta data testnya sendiri.

5.1 Data Test Start

Command start merupakan salah satu command yang digunakan pertama kali oleh pengguna pada program. Setelah menginputkan START, program akan membaca file konfigurasi default berisi list game yang dapat dimainkan. Jika program berhasil, maka program akan menampilkan pesan yang menyatakan bahwa program berhasil dibaca dan siap dijalankan.

Jika pengguna berkehendak menginputkan command START akan tetapi melakukan kesalahan, maka program akan mengeluarkan pesan kesalahan dan meminta pengguna untuk memasukkan inputan lagi.

Berikut tampilan ketika pengguna berhasil menggunakan command START



```

===== MAIN MENU =====
START
LOAD
=====

ENTER COMMAND >> START

File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan.

ENTER COMMAND >> 

```

5.2 Data Test Load

Jika pengguna memiliki file lama yang ingin diakses untuk melanjutkan progress game, pengguna dapat memasukkan command LOAD untuk melanjutkan progress game tersebut. Command LOAD memiliki format: LOAD <savefile.txt>. Ketika inputan tersebut dimasukkan, maka program akan membaca file inputan, kemudian pengguna dapat mengakses file tersebut.

Jika pengguna salah menginput command LOAD karena kesalahan dalam penginputan, maka program akan mengirimkan pesan kesalahan dan program meminta pengguna untuk melakukan input lagi sampai pengguna menginputkan command dengan benar.

Jika pengguna salah menginput file yang akan dibaca karena salah penginputan, maka program akan mengirimkan pesan bahwa file yang dimasukkan tidak ditemukan. Kemudian program akan meminta pengguna untuk melakukan input lagi, sampai program menemukan file yang tersedia.

Asumsi kami untuk kasus ini, adalah bahwa isi file yang akan dibaca oleh program selalu valid atau memiliki format yang sesuai dengan spesifikasi sehingga kami tidak membuat penanganan kondisi error untuk file yang tidak sesuai.

Berikut tampilan program ketika membaca file load dengan format dan inputan yang sesuai.

```
===== MAIN MENU =====  
START  
LOAD  
=====
```

ENTER COMMAND >> LOAD konfigurasi.txt

konfigurasi.txt berhasil dibaca. BNMO berhasil dijalankan.

ENTER COMMAND >> █

Berikut tampilan program ketika membaca command LOAD yang tidak sesuai

```
ENTER COMMAND >> load gaada.txt  
Filename gaada.txt tidak ditemukan.  
ENTER COMMAND >>
```

5.3 Data Test Save

SAVE merupakan command yang digunakan untuk menyimpan state game permainan saat ini ke dalam suatu file. Command SAVE memiliki satu argumen yang merepresentasikan nama file yang akan disimpan pada disk.

Berikut tampilan ketika pengguna berhasil

```
ENTER COMMAND >> save
Masukan tidak valid. Silahkan masukkan nama file .txt yang akan disimpan.

ENTER COMMAND >> save aaa
Masukan tidak valid. Silahkan masukkan nama file .txt yang akan disimpan.

ENTER COMMAND >> save file1.txt
file1.txt berhasil disimpan.
```

5.4 Data Test Create Game

CREATE GAME merupakan command yang digunakan untuk menambahkan game baru pada daftar game. Game custom / buatan pengguna memiliki keterbatasan, yaitu game hanya bersifat dummy. Karena buatan pengguna merupakan dummy, maka pengguna tidak dapat membuat game sesungguhnya. Inputan game pada command ini hanya akan mengisi array of Game dan judul game akan ditampilkan pada daftar game tanpa bisa dimainkan.

Nama game yang akan di input oleh pengguna tidak memiliki batasan, sehingga secara teknis tidak ada kondisi khusus untuk mengatasi kesalahan pada CREATE GAME. Jika pengguna salah menginputkan nama game sesuai dengan keinginannya, pengguna harus melakukan command DELETE pada game yang tidak sesuai. Kemudian, pengguna melakukan CREATE GAME kembali dan meninputkan nama game yang sesuai.

Berikut adalah tampilan ketika pengguna berhasil menggunakan command CREATE GAME.

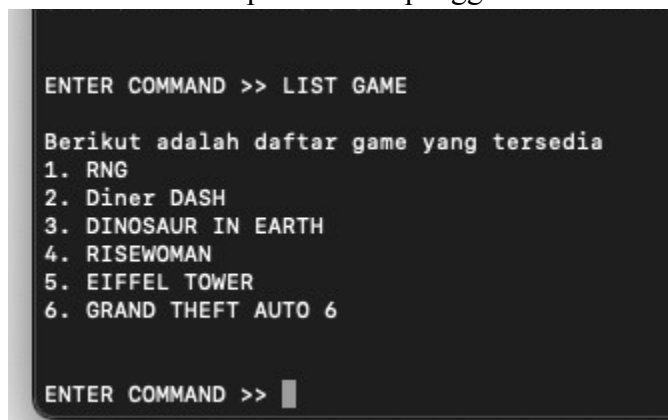
```
ENTER COMMAND >> create game
Masukkan nama game yang akan ditambahkan >> Grand Theft Auto 6
Game GRAND THEFT AUTO 6 berhasil ditambahkan.

ENTER COMMAND >> █
```

5.5 Data Test List Game

LIST GAME merupakan command yang digunakan oleh pengguna untuk menampilkan daftar game saat ini pada sistem. Tidak ada kasus khusus yang menangani kesalahan pada LIST GAME karena keterbatasan command ini hanya mencetak array of Game yang tidak pernah kosong.

Berikut adalah tampilan ketika pengguna berhasil menggunakan command LIST GAME.



```
ENTER COMMAND >> LIST GAME

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. GRAND THEFT AUTO 6

ENTER COMMAND >> █
```

5.6 Data Test Delete Game

DELET GAME merupakan command yang digunakan untuk menghapus sebuah game dari daftar game. Mekanisme penghapusan adalah, pertama, program akan menampilkan list game yang tersedia. Masing-masing game diberikan nomor sebagai parameter penghapusan game. Setelah daftar game ditampilkan, maka pengguna memasukkan nomor game yang akan dihapus. Kemudian program menghapus game dengan nomor tersebut. Terdapat beberapa aturan dalam melakukan penghapusan game. Karena aturan tersebut, maka terdapat beberapa kondisi untuk menangani error yang mungkin.

Pertama, game yang dapat dihapus hanya game yang dibuat secara custom oleh pengguna. Jika pengguna mencoba menghapus game yang di define oleh sistem, maka program akan mengeluarkan pesan kesalahan dan game gagal dihapus. Game yang didefinisi oleh sistem adalah lima game pertama, yakni : RNG, Diner DASH, DINOSAUR IN EARTH, RISEWOMAN, dan EIFFEL TOWER.

Kedua, jika pengguna memasukkan nomor game yang tidak ada pada list game, maka program akan mengeluarkan pesan kesalahan. Program gagal menghapus game.

Berikut adalah tampilan ketika pengguna berhasil menggunakan command DELETE GAME.

```
ENTER COMMAND >> delete game

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. DIABLO IMMORTAL
7. GRAND THEFT AUTO VI

Masukkan nomor game yang akan dihapus >> 6

Game berhasil dihapus.

ENTER COMMAND >> list game

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. GRAND THEFT AUTO VI

ENTER COMMAND >> █
```

Berikut adalah tampilan ketika pengguna mencoba menghapus game yang tidak ada pada daftar game.

```
ENTER COMMAND >> delete game

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. GRAND THEFT AUTO VI

Masukkan nomor game yang akan dihapus >> 10

Game gagal dihapus.

ENTER COMMAND >> list game

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. GRAND THEFT AUTO VI

ENTER COMMAND >> █
```

Berikut adalah tampilan ketika pengguna mencoba menghapus game yang didefinisikan secara default oleh sistem.

```
ENTER COMMAND >> delete game

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. GRAND THEFT AUTO VI

Masukkan nomor game yang akan dihapus >> 3

Game gagal dihapus.

ENTER COMMAND >> list game

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. GRAND THEFT AUTO VI

ENTER COMMAND >> |
```

5.7 Data Test Queue Game

QUEUE GAME merupakan command yang digunakan untuk mendaftarkan permainan ke dalam list Game. List dalam queue akan hilang ketika pengguna menjalankan command QUIT. Oleh karena itu, jika pengguna ingin menyimpan progressnya, pengguna harus menggunakan command SAVE agar queue game yang dibuatnya tidak dihapus.

Ketika pengguna menjalankan command QUEUE GAME, program akan menampilkan daftar game yang sedang berada di dalam antrian. Setelah itu, program menampilkan daftar game yang tersedia, yang dapat dimasukkan ke dalam antrian game. Setiap game diberikan nomor yang akan menjadi parameter penambahan game pada antrian game. Program dapat memasukkan permainan yang sama berulang kali ke dalam antrian game.

Jika pengguna mencoba menambahkan game yang tidak tersedia pada daftar game, maka program akan mengirimkan pesan kesalahan. Game gagal dimasukkan ke dalam antrian game.

Berikut tampilan ketika pengguna berhasil menjalankan command QUEUE GAME.

```

Berikut adalah daftar antrian game-mu
1. RNG
2. Diner DASH
Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMANIN EARTH
5. EIFFEL TOWEREARTH

Nomor game yang mau ditambahkan ke antrian >> 1

Game berhasil ditambahkan ke dalam daftar antrian.

```

Berikut tampilan ketika pengguna mencoba menambahkan game yang tidak ada pada list game.

5.8 Data Test Play Game

PLAY GAME merupakan command yang digunakan untuk memainkan permainan. Game yang dimainkan adalah game dengan urutan antrian pertama game. Game yang dapat dimainkan pengguna hanyalah: RNG, Diner DASH, dan game bonus. Game default selain RNG dan Diner DASH, ketika dimainkan, akan mengirimkan pesan bahwa game tidak dapat dimainkan.

Game custom yang dibuat menggunakan CREATE GAME pada program ini akan langsung selesai dan masuk ke tahap game over dengan skor akhir berupa integer random.

Kami tidak membuat kasus khusus ketika mencoba memainkan game dengan antrian game yang kosong. Sehingga, ketika pengguna mencoba menggunakan command PLAY GAME dengan antrian game yang kosong, program tidak melakukan apapun dan kembali meminta inputan pengguna.

Berikut adalah tampilan ketika pengguna memainkan game default selain RNG dan Diner DASH.

```

ENTER COMMAND >> play game

Berikut adalah daftar antrian game-mu
1. RNG

Loading RNG ...
RNG telah dimulai. Uji keberuntungan Anda dengan menebak angka X.

Tebakan >> 50
Lebih kecil

```

Berikut adalah tampilan ketika pengguna memainkan game custom.


```

ENTER COMMAND >> PLAY GAME

Berikut adalah daftar antrian game-mu
1. GRAND THEFT AUTO 6

Loading GRAND THEFT AUTO 6 ...

GAME OVER

Skor Anda = 41

ENTER COMMAND >> █

```

Berikut adalah tampilan ketika pengguna mencoba melakukan command PLAY GAME default selain RNG dan Diner Dash.

```

ENTER COMMAND >> skipgame 1

Berikut adalah daftar antrian game-mu
1. Diner DASH
2. RISEWOMAN

Berikut adalah daftar antrian game-mu
1. RISEWOMAN

Game RISEWOMAN masih dalam maintenance, belum dapat dimainkan.
Silahkan pilih game lain.

```

5.9 Data Test Skip Game

Pada fungsi skipgame, kita akan melewati permainan yang akan dimainkan pada antrian sebanyak n kali tergantung dari apa yang kita masukkan pada command. Terdapat 3 kondisi yang bisa di-input pada fungsi skipgame.

```

ENTER COMMAND : SKIPGAME 2
Berikut adalah daftar game-mu
1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN
Loading RISEWOMAN ...

```

pada hasil testing diatas dapat kita lihat jika kita memasukkan command “SKIPGAME 2” maka fungsi akan melewati 2 atrian game dan langsung memainkan game ketiga yaitu RISEWOMAN.

```
ENTER COMMAND : SKIPGAME b
Berikut adalah daftar game-mu
1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN
Masukkan untuk melakukan SKIPGAME salah, dimohon untuk mengganti masukkan game yang ingin di input dengan format angka.
```

Sepertinya yang bisa kita lihat diatas jika kita memasukan input tidak dikenal yaitu “b” maka masukkan akan dinyatakan tidak valid.

```
Berikut adalah daftar game-mu
1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN
Tidak ada permainan lagi dalam daftar game-mu.
```

Jika memasukkan “n” yang lebih dari banyaknya game pada antrian, maka tidak akan ada lagi permainan dalam daftar game.

5.10 Data Test Quit

QUIT merupakan command yang digunakan pengguna untuk keluar dari program. Tidak ada kondisi penanganan khusus untuk command QUIT.

Berikut tampilan ketika pengguna menggunakan command QUIT.

```
ENTER COMMAND >> quit dong

Command tidak dikenali. Silahkan masukkan command yang valid.

ENTER COMMAND >> quit

Anda keluar dari game BNMO.
Bye bye ...
```

5.11 Data Test Help

HELP merupakan command yang digunakan untuk menampilkan penjelasan mengenai command-command yang tersedia pada program ini. Tidak ada kondisi penanganan khusus untuk command HELP.

Berikut tampilan ketika pengguna menggunakan command HELP.

```

ENTER COMMAND >> HELP

===== HELP =====

START          : Membaca file konfigurasi default
LOAD <file>     : Membaca file yang berisi daftar game
SAVE <file>     : Menyimpan state game pemain saat ini ke suatu file
CREATE GAME    : Menambahkan game baru pada daftar game
LIST GAME      : Menampilkan daftar game
DELETE GAME    : Menghapus game dari daftar game
QUEUE GAME     : Mendaftarkan permainan ke dalam daftar antrian game
PLAY GAME      : Memainkan game pertama pada antrian game
SKIPGAME <n>   : Melewatkan permainan sebanyak n
QUIT           : Keluar dari program

ENTER COMMAND >> █

```

5.12 Data Test Command-command lain

Command-command selain yang disebutkan di atas tidak valid. Ketika pengguna menginputkan command lain, maka program akan menampilkan pesan kesalahan dan pengguna dapat menginputkan program lain yang valid.

Berikut adalah tampilan ketika pengguna menginputkan command lain.

```

ENTER COMMAND >> quit dong

Command tidak dikenali. Silahkan masukkan command yang valid.

```

6 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Start	memeriksa apakah start	saat main program baru berjalan, memasukkan input "start"	Data Test 1	cofiguration berhasil terbaca	sesuai hasil yang diharapkan

		berhasil dijalankan				
2	Load	memeriksa apakah load berhasil	saat main program baru berjalan. memasukan input "load"	Data Test 2	save file berhasil terbaca	sesuai hasil yang diharapkan
3	Create Game	memeriksa apakah game berhasil di create	memasukkan command create game lalu menulis game apa yang ingin dibuat	Data Test 3	game berhasil dibuat dan masuk kedalam list game	sesuai hasil yang diharapkan
4	List Game	apakah daftar list berhasil ditampilkan	memasukkan command list game	Data Test 4	list game berhasil ditampilkan	sesuai hasil yang diharapkan
5	Delete Game	memeriksa apakah game default tidak bisa di delete	memasukkan command delete game lalu memasukkan input 2	Data Test 5	game tidak bisa di delete	sesuai hasil yang diharapkan sesuai hasil yang diharapkan
		memeriksa apakah ga yang sudah kita buat bisa di delete	memasukkan command delete game lalu memasukkan input 7		game berhasil di delete	
6	Queue Game	memeriksa saat kita memasukan inputan yang tidak valid pada fungsi queue game	masuk fungsi queue game dengan menginput command "queue game". lalu memasukkan input a	Data Test 6	game tidak berhasil ditambahkan karena format salah	sesuai hasil yang diharapkan
		memeriksa saat kita memasukka				

		n fungsi yang valid	masuk fungsi queue game dengan menginput command “queue game”. lalu memasukkan input 2		game berhasil ditambahkan	sesuai hasil yang diharapkan
7	Play Game	memeriksa apakah game dinner dash dan rng bisa dimainkan	memasukkan inputan play game pada command di main program namu dengan kondisi game rng atau dinner dash terletak pertama pada antrian game	Data Test 7	game berhasil dijalankan	sesuai hasil yang diharapkan
		memeriksa output saat kita memainkan game default selain rng dan dinner dash	memasukkan inputan play game pada command di main program namu dengan kondisi game default selain rng dan dinner dash terletak pertama pada antrian game		game maintenance	
		memeriksa apa yang terjadi saat kita memainkan game yang baru di create	memasukkan inputan play game pada command di main program namu dengan kondisi game yang telah kita create terletak pertama pada antrian game			
8	Skip Game	memeriksa masukan yang tidak valid	memasukkan inputan SKIPGAME b pada command main program	Data Test 8	mengirim output yang menjelaskan bahwa format pada masukan salah	sesuai hasil yang diharapkan

		memeriksa masukkan saat lebih besar dari antrian game	memasukkan inputan SKIPGAME 7 pada command main program		mengeluarkan output bahwa tidak ada lagi game dalam antrian	sesuai hasil yang diharapkan
		memeriksa jika masukkan lebih kecil dari banyak antrian	memasukkan inputan SKIPGAME 2 pada command main program		memainkan game setelah di skip 2 kali dalam antrian game	sesuai hasil yang diharapkan
9	Quit	memeriksa apakah ada output yang tertulis saat keluar game	memasukkan inputan quit pada command main program	Data Test 9	keluar out saat keluar mainprogram	sesuai hasil yang diharapkan
10	Help	memeriksa apakah output pesan pada fungsi help bisa ditampilkan	memasukkan command help pada main program	Data Test 10	output informasi yang tertulis pada fungsi help telah ditampilkan	sesuai hasil yang diharapkan

7 Pembagian Kerja dalam Kelompok

No.	Fitur/ADT	NIM Coder	NIM Tester
1.	Program utama (main.c)	18221049, 18221073, 18221085, 18221105, 18221149	18221049, 18221073, 18221085, 18221105, 18221149
2.	ADT Array	18221049	18221049
3.	ADT Mesin Karakter dan Mesin Kata	18221049, 18221073	18221049, 18221073
4.	ADT Queue	18221105, 18221149	18221105, 18221149

5.	Fitur Start	18221049	18221049
6.	Fitur Load	18221049	18221049
7.	Fitur Save	18221049	18221149
7.	Fitur CreateGame	18221073	18221073
8.	Fitur ListGame	18221149	18221149
9.	Fitur DeleteGame	18221073	18221073, 18221149
10.	Fitur QueueGame	18221085, 18221073	18221085, 18221149
11.	Fitur PlayGame	18221149	18221149
12.	Fitur SkipGame	18221085	18221085, 18221149
13.	Fitur SkorGame	18221073	18221073
14.	Fitur Help	18221105	18221105
15.	Fitur Quit	18221085	18221085
16.	Permainan RNG	18221073	18221073
17.	Permainan Diner DASH	18221105	18221105

8 Lampiran

8.1 Deskripsi Tugas Besar

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.

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`

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







8.2 Notulen Rapat

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

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







Asisten Pembimbing : Afif Fahreza

Asistensi I

Tanggal : 4 November 2022	Catatan Asistensi:
Tempat : Zoom	
Kehadiran Anggota Kelompok: <div style="text-align: center;"> <p>1 18221049</p>  <p>2 18221073</p>  <p>3 18221085</p>  <p>4 18221105</p>  <p>5 18221149</p>  </div>	
	Tanda Tangan Asisten: 

- ElType array char atau apa? Untuk di mesin kata, tambah kata to string, di parsing dari kata ke string masukin ke eltype
- Kata disimpan masih di wadah adt, kalau string dikeluarin terus bisa dipakai sama yang lain
- Cara kerja stdin? Pakai mesin kata, mesin kata bisa dari stdin, bisa dari file, kalau loadambilnya dari file, bisa bikin copas 1 adt lagi kasih nama mesin kata command pakai stdin ada di prak 5, harus buat compare cek satu” karakter, dari panjang stringnya, ambil satu” karakternya, while sampai ketemu ‘\0’, sampai ketemu yang ga sama, langsung di matiin close
- Kalau start load file txt default yg selalu sama, kalau load beda
- Kalau load, file namanya tidak ada, harus cek
- Save kalau tidak ada nama filenya harus buat, kalau sudah ada di rewrite
- Create game cek nama apakah game yang mau ditambahkan sudah ada di daftar game

Asistensi II

Tanggal : 10 November 2022	Catatan Asistensi: <ul style="list-style-type: none"> - Main program masih belum tinggal buat manggil-manggil fungsi, tapi udah bisa load. - Kalau load filenamenya gaketemu, segmentation of fault, sebelum segmentation of fault, print kalau file ga ditemukan. - Pakai makefile atau .shell, kalau .shell copy commandnya ke .shell, dijalaninnya pakai bash, kalau ada git bash jalan, kalau gaada gabisa jalan. - Penilaiannya jalan tidak programnya, aman tidak. - Buat compile.sh, #!/bin/bash, buat test ketik echo "test s". Ke terminal, .\compile.sh, git for windows. Ganti echo "test s" dengan gcc -o, contoh gcc -o tes ./src/DinerDash/diner_dash.c ./src/DinerDash/DD2.c ./src/DinerDash/mesinkarcommand.c ./src/DinerDash/mesinkatacommand.c
Tempat : Zoom	
Kehadiran Anggota Kelompok: <div style="text-align: center;"> <p>1 18221049</p>  <p>2 18221073</p>  <p>3 18221085</p>  <p>4 18221105</p>  <p>5 18221149</p>  </div>	
	Tanda Tangan Asisten: 

8.3 Log Activity Anggota Kelompok

NIM	Nama	Log Act
18221049	Silvester Kresna W.P.P.	30/10/2022 : Membuat ADT array, mesinkata dan mesinkarakter 08/11/2022 : Membuat Load, Save, Start dan debugging serta tester 09/11/2022 : Membuat main awal 10/11/2022 : memperbaiki main 11/11/2022 : Melengkapi main, Membuat driver ADT
18221073	Jessica	08/11/2022 : Membuat RNG dan SkorGame 09/11/2022 : Membuat ADT MesinKarakter dan MesinKata untuk command 10/11/2022 : Memperbaiki dan menambah fungsi/prosedur pada MesinKarakter dan MesinKata untuk command, membuat CreateGame, membuat DeleteGame, memperbaiki QueueGame, memperbaiki ListGame 11/11/2022 : Membuat main program, memperbaiki SkipGame, membuat laporan
18221085	Nalendro N.U.A.R.F.	08/11/2022 : membuat ADT queue 09/11/2022 : selesai membuat skipgame dan queuegame 10/11/2022 : membuat main program, tes banyak fungsi, 11/11/2022 : membuat laporan, debug main program
18221105	Muhammad Aliefnaufal Permana	08/11/2022 : Menyelesaikan Diner Dash 10/11/2022 : Membuat Help 10/11/2022 : Membantu membuat main 11/11/2022 : Debug Diner Dash 11/11/2022 : Membantu membuat laporan
18221149	Rayhan Nugraha Putra	08/11/2022 : Membuat Fungsi ListGame dan PlayGame 09/11/2022 : Membuat 11/11/2022 : Tes fungsi keseluruhan (Debugging)

		11/11/2022 : Mengubah ADT Queue, Perbaiki fungsi-fungsi 11/11/2022 : Menulis laporan
--	--	--