

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

PURRMART


Dipersiapkan oleh:

Kelompok 05

Adam Joaquin Girsang	/18223089
Timothy Marvine	/18223021
A. Nurul Aqeela Amin	/18223019
Muhammad Naufal Fathan	/18223059
Khairunnisa Hurun 'lin	/18221004

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-01-05		<jml hlm>
		Revisi	<no revisi>	<Tgl release>

Daftar Isi

Ringkasan	2
1 Struktur Data (ADT)	4
1.1 ADT STACK	4
2 Program Utama	9
3 Data Test	12
3.1 Data Test Main Menu	12
Gambar 5.1.2 Tampilan menu saat START dijalankan	13
Gambar 5.1.3 Tampilan saat user sudah berhasil login	13
3.2 Data Test Profile	14
Gambar 5.2.1 Tampilan saat profile ditampilkan	14
3.3 Data Test CART ADD <nama> <n>	14
Gambar 5.3.1 Tampilan saat barang berhasil ditambahkan ke keranjang	14
Gambar 5.3.2 Tampilan saat barang tidak ada di toko	14
3.4 Data Test CART REMOVE <nama> <n>	14
Gambar 5.4.3 Tampilan pada saat barang tidak berhasil dikurangi karena barang tidak ada di dalam keranjang	15
3.5 Data Test CART SHOW	15
3.6 Data Test CART PAY	16
Gambar 5.6.2 Tampilan pada saat user gagal untuk membayar karena user tidak memiliki uang yang cukup	17
Gambar 5.6.3 Tampilan pada saat pembayaran gagal dan pengguna menginput "Tidak"	17
Gambar 5.6.4 Tampilan pada saat pembayaran gagal karena keranjang yang kosong	17
3.7 Data Test HISTORY	17
Gambar 5.7.2 Tampilan pada saat menunjukkan jumlah yang ditampilkan lebih dari total riwayat	18
Gambar 5.7.3 Tampilan pada saat menunjukkan bahwa riwayat tidak dapat ditampilkan karena user belum membeli barang apapun	18
Gambar 5.7.4 Tampilan pada saat menunjukkan bahwa riwayat tidak dapat ditampilkan karena user salah input	18
3.8 Data Test WISHLIST ADD	18
Gambar 5.8.1 Tampilan pada saat wishlist add dijalankan dan barang berhasil ditambahkan ke wishlist	19
Gambar 5.8.2 Tampilan pada saat wishlist add dijalankan namun barang tidak bisa ditambahkan karena sudah ada di list	19
Gambar 5.8.3 Tampilan pada saat wishlist add dijalankan namun barang tidak bisa ditambahkan ke wishlist karena tidak ada di list	19
3.9 Data Test WISHLIST SWAP <i> <j>	19

Gambar 5.9.2 Tampilan pada saat wishlist swap dijalankan namun gagal menukar barang karena hanya terdapat satu barang	20
3.10 Data Test WISHLIST REMOVE <i>	20
Gambar 5.10.1. Tampilan pada saat wishlist remove <i> dijalankan dan berhasil menghapus barang berdasarkan indeksnya	20
Gambar 5.10.3. Tampilan pada saat wishlist remove <i> dijalankan namun gagal dalam menghapus barang karena wishlist kosong	21
Gambar 5.10.4. Tampilan pada saat wishlist remove <i> dijalankan namun gagal karena input indeks dari user salah	21
3.11 Data Test WISHLIST REMOVE	21
3.12 Data Test WISHLIST CLEAR	22
Gambar 5.12.1. Tampilan saat wishlist clear berhasil dijalankan dan menjadi kosong	22
3.13 Data Test WISHLIST SHOW	22
Gambar 5.13.1. Tampilan pada saat wishlist show berhasil menampilkan isi wishlist dari user	22
Gambar 5.13.2 Tampilan pada saat wishlist kosong	22
4 Test Script	22
5 Pembagian Kerja dalam Kelompok	25
6 Lampiran	27
6.1 Deskripsi Tugas Besar 2	27
6.2 Notulen Rapat	27
6.3 Log Activity Anggota Kelompok	29

Ringkasan

Tugas Besar 2 IF2111 meminta mahasiswa untuk mengembangkan PURRMART, sebuah aplikasi yang menggunakan bahasa C untuk simulasi *Command Line Interface* (CLI) yang berfungsi sebagai sistem jual-beli virtual. Dalam simulasi ini, PURRMART bertindak sebagai "Toko Borma," sebuah pemasok senjata perang untuk Agen Purry, karakter fiksi yang membutuhkan senjata untuk menyelesaikan misi rahasia tanpa perlu bertemu langsung dengan pemasok. PURRMART menyediakan berbagai fitur, termasuk manajemen akun, simulasi kerja, tantangan interaktif, serta pengelolaan toko. Pengguna dapat memilih command Profile yang digunakan untuk melihat data diri pengguna.

Dalam hal pengelolaan toko, PURRMART menyediakan fitur seperti CART ADD untuk menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja, CART REMOVE untuk mengurangi barang sejumlah kuantitas tertentu dari keranjang belanja, dan CART SHOW untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang. Pengguna juga dapat membeli barang-barang yang sudah dimasukkan ke dalam keranjang dengan memastikan bahwa pengguna memiliki uang yang cukup dengan CART PAY. HISTORY untuk menunjukan riwayat pembelian seorang pengguna. Urutan penunjukan adalah dari yang paling baru ke paling tua. WISHLIST ADD digunakan untuk menambahkan suatu barang ke wishlist, WISHLIST SWAP digunakan untuk menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist dimana posisi i dan j merupakan urutan barang pada wishlist. WISHLIST REMOVE <i> digunakan untuk menghapus barang dengan posisi ke-i dari wishlist. WISHLIST REMOVE digunakan untuk menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna. WISHLIST CLEAR digunakan untuk menghapus semua barang yang terdapat di dalam WISHLIST. Terakhir, WISHLIST SHOW digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam wishlist.

File konfigurasi atau penyimpanan dapat dikelola melalui command START untuk memuat file default atau LOAD untuk file tertentu, serta SAVE untuk menyimpan status aplikasi saat ini. Aplikasi ini dirancang agar memberikan pengalaman interaktif melalui berbagai pesan dan validasi yang memastikan penggunaan fitur yang tepat. Secara

keseluruhan, PURRMART menghadirkan pengalaman simulasi jual beli yang komprehensif dengan tantangan yang menarik dan pengelolaan data yang terorganisir.

Laporan ini berisi penjelasan tentang program yang telah kami buat. Bagian pertama berisi ringkasan berisi deskripsi umum persoalan dengan singkat. Bagian kedua berisi tentang penjelasan tambahan spesifikasi tugas. Bagian ketiga tentang struktur data yang digunakan agar program dapat berjalan. Bagian keempat berisi tentang program utama. Bagian kelima berisi tentang algoritma-algoritma menarik yang digunakan di dalam program ini. Bagian keenam berisi tentang *data test*. Bagian ketujuh berisi tentang *test script* yang berisi skenario test untuk semua fitur. Bagian kedelapan berisi tentang pembagian kerja kelompok. Bagian kesembilan berisi tentang lampiran yang terkait dengan dokumen.

Program ini digunakan menggunakan bahas C dengan struktur data yang telah dipelajari pada mata kuliah IF2111 Algoritma dan Struktur Data. ADT-ADT yang digunakan yaitu Kustom, Stack, Setmap, dan Linked List.

1 Struktur Data (ADT)

Pada pembuatan aplikasi PURRMART ini, kami menggunakan beberapa struktur data untuk menyelesaikan persoalan-persoalan pada Tugas Besar, yaitu ADT Stack , ADT Setmap, ADT Linked List.

1.1 ADT STACK

1.1.1 Sketsa Struktur

```
#ifndef stackt_H
#define stackt_H

#include "boolean.h"

#define Nil -1
#define CAPACITY 100

// Define struct Purchase
typedef struct {
```

```

    char itemName[50];
    int quantity;
    float price;
} Purchase;

// Use Purchase as infotypeStack
typedef Purchase infotypeStack;
typedef int idxTypeStack;

typedef struct {
    infotypeStack T[CAPACITY]; /* tabel penyimpan elemen */
    idxTypeStack TOP;          /* alamat TOP: elemen puncak */
} Stack;

/* Definisi akses dengan Selektor */
#define Top(S) (S).TOP
#define InfoTop(S) (S).T[(S).TOP]

/* ***** Prototype ***** */
void CreateEmptyStack(Stack *S);
/* I.S. : Stack S belum terdefinisi
   F.S. : Stack S terdefinisi dengan Top(S) = Nil */

boolean IsEmptyStack(Stack S);
/* I.S. : Stack S terdefinisi
   F.S. : Mengembalikan true jika Top(S) = Nil, false jika tidak */

boolean IsFullStack(Stack S);
/* I.S. : Stack S terdefinisi
   F.S. : Mengembalikan true jika Top(S) = CAPACITY - 1, false jika tidak */

void Push(Stack *S, infotypeStack X);
/* I.S. : Stack S terdefinisi, mungkin kosong atau penuh
   F.S. : Jika Stack S tidak penuh, elemen X ditambahkan ke puncak Stack S dan
   Top(S) bertambah 1 */

void Pop(Stack *S, infotypeStack *X);
/* I.S. : Stack S terdefinisi, mungkin kosong atau tidak kosong
   F.S. : Jika Stack S tidak kosong, elemen puncak Stack S dihapus dan disimpan
   ke *X, lalu Top(S) berkurang 1 */

#endif

```

1.1.2 Persoalan yang diselesaikan

ADT Stack digunakan untuk menyimpan riwayat pembelian seorang pengguna.

1.1.3 Alasan Pemilihan

Alasan digunakannya ADT Stack karena sifat dasar Stack yaitu mengikuti prinsip Last In, First out dimana riwayat pembelian ditampilkan dari yang paling baru ke

paling lama. Lalu, ADT Stack memiliki efisiensi penambahan dan penghapusan data. Selain itu, jika pengguna ingin melihat atau membatasi riwayat ke beberapa data, ADT Stack memungkinkan untuk mengambil elemen satu per satu dari atas tumpukan.

1.1.4 Implementasi

ADT Stack diimplementasikan dalam 'src/ADT/stack.c'.

1.2 ADT MAP

1.2.1 Sketsa Struktur

```
#include "map.h"
#include <stdio.h>
#include <stdlib.h>

boolean IsEmptyMap(Map M) {
    return (Count(M) == NilMap);
}

void CreateEmptyMap(Map *M) {
    Count(*M) = NilMap;
}

void InsertMap(Map *M, keytype k, valuetype v) {
    if (M->Count < MaxElMap) {
        M->Elements[M->Count].Key = (char*)malloc((strlen(k) + 1) *
sizeof(char));
        if (M->Elements[M->Count].Key == NULL) {
            printf("Gagal mengalokasikan memori untuk Key\n");
            return;
        }
        strcpy(M->Elements[M->Count].Key, k);
        M->Elements[M->Count].Value = v;
        M->Count++;
    }
}

void DeleteMap(Map *M, keytype k) {
    address_map idx = 0;
    while (idx < Count(*M) && !isKataEqual(Elements(*M)[idx].Key, k)) {
        idx++;
    }
    if (idx < Count(*M)) {
        for (address_map i = idx; i < Count(*M) - 1; i++) {
            Elements(*M)[i] = Elements(*M)[i + 1];
        }
        Count(*M)--;
    }
}
```

```

boolean IsMemberMap(Map M, keytype k) {
    for (int i = 0; i < M.Count; i++) {
        if (isKataEqual(M.Elements[i].Key, k)) {
            return true;
        }
    }
    return false;
}

int idxMap(Map M, keytype k) {
    address_map idx = 0;
    while (idx < Count(M)) {
        if (isKataEqual(Elements(M)[idx].Key, k)) {
            return idx;
        }
        idx++;
    }
    return -1;
}

```

1.2.2 Persoalan yang diselesaikan

ADT Setmap digunakan untuk menyimpan keranjang pembelian seorang pengguna.

1.2.3 Alasan Pemilihan

Alasan digunakannya ADT Setmap karena untuk menghindari duplikasi nama barang, memungkinkan pengecekan cepat apakah nama barang sudah ada atau tidak, memastikan nama barang unik, memiliki operasi create, read, update, dan delete yang efisien.

1.2.4 Implementasi

ADT Stack diimplementasikan dalam 'src/ADT/setmap.c'

1.3 ADT Linked List

1.3.1 Sketsa Struktur

```

#ifndef LINKEDLIST_H
#define LINKEDLIST_H

#include <stdbool.h> // Untuk tipe boolean
#include <stdlib.h>  // Untuk malloc dan free
#include <string.h>  // Untuk manipulasi string

#define NIL NULL

typedef char* ElementType; // Elemen berupa string
typedef struct Node *Address;

```



```

typedef struct Node {
    ElementType info;
    Address next;
    Address prev;
} Node;

typedef struct {
    Address first;
    Address last;
} LinkedList;

// Macro untuk mempermudah akses
#define InfoL(P) (P)->info
#define NextL(P) (P)->next
#define PrevL(P) (P)->prev
#define FirstL(L) ((L).first)
#define LastL(L) ((L).last)

// Prototipe fungsi
void CreateList(LinkedList *L);
/* I.S. : LinkedList L belum terdefinisi
   F.S. : LinkedList L terdefinisi dengan FirstL(L) = NIL dan LastL(L) = NIL */

bool IsListEmpty(LinkedList L);
/* I.S. : LinkedList L mungkin kosong atau tidak kosong
   F.S. : Mengembalikan true jika FirstL(L) = NIL, false jika tidak */

Address Allocate(ElementType X);
/* I.S. : X terdefinisi
   F.S. : Mengembalikan alamat node baru dengan InfoL(node) = X, NextL(node) =
NIL, dan PrevL(node) = NIL */

void Deallocate(Address P);
/* I.S. : Address P terdefinisi dan menunjuk ke node yang valid
   F.S. : Node yang ditunjuk oleh P dibebaskan dari memori */
void InsertFirstL(LinkedList *L, Address P);
/* I.S. : LinkedList L terdefinisi, Address P menunjuk ke node yang valid
   F.S. : Node yang ditunjuk oleh P ditambahkan sebagai elemen pertama dalam
LinkedList L */

void InsertLastL(LinkedList *L, Address P);
/* I.S. : LinkedList L terdefinisi, Address P menunjuk ke node yang valid
   F.S. : Node yang ditunjuk oleh P ditambahkan sebagai elemen terakhir dalam
LinkedList L */

void DeleteFirstL(LinkedList *L, Address *P);
/* I.S. : LinkedList L terdefinisi, mungkin kosong atau tidak kosong
   F.S. : Jika LinkedList L tidak kosong, elemen pertama dihapus dan alamatnya
disimpan ke *P */

void DeletelastL(LinkedList *L, Address *P);

```

```

/* I.S. : LinkedList L terdefinisi, mungkin kosong atau tidak kosong
   F.S. : Jika LinkedList L tidak kosong, elemen terakhir dihapus dan alamatnya
   disimpan ke *P */

Address Search(LinkedList L, ElementType X);
/* I.S. : LinkedList L terdefinisi, X terdefinisi
   F.S. : Mengembalikan Address dari node yang memiliki InfoL(node) = X, atau
   NIL jika tidak ditemukan */

void PrintListForward(LinkedList L);
/* I.S. : LinkedList L terdefinisi, mungkin kosong atau tidak kosong
   F.S. : Seluruh elemen LinkedList L dicetak dari elemen pertama hingga
   terakhir */

void PrintListBackward(LinkedList L);
/* I.S. : LinkedList L terdefinisi, mungkin kosong atau tidak kosong
   F.S. : Seluruh elemen LinkedList L dicetak dari elemen terakhir hingga
   pertama */

#endif

```

1.3.2 Persoalan yang diselesaikan

ADT Linked List digunakan untuk menyimpan *wishlist* seorang pengguna.

1.3.3 Alasan Pemilihan

Alasan digunakan ADT Linked List untuk bagian wishlist adalah karena operasi WISHLIST ADD dan WISHLIST REMOVE lebih efisien karena tidak perlu geser elemen seperti pada array. Linked list cukup mengubah referensi pointer (alamat) untuk menambahkan atau menghapus kode. Lalu, pertukaran posisi antara dua elemen dapat dilakukan dengan mengubah referensi pointer tanpa perlu memindahkan data secara fisik.

1.3.4 Implementasi

ADT Linked List diimplementasikan dalam 'src/ADT/linkedlist.c'

2 Program Utama

2.1 Main Menu

Program utama dinamai "main.c", file ini mencakup semua file ADT yang telah dibuat. Program dimulai dengan menampilkan tampilan pilihan menu utama (*main menu*) yang berisikan pilihan *command* seperti START untuk memulai progres yang baru, LOAD untuk melanjutkan progres yang telah ada, dan HELP untuk melihat cara menggunakan program. Pengguna dapat mengelola akun dengan REGISTER, LOGIN, dan LOGOUT, serta melihat data diri pengguna dengan memanggil PROFILE.

Selain itu, pengguna dapat melakukan transaksi apabila terdapat barang yang diinginkan. PURRMART menyediakan keranjang untuk pengguna memilih barang-barang yang ingin dibeli, pengguna dapat memanggil CART ADD <nama barang><jumlah barang> apabila ingin menambahkan suatu barang ke dalam keranjang atau CART REMOVE <nama barang><jumlah barang> apabila ingin mengurangi atau membatalkan penambahan barang ke dalam keranjang. Pengguna dapat melihat semua isi keranjangnya dengan memanggil CART SHOW. Apabila pengguna sudah puas memilih barang, mereka dapat membeli barang-barang yang berada di dalam keranjang dengan menggunakan *command* CART PAY. Setelah melakukan pembayaran, transaksi tersebut akan tersimpan di dalam riwayat pembelian yang bisa dilihat dengan memanggil *command* HISTORY <jumlah riwayat yang ditampilkan>.

PURRMART juga menyediakan fitur *wishlist* untuk para pengguna yang ingin menandai barang yang diinginkan namun belum berencana melakukan transaksi terhadap barang tersebut. Apabila ingin menambahkan barang ke dalam *wishlist*, pengguna perlu memanggil *command* WISHLIST ADD. Pengguna juga dapat menukan posisi barang di dalam *wishlist* dengan memanggil *command* WISHLIST SWAP. Jika terdapat barang yang ingin dihapus dari *wishlist*, terdapat dua metode untuk menghapusnya, yaitu dengan memanggil WISHLIST REMOVE <i> untuk menghapus barang dengan posisi ke-i dari *wishlist* dan WISHLIST REMOVE untuk menghapus barang berdasarkan nama barang yang diinput oleh pengguna. Pengguna juga dapat menghapus semua barang yang terdapat di dalam *wishlist* dengan memanggil WISHLIST CLEAR. Jika ingin melihat barang-barang yang sudah dimasukkan ke dalam *wishlist*, pengguna hanya perlu memanggil *command* WISHLIST SHOW.

Pengguna dapat mendapatkan penghasilan melalui WORK atau tantangan seperti Tebak Angka dan WORDL3 melalui WORK CHALLENGE. Pengelolaan toko mencakup fitur STORE LIST untuk melihat nama barang beserta harganya, STORE REQUEST untuk menambahkan permintaan barang, STORE SUPPLY untuk memproses antrian permintaan, dan STORE REMOVE untuk menghapus barang. Progres aplikasi dapat disimpan menggunakan SAVE. Program ini akan berhenti looping jika user memberikan input *command* berupa QUIT.

2.2 Pembacaan dan Inisialisasi File Konfigurasi

Dalam pembacaan, perlu diketahui format file konfigurasi, spesifikasinya adalah sebagai berikut :

1. Barisan pertama adalah bilangan bulat positif N yang menunjukkan banyaknya barang di dalam sistem
2. Selanjutnya, sejumlah N baris menyatakan nama barang beserta harganya dengan format <Harga barang> <Nama barang>
3. Baris selanjutnya adalah bilangan bulat positif M yang menunjukkan banyaknya pengguna di dalam sistem
4. Selanjutnya, terdapat data M buah pengguna dengan masing-masing spesifikasi berikut:
 - a. Baris pertama adalah bilangan bulat positif K yang menunjukkan banyaknya riwayat pengguna tersebut
 - b. Selanjutnya, sejumlah K baris menyatakan nama barang beserta total biaya dengan format <Total biaya> <Nama barang>
 - c. Baris selanjutnya adalah bilangan bulat positif J yang menunjukkan banyaknya wishlist pengguna tersebut
 - d. Selanjutnya, sejumlah J baris menyatakan nama barang dengan format <Nama barang>

2.3 Pemanggilan Command

2.3.1 PROFILE

PROFILE merupakan fungsi yang digunakan untuk melihat data diri pengguna dimana profile hanya dapat dipanggil saat status pengguna telah login.

2.3.2 CART ADD <nama> <n>

CART ADD merupakan fungsi yang digunakan untuk menambahkan barang dengan kuantitas tertentu ke dalam keranjang belanja.

2.3.3 CART REMOVE <nama> <n>

CART REMOVE merupakan fungsi yang digunakan untuk mengurangi barang sejumlah kuantitas tertentu dari keranjang belanja.

2.3.4 CART SHOW

CART SHOW merupakan fungsi yang digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam keranjang.

2.3.5 CART PAY

CART PAY merupakan fungsi yang digunakan untuk membeli barang-barang yang sudah dimasukan ke dalam keranjang.

2.3.6 HISTORY

HISTORY merupakan fungsi yang digunakan untuk menunjukan riwayat pembelian seorang pengguna.

2.3.7 WISHLIST ADD

WISHLIST ADD merupakan fungsi yang digunakan untuk menambahkan suatu barang ke wishlist.

2.3.8 WISHLIST SWAP <i> <j>

WISHLIST SWAP merupakan fungsi yang digunakan untuk menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist.

2.3.9 WISHLIST REMOVE <i>

WISHLIST REMOVE merupakan fungsi yang digunakan untuk menghapus barang dengan posisi ke-i dari wishlist.

2.3.10 WISHLIST REMOVE

WISHLIST REMOVE merupakan command yang digunakan untuk menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna.

2.3.11 WISHLIST CLEAR

WISHLIST CLEAR merupakan fungsi yang digunakan untuk menghapus semua barang yang terdapat di dalam WISHLIST.

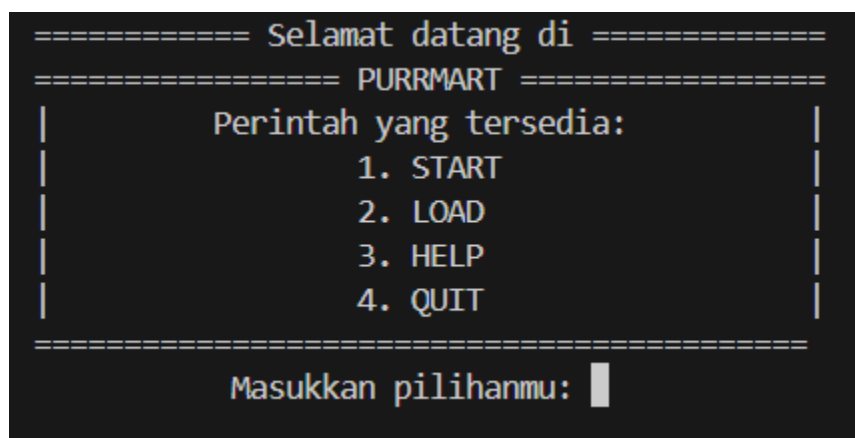
2.3.12 WISHLIST SHOW

WISHLIST SHOW merupakan fungsi yang digunakan untuk menunjukkan barang-barang yang sudah dimasukkan ke dalam wishlist.

3 Data Test

3.1 Data Test Main Menu

Pada tes ini dilakukan dilakukan pengujian untuk memastikan bahwa program dapat berjalan dengan baik dengan menampilkan tampilan awal dari aplikasi PURRMART. Tampilan awal menampilkan main menu yang berisikan pilihan START, LOAD, dan juga HELP. Cara melakukan kompilasi program ini dengan mengetikan. Adapun Gambar yaitu pada saat tampilan menu pertama saat program dijalankan, gambar yaitu pada saat tampilan menu saat start dijalankan, program akan masuk ke login menu, gambar yaitu pada saat tampilan main menu saat *user* sudah berhasil login.



Gambar 5.1.1 Tampilan menu pertama saat program dijalankan

```
File konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan
.
===== LOGIN MENU =====
|      Perintah yang tersedia:      |
|      1. LOGIN                     |
|      2. REGISTER                   |
|      3. HELP                       |
|      4. QUIT                       |
|=====|
Masukkan pilihanmu: |
```

Gambar 5.1.2 Tampilan menu saat START dijalankan

```
Anda telah login ke PURRMART sebagai tim.

===== Main Menu =====
|      1. WORK                      |
|      2. WORK CHALLENGE           |
|      3. STORE LIST                |
|      4. STORE REQUEST             |
|      5. STORE SUPPLY              |
|      6. STORE REMOVE              |
|      7. PROFILE                   |
|      8. CART ADD <nama> <n>       |
|      9. CART REMOVE <nama> <n>   |
|     10. CART SHOW                 |
|     11. CART PAY                  |
|     12. HISTORY <n>               |
|     13. WISHLIST ADD              |
|    14. WISHLIST SWAP <i> <j>       |
|    15. WISHLIST REMOVE <i>        |
|     16. WISHLIST CLEAR            |
|     17. WISHLIST SHOW             |
|     18. LOGOUT                    |
|     19. SAVE                      |
|     20. QUIT                      |
|    21. HELP -> Untuk melihat masing-masing kegunaan command |
|=====|
Masukkan pilihanmu: |
```

Gambar 5.1.3 Tampilan saat *user* sudah berhasil login

3.2 Data Test Profile

Pada tes ini akan menjalankan program profile dan memastikan bahwa fungsi berjalan dengan seharusnya. Jika fungsi berhasil dijalankan, maka akan menampilkan

Username dan jumlah koin. Adapun Gambar 5.2.1 merupakan tampilan pada saat profile ditampilkan.

```
Masukkan pilihanmu: PROFILE

== PROFILE ==
Nama   : user1
Saldo  : 100
```

Gambar 5.2.1 Tampilan saat profile ditampilkan

3.3 Data Test CART ADD <nama> <n>

Pada tes ini akan menjalankan program cart add dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.3.1 merupakan tampilan pada saat barang berhasil ditambahkan ke keranjang. Adapun Gambar 5.3.2 merupakan tampilan pada saat barang tidak ada di toko.

```
Masukkan pilihanmu: CART ADD AK 10
Berhasil menambahkan 10 AK ke keranjang belanja!
```

Gambar 5.3.1 Tampilan saat barang berhasil ditambahkan ke keranjang

```
Masukkan pilihanmu: CART ADD SOMETHING 10
Barang tidak ada di toko!
```

Gambar 5.3.2 Tampilan saat barang tidak ada di toko

3.4 Data Test CART REMOVE <nama> <n>

Pada tes ini akan menjalankan program cart remove dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.4.1 merupakan tampilan pada saat berhasil mengurangi barang tertentu dari keranjang. Adapun Gambar 5.4.2 merupakan tampilan pada saat tidak berhasil mengurangi karena kuantitas barang yang diminta pelanggan tidak ada pada keranjang. Adapun Gambar 5.4.3 merupakan tampilan pada saat barang tidak berhasil dikurangi karena barang tidak ada di dalam keranjang.

```
Masukkan pilihanmu: CART REMOVE AK 2
Berhasil mengurangi 2 AK dari keranjang belanja!
```

Gambar 5.4.1 Tampilan saat berhasil mengurangi barang tertentu dari keranjang

```
Masukkan pilihanmu: CART REMOVE DEADLINE 5
Barang tidak ada di keranjang belanja!
```

Gambar 5.4.2 Tampilan pada saat tidak berhasil mengurangi karena kuantitas barang yang diminta pelanggan tidak ada pada keranjang

```
Masukkan pilihanmu: CART REMOVE AK 20
Tidak berhasil mengurangi, hanya terdapat 10 AK pada keranjang!
```

Gambar 5.4.3 Tampilan pada saat barang tidak berhasil dikurangi karena barang tidak ada di dalam keranjang

3.5 Data Test CART SHOW

Pada tes ini akan menjalankan program cart show dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.5.1 merupakan tampilan pada saat keranjang berhasil ditampilkan karena memiliki isi. Adapun Gambar 5.5.2 merupakan tampilan pada saat keranjang kosong.

```
Masukkan pilihanmu: CART SHOW
Berikut adalah isi keranjangmu.
Kuantitas      Nama      Total
8              AK       800
Total biaya yang harus dikeluarkan adalah 800.
```

Gambar 5.5.1 Tampilan saat keranjang berhasil ditampilkan karena memiliki isi


```

1. WORK
2. WORK CHALLENGE
3. STORE LIST
4. STORE REQUEST
5. STORE SUPPLY
6. STORE REMOVE
7. PROFILE
8. CART ADD <nama> <n>
9. CART REMOVE <nama> <n>
10. CART SHOW
11. CART PAY
12. HISTORY <n>
13. WISHLIST ADD
14. WISHLIST SWAP <i> <j>
15. WISHLIST REMOVE <i>
16. WISHLIST CLEAR
17. WISHLIST SHOW
18. LOGOUT
19. SAVE
20. QUIT
21. HELP -> Untuk melihat masing-masing kegunaan command
=====
Masukkan pilihanmu: CART SHOW
Keranjang kamu kosong!

```

Gambar 5.5.2 Tampilan pada saat keranjang kosong

3.6 Data Test CART PAY

Pada tes ini akan menjalankan program start dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.6.1 merupakan tampilan pada saat user berhasil membeli barang dan pengguna memasukkan “Ya”. Adapun Gambar 5.6.2 merupakan tampilan pada saat user gagal untuk membayar karena user tidak memiliki uang yang cukup tetapi tetap menekan “Ya”. Adapun Gambar 5.6.3 merupakan tampilan pada saat pembayaran gagal dan pengguna menginput “Tidak”. Adapun Gambar 5.6.4 merupakan tampilan pada saat pembayaran gagal karena keranjang yang kosong.

```

Masukkan pilihanmu: CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas      Nama      Total
10              AK       1000
Total biaya yang harus dikeluarkan adalah 1000, apakah jadi dibeli? (Ya/Tidak)
Ya
Selamat kamu telah membeli barang-barang tersebut!

```

Gambar 5.6.1 Tampilan saat user berhasil membeli barang dan pengguna memasukkan “Ya”

```

Masukkan pilihanmu: CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas      Nama      Total
5              AK        500
Total biaya yang harus dikeluarkan adalah 500, apakah jadi dibeli? (Ya/Tidak)
Ya
Uang kamu hanya 100, tidak cukup untuk membeli keranjang!

```

Gambar 5.6.2 Tampilan pada saat user gagal untuk membayar karena user tidak memiliki uang yang cukup

```

Masukkan pilihanmu: CART PAY
Kamu akan membeli barang-barang berikut.
Kuantitas      Nama      Total
5              AK        500
Total biaya yang harus dikeluarkan adalah 500, apakah jadi dibeli? (Ya/Tidak)
Tidak

===== Main Menu =====
|                          1. WORK                          |

```

Gambar 5.6.3 Tampilan pada saat pembayaran gagal dan pengguna menginput “Tidak”

```

Masukkan pilihanmu: CART PAY
Keranjang kamu kosong!

```

Gambar 5.6.4 Tampilan pada saat pembayaran gagal karena keranjang yang kosong

3.7 Data Test HISTORY

Pada tes ini akan menjalankan program history dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.7.1 merupakan tampilan pada saat menunjukkan jumlah yang ditampilkan kurang dari total riwayat. Adapun Gambar 5.7.2 merupakan tampilan pada saat menunjukkan jumlah yang ditampilkan lebih dari total riwayat. Adapun Gambar 5.7.3 merupakan tampilan pada saat menunjukkan bahwa riwayat tidak dapat ditampilkan karena user belum membeli barang apapun.

```

>> HISTORY 4
1. AK47 40
2. AK47 100
3. Lalabu 35
4. AK47 10

```

Gambar 5.7.1 Tampilan pada saat menunjukkan jumlah yang ditampilkan kurang dari total riwayat

```
>> HISTORY 7
1. AK47 40
2. AK47 100
3. Lalabu 35
4. AK47 10
5. Meong 500
6. Ayam Goreng Crisbar 20
```

Gambar 5.7.2 Tampilan pada saat menunjukkan jumlah yang ditampilkan lebih dari total riwayat

```
>> HISTORY 1
Kamu belum membeli barang apapun!
```

Gambar 5.7.3 Tampilan pada saat menunjukkan bahwa riwayat tidak dapat ditampilkan karena user belum membeli barang apapun

```
>> HISTORY x
Error: String tidak valid untuk konversi
Input tidak valid. Harap masukkan angka positif.
```

Gambar 5.7.4 Tampilan pada saat menunjukkan bahwa riwayat tidak dapat ditampilkan karena user salah input

3.8 Data Test WISHLIST ADD

Pada tes ini akan menjalankan program wishlist add dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.8.1 merupakan tampilan pada saat wishlist add dijalankan dan barang berhasil ditambahkan ke wishlist. Adapun Gambar 5.8.2 merupakan tampilan pada saat wishlist add dijalankan namun barang tidak bisa ditambahkan karena sudah ada di list. Adapun Gambar 5.8.3 merupakan tampilan pada saat wishlist add dijalankan namun barang tidak bisa ditambahkan ke wishlist karena tidak ada di list.

```
>> WISHLIST ADD
Masukkan nama barang: Ayam Geprek Bakar Crispy Besthal
Berhasil menambahkan Ayam Geprek Bakar Crispy Besthal ke wishlist!
```

Gambar 5.8.1 Tampilan pada saat wishlist add dijalankan dan barang berhasil ditambahkan ke wishlist

```
>> WISHLIST ADD
Masukkan nama barang: Ayam Geprek Bakar Crispy Besthal
Ayam Geprek Bakar Crispy Besthal sudah ada di wishlist!
```

Gambar 5.8.2 Tampilan pada saat wishlist add dijalankan namun barang tidak bisa ditambahkan karena sudah ada di list

```
>> WISHLIST ADD
Masukkan nama barang: Ayam Geprek Crisbar
Tidak ada barang dengan nama Ayam Geprek Crisbar!
```

Gambar 5.8.3 Tampilan pada saat wishlist add dijalankan namun barang tidak bisa ditambahkan ke wishlist karena tidak ada di list

3.9 Data Test WISHLIST SWAP <i> <j>

Pada tes ini akan menjalankan program wishlist swap dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.9.1. merupakan tampilan pada saat wishlist swap dijalankan dan barang berhasil ditukar dengan posisi yang diinginkan. Adapun Gambar 5.9.2 merupakan tampilan pada saat wishlist swap dijalankan namun gagal menukar barang karena hanya terdapat satu barang.

```
Masukkan indeks pertama: 1
Masukkan indeks kedua: 2
Berhasil menukar posisi Ayam Geprek Sambalado Besthal dengan Ayam Geprek Bakar Crispy Besthal p
ada wishlist!
```

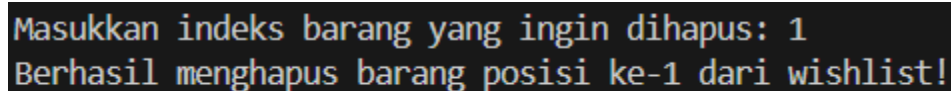
Gambar 5.9.1 Tampilan pada saat wishlist swap dijalankan dan barang berhasil ditukar dengan posisi yang diinginkan

```
Masukkan indeks pertama: 1
Masukkan indeks kedua: 2
Gagal menukar posisi! Salah satu indeks tidak valid.
```

Gambar 5.9.2 Tampilan pada saat wishlist swap dijalankan namun gagal menukar barang karena hanya terdapat satu barang

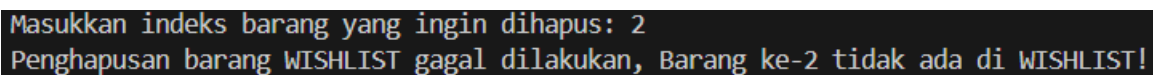
3.10 Data Test WISHLIST REMOVE <i>

Pada tes ini akan menjalankan program wishlist remove <i> dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.10.1 merupakan tampilan pada saat wishlist remove <i> dijalankan dan berhasil menghapus barang berdasarkan indeksinya. Adapun Gambar 5.10.2 merupakan tampilan pada saat wishlist remove <i> dijalankan namun gagal dalam menghapus barang dikarenakan oleh tidak adanya indeks tersebut. Adapun Gambar 5.10.3 merupakan tampilan pada saat wishlist remove <i> dijalankan namun gagal dalam menghapus barang karena wishlist kosong. Adapun Gambar 5.10.4 merupakan tampilan pada saat wishlist remove <i> dijalankan namun gagal karena input indeks dari user salah (user menginput bukan berupa angka).



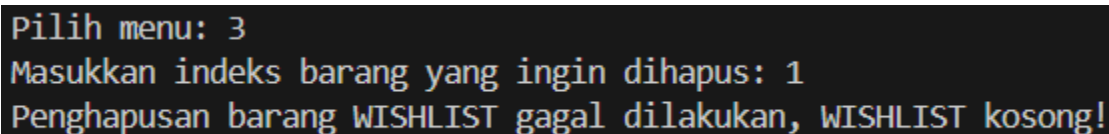
```
Masukkan indeks barang yang ingin dihapus: 1
Berhasil menghapus barang posisi ke-1 dari wishlist!
```

Gambar 5.10.1. Tampilan pada saat wishlist remove <i> dijalankan dan berhasil menghapus barang berdasarkan indeksinya



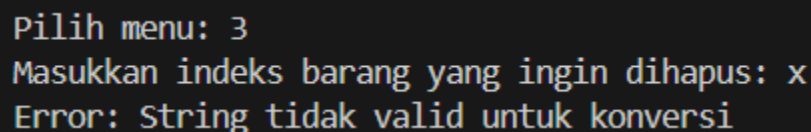
```
Masukkan indeks barang yang ingin dihapus: 2
Penghapusan barang WISHLIST gagal dilakukan, Barang ke-2 tidak ada di WISHLIST!
```

Gambar 5.10.2. Tampilan pada saat wishlist remove <i> dijalankan namun gagal dalam menghapus barang dikarenakan oleh tidak adanya indeks tersebut.



```
Pilih menu: 3
Masukkan indeks barang yang ingin dihapus: 1
Penghapusan barang WISHLIST gagal dilakukan, WISHLIST kosong!
```

Gambar 5.10.3. Tampilan pada saat wishlist remove <i> dijalankan namun gagal dalam menghapus barang karena wishlist kosong



```
Pilih menu: 3
Masukkan indeks barang yang ingin dihapus: x
Error: String tidak valid untuk konversi
```

Gambar 5.10.4. Tampilan pada saat wishlist remove <i> dijalankan namun gagal karena input indeks dari user salah

3.11 Data Test WISHLIST REMOVE

Pada tes ini akan menjalankan program wishlist remove dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.11.1 merupakan tampilan pada saat wishlist remove dijalankan dan berhasil menghapus barang di wishlist berdasarkan nama barangnya. Adapun Gambar 5.11.2 merupakan tampilan pada saat wishlist remove dijalankan, namun nama barang yang dihapus tidak ada di wishlist tersebut.

```
Pilih menu: 4
Masukkan nama barang yang akan dihapus: Ayam Geprek Bakar Crispy Besthal
Ayam Geprek Bakar Crispy Besthal berhasil dihapus dari WISHLIST!
```

Gambar 5.11.1 Tampilan saat wishlist remove dijalankan dan berhasil menghapus barang di wishlist berdasarkan nama barangnya

```
Pilih menu: 4
Masukkan nama barang yang akan dihapus: Ayam Geprek Bakar Crispy Besthal
Penghapusan barang WISHLIST gagal dilakukan, Ayam Geprek Bakar Crispy Besthal tidak ada di WISHLIST!
```

Gambar 5.11.2 Tampilan saat wishlist remove dijalankan, namun nama barang yang dihapus tidak ada di wishlist tersebut

3.12 Data Test WISHLIST CLEAR

Pada tes ini akan menjalankan program wishlist clear dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.12.1 merupakan tampilan pada saat wishlist clear berhasil dijalankan dan wishlist menjadi kosong.

```
wishlist telah dikosongkan.
```

Gambar 5.12.1. Tampilan saat wishlist clear berhasil dijalankan dan menjadi kosong

3.13 Data Test WISHLIST SHOW

Pada tes ini akan menjalankan program wishlist show dan memastikan bahwa fungsi berjalan dengan seharusnya. Adapun Gambar 5.13.1 merupakan tampilan pada saat wishlist show berhasil menampilkan isi wishlist dari *user*. Adapun Gambar 5.13.2 merupakan tampilan pada saat wishlist kosong.

```
Berikut adalah isi wishlist-mu:
1. Ayam Geprek Bakar Crispy Besthal
```

Gambar 5.13.1. Tampilan pada saat wishlist show berhasil menampilkan isi wishlist dari user

wishlist kamu kosong!

Gambar 5.13.2 Tampilan pada saat wishlist kosong

4 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur MAIN MENU	Memeriksa apakah program bisa menampilkan tampilan awal dari aplikasi PURRMART	Pada saat menjalankan program, akan keluar <i>main menu</i>	Data Test Main Menu	Memunculkan main menu yang terdapat 3 pilihan yaitu START, LOAD GAME, dan HELP	Sesuai yang diharapkan
2	Fitur PROFILE	Memeriksa apakah fitur PROFILE dapat berjalan dengan baik	Memasukkan <i>command PROFILE</i> pada main menu	Data Test Profile	Berhasil memunculkan tampilan profile	Sesuai yang diharapkan
3	Fitur CART ADD	Memeriksa apakah fitur CART ADD dapat berjalan dengan baik dan barang dapat masuk ke keranjang	Memasukkan <i>command CART ADD</i> pada main menu	Data Test Cart Add	Berhasil menambahkan keranjang	Sesuai yang diharapkan
4	Fitur CART REMOVE	Memeriksa apakah fitur CART REMOVE dapat berjalan dengan baik dan barang dapat dihilangkan dari keranjang	Memasukkan <i>command CART REMOVE</i> pada main menu	Data Test Cart Remove	Berhasil menghapus barang dari keranjang	Sesuai yang diharapkan
5	Fitur CART SHOW	Memeriksa apakah fitur CART SHOW	Memasukkan <i>command CART SHOW</i> pada main menu	Data Test Cart Show	Berhasil menampilkan isi keranjang	Sesuai yang diharapkan

		dapat berjalan dengan baik dan dapat menampilkan isi keranjang				
6	Fitur CART PAY	Memeriksa apakah fitur CART PAY dapat berjalan dengan baik dan isi keranjang dapat dibayar	Memasukkan <i>command CART PAY</i> pada main menu	Data Test Cart Pay	Berhasil melakukan command untuk membeli barang-barang yang sudah dimasukkan ke keranjang	Sesuai yang diharapkan
7	Fitur HISTORY	Memeriksa apakah fitur HISTORY dapat berjalan dengan baik dan dapat menampilkan seluruh riwayat	Memasukkan <i>command HISTORY</i> pada main menu	Data Test History	Berhasil menampilkan riwayat pembelian seorang pengguna	Sesuai yang diharapkan
8	Fitur WISHLIST ADD	Memeriksa apakah fitur WISHLIST ADD dapat berjalan dengan baik dan dapat menampilkan penambahan wishlist	Memasukkan <i>command WISHLIST ADD</i> pada main menu	Data Test Wishlist Add	Berhasil menjalankan command untuk menambahkan suatu barang ke wishlist	Sesuai yang diharapkan
9	Fitur WISHLIST SWAP <i><j>	Memeriksa apakah fitur WISHLIST SWAP dapat berjalan dengan baik dan dapat menukar wishlist	Memasukkan <i>command WISHLIST SWAP <i><j></i> pada main menu	Data Test Wishlist Swap	Berhasil menjalankan command untuk menukar barang posisi ke-i dengan barang posisi ke-j pada wishlist	Sesuai yang diharapkan
10	Fitur WISHLIST REMOVE <i>	Memeriksa apakah fitur WISHLIST REMOVE dapat	Memasukkan <i>command WISHLIST REMOVE</i> pada main menu	Data Test Wishlist Remove <i>	Berhasil menjalankan command untuk menghapus	Sesuai yang diharapkan

		berjalan dengan baik dan dapat menghapus wishlist pada list wishlist			barang dengan posisi ke-i dari wishlist	
11	Fitur WISHLIST REMOVE	Memeriksa apakah fitur WISHLIST REMOVE dapat berjalan dengan baik dan dapat menghapus wishlist pada list wishlist	Memasukkan <i>command WISHLIST REMOVE</i> pada main menu	Data Test Wishlist Remove	Berhasil menjalankan command untuk menghapus barang dari wishlist berdasarkan nama barang yang dimasukkan pengguna	Sesuai yang diharapkan
12	Fitur WISHLIST CLEAR	Memeriksa apakah fitur WISHLIST CLEAR dapat berjalan dengan baik dan dapat menghapus semua barang yang terdapat di dalam WISHLIST.	Memasukkan <i>command WISHLIST CLEAR</i> pada main menu	Data Test Wishlist Clear	Berhasil menjalankan command untuk menghapus seluruh barang yang terdapat pada wishlist.	Sesuai yang diharapkan
13	Fitur WISHLIST SHOW	Memeriksa apakah fitur WISHLIST SHOW dapat berjalan dengan baik dan dapat menampilkan barang-barang yang sudah dimasukkan ke dalam WISHLIST	Memasukkan <i>command WISHLIST SHOW</i> pada main menu	Data Test Wishlist Show	Berhasil menjalankan command untuk menunjukkan barang-barang yang sudah dimasukkan ke wishlist.	Sesuai yang diharapkan

5 Pembagian Kerja dalam Kelompok

Berikut merupakan penjelasan pembagian kerja dalam kelompok.

No	Fitur/ADT	Nim Coder	Nim Tester
1.	ADT Stack	18223021	18221059, 18221004, 18223021, 18223019, 18223089
2.	ADT Setmap	18223059, 18221104	1823021, 18223059, 18221004, 18223019, 18223089
3.	ADT Linked List	18223089, 18223021	18223021, 18223059, 18221004, 18223019, 18223089
4.	Profile	18223019	18223059, 18221004, 18223021, 18223019, 18223089
5.	CART ADD <nama> <n>	18223059	18223059, 18221004, 18223021, 18223019, 18223089
6.	CART REMOVE <nama> <n>	18223059	18223021, 18221004, 18223019, 18223059, 18223089
7.	CART SHOW	18221104	18223059, 18221004, 18223021, 18223019, 18223089
8.	CART PAY	18221104	18221004, 18223021, 18223019, 18223059, 18223089
9.	HISTORY	18223021	18223089, 18221004, 18223021, 18223019, 18223059
10/	WISHLIST ADD	18223021	18223089, 18221004, 18223021, 18223019, 18223059

11.	WISHLIST SWAP <i><j>	18223089	18223089, 18221004, 18223021, 18223019, 18223059
12.	WISHLIST REMOVE <i>	18223089	18223021, 18221004, 18223019, 18223059, 18223089
13.	WISHLIST REMOVE	18223089	18223021, 18221004, 18223019, 18223059, 18223089
14.	WISHLIST CLEAR	18223089	18221004, 18223021, 18223019, 18223059, 18223089
15.	WISHLIST SHOW	18223089	18221004, 18223021, 18223019, 18223059, 18223089

6 Lampiran

6.1 Deskripsi Tugas Besar 2

PURRMART sebuah sistem e-commerce yang dibuat untuk membantu OWCA membeli suplai perang dari Toko Borma secara digital. Sistem ini menggantikan transaksi tatap muka dengan fitur-fitur seperti pemesanan barang digital, manajemen stok, efisiensi operasi. PURRMART menjadi solusi logistik OWCA untuk tetap bertahan dan mencapai kemenangan tanpa hambatan. Pada milestone ini terdapat beberapa fitur tambahan diantara lain profile, cart, history, wishlist.

6.2 Notulen Rapat

**Form Asistensi Tugas Besar
IF2111/Algoritma dan Struktur Data STI
Sem. 1 2024/2025**

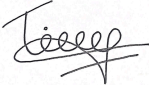




No. Kelompok/Kelas : 05/01
Nama Kelompok :
Anggota Kelompok (Nama/NIM) :
1. Timothy Marvine/18223021
2. Muhammad Naufal Fathan/18223059
3. Adam Joaquin Girsang/18223089

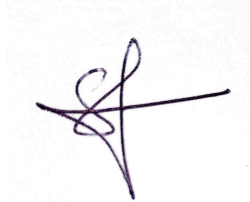
4. A.Nurul Aqeela Amin/18223019
5. Khairunnisa Hurun 'In/18221004

Asisten Pembimbing

: Jonathan Arthurito Aldi Sinaga

Asistensi II

Tanggal : 17 Desember 2024	Catatan Asistensi:
Tempat : Google Meet	1. Progres :
Kehadiran Anggota Kelompok:	<ul style="list-style-type: none">- sudah membagi tugas, dan masing-masing command sudah selesai, sedang diintegrasikan ke dalam satu program
1 18223021 Timothy Marvine 	2. Pertanyaan :
2 18223059 Muhammad Naufal Fathan 	<ul style="list-style-type: none">- File namanya apakah harus default atau dibebaskan? jawab: bebas aja sii di kalian- Apakah sama dengan penamaan command? jawab: command nya diterangkan di READ ME- apakah setiap ADT wajib memiliki driver? isinya ada ketentuan ga? jawab: iya, wajib ada, isi nya bebas gaada ketentuan- Untuk stack dan queue, apakah harus dalam bentuk linkedlist? jawab: iya- dengan bertambahnya command, berarti main menu nya bertambah ya kak? jawab: iya ditambah- Apakah ADT yang digunakan harus sesuai yang disarankan spek? (contohnya: penggunaan map pada cart) jawab:- Cart pay, langsung check out semua barang ke keranjang, apakah berarti history sesuai barangnya dalam cart atau bagaimana? jawab: urutan paling atas adalah yang paling baru- untuk ADT custom menggantikan ADT user? jawab: iya cuman ada satu ADT user
3 18223090 Adam Joaquin Girsang 	3. Summary :
4 18223019 A.Nurul Aqeela Amin 	<ul style="list-style-type: none">- Buat folder baru seperti utile untuk command 'menu, help, load, etc'
5 18221004 Khairunnisa Hurun 'In 	

	Tanda Tangan Asisten: 

6.3 Log Activity Anggota Kelompok

Berikut merupakan penjelasan pembagian kerja dalam kelompok.

No.	Tanggal	Kegiatan	Hasil
1.	07-12-2024	- Pembagian tugas	Masing-masing anggota menerima pembagian tugas dan mulai mengerjakan bagian masing-masing pada Github
2.	16-12-2024	- <i>Offline meeting</i>	Membagikan progres masing-masing dan mengerjakan bersama
3.	17-12-2024	- <i>Offline meeting</i> - Asistensi I	Membagikan progres masing-masing dan mengerjakan bersama serta melakukan penyesuaian berdasarkan hasil asistensi
4.	18-12-2024	- <i>Offline meeting</i>	Melanjutkan progres
5.	20-12-2024	- <i>Offline meeting</i>	Melanjutkan progres dan finalisasi (integrasi)
6.	21-12-2024	- <i>Offline meeting</i>	Finalisasi dan pengumpulan

LAMPIRAN

- Sebagian besar kode kami dikirim melalui line dikarenakan untuk menghindari konflik. Berikut bukti-bukti bahwa kami mengerjakan sendiri dan mengunggah code kami ke line untuk digabungkan (Screenshot oleh Timothy)

