

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data


BNMO

Dipersiapkan oleh:

KELOMPOK 05

Ahmad Rizki	18221071
Nadine Aliya Putri	18221081
Clara Alrosa Fernanda Sinaga	18221099
Pramaditya Fajri Migfar	18221111
Naura Valda Prameswari	18221173

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-05</i>		<i>37</i>
		<i>Revisi</i>	<i>00</i>	<i>11/11/2022</i>

Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	5
2.1 RNG	5
2.2 Bonus Game (“STI Mencari Jodoh”)	5
3 Struktur Data (ADT)	6
3.1 ADT arrayGame	6
3.2 ADT mesin_input	6
3.3 ADT mesin_kalimat	7
3.4 ADT mesin_kar	7
3.5 ADT queueDinerDash	8
3.6 ADT queueGame	8
4 Program Utama	9
4.1 Main Menu	9
4.2 Pembacaan dan Inisialisasi File Konfigurasi	9
4.3 Pembacaan Command	10
4.4 Command “CREATE GAME”	10
4.5 Command “LIST GAME”	10
4.6 Command “DELETE GAME”	10
4.7 Command “QUEUE GAME”	10
4.8 Command “PLAY GAME”	11
4.9 Command “SKIP GAME <n>”	11
4.10 Command “HELP”	11
4.11 Command “SAVE <namafile>”	11
4.12 Command “QUIT”	11
4.13 Game “RNG”	12
4.14 Game “Diner Dash”	12
4.15 Game “STI Mencari Jodoh”	12
5 Algoritma-Algoritma Menarik	13
5.1 Algoritma scan parser	13
5.2 Mesin Input	13
6 Data Test	14

6.1 Data Test 1	14
6.2 Data Test 2	14
6.3 Data Test 3	14
6.4 Data Test 4	15
6.5 Data Test 5	15
6.6 Data Test 6	15
6.7 Data Test 7	16
6.8 Data Test 8	16
6.9 Data Test 9	16
6.10 Data Test 10	17
6.11 Data Test 11	17
6.12 Data Test 12	18
6.13 Data Test 13	19
6.14 Data Test 14	19
6.15 Data Test 15	20
6.16 Data Test 16	21
6.17 Data Test 17	21
6.18 Data Test 18	22
6.19 Data Test 19	22
6.20 Data Test 20	23
6.21 Data Test 21	23
6.22 Data Test 22	23
7 Test Script	25
8 Pembagian Kerja dalam Kelompok	27
9 Lampiran	28
9.1 Deskripsi Tugas Besar	28
9.2 Notulen Rapat	31
9.3 Log Activity Anggota Kelompok	36

1 Ringkasan

Indra dan Doni yang merupakan pemilik dari robot video game console Binomo sedang kebingungan dikarenakan robot mereka yang sudah diperbaiki dua bulan yang lalu mengalami kerusakan yang tambah parah. Karena mereka sudah tidak tahu apa yang harus dilakukan lagi, Indra dan Doni akhirnya meminta bantuan kepada 5 temannya yang berada di jurusan Sistem Teknologi dan Informasi untuk memprogram ulang robot video game console kesayangannya agar dapat berfungsi seperti semula.

BNMO merupakan sebuah robot game console yang dapat menjalankan berbagai jenis permainan. Binomo diprogram menggunakan bahasa C dan merupakan permainan yang berbasis CLI (command-line interface). Program Binomo ini menggunakan berbagai struktur data terkait list, array, mesin karakter, mesin kata, serta queue agar dapat memenuhi kebutuhan program Binomo. Alur program Binomo dimulai dari penulisan command *start* yang akan memulai program Binomo. Setelah itu, masukan command *load* agar dapat memasukan nama file yang akan dibuka. Selain itu, kita juga dapat memainkan game, menambahkan game, menghapus game, serta mengurutkan game yang akan dimainkan sesuai dengan command yang diisi. Program akan berakhir ketika command *quit* dimasukkan. Sebelum itu, kita dapat menyimpan state game saat ini ke dalam suatu file dengan memasukkan command *save*.

Tugas besar ini bertujuan untuk menambah wawasan mahasiswa tentang *Abstract Data Type* serta meningkatkan pengetahuan mahasiswa tentang bahasa pemrograman C. Tugas besar ini juga sangat bermanfaat karena dapat melatih pola pikir mahasiswa saat bertemu dengan persoalan yang lumayan kompleks. Oleh karena itu, pemberian tugas besar ini banyak memberikan manfaat bagi mahasiswa.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 RNG

Spesifikasi Tugas meminta ditampilkan skor pemain sesuai dengan kecepatan pemain menebak angka dengan formula yang dibebaskan. Untuk ini, kami menetapkan batasan pemain menebak sebanyak 10 kali. Di awal permainan, pemain memiliki 1000 poin dan untuk tebakan yang tidak tepat poin tersebut akan dikurang 100 poin. Hal ini membuat poin tertinggi yang bisa diperoleh pemain adalah 1000 dan poin terendahnya adalah 0.

2.2 Bonus Game (“STI Mencari Jodoh”)

Di awal permainan, pemain akan diminta memasukkan nama dan jenis kelaminnya. Bila, pemain memasukkan input jenis kelamin selain “cowo” dan “cewe” maka, prosedur akan menampilkan pesan kesalahan dan meminta pemain memasukkan jenis kelamin yang tepat.

Jika pemain memasukkan jenis kelamin “cewe” maka, prosedur akan memanggil fungsi STARTCEWE. Jika pemain memasukkan jenis kelamin “cowo” maka, prosedur akan memanggil fungsi STARTCOWO. Kedua fungsi akan membaca file berisi nama-nama lawan jenis dari input misal, fungsi STARTCEWE akan membaca suatu file bernama jodohcewe.txt yang berisi kumpulan nama laki-laki dan begitu pula sebaliknya untuk fungsi STARTCOWO. Lalu dengan menggunakan *random number generator* bawaan C, prosedur akan *generate* nomor acak yang kemudian dijadikan indeks akses list nama lawan jenis tersebut. Terakhir, prosedur akan menampilkan nama lawan jenis sesuai indeks hasil *random number* tersebut sebagai jodoh sang pemain.

3 Struktur Data (ADT)

BNMO merupakan suatu robot game console yang dapat menjalankan permainan. BNMO merupakan robot yang cukup kompleks dari segi algoritmanya, karena diperlukan beberapa ADT agar robot game console BNMO dapat berjalan dengan lancar. Dalam algoritmanya, BNMO menggunakan 6 jenis ADT yang berbeda, yaitu ADT arrayGame, ADT mesin_input, ADT mesin_kalimat, ADT mesin_kar, ADT queueDinerDash, serta ADT queueGame.

3.1 ADT arrayGame

- Sketsa struktur data
Pada ADT arrayGame, terdapat struktur data TabGame yang menyimpan nilai TG berupa array of ElTypeG dan Neff yang merupakan banyaknya elemen efektif. Type dari ElTypeG merupakan sebuah Kalimat, sedangkan Neff merupakan sebuah integer. Prototype pada ADT berupa konstruktor untuk membuat arrayGame yang kosong, lalu terdapat selektor untuk mengakses index pertama, terakhir, dan element game. Ada juga primitif lain yaitu selektor set yang berguna untuk mengeset nilai TabGame, elemen, serta Neff. Terakhir, terdapat primitif untuk menentukan apakah indeks yang dimasukan valid, efektif, apakah sebuah array kosong dan penuh.
- Persoalan yang diselesaikan
ADT arrayGame menyelesaikan persoalan tentang bagaimana kumpulan game yang tersedia di BNMO dapat disimpan, ditambahkan sebuah game, diakses, dihapus, dan ditampilkan ke user.
- Alasan pemilihan
Alasan pemilihan arrayGame (array of kalimat) adalah untuk menyimpan hasil pembacaan file config atau savefile ke program, sehingga kumpulan list dapat diberi perlakuan sesuai dengan sistem game pada umumnya.
- Implementasi
Diimplementasikan sebagai ADT arrayGame dengan nama *file header* “arrayGame.h”

3.2 ADT mesin_input

- Sketsa struktur data
pada ADT mesin_input, terdapat struktur data Input yang menyimpan nilai TabInput berupa array of character dan Length yang merupakan panjang dari input. Prototype yang digunakan adalah STARTINPUT yang menandakan mesin siap dioperasikan, dan ADVINPUT yang menandakan pita dimajukan satu karakter. Selain itu, terdapat primitif IgnoreBlank, SalinInput, InputToString, InputtoInt, CopyInput, strlen, dan strconcat.
- Persoalan yang diselesaikan
ADT mesin_input ini berguna untuk mengoperasikan pita karakter yang diinput oleh user. ADT mesin input memanfaatkan mesin karakter untuk membaca karakter yang diinput

oleh user dengan stdin. Sehingga mesin input dapat memfasilitasi prosedur scan parser, scan string, hingga scan integer.

- Alasan pemilihan
Pemilihan ADT didasarkan pada kebutuhan scan pita karakter yang diinput oleh user. Dengan adanya ADT mesin_input kita dapat membaca pita karakter yang diinput user sebagai kata dan kalimat, selain itu bisa disimpan dalam string.
- Implementasi
Diimplementasikan sebagai ADT mesin_input dengan nama *file header* “mesin_input.h”

3.3 ADT mesin_kalimat

- Sketsa struktur data
Pada ADT mesin_kalimat, terdapat struktur data Kalimat yang menyimpan nilai *TabKalimat* berupa *array of character* serta *Length* yang berupa integer. Prototype yang digunakan adalah *STARTKALIMAT* dan *STARTKALIMATFILE* yang menandakan bahwa mesin siap untuk dioperasikan serta *ADV KALIMAT* dan *ADV KALIMATFILE* yang menandakan pita dimajukan. Selain itu, terdapat juga primitif *IgnoreNewLine*, *SalinKalimat*, *SalinKalimatFile*, *KalimatToString*, serta *strToInt*.
- Persoalan yang diselesaikan
ADT mesin_kalimat menyelesaikan persoalan dalam memperlakukan kumpulan game pada BNMO sebagai suatu antrian.
- Alasan pemilihan
Alasan pemilihan mesin kalimat adalah karena game yang tersimpan pada data config dan savefile merupakan kalimat yang diakhiri dengan *new line*.
- Implementasi
Diimplementasikan sebagai ADT mesin_kalimat dengan nama *file header* “mesin_kalimat.h”

3.4 ADT mesin_kar

- Sketsa struktur data
Pada ADT mesin_kar, terdapat nilai *extern EOP* yang bernilai *boolean*, serta nilai *extern Current Character (CC)* yang bernilai *character*. Prototype yang digunakan adalah *START* yang menandakan bahwa mesin siap untuk dioperasikan, serta *ADV* yang menandakan bahwa pita dimajukan satu karakter, terdapat juga prototype *STARTFILE* dan *ADVFILE*.
- Persoalan yang diselesaikan
ADT mesin karakter menyelesaikan persoalan input string, membaca karakter dalam file, serta menjadi dasar dalam membuat mesin kalimat dan mesin input.

- Alasan pemilihan: Pengimplementasian mesin kalimat dan mesin input membutuhkan fungsi dari mesin karakter untuk membuat fungsi builder-nya.
- Implementasi
Diimplementasikan sebagai ADT mesin_kar dengan nama *file header* “mesin_kar.h”

3.5 ADT queueDinerDash

- Sketsa struktur data
Pada ADT queueDinerDash, terdapat struktur data ElType dan Queue. Struktur data Eltype menyimpan nilai makanan, durasi, ketahanan, serta harga yang berupa sebuah integer. Selain itu, pada struktur data Queue menyimpan ElType buffer, idxHead yang berupa integer, serta idxTail berupa integer. Prototype pada ADT berupa CreateQueue untuk membuat queue kosong, terdapat juga primitif length, enqueue, dequeue, displayqueue, serta primitif untuk memeriksa apakah sebuah queue kosong/full.
- persoalan yang diselesaikan
ADT queueDinerDash menyelesaikan persoalan dalam membuat game Diner Dash yang berbasis antrian dalam menerima, memasak, dan menyajikan masakan pesanan.
- Alasan pemilihan
Queue of Eltype (makanan, durasi, ketahanan, harga) berguna untuk pengimplementasian game Diner Dash.
- Implementasi
Diimplementasikan sebagai ADT queueDinerDash dengan nama *file header* “queueDinerDash.h”

3.6 ADT queueGame

- Sketsa struktur data
Pada ADT queueGame, terdapat struktur data QueueGame yang menyimpan nilai bufferG yang merupakan array of ElTypeG, idxHeadG yang berupa integer, serta idxTailG yang berupa integer. Terdapat juga kalimat yang berupa ElTypeG. Ada prototype CreateQueueGame untuk membuat queuegame yang kosong, primitif lengthgame untuk mengitinkan banyaknya elemen queue, enqueuegame untuk menambahkan val pada q dengan aturan FIFO, dequeuegame untuk menghapus val pada q, displayqueuegame untuk menuliskan isi queue dengan traversal, serta primitif untuk memeriksa apakah queuegame kosong/full.
- Persoalan yang diselesaikan
ADT queueGame menyelesaikan persoalan tentang bagaimana kumpulan game diurutkan untuk dimainkan.
- Alasan pemilihan

Alasan pemilihan queueGame (queue of kalimat dengan alternatif 1) adalah untuk memberi perlakuan kumpulan game sesuai dengan sistem game pada umumnya. Queue alternatif 1 dipilih supaya saat setelah di-dequeue dan di-display queue index dimulai dari 0 (ditampilkan 1).

- Implementasi
Diimplementasikan sebagai ADT queueGame dengan nama *file header* “queueGame.h”

4 Program Utama

4.1 Main Menu

Ketika program pertama kali dijalankan, BNMO akan memperlihatkan main menu yang berisi *welcome page* dan empat menu pilihan yaitu START, LOAD, HELP, dan QUIT. Pada file main program, program akan menggunakan prosedur *display_welcoming* dan *mainmenu* untuk menampilkan pesan selamat datang dan *main menu*. Saat prosedur tersebut dipanggil, program akan meminta *input command* kepada *user*. Apabila *user* memasukkan *command* selain START, LOAD, HELP, dan QUIT, program akan menampilkan pesan kesalahan dan meminta *user* untuk memasukkan *command* lagi.

Command START digunakan jika *user* baru pertama kali memainkan BNMO atau me-restart permainan. Command LOAD <savefile> digunakan jika *user* sudah pernah memainkan BNMO. Command HELP di *main menu* memberikan panduan mengenai empat *command* yang tersedia. Command QUIT memungkinkan pemain untuk keluar dari program. Prosedur *scanParserStartStr* akan membaca input dari pengguna berupa pita karakter yang ditranslasikan menjadi string dan mengubah nilai string pada parameter input output prosedur.

4.2 Pembacaan dan Inisialisasi File Konfigurasi

Pada program yang kami buat, kami memutuskan untuk membuat dua buah prosedur pembacaan file yaitu prosedur *readConfig* dan *readSaveFile*. Prosedur *readConfig* digunakan saat pemain memasukkan *command* START. Program akan langsung membaca file konfigurasi awal (*config.txt*) pada folder data yang berisikan daftar permainan. Daftar permainan pada file konfigurasi awal tersebut akan dibaca tiap barisnya dan akan dimasukkan kedalam array yang berisikan daftar permainan. Kemudian, array yang telah terisi tersebut akan ditampilkan ke layar.

Sedikit berbeda dengan prosedur *readConfig* yang langsung membaca file *config.txt*, prosedur *readSaveFile* akan membaca file hasil input dan melakukan algoritma yang sama dengan prosedur *readConfig*.

4.3 Pembacaan Command

Setelah pemain melakukan START atau LOAD, maka program akan menampilkan daftar command yang tersedia yang bisa dipilih user dengan fungsinya masing-masing. Setelah itu, program akan meminta masukan dari pemain. Apabila masukan pemain tidak valid (command tidak dikenali) maka, program akan meminta masukan ulang hingga masukan valid. Pada pembacaan command, kami menggunakan prosedur `scanParserStr` dengan meng-include ADT `mesin_input`. Pada prosedur `scanParserStr`, program akan meminta input dari pemain dengan menggunakan prosedur `STARTINPUT2`, lalu program akan memanggil fungsi `isSameString` untuk melakukan perbandingan antara input pemain dengan nama command, jika True maka program akan menjalankan prosedur sesuai dengan command terkait.

4.4 Command “CREATE GAME”

Command ini akan meminta input string berupa nama game yang akan ditambahkan dari user dan kemudian menambahkan nama game tersebut ke dalam *array of games*. Game baru yang ditambahkan pemain bisa dihapus dan jika dijalankan akan menampilkan pesan bahwa game tersebut masih dalam *maintenance*.

4.5 Command “LIST GAME”

Command ini akan menampilkan daftar game yang tersimpan dalam sistem.

4.6 Command “DELETE GAME”

Command ini akan menampilkan list game yang tersedia oleh sistem. Kemudian, akan meminta input integer berupa nomor game yang ingin dihapus oleh pemain. Setelah menerima input, input user akan divalidasi. Jika nomor game terdapat dalam list game yang tersedia dan memenuhi syarat hapus maka, game akan dihapus dari list game yang tersedia. Jika game berhasil dihapus maka akan menghasilkan output “Game berhasil dihapus” dan jika tidak akan menghasilkan output “Game gagal dihapus”.

4.7 Command “QUEUE GAME”

Command “QUEUE GAME” digunakan untuk mendaftarkan *game* yang tersedia dalam sistem ke dalam antrian *game* yang akan dimainkan. Untuk mendaftarkan *game* ke dalam antrian *game*, pemain diminta untuk memasukkan nomor dari daftar *game* yang tersedia yang bertipe *integer*. Jika *input integer* terdapat pada daftar *game* yang tersedia, sistem akan menambahkan *game* tersebut ke dalam daftar antrian. Jika *input integer* tidak terdapat pada daftar *game* yang tersedia, sistem akan menampilkan pesan bahwa nomor *game* tidak valid dan meminta untuk memasukkan ulang *input integer* tersebut. Pada command ini, *game* yang sudah terdapat di antrian *game* bisa ditambahkan lagi ke dalam

antrian *game*. Daftar antrian *game* akan hilang ketika pemain menjalankan *command* QUIT.

4.8 Command “PLAY GAME”

Command ini digunakan untuk memainkan sebuah permainan dari daftar *game* yang tersedia. *Game* yang dimainkan yaitu *game* yang berada pada urutan pertama dari daftar *game* yang tersedia. Ketika salah satu *game* dimainkan, sistem akan menjalankan *game* tersebut sesuai dengan *section* “Spesifikasi Game”. Dalam artian, jika *game* yang dijalankan tidak terdapat pada “Spesifikasi Game”, sistem akan menampilkan pesan bahwa *game* tersebut tidak dapat dijalankan. *Game* yang sudah dijalankan, akan dihapus dari daftar antrian *game* dan jika *game* yang dipilih tersebut dalam *maintenance*, *game* tersebut akan dihapus juga dari daftar antrian *game*.

4.9 Command “SKIP GAME <n>”

Command ini digunakan untuk melewati *game* pada daftar antrian *game*. Jumlah *game* yang dilewatkan akan sebanyak “n” sesuai dengan nilai *input integer* positif oleh *user*. *Game* yang dilewatkan berarti akan dihapus dari daftar antrian *game*. Jika *input integer* tersebut lebih dari daftar antrian *game*, sistem akan menghapus semua *game* pada daftar antrian.

4.10 Command “HELP”

Command HELP pada BNMO akan menampilkan seluruh *command* yang valid beserta penjelasannya untuk membantu user yang kurang memahami fungsi-fungsi dari setiap *command*.

4.11 Command “SAVE <namafile>”

Command SAVE adalah *command* untuk menyimpan nama *game* yang telah ditambahkan menggunakan *command* CREATE GAME ke file konfigurasi. Program akan membuka file <namafile>.txt dengan fungsi fopen() dan akan memastikan apakah file berhasil di buka atau tidak. Bila file konfigurasi berhasil dibuka, maka program akan menuliskan jumlah *game* yang disimpan dan nama-nama *game* yang ada di sistem pada file <namafile>.txt . Setelah semua nama *game* sudah ditulis di file <namafile>.txt, program akan menutup file dengan fungsi fclose().

4.12 Command “QUIT”

Command QUIT adalah *command* untuk keluar dari program BNMO setelah user sudah selesai menggunakan program.

4.13 Game “RNG”

Game RNG adalah game tebak-tebakan yang dimenangkan oleh user dengan cara menebak *random number* yang digenerate oleh program. Pemain memiliki 10 kesempatan dan 1000 poin di awal permainan. Untuk tiap tebakkan salah, poin pemain akan dikurang 100. Pada awal permainan, time diinisialisasi null untuk menjamin *random number generator* yang kemudian akan digunakan benar-benar acak. Kemudian, program akan meminta *input integer* dari pemain dengan memanggil fungsi *scanint*. Dengan menggunakan *random number generator* bawaan bahasa C, program akan menghasilkan suatu angka acak dengan batasan 100. Program akan memanggil prosedur tebakkan yang akan melakukan *comparison* antara *input* tebakkan pemain dengan angka acak hasil *random number generator*. Bila *input* pemain lebih besar maka, program akan menampilkan “Lebih besar”. Bila *input* pemain lebih kecil maka, program akan menampilkan “Lebih kecil”. Bila *input* pemain sama besar maka, program akan menampilkan “Ya, X adalah <*input* pemain>”.

4.14 Game “Diner Dash”

Game Diner Dash adalah permainan pelayanan makanan yang memiliki tiga perintah utama yaitu “COOK”, “SERVE”, dan “SKIP”. Permainan dimulai dengan diinisialisasi tiga “pelanggan” di antrian. Masing-masing pelanggan sudah memesan sebuah makanan yang diwakilkan ADT Meal. Setiap Meal memiliki informasi nama, durasi yang dibutuhkan untuk dimasak, durasi ketahanan makanan, dan harga makanan. Permainan akan berakhir ketika user berhasil SERVE 15 orang dengan score total harga makanan yang diserve. Permainan juga akan berakhir bila antrian sudah melebihi 7 orang dengan score 0 (dianggap kalah).

4.15 Game “STI Mencari Jodoh”

Di awal permainan, pemain akan diminta memasukkan nama dan jenis kelaminnya. Bila, pemain memasukkan input jenis kelamin selain “cowo” dan “cewe” maka, prosedur akan menampilkan pesan kesalahan dan meminta pemain memasukkan jenis kelamin yang tepat.

Jika pemain memasukkan jenis kelamin “cewe” maka, prosedur akan memanggil fungsi STARTCEWE. Jika pemain memasukkan jenis kelamin “cowo” maka, prosedur akan memanggil fungsi STARTCOWO. Kedua fungsi akan membaca file berisi nama-nama lawan jenis dari input misal, fungsi STARTCEWE akan membaca suatu file bernama jodohcewe.txt yang berisi kumpulan nama laki-laki dan begitu pula sebaliknya untuk fungsi STARTCOWO. Lalu dengan menggunakan *random number generator* bawaan C, prosedur akan *generate* nomor acak yang kemudian dijadikan indeks akses list nama lawan jenis tersebut. Terakhir, prosedur akan menampilkan nama lawan jenis sesuai indeks hasil *random number* tersebut sebagai jodoh sang pemain.

5 Algoritma-Algoritma Menarik

5.1 Algoritma scan parser

```
void scanParserStr(char* *sInput1, char* *sInput2, char* *sInput3){
    STARTINPUT2();
    *sInput1 = (char*)malloc(sizeof(char)* CInput.Length);
    InputToString(CInput,*sInput1);
    ADVINPUT2();
    if (isSameString(*sInput1, "SAVE") || isSameString(*sInput1, "CREATE") || isSameString(*sInput1, "LIST") || is
        *sInput2 = (char*)malloc(sizeof(char)* CInput.Length);
        InputToString(CInput,*sInput2);
        if (CInput.Length != 0){
            ADVINPUT2();
            *sInput3 = (char*)malloc(sizeof(char)* CInput.Length);
            InputToString(CInput,*sInput3);
        }
    } else {
        *sInput2 = (char*)malloc(sizeof(char)* CInput.Length);
        InputToString(CInput,*sInput2);
    }
    while (CInput.Length != 0){
        ADVINPUT2();
    }
}
```

Algoritma scan parser adalah algoritma yang dapat mengubah pita input 2 kata atau lebih menjadi 2 string (masing-masing string berisi 1 kata). Prosedur scan parser digunakan pada saat harus membaca input command dari user. Algoritma ini menarik karena membuat pembacaan input menjadi lebih mudah dioperasikan. Kegunaan dari algoritma ini hampir sama dengan kegunaan dari fungsi bawaan bahasa C yaitu `sscanf()`.

5.2 Mesin Input

```
boolean EndInput;
Input CInput;

void IgnoreNewline(){
    while (CC == NEWLINE)
    {
        ADV();
    }
}

void IgnoreBlank(){
    while (CC==BLANK)
    {
        ADV();
    }
}
```

Pada ADT mesin input, terdapat prosedur `IgnoreNewline` (terdapat pada mesin kalimat) dan `IgnoreBlank` (terdapat pada mesin kata). Hal ini membuat ADT mesin input menjadi menarik karena merupakan gabungan dari mesin kata dan mesin kalimat. ADT mesin input ini digunakan untuk membaca input dari user sebagai kata atau sebagai kalimat.

- Hasil yang ditampilkan

```
-----
ENTER COMMAND: LOAD nofile.txt
File tidak ditemukan!
-----
```

6.4 Data Test 4

- Fitur yang dites
Test ini dilakukan untuk membaca file yang dituliskan di command, namun kondisinya jika file yang dimasukkan tersedia

- Hasil yang diharapkan

```
ENTER COMMAND: LOAD savefile1.txt
Save file berhasil dibaca. BNMO berhasil dijalankan.
```

- Hasil yang ditampilkan

```
ENTER COMMAND: LOAD savefile1.txt

Load file berhasil dibaca. BNMO berhasil dijalankan.

-----
```

6.5 Data Test 5

- Fitur yang dites
Test ini dilakukan untuk menampilkan bantuan berupa command-command yang tersedia pada main menu
- Hasil yang diharapkan
Menampilkan bantuan command start, load, quit, help
- Hasil yang ditampilkan

```
-----
ENTER COMMAND: HELP
START -> Untuk memulai petualanganmu bersama BNMO! Memungkinkan file konfigurasi default yang berisi list game dimainkan
LOAD -> Pilih filename yang berisi list game yang ingin dimainkan.
QUIT -> Memungkinkanmu keluar dari program.
HELP -> Bantuan untuk kamu yang kebingungan dengan command-command yang tersedia!
-----
```

6.6 Data Test 6

- Fitur yang dites
Test ini dilakukan untuk menampilkan list command yang dapat dijalankan
- Hasil yang diharapkan
Menampilkan menu program
- Hasil yang ditampilkan

```
----- MENU -----
1. SAVE [file name]
2. CREATE GAME
3. LIST GAME
4. DELETE GAME
6. QUEUE GAME
7. PLAY GAME
8. SKIPGAME [jumlah skip]
9. HELP
0. QUIT
-----
```

6.7 Data Test 7

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat menyimpan state game permainan saat ini ke dalam suatu file

- **Hasil yang diharapkan**

```
ENTER COMMAND: SAVE savefile1.txt
Save file berhasil disimpan.
```

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: SAVE savefile1.txt
Berhasil menyimpan state ke path ../data/savefile1.txt.
-----
```

6.8 Data Test 8

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat menambahkan game baru pada daftar game

- **Hasil yang diharapkan**

```
ENTER COMMAND: CREATE GAME
Masukkan nama game yang akan ditambahkan: EXTRA1
Game berhasil ditambahkan
```

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: CREATE GAME
Masukkan nama game yang akan ditambahkan: EXTRA1
Game berhasil ditambahkan
-----
```

6.9 Data Test 9

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat menampilkan daftar game yang tersedia

- **Hasil yang diharapkan**


```
ENTER COMMAND: LIST GAME
Berikut adalah daftar game yang tersedia
1. RNG
2. LUNCH SLOW
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
```

- Hasil yang ditampilkan

```
-----
ENTER COMMAND: LIST GAME
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
-----
```

6.10 Data Test 10

- Fitur yang dites

Test ini dilakukan untuk mengecek apakah program dapat menghapus game dari list, namun kondisinya jika nomor game yang ingin dihapus tidak valid

- Hasil yang diharapkan

Mengeluarkan pesan error

- Hasil yang ditampilkan

```
-----
ENTER COMMAND: DELETE GAME
Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
Masukkan nomor game yang akan dihapus: 6

Nomor game tidak valid!
-----
```

6.11 Data Test 11

- Fitur yang dites

Test ini dilakukan untuk mengecek apakah program dapat menghapus game dari list, namun kondisinya jika nomor game yang ingin dihapus merupakan 5 game pertama pada file (Peraturannya 5 game pertama dalam file tidak dapat dihapus)

- Hasil yang diharapkan

```
ENTER COMMAND: DELETE GAME
Berikut adalah daftar game yang tersedia
1.  RNG
2.  LUNCH SLOW
3.  DINOSAUR IN EARTH
4.  RISEWOMAN
5.  EIFFEL TOWER
Masukkan nomor game yang akan dihapus: 1

Game gagal dihapus
```

- Hasil yang ditampilkan

```
ENTER COMMAND: DELETE GAME
Berikut adalah daftar game yang tersedia
1.  RNG
2.  Diner DASH
3.  DINOSAUR IN EARTH
4.  RISEWOMAN
5.  EIFFEL TOWER
Masukkan nomor game yang akan dihapus: 3

Game gagal dihapus!
```

6.12 Data Test 12

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat menghapus game dari list, namun kondisinya jika nomor game yang ingin dihapus merupakan game yang dibuat secara custom

- **Hasil yang diharapkan**

```
ENTER COMMAND: DELETE GAME
Berikut adalah daftar game yang tersedia
1.  RNG
2.  LUNCH SLOW
3.  DINOSAUR IN EARTH
4.  RISEWOMAN
5.  EIFFEL TOWER
6.  CUSTOM GAME 1
Masukkan nomor game yang akan dihapus: 6

Game berhasil dihapus
```

- Hasil yang ditampilkan

```

-----
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. EXTRA1
Masukkan nomor game yang akan dihapus: 6

Game berhasil dihapus.
-----

```

6.13 Data Test 13

- Fitur yang dites

Test ini dilakukan untuk mengecek apakah program dapat menambahkan game ke dalam daftar antrian, namun kondisinya jika nomor game yang ingin ditambahkan tidak valid

- Hasil yang diharapkan

```

ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
1. EIFFEL TOWER
2. RISEWOMAN
3. LUNCH SLOW

Berikut adalah daftar game yang tersedia
1. RNG
2. LUNCH SLOW
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER

Nomor Game yang mau ditambahkan ke antrian: 9

Nomor permainan tidak valid, silahkan masukkan nomor game pada list.

```

- Hasil yang ditampilkan

```

-----
ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
-> Antrian game kosong
Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
Nomor Game yang mau ditambahkan ke antrian: 6

Nomor permainan tidak valid, silahkan masukkan nomor game pada list.
-----

```

6.14 Data Test 14

- Fitur yang dites

Test ini dilakukan untuk mengecek apakah program dapat menambahkan game ke dalam daftar antrian, namun kondisinya jika nomor game yang ingin ditambahkan valid

- Hasil yang diharapkan

```
ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
1. EIFFEL TOWER
2. RISEWOMAN
3. LUNCH SLOW

Berikut adalah daftar game yang tersedia
1. RNG
2. LUNCH SLOW
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER

Nomor Game yang mau ditambahkan ke antrian: 2

Game berhasil ditambahkan ke dalam daftar antrian.
```

- Hasil yang ditampilkan

```
-----
ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
-> Antrian game kosong
Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
Nomor Game yang mau ditambahkan ke antrian: 3

Game berhasil ditambahkan kedalam daftar antrian.
-----
```

6.15 Data Test 15

- Fitur yang dites

Test ini dilakukan untuk mengecek apakah program dapat memainkan game yang terdapat di urutan pertama pada antrian game, namun kondisinya jika daftar antrian kosong

- Hasil yang diharapkan

Mengeluarkan pesan error

- Hasil yang ditampilkan

```
-----
ENTER COMMAND: PLAY GAME
Berikut adalah daftar game-mu:
-> Antrian game kosong

Tidak ada game yang dapat dimainkan. Queue game terlebih dahulu!
-----
```

6.16 Data Test 16

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat memainkan game yang terdapat di urutan pertama pada antrian game, namun kondisinya jika game sedang dalam maintenance

- **Hasil yang diharapkan**

```
ENTER COMMAND: PLAY GAME
Berikut adalah daftar Game-mu
1. RISEWOMAN
2. RISEWOMAN
3. LUNCH SLOW

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

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: PLAY GAME
Berikut adalah daftar game-mu:
Berikut adalah daftar antrian game-mu
1. DINOSAUR IN EARTH
2. RNG

Game DINOSAUR IN EARTH masih dalam maintenance, belum dapat dimainkan. Silahkan pilih game lain.
-----
```

6.17 Data Test 17

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat memainkan game yang terdapat di urutan pertama pada antrian game, namun kondisinya jika game dapat dimainkan

- **Hasil yang diharapkan**

```
ENTER COMMAND: PLAY GAME
Berikut adalah daftar Game-mu
1. EIFFEL TOWER
2. RISEWOMAN
3. LUNCH SLOW

Loading EIFFEL TOWER ...
```

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: PLAY GAME
Berikut adalah daftar game-mu:
Berikut adalah daftar antrian game-mu
1. RNG

Loading RNG ...
RNG Telah dimulai.
Uji keberuntungan Anda dengan menebak X (0 <= X <= 100)
Anda dapat menebak maksimal 10 kali!
Tebakan:
```

6.18 Data Test 18

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat melewati permainan sebanyak yang diminta, namun kondisinya jika masukan valid

- **Hasil yang diharapkan**

```
ENTER COMMAND: SKIP GAME 2
Berikut adalah daftar Game-mu
1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN

Loading RISEWOMAN ...
```

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: SKIP GAME 2
Berikut adalah daftar antrian game-mu
1. RNG
2. Diner DASH
3. STI MENCARI JODOH

Loading STI MENCARI JODOH ...
Cari Jodohmu disini!!
Masukkan nama kamu:
```

6.19 Data Test 19

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat melewati permainan sebanyak yang diminta, namun kondisinya jika masukan tidak valid

- **Hasil yang diharapkan**

```
ENTER COMMAND: SKIP GAME 5
Berikut adalah daftar Game-mu
1. RISEWOMAN
2. LUNCH SLOW
3. RISEWOMAN

Tidak ada permainan lagi dalam daftar game-mu.
```

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: SKIP GAME 5
Berikut adalah daftar antrian game-mu
1. RNG
2. Diner DASH
3. STI MENCARI JODOH

Tidak ada game yang dapat dimainkan. Queue game terlebih dahulu!
-----
```

6.20 Data Test 20

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah dapat keluar dari program jika memberikan command quit

- **Hasil yang diharapkan**

```
ENTER COMMAND: QUIT

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

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: QUIT
Anda keluar dari game BNMO.
Bye bye ...
----- TERIMA KASIH -----
----- TELAH MENGGUNAKAN BNMO -----
```

6.21 Data Test 21

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program dapat menampilkan daftar command yang tersedia

- **Hasil yang diharapkan**

Mengeluarkan bantuan berupa command-command yang tersedia pada program

- **Hasil yang ditampilkan**

```
-----
ENTER COMMAND: HELP
SAVE -> Simpan state game-mu dengan command ini!
CREATE GAME -> Ingin menambahkan game baru? Command ini jawabannya.
LIST GAME -> Untuk melihat daftar game yang tersedia.
DELETE GAME -> Hapus game yang kamu tidak suka dengan command ini.
QUEUE GAME -> Lihat dan tambahkan permainan yang ingin kamu mainkan ke dalam list!
PLAY GAME -> Mulai memainkan game sesukamu dengan command ini!
SKIP GAME -> Gunakan command ini untuk melewati permainan sebanyak n kali.
QUIT -> Memungkinkanmu keluar dari program.
HELP -> Bantuan untuk kamu yang kebingungan dengan command-command yang tersedia!
-----
```

6.22 Data Test 22

- **Fitur yang dites**

Test ini dilakukan untuk mengecek apakah program akan mengirimkan pesan error apabila pemain memasukkan command yang tidak valid

- **Hasil yang diharapkan**

```
ENTER COMMAND: COMMAND_ANEH
```

Command tidak dikenali, silahkan masukkan command yang valid.
ENTER COMMAND:

- Hasil yang ditampilkan

ENTER COMMAND: COMMANDANEH

Command tidak dikenali, silahkan memasukkan command yang valid.

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1.	MAIN MENU	Memeriksa apakah program dapat menampilkan tampilan utama	Melakukan compile file program di terminal	Data Test 1	Menampilkan tampilan utama program	Sesuai dengan hasil yang diharapkan
2.	START	Memeriksa apakah file config.txt berhasil terbaca pada listgame di main program	Memasukkan command START untuk mengkonfigurasi file	Data Test 2	File konfigurasi sistem berhasil dibaca	Sesuai dengan hasil yang diharapkan
3.	LOAD [file name]	Memeriksa apakah file savefile1.txt terbaca dan tersimpan di listgame pada main program	Memasukkan command LOAD untuk membaca file	Data Test 3 Data Test 4	File berhasil dibaca. BNMO berhasil dijalankan	Sesuai dengan hasil yang diharapkan
4.	HELP MAIN MENU	Memeriksa apakah program dapat memberikan bantuan command pada main menu	Memasukkan command HELP	Data Test 5	Program mengeluarkan bantuan berupa command-command yang tersedia pada main menu	Sesuai dengan hasil yang diharapkan
5.	MENU	Memeriksa apakah program dapat menampilkan main menu yang tersedia	Otomatis jalan saat main program dieksekusi	Data Test 6	Program mengeluarkan list command yang dapat dimasukkan	Sesuai dengan hasil yang diharapkan
6.	SAVE [file name]	Memeriksa apakah current state program (isi listgame) tersimpan di savefile1.txt sesuai dengan format	Memasukkan command SAVE <filename>	Data Test 7	File berhasil disimpan	Sesuai dengan hasil yang diharapkan
7.	CREATE GAME	Memeriksa apakah game yang input dapat ditambahkan ke listgame	Memasukkan command CREATE GAME memasukkan nama game yang ingin ditambah	Data Test 8	Game berhasil ditambahkan	Sesuai dengan hasil yang diharapkan
8.	LIST GAME	Memeriksa apakah dapat menampilkan game yang tersedia	Memasukkan command LIST GAME	Data Test 9	Menunjukan daftar game yang tersedia	Sesuai dengan hasil yang diharapkan
9.	DELETE GAME	Memeriksa apakah dapat menghapus	Memasukkan command	Data Test 10	Menghapus game yang diinginkan	Sesuai dengan hasil yang diharapkan

		sebuah game dari daftar game	DELETE GAME Memasukkan nomor game yang ingin dihapus	Data Test 11 Data Test 12		
10.	QUEUE GAME	Memeriksa apakah dapat mendaftarkan permainan ke dalam list	Memasukkan command QUEUE GAME Memasukkan nomor game yang ingin ditambahkan ke antrian	Data Test 13 Data Test 14	Menambahkan game ke dalam daftar antrian	Sesuai dengan hasil yang diharapkan
11.	PLAY GAME	Memeriksa apakah program dapat memainkan sebuah permainan	Memasukkan command PLAY GAME	Data Test 15 Data Test 16 Data Test 17	Memainkan sebuah permainan	Sesuai dengan hasil yang diharapkan
12.	SKIPGAME [jumlah skip]	Memeriksa apakah program dapat melewati permainan sebanyak n	Memasukkan command SKIPGAME [jumlah skip]	Data Test 18 Data Test 19	Melewatkan permainan sebanyak yang diinput	Sesuai dengan hasil yang diharapkan
13.	QUIT	Untuk keluar dari program	Memasukkan command QUIT	Data Test 20	Keluar dari program	Sesuai dengan hasil yang diharapkan
14.	HELP MENU	Memeriksa apakah program dapat menampilkan menu command yang tersedia	Memasukkan command HELP	Data Test 21	Menampilkan daftar command yang tersedia	Sesuai dengan hasil yang diharapkan
15.	COMMAND LAIN	Memeriksa program mengeluarkan pesan error jika command tidak sesuai	Memasukkan command yang tidak valid	Data Test 22	Menampilkan pesan error	Sesuai dengan hasil yang diharapkan

8 Pembagian Kerja dalam Kelompok

Nama	Tugas
18221071/Ahmad Rizki	<ul style="list-style-type: none">• Membuat main program• Membuat fungsi <i>start, load, save, quit, playgame</i>• Membuat game bonus• Membuat fungsi tambahan untuk program
18221081/Nadine Aliya	<ul style="list-style-type: none">• Membuat game RNG• Membuat fungsi <i>skipgame, queuegame, commandlain, help</i>• Membuat laporan
18221099/Clara Alrosa Fernanda Sinaga	<ul style="list-style-type: none">• Membuat game RNG• Membuat fungsi <i>skipgame, queuegame, commandlain, help</i>• Membuat laporan
18221111/Pramaditya Fajri M.	<ul style="list-style-type: none">• Membuat fungsi <i>creategame, listgame, deletgame</i>• Membuat game Diner Dash• Membuat file driver
18221173/Naura Valda Prameswari	<ul style="list-style-type: none">• Membuat main program• Membuat fungsi <i>start, load, save, quit, playgame</i>• Membuat fungsi tambahan untuk program• Membuat laporan

9 Lampiran

9.1 Deskripsi Tugas Besar

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

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:

a. START

START merupakan 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.

b. LOAD <filename>

LOAD merupakan 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, lebih detailnya bisa dilihat pada [Konfigurasi Sistem](#).

c. **SAVE <filename>**

SAVE merupakan 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.

d. **CREATEGAME**

CREATEGAME merupakan command yang digunakan untuk menambahkan game baru pada daftar game. Spesifikasi game yang dibuat dapat dilihat pada section [Spesifikasi Game](#)

e. **LISTGAME**

LISTGAME merupakan command yang digunakan untuk menampilkan daftar game yang disediakan oleh sistem.

f. **DELETGAME**

DELETGAME merupakan command yang digunakan untuk menghapus sebuah game dari daftar game. Adapun aturan penghapusan game adalah:

- Game yang dapat dihapus hanya game yang dibuat secara custom oleh pengguna.
- 5 game pertama pada file konfigurasi tidak dapat dihapus.
- Game yang saat itu terdapat di dalam queue game tidak dapat dihapus.

g. **QUEUEGAME**

QUEUEGAME merupakan command yang digunakan untuk mendaftarkan permainan kedalam list. List dalam queue akan hilang ketika pemain menjalankan command **QUIT**.

h. **PLAYGAME**

PLAY GAME merupakan command yang digunakan untuk memainkan sebuah permainan. Game yang dimainkan adalah game dengan urutan pertama di antrian game. Ketika salah satu permainan dimulai, sistem akan menjalankan game sesuai pada section [Spesifikasi Game](#). Permainan selain yang dispesifikasikan pada [Spesifikasi Game](#) akan menampilkan pesan bahwa game tidak dapat dimainkan.

- i. **SKIPGAME <n>**
SKIPGAME merupakan command yang digunakan untuk melewati permainan sebanyak n.
- j. **QUIT**
Keluar dari program.
- k. **HELP**
Bantuan command-command yang disebutkan di atas. Tampilan dan kata-kata dibebaskan.
- l. **COMMAND LAIN**
Command-command lain selain yang disebutkan diatas tidak valid.

4. **Konfigurasi sistem**

File konfigurasi akan dibaca saat memulai permainan. File ini menyimpan data-data yang disimpan ketika sistem dijalankan sebelumnya

9.2 Notulen Rapat



Tanggal	Kesimpulan	Anggota yang terlibat
29 November 2022	<ul style="list-style-type: none">- Memahami spesifikasi tugas- Pembagian kerja untuk membuat <i>command</i> yang diperlukan.- Menghubungi asisten perihal asistensi 1	Semua Anggota
9 November 2022	<ul style="list-style-type: none">- Membuat laporan- Menyelesaikan program	Semua Anggota
10 November 2022	<ul style="list-style-type: none">- Mengecek program secara garis besar- Melengkapi laporan- Mengurus bug-bug yang masih ada pada program	Semua Anggota



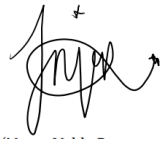

Form Asistensi Tugas Besar
IF2110/Algoritma dan Struktur Data
Sem. 1 2022/2023

No. Kelompok/Kelas : 05/01
 Nama Kelompok : DoaWibu
 Anggota Kelompok (Nama/NIM) :
 1. Ahmad Rizki/18221071
 2. Nadine Aliya Putri/18221081
 3. Clara Alrosa Fernanda Sinaga/18221099
 4. Pramaditya Fajri Migfar/18221111
 5. Naura Valda Prameswari/18221173





Asisten Pembimbing : Jason Stanley Yoman



Asistensi I

Tanggal : Kamis, 3 November 202	Catatan Asistensi: Amriz : Ada 2 konfigurasi di spesifikasi, konfigurasi yang save file ada history awal, tapi dihilangkan jadi bedanya konfigurasi awal sama file yang disimpan oleh sistem itu apa kak? Jawaban: **belum terjawab Amriz : Di konfigurasi gada MARK nya kak, jadinya aku dan teman-teman memutuskan untuk menambah MARK berupa ; kira-kira ada ide lain ga ya kak? Untuk mesin kata kita, BLANK dihitung kata kak. Jawaban : Tidak perlu MARK yang pakai titik koma, tapi pakai EOF Di TXT ada ' /0' atau ngga Jawaban: Gaada, itu khusus string yang diisi aja Saran: Kalo di c ada yang namanya end of file, kalo disitu kaya MARK untuk akhir dari sebuah file, jadi kalian boleh cari tau cara pakai EOF itu gimana EOF itu untuk fungsinya lagi atau sudah tersedia di c? Jawaban: sudah tersedia, itu cara kerjanya mirip mark, tp mark khusus file
Tempat : Google Meet	
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1  Ahmad Rizki 18221071 2  Nadine Aliya Putri 18221081 3	

 <p>Clara Alrosa Fernanda Sinaga 18221099</p> <p>4</p>  <p>Pramaditya Fajri Migfar 18221111</p>  <p>Naura Valda Prameswari 18221173</p>	<p>Makefile sebenarnya buat apa Jawaban: Makefile untuk mempermudah compile biar ga panjang-panjang banget. Jadinya di makefile bisa buat alias. Jadinya tidak perlu gcc terus sebut masing-masing file</p> <p>Clara : Yang RNG sejauh ini progress kita sudah selesai, tapi ada satu spesifikasi yang bingung, karena kita masih pakai random generator langsung dari c, namun di spesifikasi ga bener-bener acak. Maksud dari spesifikasinya itu apa ya kak?</p> <p>Jawaban: Kalo tidak pakai srand time null kalo dijalankan berulang kali hasil randomnya sama karena diinisiasi dengan nomor yang sama. Time null itu maksudnya time stand jam sekarang. Jadinya tiap dijalankan asalkan waktunya beda, random number nya bakal berbeda</p>
	<p>Tanda Tangan Asisten:</p>  <p>Jason Stanley Yoman 13519019</p>

Asistensi II

Tanggal : Kamis, 10 November 2022	Catatan Asistensi:
Tempat : Google Meet	
Kehadiran Anggota Kelompok:	
No NIM Tanda tangan 1  Ahmad Rizki 18221071 2  Nadine Aliya Putri 18221081 3  Clara Alrosa Fernanda Sinaga 18221099 4  Pramaditva Fairi Migfar	

<p>18221111</p>  <p>Naura Valda Prameswari 18221173</p>	
	<p>Tanda Tangan Asisten:</p>  <p>Jason Stanley Yoman 13519019</p>

9.3 Log Activity Anggota Kelompok

Nama	Tanggal	Aktivitas
Ahmad Rizki / 18221071	28 Oktober 2022	Membuat repository github, dan menambah kolaborator, membuat welcome page
Naura Valda P/ 18221173	30 Oktober 2022	Membuat ADT array, arrayGame, queue, mesin karakter, dan mesin kalimat. Membuat fungsi start, load, dan display list
Ahmad Rizki/ 18221071	30 Oktober 2022	Membuat ADT array, arrayGame, queue, mesin karakter, dan mesin kalimat. Membuat fungsi start, load, dan display list
Nadine Aliya Putri / 18221081	31 Oktober 2022	Membuat game RNG
Clara Alrosa Fernanda Sinaga / 18221099	31 Oktober 2022	Membuat game RNG dan menambahkan fungsi help
Pramaditya Fajri Migfar / 18221111	31 Oktober 2022	Mengupdate ADT queue untuk game Dinner Dash
Pramaditya Fajri Migfar / 18221111	1 November 2022	Membuat game Dinner Dash
Clara Alrosa Fernanda Sinaga / 18221099	2 November 2022	Menambahkan fungsi queue game, melakukan beberapa perubahan pada game RNG
Nadine Aliya Putri/18221081	2 November 2022	Menambahkan fungsi skipgame, update game rng, update program command lain
Ahmad Rizki / 18221071	3 November 2022	Menambahkan prosedur save, membenarkan ADT, prosedur playgame, mesin input, scan input dari pengguna dan parser, check semua fungsi yang error
Naura Valda P / 18221173	3 November 2022	Menambahkan prosedur save, membenarkan ADT, prosedur playgame, mesin input, scan input dari pengguna dan parser, check semua

		fungsi yang error
Ahmad Rizki / 18221071	4 November 2022	Menggabungkan program dalam console dan main
Naura Valda P / 18221173	4 November 2022	Menggabungkan program dalam console dan main
Ahmad Rizki / 18221071	10 November 2022	Debug Program
Naura Valda P / 18221071	10 November 2022	Debug Program
Clara Alrosa Fernanda Sinaga / 18221099	10 November 2022	Debug Program
Nadine Aliya Putri/18221081	10 November 2022	Debug Program
Pramaditya Fajri Migfar/ 18221111	10 November 2022	Debug Program