

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data

<BNMO>

Dipersiapkan oleh:

Kelompok 12

Ghaylan Muhammad Fatih 18221042

Gracia Theophilia 18221078


Fikri Naufal Hamdi 18221096

Reinhart Wisely Lim 18221154

Esther Regina 18221086

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB1-12</i>		<i><jml hlm></i>
		<i>Revisi</i>	<i>1</i>	<i>11 November 2022</i>

Daftar Isi

1 Ringkasan	3
2 Penjelasan Tambahan Spesifikasi Tugas	4
2.1 Mekanisme Load	4
2.2 Spesifikasi Game RNG	4
2.3 Banner BNMO	5
2.4 Mekanisme Queue Game	5
2.5 Mekanisme Save	5
3 Struktur Data (ADT)	5
3.1 ADT Arraydin (Array Dinamis)	5
3.2 ADT ArraydinInt	7
3.3 ADT Mesin Karakter	8
3.4 ADT Mesin Kata	8
3.5 ADT Queue	10
3.6 ADT QueueInt	10
4 Program Utama	11
5 Algoritma-Algoritma Menarik	12
5.1 Command	12
5.2 RNG (Random Number Generator)	14
6 Data Test	14
6.1 Data Test START	14
6.2 Data Test LOAD	15
6.3 Data Test SAVE	16
6.4 Data Test CREATE GAME	16
6.5 Data Test LIST GAME	17
6.6 Data Test DELETE GAME	17
6.7 Data Test QUEUE GAME	19
6.8 Data Test PLAY GAME	20
6.9 Data Test SKIP GAME	21
6.10 Data Test QUIT	22
6.11 Data Test HELP	22
6.12 Data Test COMMAND LAIN	23

6.13 Data Test RNG	23
6.14 Data Test DINER DASH	24
7 Test Script	28
8 Pembagian Kerja dalam Kelompok	31
9 Lampiran	31
9.1 Deskripsi Tugas Besar 2	31
9.2 Notulen Rapat	31
9.3 Log Activity Anggota Kelompok	32
9.4 Lain-Lain	35

1 Ringkasan

BNMO adalah sebuah *robot video game console* berbasis CLI (*command-line interface*) yang dapat menjalankan permainan. BNMO memiliki beberapa fitur utama, yaitu memainkan game, menambah game, menghapus game, dan mengurutkan game yang akan dimainkan. Pada setiap kali permainan, pemain dapat memasukkan berbagai command seperti START, LOAD <filename>, SAVE <filename>, CREATEGAME, LISTGAME, DELETEGAME, QUEUEGAME, PLAYGAME, SKIPGAME <n>, QUIT, dan HELP. Selain command-command yang telah disebutkan, program tidak akan mengenali command tersebut dan akan meminta pemain memasukkan command yang valid.

Permainan BNMO dibuat dengan bahasa C dengan memanfaatkan struktur data yang sudah dipelajari dalam mata kuliah Algoritma dan Struktur Data. Library yang boleh digunakan pada tugas besar kali ini adalah stdio.h, stdlib.h, time.h, dan math.h. Berdasarkan spesifikasi permainan yang ada, berikut beberapa ADT yang dapat digunakan:

1. ADT Array
2. ADT Mesin Karakter
3. ADT Mesin Kata
4. ADT Queue

Penjelasan mengenai ADT yang digunakan dan modifikasi yang kami lakukan pada ADT akan dijelaskan pada Bagian 3 Struktur Data pada laporan ini.

Pada program yang telah kami buat, program ini terdiri dari beberapa game yaitu RNG (*Random Number Generator*) dan Diner Dash. RNG adalah permainan yang mengharuskan pemain untuk menebak angka acak. Diner Dash adalah permainan mengantar makanan terurut berdasarkan prioritasnya.

Secara umum, laporan ini berisi ringkasan mengenai program yang kami buat, penjelasan tambahan mengenai spesifikasi tugas, struktur data yang digunakan dalam membuat permainan BNMO, program utama, algoritma-algoritma menarik dalam program, data test, test script, pembagian kerja dalam kelompok, dan lampiran-lampiran yang berisi notulen rapat, log activity anggota kelompok, dan lain-lain.

2 Penjelasan Tambahan Spesifikasi Tugas

Dalam membuat dan menjalankan program, terdapat beberapa fitur atau command yang memerlukan spesifikasi tambahan agar tidak terjadi miskonsepsi terhadap program tersebut. Berikut penjelasan tambahan terkait beberapa spesifikasi fitur maupun command.

2.1 Mekanisme Load

Pada mekanisme load, jika daftar game kosong, load tidak dapat dijalankan. Konfigurasi games harus dimasukkan terlebih dahulu ke dalam file konfigurasi.

2.2 Spesifikasi Game RNG

Pada game RNG, angka yang di-random berada pada range 0-100. Kesempatan setiap pemain untuk menebak angka yang diacak tersebut adalah 10 kali. Program akan mengeluarkan pesan apakah tebakan pemain lebih besar/lebih kecil/sudah benar. Jika pemain tidak dapat

menebak angka selama 10 kesempatan, maka program akan mengeluarkan pesan bahwa kesempatan menebak telah habis dan program akan memberi tahu pemain jawaban yang benar.

2.3 Banner BNMO

Pada awal program, ditampilkan sebuah banner yang bertuliskan “BNMO”. Fitur ini merupakan fitur tambahan untuk memperindah tampilan pada awal program sekaligus dapat merepresentasikan program permainan yang kami buat, yaitu BNMO. Selain banner BNMO, program juga akan menampilkan tulisan “Selamat Datang di BNMO” dan menampilkan panduan singkat command untuk memulai program.

2.4 Mekanisme Queue Game

Pada mekanisme Queue Game, pemain dimungkinkan untuk menambahkan game yang sama lebih dari satu kali ke dalam daftar antrian game. Ketika pemain menambahkan game yang sama lebih dari satu kali, game tersebut akan terdaftar sebanyak jumlah penambahan game tersebut oleh pemain.

2.5 Mekanisme Save

Pada mekanisme Save, jika pemain memasukkan file konfigurasi tanpa “.txt”, file konfigurasi tetap akan terbentuk dan *state game* akan tetap berhasil disimpan dalam file tersebut.

3 Struktur Data (ADT)

Pada program ini, kami menggunakan 4 buah ADT. Berikut daftar dari beberapa ADT yang kami gunakan dalam program ini.

3.1 ADT Arraydin (Array Dinamis)

ADT Array Dinamis terdiri dari struktur ArrayDin yang terdiri atas array of char, capacity (integer), dan Neff (integer). Prototype pada ADT Array Dinamis terdiri atas konstruktor pembuat tabel kosong (MakeArrayDin), destruktur dealokasi array dinamis (DeallocateArrayDin), selektor yang terdiri atas IsEmpty (mengecek apakah banyaknya elemen efektif array bernilai 0), Length (banyaknya elemen array), Get (mengembalikan indeks logik elemen array), GetCapacity (mendapatkan kapasitas yang tersedia), dan operasi-operasi Array Dinamis, yaitu InsertAt (menambah elemen baru di indeks tertentu pada array), InsertLast (menambah elemen baru di akhir array), InsertFirst (menambah elemen baru di awal array), DeleteAt (menghapus elemen di indeks tertentu pada array), DeleteLast (menghapus elemen terakhir pada array), DeleteFirst (menghapus elemen pertama pada array), PrintArrayDin (melakukan *print* suatu array), ReverseArrayDin (melakukan *reverse* suatu array), CopyArrayDin (melakukan *copy* suatu array), dan SearchArrayDin (mencari elemen tertentu dalam array).

ADT Array Dinamis digunakan pada program utama untuk membuat list game yang kosong menggunakan konstruktor MakeArrayDin. Selain itu, ADT Array Dinamis juga digunakan pada prosedur STARTBNMO, LOADBNMO, dan CREATEGAME. Pada prosedur CREATEGAME, operasi InsertLast digunakan untuk menambahkan nama game yang ingin

ditambahkan oleh pemain. ADT Array Dinamis diimplementasikan dengan nama *file header* “arraydin.h”.

Sketsa pada ADT Array Dinamis terdiri dari :

- **ArrayDin MakeArrayDin()**
Digunakan untuk membuat ArrayDin kosong dengan ukuran InitialSize. Pada program yang kami buat, fungsi MakeArrayDin digunakan pada program utama untuk membuat daftar game yang kosong pada awal program dijalankan.
- **void DeallocateArrayDin(ArrayDin *array)**
Digunakan untuk melakukan dealokasi terhadap ArrayDin.
- **boolean IsEmpty(ArrayDin array)**
Digunakan untuk mengetahui apakah suatu array kosong. Pada program yang kami buat, fungsi IsEmpty digunakan pada program utama untuk mengecek apakah daftar antrian game kosong.
- **int Length(ArrayDin array)**
Digunakan untuk mengetahui banyaknya elemen efektif dalam ArrayDin. Pada program yang kami buat, fungsi Length digunakan dalam prosedur LISTGAME, QUEUEGAME, dan SAVEBNMO sebagai kondisi berhenti dalam suatu loop.
- **ElType Get(ArrayDin array, IdxType i)**
Digunakan untuk mengembalikan elemen array ke-i (indeks logik).
- **int GetCapacity (ArrayDin array)**
Digunakan untuk mendapatkan kapasitas array yang tersedia.
- **void InsertAt(ArrayDin *array, ElType el, IdxType,i)**
Digunakan untuk menambahkan elemen baru di indeks ke-i.
- **void InsertLast(ArrayDin *array, ElType el)**
Digunakan untuk menambahkan elemen baru di akhir array. Pada program yang kami buat, prosedur InsertLast digunakan untuk menambahkan game pada daftar game.
- **void InsertFirst(ArrayDin *array, ElType el)**
Digunakan untuk menambahkan elemen baru di awal ArrayDin.
- **void DeleteAt(ArrayDin *array, IdxType i)**
Digunakan untuk menghapus elemen di indeks ke-i pada ArrayDin.
- **void DeleteLast(ArrayDin *array)**
Digunakan untuk menghapus elemen terakhir ArrayDin.
- **void DeleteFirst(ArrayDin *array)**
Digunakan untuk menghapus elemen pertama ArrayDin.
- **void PrintArrayDin(ArrayDin array)**
Digunakan untuk melakukan *print* atau menampilkan suatu ArrayDin.
- **void ReverseArrayDin(ArrayDin *array)**
Digunakan untuk melakukan *reverse* suatu ArrayDin.
- **ArrayDin CopyArrayDin(ArrayDin array)**
Digunakan untuk melakukan *copy* suatu ArrayDin.
- **Idx SearchArrayDin(ArrayDin array, ElType el)**
Digunakan untuk melakukan pencarian ElType yang bernilai el pada ArrayDin dan mengembalikan indeks el ditemukan pada ArrayDin.

3.2 ADT ArraydinInt

Secara singkat, ADT ArraydinInt merupakan ADT Arraydin yang dimodifikasi menjadi array of integer. ADT ArraydinInt terdiri dari struktur ArrayDinInt yang terdiri atas array of integer, capacity (integer), dan Neff (integer). Prototype pada ADT ArraydinInt terdiri atas konstruktor pembuat tabel kosong (MakeArrayDin_AInt), destruktur dealokasi array dinamis (DeallocateArrayDin_AInt), selektor yang terdiri atas IsEmpty_AInt, Length_AInt, Get_AInt, dan GetCapacity_AInt, serta berbagai operasi seperti InsertAt_AInt, InsertLast_AInt, InsertFirst_AInt, DeleteAt_AInt, DeleteLast_AInt, DeleteFirst_AInt, PrintArrayDin_AInt, ReverseArrayDin_AInt, CopyArrayDin_AInt, dan SearchArrayDin_AInt.

Sketsa pada ADT ArraydinInt terdiri dari:

- **ArrayDinInt MakeArrayDin_AInt()**
Digunakan untuk membuat ArrayDinInt kosong dengan ukuran InitialSize. Pada program yang kami buat, fungsi MakeArrayDin_AInt digunakan dalam program Diner Dash untuk membuat array dinamis integer yang kosong pada awal program yang bernama makan3, durasi3, tahan3, dan harga3.
- **void DeallocateArrayDin_AInt(ArrayDinInt *array)**
Digunakan untuk melakukan dealokasi terhadap ArrayDinInt.
- **boolean IsEmpty_AInt(ArrayDinInt array)**
Digunakan untuk mengetahui apakah suatu ArrayDinInt kosong. Pada program yang kami buat, fungsi IsEmpty_AInt digunakan untuk memeriksa apakah daftar pesanan dan makanan kosong.
- **int Length_AInt(ArrayDinInt array)**
Digunakan untuk mengetahui banyaknya elemen efektif dalam ArrayDinInt. Pada program yang kami buat, fungsi Length_AInt digunakan sebagai kondisi berhenti pada prosedur header2.
- **int Get_AInt(ArrayDinInt array, IdxType i)**
Digunakan untuk mengembalikan elemen array ke-i (indeks logik) pada ArrayDinInt..
- **int GetCapacity_AInt(ArrayDinInt array)**
Digunakan untuk mendapatkan kapasitas ArrayDinInt yang tersedia.
- **void InsertAt_AInt(ArrayDinInt *array, int el, IdxType i)**
Digunakan untuk menambahkan elemen baru di indeks ke-i.
- **void InsertLast_AInt(ArrayDinInt *array, int el)**
Digunakan untuk menambahkan elemen baru di akhir ArrayDinInt. Pada program yang kami buat, prosedur InsertLast_AInt digunakan pada prosedur decrement cook.
- **void InsertFirst_AInt(ArrayDinInt *array, int el)**
Digunakan untuk menambahkan elemen baru di awal ArrayDinInt.
- **void DeleteAt_AInt(ArrayDinInt *array, IdxType i)**
Digunakan untuk menghapus elemen indeks ke-i pada ArrayDinInt. Pada program yang kami buat, prosedur DeleteAt_AInt digunakan pada prosedur serve untuk menghapus makanan yang sudah di-serve, ketahanannya, dan harganya.
- **void DeleteLast_AInt(ArrayDinInt *array)**

Digunakan untuk menghapus elemen di akhir ArrayDinInt.

- `void DeleteFirst_AInt(ArrayDinInt *array)`
Digunakan untuk menghapus elemen pertama ArrayDinInt.
- `void PrintArrayDin_AInt(ArrayDinInt array)`
Digunakan untuk melakukan *print* atau menampilkan suatu ArrayDinInt.
- `void ReverseArrayDin_AInt(ArrayDinInt *array)`
Digunakan untuk melakukan *reverse* suatu ArrayDinInt.
- `ArrayDinInt CopyArrayDin_AInt(ArrayDinInt array)`
Digunakan untuk melakukan *copy* suatu ArrayDinInt.
- `IdxType SearchArrayDin_AInt(ArrayDinInt array, int el)`
Digunakan untuk melakukan pencarian suatu elemen pada ArrayDinInt dan mengembalikan indeks elemen tersebut ditemukan pada ArrayDinInt.

3.3 ADT Mesin Karakter

ADT Mesin Karakter terdiri dari variabel extern bertipe char bernama currentChar, variabel extern bertipe boolean bernama EOP, dan variabel extern FILE bernama pita. Prototype pada ADT Mesin Karakter adalah START, ADV, GetCC, IsEOP, COMMAND, dan ADVC. Persoalan yang diselesaikan menggunakan ADT Mesin Karakter adalah ADT Mesin Kata. Alasan pemilihan penggunaan ADT Mesin Karakter adalah untuk pembuatan ADT Mesin Kata yang memerlukan primitif-primitif ADT Mesin Karakter untuk pemrosesan kata. Adapun, kami menambahkan prosedur baru dalam ADT Mesin Karakter untuk membaca masukan dari terminal yaitu prosedur COMMAND yang menyiapkan pita command dan prosedur ADVC yang memajukan pita sebanyak satu karakter. ADT Mesin Karakter diimplementasikan dengan nama file header “mesinkarakter.h”.

Sketsa pada ADT Mesin Karakter terdiri dari:

- `void START(char *directory)`
Digunakan untuk menyiapkan pita karakter untuk dibaca.
- `void ADV()`
Digunakan untuk memajukan pita sebanyak satu karakter.
- `char GetCC()`
Digunakan untuk mengirimkan currentChar.
- `boolean IsEOP()`
Digunakan untuk menandakan apakah currentChar sudah mencapai MARK.
- `void COMMAND()`
Digunakan untuk menyiapkan pita command untuk dibaca (Berupa pembacaan dari terminal)
- `void ADVC()`
Digunakan untuk memajukan pita command sebanyak satu karakter.

3.4 ADT Mesin Kata

ADT Mesin Kata adalah ADT untuk membaca kata dari masukan pemain dengan memanfaatkan ADT Mesin Karakter. ADT Mesin Kata terdiri dari struktur Word yang terdiri

dari array of character dan Length. Selain itu, ADT Mesin Kata terdiri dari variabel extern bertipe boolean bernama EndWord, variabel extern Word bernama currentWord, dan variabel extern Word bernama currentCMD. Prototype pada ADT Mesin Kata adalah IgnoreBlanks, STARTWORD, ADVWORD, CopyWord, PrintWord, CopyLine, ADVLine, IgnoreEnter, IgnoreBlanksCMD, STARTCOMMAND, ADVCOMMAND, CopyCommand, CopyCommandGame, dan STARTCOMMANDGAME.

ADT Mesin Kata digunakan pada prosedur STARTBNMO, LOADBNMO, CREATEGAME, DELETEGAME, dan QUEUEGAME. Pada prosedur STARTBNMO, prosedur STARTWORD digunakan untuk memulai pembacaan kata pada file .txt dan prosedur ADVLine digunakan untuk mengakuisisi baris. Pada prosedur LOADBNMO, STARTWORD digunakan untuk memulai pembacaan kata pada path dan ADVLine untuk mengakuisisi baris. Pada prosedur CREATEGAME, STARTCOMMANDGAME digunakan untuk memulai pembacaan command game yang dimasukkan pemain. Pada DELETEGAME dan QUEUEGAME, STARTCOMMAND digunakan untuk memulai pembacaan command yang dimasukkan oleh pemain. Selain itu pada games RNG, STARTCOMMAND juga digunakan untuk memulai pembacaan command yang dimasukkan oleh pemain. Alasan penggunaan ADT Mesin Kata adalah untuk melakukan pembacaan kata pada file pada saat command START dijalankan dan untuk membaca masukan pemain pada terminal. ADT Mesin Kata diimplementasikan dengan nama file header “mesinkata.h”.

Sketsa pada ADT Mesin Kata terdiri dari:

- **void IgnoreBlanks()**
Digunakan untuk mengabaikan satu atau beberapa blank.
- **void STARTWORD (char *filename)**
Digunakan untuk memulai pembacaan kata.
- **void ADVWORD()**
Digunakan untuk mengakuisisi kata menggunakan prosedur SalinWord.
- **void CopyWord ()**
Digunakan untuk mengakuisisi kata dan menyimpan dalam CurrentWord.
- **void CopyLine()**
Digunakan untuk mengakuisisi baris, menyimpan dalam currentWord.
- **void ADVLine()**
Digunakan untuk mengakuisisi kata menggunakan CopyLine.
- **void IgnoreEnter()**
Digunakan untuk mengabaikan satu atau beberapa ENTER.
- **void IgnoreBlanksCMD()**
Digunakan untuk mengabaikan satu atau beberapa BLANK pada command dari terminal.
- **void STARTCOMMAND()**
Digunakan untuk memulai pembacaan command.
- **void ADVCOMMAND()**
Digunakan untuk mengakuisisi kata menggunakan prosedur CopyCommand.
- **void CopyCommand()**
Digunakan untuk mengakuisisi command dan menyimpan dalam currentCMD.
- **void CopyCommandGame()**
Digunakan untuk mengakuisisi command dan menyimpan dalam currentCMD.

- `void STARTCOMMANDGAME()`
Digunakan untuk memulai pembacaan command line.

3.5 ADT Queue

ADT Queue terdiri dari struktur Queue yang terdiri dari array of char, `idxHead`, dan `idxTail`. Selain itu, ADT Queue mendefinisikan `IDX_UNDEF` adalah -1 dan `CAPACITY` adalah 100. Kreator dari ADT Queue adalah `CreateQueue`. Prototype dari ADT Queue adalah `isEmpty`, `isFull`, dan `length`. Selain itu, terdapat primitif Add dalam ADT Queue yaitu `enqueue` dan Delete dalam ADT Queue yaitu `dequeue`.

ADT Queue digunakan pada prosedur `QUEUEGAME` untuk mengecek apakah daftar antrian game kosong menggunakan `isEmpty` dan `enqueue` untuk menambahkan game ke dalam antrian. Selain itu, ADT Queue juga digunakan pada prosedur `QUIT` untuk melakukan pengosongan (`dequeue`) pada daftar antrian game saat command `QUIT` dijalankan. Alasan penggunaan ADT Queue adalah untuk membuat queue untuk daftar antrian game, menambahkan daftar antrian game, dan mengosongkan daftar antrian game. ADT Queue diimplementasikan dengan nama file header “`queue.h`”.

Sketsa pada ADT Queue terdiri dari:

- `CreateQueue(Queue *q)`
Digunakan untuk melakukan alokasi dan membuat sebuah `q` yang kosong.
- `isEmpty(Queue q)`
Digunakan untuk mengecek apakah `q` kosong. Pada program yang kami buat, `isEmpty` digunakan untuk mengecek apakah daftar antrian game kosong.
- `isFull(Queue q)`
Digunakan untuk mengecek apakah `q` sudah penuh.
- `length(Queue q)`
Digunakan untuk mengirimkan banyaknya elemen `q`. Pada program yang kami buat, `length` digunakan pada prosedur `DELETGAME`, `QUEUEGAME`, `SKIPGAME`, dan `PLAYGAME` sebagai kondisi berhenti dalam suatu loop.
- `enqueue(Queue *q, ElementType val)`
Digunakan untuk menambahkan `val` pada `q`. Pada program yang kami buat, `enqueue` digunakan dalam `QUEUEGAME` untuk menambahkan game pada antrian.
- `dequeue(Queue *q, ElementType *val)`
Digunakan untuk menghapus `val` pada `q`. Pada program yang kami buat, `dequeue` digunakan untuk mengosongkan daftar antrian game pada saat pemain melakukan `QUIT` dari program tersebut.
- `displayQueue(Queue q)`
Digunakan untuk menuliskan isi Queue dengan transversal.

3.6 ADT QueueInt

Secara singkat, ADT `QueueInt` merupakan ADT Queue yang dimodifikasi menjadi array of integer. ADT `QueueInt` terdiri dari struktur `QueueInt` yang terdiri atas array of integer, `idxHead`, dan `idxTail`. ADT ini mendefinisikan `IDX_UNDEF` sebagai -1 dan `CAPACITY` adalah 100. Kreator dari ADT `QueueInt` adalah `CreateQueueInt`. ADT `QueueInt` juga memiliki 3

prototype, yaitu isEmptyQInt, isFullQInt, dan lengthQInt. Selain itu, terdapat juga primitif Add berupa enqueueQInt, primitif Delete berupa dequeueQInt, serta displayQueueInt.

ADT QueueInt digunakan dalam game Diner Dash. Pada game Diner Dash, prosedur CreateQueueInt digunakan untuk membuat daftar pesanan dan makanan. Selain itu, prosedur isEmptyQInt digunakan untuk mengecek apakah daftar pesanan dan makanan kosong. Pada game Diner Dash, prosedur lengthQInt digunakan sebagai kondisi berhenti dalam suatu loop. Selain itu digunakan juga prosedur enqueueQInt dan dequeueQInt. Prosedur enqueueQInt digunakan untuk menambahkan daftar pesanan dan makanan, sedangkan dequeueQInt digunakan untuk pengosongan daftar pesanan dan makanan.

Sketsa pada ADT QueueInt terdiri dari:

- **void CreateQueueInt(QueueInt *q)**
Digunakan untuk melakukan alokasi dan membuat QueueInt kosong. Pada program yang kami buat, CreateQueueInt digunakan untuk membuat daftar pesanan dan makanan dalam game Diner Dash.
- **boolean isEmptyQInt(QueueInt q)**
Digunakan untuk mengecek apakah QueueInt kosong. Pada program yang kami buat, isEmptyQInt digunakan untuk mengecek apakah daftar pesanan dan makanan kosong.
- **boolean isFullQInt(QueueInt q)**
Digunakan untuk mengecek apakah QueueInt sudah penuh.
- **int lengthQInt(QueueInt q)**
Digunakan untuk mengirimkan banyaknya elemen QueueInt. Pada program yang kami buat, lengthQInt digunakan sebagai kondisi berhenti dalam suatu loop.
- **void enqueueQInt(QueueInt *q, int val)**
Digunakan untuk menambahkan val pada QueueInt. Pada program yang kami buat, enqueueQInt digunakan untuk menambahkan daftar pesanan dan makanan.
- **void dequeueQInt(QueueInt *q, int *val)**
Digunakan untuk menghapus val pada QueueInt. Pada program yang kami buat, dequeueQInt digunakan untuk mengosongkan daftar pesanan dan makanan.
- **void displayQueueInt(QueueInt q)**
Digunakan untuk menuliskan isi QueueInt secara traversal.

4 Program Utama

Program utama dibuat dalam file “main.c” dengan file header “console.h” dan stdio.h. Program utama akan membuat array dinamis yang kosong bernama GamesList dan membuat queue yang kosong yang bernama GamesQueue. Program utama juga terdiri dari boolean bernama EndProgram yang diinisiasi sebagai false di awal program dijalankan sebagai penanda bahwa program belum selesai. Kemudian, program utama akan menampilkan tampilan awal yang berisi “Selamat datang di BNMO” dan keterangan singkat permainan. Selanjutnya, program akan memasuki loop dan akan meminta command dari pemain. Command yang dapat dimasukkan pemain adalah START, LOAD <filename>, SAVE <filename>, CREATEGAME, LISTGAME, DELETGAME, QUEUEGAME, PLAYGAME, SKIPGAME <n>, QUIT, dan

HELP. Program akan mulai membaca command dari pemain dan command yang diterima akan diubah menjadi string menggunakan fungsi WordToString. Setelah itu, program akan mengecek apakah daftar games kosong atau tidak. Jika daftar games kosong dan command yang diberikan user adalah START maka program utama akan menjalankan START. Begitu pula dengan LOAD. Apabila daftar games kosong dan command yang dimasukkan command selain START atau LOAD, maka program akan memuat pesan agar pemain memasukkan games terlebih dahulu dengan LOAD.

Jika command yang dimasukkan adalah command yang valid, maka program akan menjalankan command sesuai yang diperintahkan pemain. Namun, jika command yang dimasukkan tidak valid, program akan mengeluarkan pesan bahwa command tidak dikenali dan akan terus meminta pemain memasukkan command hingga command yang dimasukkan valid.

5 Algoritma-Algoritma Menarik

Dalam pengerjaan tugas besar ini, kami menemukan beberapa algoritma menarik yang telah kami buat. Berikut penjelasan mengenai berbagai algoritma menarik yang kami temukan.

5.1 Command

Dalam spesifikasi tugas besar, dijelaskan bahwa pembacaan input dari user dalam program utama tidak diperbolehkan menggunakan scanf maupun fscanf. Oleh karena itu, kami membuat suatu algoritma baru untuk membaca input user. Algoritma tersebut terdapat dalam ADT Mesin Karakter (COMMAND dan ADVC) dan ADT Mesin Kata (IgnoreBlanksCMD, STARTCOMMAND, STARTCOMMANDGAME, ADVCOMMAND, CopyCommand, dan CopyCommandGame). Dalam program utama kami, kami mengimplementasikannya dalam beberapa prosedur, yaitu CREATEGAME, DELETGAME, dan QUEUEGAME.

Algoritma pembacaan input dari user ini menarik karena umumnya kami menggunakan scanf atau fscanf secara langsung tanpa harus menggunakan ADT mesin karakter dan ADT mesin kata dalam implementasinya. Dengan ini, kami dapat mempelajari algoritma baru untuk pembacaan input dari user ke program utama dengan mengimplementasikan ADT mesin karakter dan ADT mesin kata dalam pembacaan dari terminal.

Berikut algoritma dari pembacaan input pemain dari terminal:

```
1. void COMMAND() {
    command = stdin;
    ADVC();
}
2. void ADVC() {
    retval = fscanf(command,"%c",&currentChar);
}
3. void IgnoreBlanksCMD(){
    while (GetCC() == BLANK){
        ADVC();
    }
}
```

```

4. void STARTCOMMAND(){
    COMMAND();
    IgnoreBlanksCMD();
    if (currentChar == ENTER){
        EndWord = true;
    } else {
        EndWord = false;
        CopyCommand();
    }
}

5. void STARTCOMMANDGAME(){
    COMMAND();
    IgnoreBlanksCMD();
    if(currentChar == ENTER){
        EndWord = true;
    } else {
        EndWord = false;
        CopyCommandGame();
    }
}

6. void ADVCOMMAND(){
    IgnoreBlanksCMD();
    if (currentChar == ENTER && !EndWord){
        EndWord = true;
    } else{
        CopyCommand();
        IgnoreBlanksCMD();
    }
}

7. void CopyCommand(){
    int i = 0;
    while ((currentChar != BLANK) && (currentChar != ENTER) && i
    != NMax) {
        currentCMD.TabWord[i] = currentChar;
        ADVC();
        i++;
    }
    if (i >= NMax) currentCMD.Length = NMax;
    else currentCMD.Length = i;
}

8. void CopyCommandGame(){
    int i = 0;
    while ((currentChar != ENTER) && i != NMax) {

```

```

        currentCMD.TabWord[i] = currentChar;
        ADVG();
        i++;
    }
    if (i >= NMax) currentCMD.Length = NMax;
    else currentCMD.Length = i;
}

```

5.2 RNG (Random Number Generator)

Dalam pembuatan program Random Number Generator, terdapat fungsi randomnumber yang akan men-*generate* angka acak dan mengembalikan angka tersebut. Dalam pembuatan fungsi randomnumber, kami meng-*include* library time.h yang merupakan file header yang didefinisikan dalam C Standard Library. Dalam pembuatan fungsi randomnumber, kami menggunakan fungsi rand untuk men-*generate* angka acak. Akan tetapi, angka yang dihasilkan akan sama setiap kali program dijalankan. Oleh karena itu, fungsi srand digunakan untuk memberikan seed pada fungsi rand. Kemudian, kami menggunakan fungsi time untuk melakukan seed pada randomizer sehingga nilai menjadi berubah-ubah. Oleh karena itu, setiap kali kita memanggil fungsi rand, nilainya akan selalu berubah ubah setiap waktunya.

Berikut adalah algoritma dari fungsi randomnumber:

```

#include <time.h>
int randomnumber (){
    int angka;
    srand(time(NULL));
    angka = rand % 100+1;
    return angka;
}

```

6 Data Test

Dalam program yang telah kami buat, yaitu BNMO, terdapat beberapa fitur atau command yang dijalankan oleh pemain. Untuk memastikan berbagai fitur dan command dapat dijalankan dengan baik dan benar, perlu dilakukan *testing* terhadap beberapa kondisi. Berikut *testing* dari beberapa fitur dan command dalam program kami, Selain itu, terdapat juga penjelasan data test, input, serta output yang akan diberikan program dalam tiap kondisi.

6.1 Data Test START

Pada awal program dijalankan, command yang dapat dijalankan adalah START atau LOAD. Pada saat pemain memilih command START, file konfigurasi default yang tersimpan dalam config.txt akan dibaca. File konfigurasi default tersebut berisi list game yang dapat dimainkan. Output yang baik adalah program akan membaca file konfigurasi dan menampilkan pesan “File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan.”.

```
'#####:##:: #: '#####:
##... #: #: #: #: #: '#####:##... #:
##:: #: #: #: #: #: '#####:##:: #:
#####: ## # #: # # #: #: #: #:
##... #: #: #: #: #: #: #: #: #:
##:: #: #: #: #: #: #: #: #: #:
#####: #: #: #: #: #: #: #:
.....:.....:.....:.....:.....:
===== SELAMAT DATANG DI BNMO! =====
Masukkan command START atau LOAD <filename> untuk memulai permainan!
Note: START untuk konfigurasi default dan LOAD <filename> untuk konfigurasi sesuai filename.

ENTER COMMAND: START
File konfigurasi sistem berhasil dibaca. BNMO berhasil dijalankan.
```

Gambar 6.1.1 Tampilan START jika file konfigurasi berhasil dibaca dan BNMO berhasil

6.2 Data Test LOAD

Selain command START, pengguna juga diperbolehkan untuk memulai program dengan command LOAD. Namun, perlu diperhatikan bahwa LOAD harus diikuti dengan nama file yang ingin dibuka oleh pemain. File tersebut berisi list game yang dapat dimainkan. Pada saat pemain memilih command LOAD, file konfigurasi default dalam bentuk config.txt tersebut akan dibaca. Command tersebut diketahui bekerja dengan baik dan benar ketika file berhasil dibaca, BNMO berhasil dijalankan, serta disertai output “Save file berhasil dibaca. BNMO berhasil dijalankan”.

```
'#####:##:: #: '#####:
##... #: #: #: #: #: '#####:##... #:
##:: #: #: #: #: #: '#####:##:: #:
#####: ## # #: # # #: #: #: #:
##... #: #: #: #: #: #: #: #: #:
##:: #: #: #: #: #: #: #: #: #:
#####: #: #: #: #: #: #: #:
.....:.....:.....:.....:.....:
===== SELAMAT DATANG DI BNMO! =====
Masukkan command START atau LOAD <filename> untuk memulai permainan!
Note: START untuk konfigurasi default dan LOAD <filename> untuk konfigurasi sesuai filename.

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

Gambar 6.2.1 Tampilan LOAD jika terdapat file konfigurasi yang sesuai dengan command pengguna dan file berhasil dibaca

Jika file konfigurasi tidak ditemukan, program akan mengidentifikasinya sebagai command yang salah, seperti tampilan di bawah ini.

```
ENTER COMMAND: LOAD savefile3.txt
Command tidak dikenali, silahkan masukkan command yang valid.
```

Gambar 6.2.2 Tampilan LOAD jika tidak ditemukan file konfigurasi yang sesuai dengan command pengguna

6.3 Data Test SAVE

Untuk menyimpan *state game* pemain dalam file, pemain dapat menggunakan command SAVE. Perlu diperhatikan bahwa SAVE harus diikuti dengan nama file yang akan disimpan pada disk. Command SAVE diketahui bekerja dengan baik dan benar jika *state game* berhasil disimpan dalam file yang telah disebutkan namanya. File yang disebutkan dapat berupa file lama yang akan di-"update" atau file baru yang ingin dibentuk.

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

ENTER COMMAND: SAVE savefile2.txt
Save file berhasil disimpan.
```

Gambar 6.3.1 Tampilan SAVE jika *state game* berhasil disimpan dalam file tertentu

```
data > ≡ savefile2.txt
1  8
2  RNG
3  Diner DASH
4  DINOSAUR IN EARTH
5  RISEWOMAN
6  EIFFEL TOWER
7  Alstrukdat
8  BNMO
9  STUMBLE
```

Gambar 6.3.2 Tampilan file yang telah berhasil menyimpan *state game* terkini

6.4 Data Test CREATE GAME

Untuk menambahkan daftar game pada list game, pemain dapat menggunakan command CREATE GAME. Setelah memasukkan command CREATE GAME, pemain akan diminta untuk memasukkan nama game yang ingin ditambahkan. Nama game yang baru dimasukkan akan muncul pada daftar game pada urutan paling akhir.

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

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. Stumble
```

Gambar 6.4.1 Tampilan jika permainan berhasil ditambahkan pada list game

6.5 Data Test LIST GAME

Ketika pemain ingin melihat daftar permainan yang tersedia, pemain dapat menggunakan command LIST GAME. Setelah pemain memasukkan command LIST GAME, daftar game yang tersedia akan langsung ditampilkan.

```
ENTER COMMAND: LOAD savefile2.txt
Save file berhasil dibaca. BNMO berhasil dijalankan.

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. Alstrukdat
7. BNMO
```

Gambar 6.5.1 Tampilan jika daftar permainan berhasil ditampilkan dengan lengkap

Testing pada command LIST GAME dapat dilakukan dengan menyesuaikan data daftar permainan yang terkandung dalam file konfigurasi yang sebelumnya sudah disimpan.

```
data > ≡ savefile2.txt
1 7
2 RNG
3 Diner DASH
4 DINOSAUR IN EARTH
5 RISEWOMAN
6 EIFFEL TOWER
7 Alstrukdat
8 BNMO
```

Gambar 6.5.2 Tampilan data daftar permainan dalam file konfigurasi

6.6 Data Test DELETE GAME

Untuk menghapus daftar game yang dibuat oleh pemain, pemain dapat menggunakan command DELETE GAME. Game yang dapat dihapus pada daftar game adalah game yang dibuat secara custom oleh pemain. Jika pemain mencoba melakukan penghapusan pada 5 game pertama pada file konfigurasi, maka program akan menampilkan pesan bahwa game gagal dihapus. Selain itu, game akan gagal dihapus jika pemain memasukkan input nomor daftar game yang tidak valid (angka yang tidak ada pada daftar game). Berikutnya, game akan gagal dihapus jika pemain ingin menghapus game yang berada dalam daftar antrian game. Selain itu, jika daftar game yang hendak dihapus adalah game buatan pemain, maka program akan menghapus game dari daftar game dan menampilkan pesan bahwa game berhasil dihapus.

```

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. Stumble

Masukkan nomor game yang akan dihapus: 3

Game gagal dihapus

```

Gambar 6.6.1 Tampilan jika pemain menghapus 5 game pertama pada file konfigurasi

```

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. Stumble

Masukkan nomor game yang akan dihapus: 6

Game berhasil dihapus

```

Gambar 6.6.2 Tampilan jika pemain menghapus game custom buatan pemain

```

ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
Daftar antrian game-mu kosong, silahkan tambahkan game ke antrian.

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: 4

Game berhasil ditambahkan ke dalam daftar antrian.

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: 4

Game gagal dihapus

```

Gambar 6.6.3 Tampilan jika pemain hendak menghapus daftar game yang berada pada antrian game

6.7 Data Test QUEUE GAME

Pemain dapat menambahkan game yang akan dimainkan menggunakan command QUEUE GAME. Setelah command QUEUE GAME dimasukkan, pemain akan diminta nomor game yang mau ditambahkan ke dalam antrian. Jika nomor game yang dimasukkan valid (nomor game yang tertera pada list), game akan dimasukkan ke daftar antrian game. Game yang dimasukkan paling akhir akan dimasukkan ke dalam daftar antrian paling belakang. Jika nomor game yang dimasukkan tidak valid, program akan menampilkan pesan “Nomor permainan tidak valid, silahkan masukkan nomor game pada list.”.

```
ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
Daftar antrian game-mu kosong, silahkan tambahkan game ke antrian.

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 ke dalam daftar antrian.
```

Gambar 6.7.1 Tampilan jika nomor game dimasukkan valid dan queue masih kosong

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

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: 1

Game berhasil ditambahkan ke dalam daftar antrian.
```

Gambar 6.7.2 Tampilan jika nomor game yang dimasukkan valid dan queue tidak kosong

```

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

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: 10

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

```

Gambar 6.7.3 Tampilan jika nomor game yang dimasukkan tidak valid

```

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

Berikut adalah daftar game yang tersedia
1. RNG
2. Diner DASH
3. DINOSAUR IN EARTH
4. RISEWOMAN
5. EIFFEL TOWER
6. Alstrukdat
7. BNMO
8. STUMBLE
Nomor Game yang mau ditambahkan ke antrian: 1

Game berhasil ditambahkan ke dalam daftar antrian.

ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
1. RNG
2. RNG

```

Gambar 6.7.4 Tampilan jika game yang sama ditambahkan lebih dari sekali

6.8 Data Test **PLAY GAME**

Pemain dapat memainkan game yang telah dimasukkan ke dalam daftar antrian game menggunakan command **PLAY GAME**. Jika belum ada game yang dimasukkan ke dalam daftar antrian, program akan menampilkan pesan “Tidak ada game dalam antrian permainan Anda!”. Jika daftar antrian game tidak kosong, game yang akan dimainkan adalah game dengan urutan paling awal terlebih dahulu. Game yang dapat dimainkan hanya RNG dan Dinner Dash. Jika game yang dimainkan selain RNG atau Diner Dash, program akan menampilkan pesan bahwa game sedang dalam maintenance dan belum dapat dimainkan.

```

ENTER COMMAND: PLAY GAME
Tidak ada game dalam antrian permainan Anda!

```

Gambar 6.8.1 Tampilan jika tidak ada game dalam daftar antrian game

```

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

Game DINOSAUR IN EARTH masih dalam maintenance, belum dapat dimainkan.

```

Gambar 6.8.2 Tampilan jika game yang akan dimainkan bukan RNG/Diner Dash

```

ENTER COMMAND: PLAY GAME
Berikut adalah daftar Game-mu
1. RNG

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

```

Gambar 6.8.3 Tampilan jika game yang dimainkan RNG/Diner Dash

6.9 Data Test SKIP GAME

Untuk melewati permainan, pemain dapat menggunakan command SKIP GAME <n> dimana n adalah banyaknya permainan yang ingin dilewat. Ketika pemain memasukkan command SKIP GAME <n>, maka program akan melewati sebanyak n game dan akan memainkan game ke n+1 pada queue. Program juga akan menampilkan daftar antrian game yang baru yaitu daftar antrian yang telah melewati n game. Jika n yang dimasukkan lebih besar atau sama dengan banyak daftar antrian game, program akan menuliskan pesan “Daftar antrian game-mu kosong, silahkan tambah game ke antrian”.

```

ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
1. DINOSAUR IN EARTH
2. RISEWOMAN
3. EIFFEL TOWER
4. EIFFEL TOWER
5. Diner DASH

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: 1

Game berhasil ditambahkan ke dalam daftar antrian.

ENTER COMMAND: SKIP GAME 1
Berikut adalah daftar Game-mu
1. RISEWOMAN
2. EIFFEL TOWER
3. EIFFEL TOWER
4. Diner DASH
5. RNG

Game RISEWOMAN masih dalam maintenance, belum dapat dimainkan.

```

Gambar 6.9.1 Tampilan jika n yang dimasukkan berada dalam range jumlah antrian game sehingga masih ada daftar game yang dapat dijalankan

```

ENTER COMMAND: QUEUE GAME
Berikut adalah daftar antrian game-mu
1. DINOSAUR IN EARTH
2. RISEWOMAN

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: 5

Game berhasil ditambahkan ke dalam daftar antrian.

ENTER COMMAND: SKIP GAME 3
Tidak ada game dalam antrian permainan Anda!

```

Gambar 6.9.2 Tampilan jika n yang dimasukkan di luar range jumlah antrian game sehingga sudah tidak ada daftar game yang dapat dijalankan

6.10 Data Test QUIT

Untuk keluar dari program, pemain harus menggunakan command QUIT. Setelah pemain memasukkan command QUIT, terdapat pesan “Anda keluar dari game BNMO. Bye bye ...” dan pemain akan keluar dari program.

```

ENTER COMMAND: QUIT
Anda keluar dari game BNMO.
Bye bye ...

```

Gambar 6.10.1 Tampilan jika pemain berhasil keluar dari program

6.11 Data Test HELP

Command HELP digunakan untuk membantu user dalam mengidentifikasi berbagai command yang valid. Setelah pemain memasukkan command HELP, ditampilkan bantuan untuk berbagai command yang terdefinisi.

```

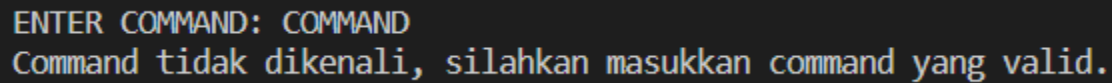
ENTER COMMAND: HELP
-----HELP-----
START                : Command yang membaca file konfigurasi untuk menjalankan BNMO.
LOAD <filename>      : Command yang membaca save file <filename> yang berisi list game yang dapat dimainkan, histori, dan scoreboard game.
SAVE <filename>      : Command yang digunakan untuk menyimpan state game pemain saat ini ke dalam suatu file.
CREATE GAME          : Command yang digunakan untuk menambahkan game baru pada daftar game.
LIST GAME            : Command yang digunakan untuk menampilkan daftar game yang disediakan oleh sistem.
DELETE GAME          : Command yang digunakan untuk menghapus sebuah game dari daftar game.
QUEUE GAME           : Command yang digunakan untuk mendaftarkan permainan ke dalam list.
PLAY GAME            : Command yang digunakan untuk memainkan sebuah permainan.
SKIP GAME <n>        : Command yang digunakan untuk melewati permainan sebanyak n.
HELP                 : Bantuan command command yang ada.
QUIT                 : Command untuk keluar dari program.

```

Gambar 6.11.1 Tampilan jika bantuan command sudah valid

6.12 Data Test COMMAND LAIN

Command Lain merupakan suatu fitur yang memeriksa apakah command yang digunakan oleh pemain valid atau tidak. Jika command tidak valid, pemain akan diminta untuk memasukkan command lagi.

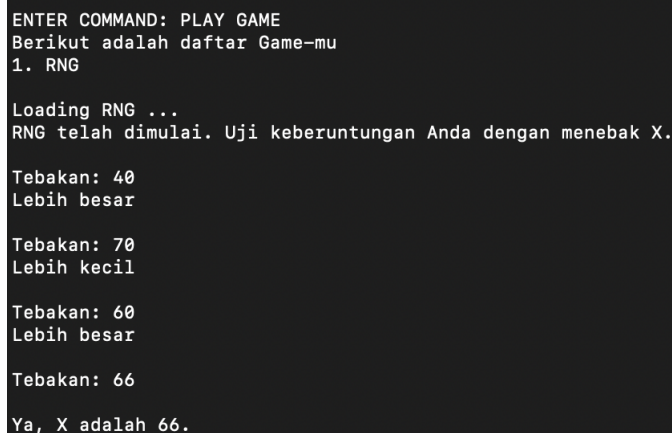


```
ENTER COMMAND: COMMAND
Command tidak dikenali, silahkan masukkan command yang valid.
```

Gambar 6.12.1 Tampilan jika command yang digunakan pemain tidak valid

6.13 Data Test RNG

RNG adalah sebuah permainan di mana pemain harus menebak angka yang telah diacak oleh program. Setiap pemain diberikan 10 kali kesempatan untuk menebak angka. Jika kesempatan menebak sudah habis, maka program akan mengeluarkan pesan bahwa kesempatan menebak sudah habis dan program akan memberi tahu jawaban yang benar kepada pemain. RNG akan dimainkan jika RNG berapa pada urutan nomor 1 daftar antrian game.



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

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

Tebakan: 40
Lebih besar

Tebakan: 70
Lebih kecil

Tebakan: 60
Lebih besar

Tebakan: 66

Ya, X adalah 66.
```

Gambar 6.13.1 Tampilan jika pemain berhasil menebak X


```

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

Tebakan: 100
Lebih kecil

Tebakan: 90
Lebih kecil

Tebakan: 70
Lebih kecil

Tebakan: 60
Lebih kecil

Tebakan: 50
Lebih besar

Tebakan: 40
Lebih besar

Tebakan: 30
Lebih besar

Tebakan: 20
Lebih besar

Tebakan: 10
Lebih besar

Tebakan: 9
Lebih besar
Maaf kesempatan untuk menebak sudah habis.
Jawaban yang benar adalah 53.

```

Gambar 6.13.2 Tampilan jika pemain gagal menebak X

6.14 Data Test DINER DASH

Diner Dash adalah sebuah permainan dimana pemain dapat memasukkan command yaitu COOK, SERVE, atau SKIP. Ketika game diner dash baru dijalankan, pemain akan disajikan tabel daftar pesanan yang berisi makanan, durasi memasak, ketahanan dan harga, serta saldo awal yang nilainya 0. Setiap kali pemain melakukan SKIP, maka jumlah pelanggan akan tetap bertambah. Selain itu, command yang tidak valid akan menampilkan pesan “INVALID COMMAND” dan jumlah pelanggan tidak akan bertambah. Jika command yang dimasukkan pada game Diner Dash valid, program akan melakukan perintah. Makanan yang dapat disajikan (serve) harus berdasarkan urutan pesanan. Jika makanan belum diserve hingga ketahanan makanan habis, makanan harus dimasak ulang. Program akan berhenti jika pelanggan telah melebihi 7 atau telah melayani 15 pelanggan.


```

ENTER COMMAND: PLAY GAME
Berikut adalah daftar Game-mu
1. Diner DASH

Loading Diner DASH ...
Selamat Datang di Diner Dash!
=====

Saldo: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----|-----|-----|-----
M0 | 3 | 5 | 17371
M1 | 4 | 2 | 18761
M2 | 4 | 2 | 49526

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----|-----
|

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----|-----
|

Masukkan Command: COMMAND
INVALID COMMAND

Masukkan Command: COOK
INVALID COMMAND

Masukkan Command: SERVE
INVALID COMMAND

```

Gambar 6.14.1 Tampilan awal diner dash dan command yang tidak valid untuk game Diner Dash

```

Masukkan Command: COOK M0
berhasil memasak M0

=====

Saldo: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----|-----|-----|-----
M0 | 3 | 5 | 17371
M1 | 4 | 2 | 18761
M2 | 4 | 2 | 49526
M3 | 1 | 4 | 41118

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----|-----
M0 | 3
|

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----|-----
|

Masukkan Command: COOK M1
berhasil memasak M1

```

Gambar 6.14.2 Tampilan jika command valid, program akan menjalankan perintah dan pesanan akan bertambah 1

```
Saldo: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----|-----|-----|-----
M0 | 3 | 5 | 17371
M1 | 4 | 2 | 18761
M2 | 4 | 2 | 49526
M3 | 1 | 4 | 41118
M4 | 4 | 3 | 31268

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----|-----
M0 | 2
M1 | 4

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----|-----
|

Masukkan Command: SERVE M3
Tidak ada makanan yang dapat disajikan

Masukkan Command: SERVE M0
Tidak ada makanan yang dapat disajikan
```

Gambar 6.14.3 Tampilan jika command valid tetapi tidak ada makanan yang disajikan

```
=====
Saldo: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----|-----|-----|-----
M0 | 3 | 5 | 17371
M1 | 4 | 2 | 18761
M2 | 4 | 2 | 49526
M3 | 1 | 4 | 41118
M4 | 4 | 3 | 31268
M5 | 4 | 3 | 35349
M6 | 1 | 4 | 37014

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----|-----
M1 | 2
M2 | 3
M3 | 1

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----|-----
M0 | 5

Masukkan Command: SERVE M0
Berhasil menyajikan M0

Makanan M3 telah selesai dimasak
```

Gambar 6.14.4 Tampilan jika command serve berhasil dijalankan, program juga akan menampilkan makanan yang telah selesai masak jika terdapat pesanan yang sudah selesai dimasak

```

Masukkan Command: COOK M0
berhasil memasak M0

=====
Saldo: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M0 | 5 | 4 | 18883
M1 | 1 | 1 | 49879
M2 | 5 | 2 | 16359
M3 | 3 | 2 | 27670

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----
M0 | 5

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----
|

Masukkan Command: SKIP
=====
Saldo: 0

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M0 | 5 | 4 | 18883
M1 | 1 | 1 | 49879
M2 | 5 | 2 | 16359
M3 | 3 | 2 | 27670
M4 | 4 | 3 | 36538

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----
M0 | 4

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----
|

Masukkan Command:

```

Gambar 6.14.5 Tampilan jika pemain memasukkan SKIP, pelanggan akan bertambah, durasi memasak akan berkurang

```

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M0 | 3 | 5 | 17371
M1 | 4 | 2 | 18761
M2 | 4 | 2 | 49526
M3 | 1 | 4 | 41118
M4 | 4 | 3 | 31268
M5 | 4 | 3 | 35349
M6 | 1 | 4 | 37814

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----
M1 | 2
M2 | 3
M3 | 1

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----
M0 | 5

Masukkan Command: SERVE M0
Berhasil menyajikan M0

Makanan M3 telah selesai dimasak

=====
Saldo: 17371

Daftar Pesanan
Makanan | Durasi Memasak | Ketahanan | Harga
-----
M1 | 3 | 5 | 17371
M2 | 4 | 2 | 18761
M3 | 4 | 2 | 49526
M4 | 1 | 4 | 41118
M5 | 4 | 3 | 31268
M6 | 4 | 3 | 35349
M7 | 1 | 4 | 37814

Daftar Makanan yang sedang dimasak
Makanan | Sisa Durasi Memasak
-----
M1 | 1
M2 | 2

Daftar makanan yang dapat disajikan
Makanan | Sisa Ketahanan Memasak
-----
M3 | 4

Game Over!
Saldo: 17371

```

Gambar 6.14.6 Tampilan jika pelanggan sudah melebihi 7 maka game Diner Dash akan selesai dan program akan menampilkan saldo akhir

7 Test Script

Seperti yang telah disebutkan sebelumnya, diperlukan *testing* untuk memastikan bahwa fitur-fitur dan command-command yang ada dalam program dapat berjalan dengan baik dan benar. Oleh karena itu, dalam bagian Test Script ini akan dijelaskan beberapa skenario test yang dimungkinkan untuk semua fitur dan command.

Berikut skenario test dijelaskan lebih lanjut dalam bentuk tabel.

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Start	Memeriksa apakah permainan dapat dijalankan	Melakukan compile dan masukkan command START	6.1 Data Test Start	Dibaca file konfigurasi default yang berisi list game yang dapat dimainkan	Sesuai Gambar 6.1.1
2	Load	Memeriksa apakah file berhasil dibaca dan BNMO berhasil dijalankan	Melakukan compile dan memasukkan command LOAD yang disertai nama file.	6.2 Data Test Load	Jika terdapat save file yang tersedia, save file yang dipilih berhasil dibaca dan permainan berhasil dijalankan. Apabila file tidak tersedia, menampilkan pesan bahwa File tidak exist.	Sesuai Gambar 6.2.1
3	Save	Memeriksa apakah state game berhasil disimpan dalam file	Melakukan compile dan masukkan command SAVE beserta nama file.	6.3 Data Test Save	Menyimpan state game pemain ke dalam suatu file.	Sesuai Gambar 6.3.1
4	Create Game	Memeriksa apakah permainan baru dapat dibuat	Memasukkan command CREATE GAME. Lalu, masukkan nama game yang ingin ditambahkan.	6.4 Data Test Create Game	Menampilkan pesan bahwa daftar game berhasil ditambah dan menambahkan nama game ke daftar game.	Sesuai Gambar 6.4.1

5	List Game	Memeriksa apakah daftar permainan berhasil ditampilkan	Memasukkan command LIST GAME.	6.5 Data Test List Game	Daftar permainan berhasil ditampilkan secara lengkap	Sesuai Gambar 6.5.1
6	Delete Game	Memeriksa apakah permainan berhasil dihapus	Memasukkan command DELETE GAME. Lalu, memasukkan nomor game yang ingin dihapus.	6.6 Data Test Delete Game	Permainan yang nomornya sesuai dengan input pemain yang valid berhasil dihapus	Sesuai Gambar 6.6.1, 6.6.2, dan 6.6.3
7	Queue Game	Memeriksa apakah permainan berhasil didaftarkan ke dalam antrian	Memasukkan command QUEUE GAME. Lalu, memasukkan nomor game yang ingin dimasukkan ke dalam antrian game.	6.7 Data Test Queue Game	Permainan yang nomornya sesuai dengan input pemain yang valid berhasil didaftarkan ke dalam antrian permainan	Sesuai Gambar 6.7.1, 6.7.2, 6.7.3, dan 6.7.4
8	Play Game	Memeriksa apakah permainan yang sesuai spesifikasi dapat dijalankan	Memasukkan command PLAY GAME.	6.8 Data Test Play Game	Menjalankan game dengan urutan antrian paling atas terlebih dahulu. Jika games adalah RNG/Dinner Dash, maka games akan dimainkan. Apabila games selain RNG atau Dinner Dash, menampilkan pesan error bahwa games sedang dalam maintenance.	Sesuai Gambar 6.8.1, 6.8.2, dan 6.8.3
9	Skip Game	Memeriksa apakah permainan berhasil dilewatkan	Memasukkan command SKIP GAME dan banyak permainan yang ingin dilewat.	6.9 Data Test Skip Game	Melewati permainan. Menampilkan loading game yang akan dimainkan berikutnya atau tidak ada permainan lagi	Sesuai Gambar 6.9.1 dan 6.9.2

					dalam daftar game.	
10	Quit	Memeriksa apakah pemain dapat keluar dari program	Memasukkan command QUIT	6.10 Data Test Quit	Keluar dari program dan menampilkan pesan bahwa pemain telah keluar dari BNMO.	Sesuai Gambar 6.10.1
11	Help	Memeriksa apakah tampilan bantuan sudah sesuai	Memasukkan command HELP	6.11 Data Test Help	Menampilkan deskripsi tentang command command yang valid.	Sesuai Gambar 6.11.1
12	Command lain	Memeriksa apakah fitur akan memeriksa command hingga valid	Memasukkan command selain START, LOAD <filename>, SAVE <filename>, CREATE GAME, LIST GAME, DELETE GAME, QUEUE GAME, PLAY GAME, SKIP GAME <n>, QUIT, dan HELP.	6.12 Data Test Command Lain	Menampilkan pesan bahwa command tidak dikenali, silahkan masukkan command yang valid. Setelah itu, program akan meminta command lagi.	Sesuai Gambar 6.12.1
13	RNG	Memeriksa apakah program RNG dapat dimainkan dan mengeluarkan output yang sesuai	Melakukan queue game dan memasukkan nomor 1 pada game yang ingin dimasukkan pada antrian. Lalu, RNG akan dimainkan jika pemain memasukkan command PLAY GAME dan RNG merupakan urutan teratas pada queue game. Kemudian, pemain memasukkan tebakan angkanya.	6.13 Data Test RNG	Menampilkan apakah pemain berhasil menebak angka random dalam 10 kesempatan/tidak. Setiap pemain salah menjawab, program akan menampilkan clue apakah angka tebakan lebih besar atau lebih kecil dari angka yang dirandom.	Sesuai Gambar 6.13.1 dan 6.13.2
14	Diner Dash	Memeriksa apakah game Diner Dash dapat dimainkan dan	Melakukan queue game dan memasukkan nomor 1 pada game yang ingin dimasukkan pada antrian. Lalu, Diner Dash akan dimainkan jika pemain	6.14 Data Test Diner Dash	Menampilkan saldo akhir pemain yang berdasarkan pelanggan dan	Sesuai Gambar 6.14.1, 6.14.2, 6.14.3, 6.14.4, 6.14.5, dan 6.14.6

		mengeluarkan output yang sesuai	memasukkan command PLAY GAME dan Diner Dash merupakan urutan teratas pada queue game. Kemudian, pemain memasukkan salah satu dari 3 command pada game Diner Dash diantaranya yaitu COOK, SERVE, SKIP.		harga makanannya.	
--	--	---------------------------------	---	--	-------------------	--

8 Pembagian Kerja dalam Kelompok

Nama	NIM	Pembagian Kerja
Ghaylan Muhammad Fatih	18221042	PLAY GAME, SKIP GAME, Diner Dash
Gracia Theophilia	18221078	LIST GAME, DELETE GAME, QUEUE GAME, membuat laporan, mengontak asisten
Fikri Naufal Hamdi	18221096	CREATE GAME, ADT, Main Program
Reinhart Wisely Lim	18221154	START, LOAD, SAVE
Esther Regina	18221086	QUIT, HELP, COMMAND LAIN, RNG, membuat laporan

9 Lampiran

9.1 Deskripsi Tugas Besar 2

BNMO (dibaca: Binomo) adalah sebuah robot video game console yang dimiliki oleh Indra dan Doni. Dua bulan yang lalu, ia mengalami kerusakan dan telah berhasil diperbaiki. Sayangnya, setelah diperbaiki ia justru mendapatkan lebih banyak bug dalam sistemnya. Oleh karena itu, Indra dan Doni mencari programmer lain yang lebih handal untuk ulang memprogram robot video game console kesayangannya.

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.

9.2 Notulen Rapat

Rapat 1 : 31 Oktober 2022

- Topik : Pembagian Tugas
- Media komunikasi : LINE
- Kesimpulan :
Esther = QUIT, HELP, COMMAND LAIN, RNG
Lim = START, LOAD, SAVE
Gracia = LISTGAME, DELETGAME, QUEUEGAME
Ghaylan = PLAYGAME, DINNER DASH, SKIPGAME
Nopal = CREATEGAME, ADT, MAIN PROGRAM

9.3 Log Activity Anggota Kelompok

NIM	Deskripsi	Tanggal
18221154	Membuat grup line dan membuat google drive untuk keperluan tugas besar	28 Oktober 2022
18221096	Membuat repository github untuk pengerjaan tugas besar	28 Oktober 2022
18221042 18221078 18221096 18221154 18221086	Melakukan pembagian tugas	31 Oktober 2022
18221096	Membuat ADT Mesin Karakter, Mesin Kata, Queue	1 November 2022
18221042	Membuat Play Game & Skip Game	1 November 2022
18221096	Membuat main.c, maing.c (testing play game & skip game)	2 November 2022
18221154	Membuat STARTBNMO, update mesin karakter, mesin kata,	2 November 2022
18221086	Membuat help, quit, RNG	2 November 2022
18221154	Fix STARTBNMO, update mesin kata, update mesin karakter	3 November 2022

18221096	Menambahkan Print Word pada mesin kata, update ADT Mesin Kata & Karakter untuk Command	3 November 2022
18221154	Menambahkan length, concat, load, membuat SAVE	3 November 2022
18221096	Benerin IgnoreBlanksCMD, update comparestring, implementasi load di main.c	3 November 2022
18221078	Membuat list game, delete game, queue game	4 November 2022
18221042	Update play game, main, skip game	4 November 2022
18221086	Mengosongkan antrian game di quit, update RNG	4 November 2022
18221096	Update main.c (menambahkan help, quit, create game), update README, benerin dequeue error di quit	4 November 2022
18221154	Update STARTBNMO, test savefile, update main.c	4 November 2022
18221086	Update RNG	5 November 2022
18221042 18221078 18221096 18221154 18221086	Asistensi 1	5 November 2022
18221042	Membuat dinner dash	8 November 2022
18221078	Update console.c & console.h	8 November 2022
18221086	Membuat laporan bagian ringkasan	8 November 2022
18221086	Membuat laporan bagian ADT Array Dinamis, Mesin Kata, Mesin Karakter, Queue, Program Utama	9 November 2022

18221078	Menambahkan laporan bagian ringkasan, ADT Array Dinamis, ADT Queue dan ADT Mesin Kata	9 November 2022
18221042	Update diner dash, update skip game, play game	9 November 2022
18221086	Update RNG.c dan RNG.h	9 November 2022
18221154	Update console (save), update mesin (menambahkan driver mesinkar, mesinkata)	10 November 2022
18221078	Membuat laporan bagian spesifikasi mekanisme load, algoritma menarik Command, ADT QueueInt, menambahkan ADT Mesin Karakter, ADT Mesin Kata, dan ADT ArrayDin	10 November 2022
18221086	Membuat laporan bagian spesifikasi game RNG, algoritma menarik RNG	10 November 2022
18221096	Update main, console, makefile, membuat ADT QueueInt, fixing Printing GameQueue Printing Order di Play Game, update main (add BNMO CLI Banner)	10 November 2022
18221078 18221086	Membuat laporan bagian data test dan test script serta menambahkan tambahan spesifikasi banner BNMO dan mekanisme Queue Game	11 November 2022
18221096	Menambahkan ADT ArraydinInt, Update Diner Dash	11 November 2022
18221042	Update skip game	11 November 2022
18221078	Membuat laporan bagian ADT ArraydinInt	11 November 2022
18221042	Finalisasi main program, finalisasi	11 November 2022

18221078 18221096 18221154 18221086	diner dash	
18221042 18221078 18221096 18221154 18221086	Asistensi 2	11 November 2022


9.4 Lain-Lain






9.4.1 Form Asistensi 1

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

No. Kelompok/Kelas : 12/K02
 Nama Kelompok : Kelompok 12
 Anggota Kelompok (Nama/NIM) :
 1. Ghaylan Muhammad Fatih/18221042
 2. Gracia Theophilia/18221078
 3. Fikri Naufal Hamdi/18221096
 4. Reinhart Wisely Lim/18221154
 5. Esther Regina/18221086
 Asisten Pembimbing : Jason Stanley Yoman

Asistensi I

Tanggal : 4 November 2022	Catatan Asistensi: 1. Untuk skip game, game yang sebelumnya didequeue? Iya. Terus yang masuk history yang dimainkan aja. 2. Jadi kan di contoh spek contoh command kaya creategame/listgame itu posisinya command pisah sama spasi. Nah kalo misalnya commandnya ga dikasih spasi perlu dihandle ga? Engga, commandnya disesuaikan sama yang dispek aja. Yang dihandle cuma command lain kayak tanpa spasi itu dianggepnya command lain. 3. Untuk skipgame kalo gamenya dalam maintenance bakal ngeluarin pesan gak? Gak perlu jadi kalo kaya gitu
Tempat : Zoom	
Kehadiran Anggota Kelompok: No NIM Tanda tangan 1 18221042 	

<p>2 18221078</p>  <p>3 18221096</p>  <p>4 18221154</p>  <p>5 18221086</p> 	<p>ke-dequeue tapi ga di-play.</p> <p>4. Kan di config sama save file gada mark, jadi kita pake end of file? Boleh atau modif tanpa mark. Bisa juga dikasih kaya mark manual gitu.</p> <p>5. Aku udah commit dan ngepush di github tapi di contributor gada username aku tapi di history commitnya ada. Nah itu gimana yah? Cek config dari git karena config ga sesuai dengan email yang dipake. git config --global user.email , git config --global user.name. cek apakah sudah sesuai git config --global user.email "your_email@youreemail.com"</p>
	<p>Tanda Tangan Asisten:</p>  <p>Jason Stanley Yoman</p>

9.4.2 Form Asistensi 2

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

No. Kelompok/Kelas : 12/K02
Nama Kelompok : Kelompok 12
Anggota Kelompok (Nama/NIM) : 1. Ghaylan Muhammad Fatih/18221042
2. Gracia Theophilia/18221078
3. Fikri Naufal Hamdi/18221096





4. Reinhart Wisely Lim/18221154



5. Esther Regina/18221086

Asisten Pembimbing

: Jason Stanley Yoman

Asistensi 2

Tanggal : 11 November 2022	Catatan Asistensi: <ol style="list-style-type: none">1. Buat di laporan yang nomor dokumen, nomor revisi, dan tanggal rilis itu diisi apa ya? Untuk nomor revisi ditulis aja 1 dan kalau ada revisi nanti ditulis aja 2. Untuk tanggal rilis, ditulis tanggal laporannya selesai (hari ini). Untuk nomor dokumen, ditulis 1 aja sebagai dokumen pertama.2. Untuk di diner dash, serve tergantung daftar pesanan atau daftar pelanggan? Jadi, ga peduli urutan makanan ke berapa asal serve orang pertama (berdasarkan urutan pesanan).3. Untuk cook di diner dash bisa pesanan yang sama? Bisa asal ada di daftar pesanan. Walaupun lagi cook makanan yang satu bisa sambil cook makanan yang lain juga.
Tempat : Zoom	
Kehadiran Anggota Kelompok: <p>No NIM Tanda tangan</p> <p>1 18221042</p>  <p>2 18221078</p>  <p>3 18221096</p>  <p>4 18221154</p>  <p>5 18221086</p>	

	
	<p>Tanda Tangan Asisten:</p>  <p>Jason Stanley Yoman</p>