

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI

Purrmart


Dipersiapkan oleh:

Kelompok 3

18223109	Fadhil Rifqi Rabbani Pane
18223110	Desati Dinda Saraswati
18223136	Geraldo Linggom Samuel T.
18223125	Matilda Angelina Sumaryo
18222133	Hanan Fitra Salam

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-03-03		34
		Revisi	<no revisi>	25-11-2024

STEI- ITB	IF2111-TB-03-03	Halaman 1 dari 34 halaman
Template dokumen ini dan informasi yang dimilikinya adalah milik Sekolah Teknik Elektro dan Informatika ITB dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Sekolah Teknik Elektro dan Informatika ITB.		

Daftar Isi

Daftar Gambar.....	4
1 Ringkasan.....	4
2 Penjelasan Tambahan Spesifikasi Tugas.....	5
2.1 Perincian spesifikasi pada TEBAK ANGKA.....	5
2.2 <Spesifikasi Fitur Tambahan 2>.....	6
3 Struktur Data (ADT).....	6
3.1 ADT Kustom.....	6
3.1.1 ADT User.....	6
3.1.2 ADT Barang.....	6
3.2 ADT List.....	6
3.2.1 ADT List Statis untuk User.....	6
3.2.2 ADT List Dinamis untuk Barang.....	7
3.3 ADT Mesin Kata.....	7
3.4 ADT Mesin Karakter.....	7
3.5 ADT Mesin Baris.....	8
3.6 ADT Queue.....	8
3.7 ADT menu.....	9
4 Program Utama.....	9
5 Algoritma-Algoritma Menarik.....	11
5.1 Makefile.....	11
6 Data Test.....	11
6.1 Start.....	11
6.2 Load.....	12
6.3 Login.....	13
6.4 Logout.....	14
6.5 Register.....	14
6.6 Work.....	15
6.7 Tebak Angka (work challenge).....	15
6.8 WORDL3 (work challenge).....	17
6.9 Store List.....	21
6.10 Store Request.....	22
6.11 Store Supply.....	23
6.12 Store Remove.....	24
6.13 Help.....	25

6.14 Save.....	26
6.15 Quit.....	27
7 Test Script.....	27
8 Pembagian Kerja dalam Kelompok.....	30
9 Lampiran.....	31
9.1 Deskripsi Tugas Besar.....	31
9.2 Notulen Rapat.....	31
9.2.1 Rapat I.....	31
9.2.2 Rapat II.....	31
9.2.3 Rapat III.....	31
9.3 Log Activity Anggota Kelompok.....	32
9.4 Notulensi Asistensi.....	32
9.4.1 Notulensi Asistensi I.....	32

Daftar Gambar

Gambar 6.3.1 Test Case jika berhasil login.....	12
Gambar 6.3.2 Test Case jika tidak berhasil login.....	13
Gambar 6.4.1 Test case jika berhasil logout.....	13
Gambar 6.5.1 Test Case jika berhasil register.....	14
Gambar 6.5.2 Test case jika tidak berhasil register.....	14
Gambar 6.6.1 Test Case Pada work 1 (Evil Lab Assistant).....	15
Gambar 6.7.1 Test case pada TEBAK ANGKA jika user berhasil menebak.....	16
Gambar 6.7.2 Test case pada TEBAK ANGKA jika user tidak berhasil menebak.....	17
Gambar 6.8.1 Tampilan pada WORDL3 ketika user berhasil menebak kata.....	20
Gambar 6.8.2 Tampilan pada WORDL3 ketika user gagal menebak kata.....	21
Gambar 6.10.1 Tampilan ketika user merequest barang yang belum ada di antrian shop.....	22
Gambar 6.10.2 Tampilan ketika user merequest barang yang sudah berada dalam antrian di shop.....	22
Gambar 6.11.1 Tampilan ketika user ingin menambahkan barang pada antrian paling belakang akan diminta harga.....	23
Gambar 6.11.2 Tampilan ketika user meminta menambah barang padahal tidak ada barang pada antrian.....	23
Gambar 6.12.1 Tampilan ketika user menghapus barang yang sudah ada pada shop dengan “STORE REMOVE”.....	24
Gambar 6.12.2 Tampilan ketika user ingin menghapus barang yang tidak ada pada shop.....	24
Gambar 6.13.2 Tampilan menu HELP pada welcome menu.....	26
Gambar 6.13.2 Tampilan menu HELP pada login menu.....	26
Gambar 6.13.3 Tampilan menu HELP pada main menu.....	26
Gambar 6.14.1 Test case jika berhasil save.....	27

1 Ringkasan

Sebagai tim programmer yang dipercaya oleh OWCA untuk mengatasi permasalahan yang dialami OWCA yaitu tidak memiliki transportasi untuk pergi ke Bojongsoang menghampiri supermarket Borma karena Borma adalah komponen penting agar bisa membawa kemenangan untuk OWCA pada pertarungan sengit melawan Dr. Asep Spakbor. Untuk itu solusi yang paling tepat adalah membuat dan merancang PURRMART, sebuah aplikasi sistem jual beli untuk mempermudah Agen Purry menyelesaikan misi terbarunya menggunakan bahasa C.

Saat program dijalankan, pengguna akan disambut oleh menu utama yang mencakup berbagai perintah seperti **START**, **LOAD**, **HELP**, dan **QUIT** serta dapat memasukkan perintah lainnya untuk memulai atau melanjutkan aktivitas di PURRMART,.

Aplikasi PURRMART, memiliki beberapa fitur unggulan, seperti memilih pekerjaan untuk mendapatkan uang, menyelesaikan tantangan **Tebak Angka** dan **WORDL3**, selain itu pengguna juga dapat memilih pekerjaan dengan memanfaatkan ADT mesin kata, mesin karakter, serta list. Selain itu, pengguna dapat menambahkan barang ke dalam antrian menggunakan *Queue*, menghapus barang dari antrian, membuat permintaan barang, mengatur harga barang, hingga menyimpan progres aplikasi melalui perintah **SAVE** sebelum keluar dengan perintah **QUIT**. Program ini dirancang menggunakan berbagai struktur data seperti list dinamis, queue, dan mesin karakter, yang memungkinkan pengelolaan data dan operasi aplikasi secara efisien.

Laporan Tugas Besar ini secara umum berisi tentang ringkasan tentang sistem jual beli yang dapat di akses oleh pengguna pada aplikasi PURRMART, penjelasan tambahan spesifikasi yang kelompok kami gunakan, penjelasan mengenai struktur data yang dipakai pada keseluruhan fungsi, penjelasan dan bukti *data test*, beserta dengan lampiran-lampiran yang mendukung kebutuhan tugas ini, seperti : dekripsi tugas, notulen rapat, *log activity* anggota kelompok, dan notulen asistensi.

Tugas besar ini secara umum menghasilkan suatu aplikasi sederhana, Purrmart, yang memuat beberapa fitur yang tidak sederhana. Aplikasi ini hanya berjalan di terminal sebagai output dengan *user*. Purrmart memuat fitur seperti *store* yang dapat melihat, menambah, menghapus, dan memverifikasi permintaan untuk menambah. Selain itu ada *work* yang dapat digunakan untuk menambah uang *user* dengan berbagai cara menarik. *User* juga mampu menyimpan dan melanjutkan permainan dengan kombinasi fitur *load*, *login*, dan *save*. Selain fitur-fitur yang disebutkan terdapat fitur penopang seperti *start*, *logout*, *exit*, *help*, dan *register*.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Perincian spesifikasi pada **TEBAK ANGKA**

Pada spesifikasi yang diberikan pada **TEBAK ANGKA**, hanya dijelaskan bahwa Jumlah kesempatan yang dipakai akan mempengaruhi uang yang didapatkan, menurut kami kalimat disini sedikit rancu, sehingga kami menentukan untuk memberikan hadiah sebesar 50*sisa kesempatan yang user miliki, contohnya akan tercantum pada [data test **TEBAK ANGKA**](#)

3 Struktur Data (ADT)

Pada Bab ini akan dijelaskan mengenai jenis jenis ADT (*Abstract Data Type*) yang digunakan, alasan pemilihan ADT tersebut untuk mengatasi permasalahan pada tiap fungsi, berikut jenis ADT yang digunakan antara lain :

3.1 ADT Kustom

3.1.1 ADT User

```
typedef struct {
    char name[MAX_LEN];
    char password[MAX_LEN];
    int money;
} User;
```

ADT Kustom User ini berfungsi untuk mendefinisikan dan merepresentasikan data pengguna. ADT ini menyimpan nama *user* dalam *char name*, *password user* dalam *char password*, dan menyimpan uang *user* dalam *int money*.

3.1.2 ADT Barang

```
typedef struct {
    char name[MAX_LEN];
    int price;
} Barang;
```

ADT Kustom ini memiliki fungsi yang serupa dengan ADT User. ADT ini berfungsi untuk mendefinisikan dan merepresentasikan data suatu barang. ADT ini menyimpan nama barang dalam *char name* dan harganya dalam *int price*.

3.1.2 ADT Menu

```
typedef enum {
    menuforwelcome,
    menuforlogin,
    menuutama,
    menustore,
    menuwork,
    menuworkchallenge,
} MenuState;
```

Menu digunakan untuk membedakan help pada setiap menu. Ini digunakan untuk memudahkan dan mempersingkat kode pada pemanggilan fungsi. Sehingga pada saat pemanggilan, akan dicegah adanya konflik antar fungsi. Penggunaan ini juga mengurangi jumlah fungsi yang dibuat.

3.2 ADT List

3.2.1 ADT List Statis untuk User

```
typedef struct {
    char name[MAX_LEN];
    char password[MAX_LEN];
    int money;
} User;
```

```
typedef struct {
    User users[MAX_USERS];
    int count; // Jumlah pengguna saat ini
    int currentUserIndex;
} UserList;
```

UserList adalah ADT berupa *list* statis yang digunakan untuk menyimpan informasi nama, *password*, dan uang yang dimiliki setiap *user* yang telah *login* yang berada di dalam struktur *User* beserta jumlah saat ini, dan posisi *user* di *list*. Kami memilih ADT ini karena dapat menyimpan informasi dari setiap *user* tentunya lebih dari satu. ADT ini diimplementasikan sebagai ADT *list* dengan nama *file header* “user.h”.

3.2.2 ADT List Dinamis untuk Barang

```
typedef struct {
    char name[MAX_LEN];
    int price;
} Barang;
```

```
typedef struct {
    Barang *items; // Pointer ke array dinamis
    int count;      // Jumlah barang saat ini
    int capacity;   // Kapasitas maksimum saat ini
} BarangList;
```

BarangList adalah ADT berupa *list* dinamis yang digunakan untuk menyimpan informasi nama barang, harga, jumlah barang, dan kapasitas maksimum tiap barang. Kami memilih ADT ini karena dapat menyimpan informasi dari setiap input barang yang tidak terbatas. ADT ini diimplementasikan sebagai ADT *list* dengan nama *file header* “barang.h”.

3.3 ADT Mesin Kata

```
typedef struct
{
    char TabWord[NMax];
    int Length;
} Word;
```

Word yang menggunakan ADT dari Mesin Kata ini berguna untuk menyelesaikan permasalahan mengenai keberadaan *string* pada bahasa pemrograman C. ADT ini digunakan untuk melakukan *input string*, *parsing command* program dan membaca *file* konfigurasi ke dalam aplikasi. ADT ini juga menjadi dasar dalam membuat mesin kata. Dengan membuat Word ini memiliki sebuah *array of char*, Word dapat menggantikan peran *string* dan dapat dimanfaatkan untuk penulisan kata pada program. Word juga memiliki *length* yang menyimpan informasi mengenai panjang huruf dari suatu “kata”. Membaca informasi dari *file external* membutuhkan akuisisi per kata. Mesin ini dikembangkan sebagai pengganti *scanf*. Namun, pada

tugas ini, Mesin Kata digunakan untuk membaca *command* yang diinput oleh pengguna ADT ini diimplementasikan sebagai ADT Mesin Kata dengan nama *file header* “mesinkata.h”.

3.4 ADT Mesin Karakter

```
#define MARK '\r'
/* State Mesin */
extern char currentChar;
extern boolean EOP;
static int retval;
```

ADT Mesin Karakter tidak memiliki struktur khusus karena pada dasarnya struktur ini hanya merepresentasikan sebuah karakter. Variabel *currentChar* merupakan struktur dari ADT Mesin Karakter ini yang bertipe data *char*. ADT Mesin Kata berperan sebagai dasar landasan ADT Mesin Kata maupun ADT Mesin Baris. ADT ini Mesin karakter digunakan untuk membaca karakter yang ada, baik dari *file* yang sudah ada maupun dari input pengguna. Tugas ini pada dasarnya membutuhkan mesin karakter untuk pengerjaannya. *Input command* pengguna, menggunakan Mesin Karakter. Membaca isi *file* konfigurasi atau *file save* juga menggunakan mesin karakter yang lebih lanjut akan diimplementasikan di mesin kata dan mesin baris. ADT diimplementasikan dengan nama *file header* “mesinkarakter.h”.

3.5 ADT Mesin Baris

```
typedef struct
{
    char kalimat[Nmax];
    int lengthsentence;
} Sentence;
```

ADT Mesin Baris berfungsi untuk membaca, memproses, dan mengelola data berupa teks atau string dalam bentuk baris. Dalam ADT ini, *Sentence* berguna untuk menyimpan suatu kalimat. *Sentence* juga berguna untuk menyimpan dan mengelola informasi baris. Konsep dasar dari ADT Mesin Kata digunakan dalam ADT ini, tetapi dimodifikasi untuk menggunakan *Mark* yang terdapat pada ADT Mesin Kata.

3.6 ADT Queue

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

Queue digunakan untuk menyelesaikan persoalan mengenai antrian pada menu *store*. Queue menyimpan informasi barang-barang yang user masukkan ke dalam antrian seperti informasi nama barang, nantinya bisa menambahkan barang baru ke dalam toko berdasarkan antrian permintaan. Barang yang berada pada antrian paling depan akan dimasukan ke toko. Pengguna dapat menerima, menunda, atau menolak permintaan. ADT Queue dengan nama *file header* “queue.h”. ADT Queue ini dipilih karena bisa menyimpan elemen (informasi lagu) yang berurutan sesuai penyisipannya dengan konsep FIFO (*First In First Out*) sesuai dengan spesifikasi pada *menu store request* dan *store supply* ADT Queue dengan nama *file header* “queue.h”.

4 Program Utama

Program utama PURRMART akan menampilkan sebuah *welcome page* dan beberapa *command awal* yang dapat diakses *user* yaitu START, LOAD, dan QUIT. Command START adalah command yang bisa digunakan *user* pertama kali. Jika *user* pada *welcome page* memilih command START maka program akan membaca file default konfigurasi. Lalu, Command LOAD adalah command yang bisa digunakan *user* ketika pernah menyimpan sebuah progress aktivitas aplikasi PURRMART. Sedikit berbeda isi dari file default konfigurasi, tapi fungsi ini berguna untuk membaca file teks juga. Sedangkan command HELP digunakan oleh *user* untuk melihat apa saja fungsi yang tersedia pada aplikasi PURRMART. Command yang dapat dipanggil pada PURRMART setelah *user* memilih START atau LOAD sebagai berikut:

- LOGIN, Untuk masuk ke dalam akun dan memulai sesi
- REGISTER, Untuk melakukan pendaftaran akun baru
- QUIT, Untuk keluar dari program
- MENU, Untuk kembali ke menu awal

Kemudian setelah *user* login *user* bisa memilih beberapa Menu yang ada pada PURRMART, seperti :

- WORK, Untuk pekerjaan biasa
- WORK CHALLENGE, Untuk pekerjaan spesial
- STORE, Untuk berbelanja, dan mengatur barang yang mau dijual seperti menghapus, menambah, dan meminta penambahan barang
- LOGOUT, Untuk keluar dari sesi
- SAVE, Untuk menyimpan state ke dalam file

Berikut menu-menu yang bisa user akses pada STORE :

- a. STORE LIST, Untuk melihat barang-barang di toko
- b. STORE REQUEST, Untuk meminta penambahan barang
- c. STORE SUPPLY, Untuk menambahkan barang dari permintaan
- d. STORE REMOVE, Untuk menghapus barang
- e. MENU, Untuk kembali ke menu sebelumnya

Berikut menu-menu yang bisa user akses pada WORK :

- a. TEBAK ANGKA, Tebak kata dengan biaya bermain 200 dan kesempatan memenangkan hingga 500
- b. WORDL3, Tebak kata dengan biaya bermain 500 dan kesempatan memenangkan 1500
- c. MENU, Untuk kembali ke menu sebelumnya

Beberapa command yang dapat dipanggil pada welcome page, command START dan LOAD, tidak bisa digunakan kembali jika sudah memanggil salah satu diantara kedua command tersebut. Selain dari kedua command sebelumnya dapat digunakan lebih dari satu kali. Setelah user menggunakan berbagai command yang tertera pada PURRMART, user dapat memilih untuk mengakhiri aktivitas pada aplikasi PURRMART. Command yang bisa dipilih untuk menutup program PURRMART adalah SAVE dan QUIT. Command SAVE digunakan untuk menyimpan perubahan yang sudah dilakukan pada aplikasi PURRMART. Command ini akan menyimpan state aplikasi terbaru ke dalam suatu file. Sedangkan command QUIT digunakan untuk keluar dari aplikasi PURRMART tanpa menyimpan perubahan yang telah dilakukan. Dalam menjalankan beberapa command diatas, dapat terjadi kasus dimana pengguna memasukan inputan yang salah, memilih command yang tidak tertera pada program, dll. Hal tersebut dapat membuat program mengeluarkan pesan berupa “Invalid command. Please try again.” yang mengartikan bahwa terdapat sesuatu ketidaktepatan input yang user lakukan.

Berikut source code kode pada program utama (main)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "../ADT/User_Barang/user.h"
#include "../ADT/Mesin_Kata/mesinkata.h"
#include "../ADT/User_Barang/barang.h"
#include "../ADT/queue/queue.h"

#include "animasi/animasi.h"
#include "start/start.h"
#include "load/load.h"
#include "login/login.h"
#include "register/register.h"
#include "save/save.h"
#include "work/work.h"
#include "workchallenge/tebakangka/tebakangka.h"
```

```

#include "workchallenge/wordl3/wordl3.h"
#include "Store_List/Store_List.h"
#include "Store_Request/Store_Request.h"
#include "Store_Supply/Store_Supply.h"
#include "Store_Remove/Store_Remove.h"
#include "Help/help.h"
#include "Logout/logout.h"
#include "quit/quit.h"
#include "handler/handler.h"

// Main Function
int main() {
    boolean running = true;

    UserList userList;
    BarangList barangList;
    Queue request;
    CreateUserList(&userList);
    CreateBarangList(&barangList, 2);
    CreateEmptyRequest(&request);

    MenuState currentMenu = menuforwelcome;

    while (running) {
        handleMainMenu(&userList, &barangList, &running, &request, &currentMenu);
    }

    FreeBarangList(&barangList);
    return 0;
}

```

5 Algoritma-Algoritma Menarik

5.1 Makefile

Menurut kami algoritma Makefile menarik karena memiliki sintaksis-sintaksis tersendiri. Dalam pengimplementasiannya juga melewati banyak kesulitan hingga menemukan solusi untuk masalah ini. Alasan kenapa menggunakan karena algoritma ini sangat mempermudah kita dalam meng-*compile* suatu file atau program kita, dari yang seharusnya panjang dengan gcc dan semua file yang diminta menjadi hanya 'make file'. Makefile ini sepertinya memiliki banyak cara yang berbeda untuk setiap *operating system* (OS), seperti pada windows atau linux itu berbeda dan memiliki sintaksnya sendiri-sendiri. Mempelajari cara bekerja Makefile ini sangat membantu dalam pembuatan program keseluruhan.

6 Data Test

Pada bagian Data test akan dilampirkan hasil yang diharapkan dan hasil yang ditampilkan dan penjelasan apabila hasilnya tidak sesuai

6.1 Start

Fitur yang di tes: START

Pada START ini akan di baca file default konfigurasi

Hasil yang diharapkan:

```
>> START  
File konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan.
```

Hasil yang didapatkan:

```
Default configuration successfully loaded. PURRMART is ready.  
Configuration loaded successfully!
```

Gambar 6.1 Test case jika berhasil melakukan start

6.2 Load

Fitur yang di tes: LOAD

Hasil yang diharapkan:

```
>> LOAD filenihil.txt;  
// Meload /save/filenihil.txt  
  
Save file tidak ditemukan. PURRMART gagal dijalankan.
```

```
>> LOAD savefile.txt;  
// Meload /save/savefile.txt  
  
Save file berhasil dibaca. PURRMART berhasil dijalankan.
```

Hasil yang didapatkan:

```
1. START          2. LOAD          3. HELP          4. QUIT

Enter command: load
Enter the file name to load: load/dummy.txt
testes2Jumlah barang: 3
Jumlah pengguna: 7

Config file load/dummy.txt successfully loaded.

Daftar Barang:
Barang 1: Nama: AK47, Harga: 10
Barang 2: Nama: Lalabu, Harga: 20
Barang 3: Nama: Ayam Goreng Crisbar, Harga: 20

Daftar Pengguna:
User 1: Nama: admin, Password: alstruktakeren, Uang: 100
User 2: Nama: kataka, Password: kaaaaa, Uang: 90
User 3: Nama: la, Password: l, Uang: 2000
User 4: Nama: k, Password: a, Uang: 90
User 5: Nama: praktikan, Password: kerenbangetkak, Uang: 25
User 6: Nama: a, Password: k, Uang: 90
User 7: Nama: l, Password: a, Uang: 0
```

Gambar 6.2.1. Test Case ketika load berhasil

6.3 Login

Fitur yang di tes: LOGIN

Hasil yang diharapkan:

// Contoh login yang berhasil untuk username johndoe dan password janedoe

>> **LOGIN**

Username: **johndoe**

Password: **janedoe**

Anda telah login ke PURRMART sebagai johndoe.

// Contoh login yang gagal karena password salah

>> **LOGIN**

Username: **johndoe**

Password: **johndoe**

Username atau password salah.

// Contoh login yang gagal karena belum ada pengguna dengan username janedoe

>> **LOGIN**

Username: **janedoe**

Password: **johndoe**

Username atau password salah.

// Contoh login yang gagal karena pengguna belum LOGOUT

>> **LOGIN**

Username: **janedoe**

Password: **johndoe**

Anda masih tercatat sebagai johndoe. Silakan LOGOUT terlebih dahulu.

Hasil yang didapatkan :

```
Enter your command: LOGIN
Users loaded successfully.
Enter Username: admin
Enter Password: alstrukdatkeren
Login successful! Welcome, admin.
```

Gambar 6.3.1 Test Case jika berhasil login

```
Enter command: login
Users loaded successfully.
Enter Username: hanan3
Enter Password: teshanan2
Invalid username or password. Please try again.
```

Gambar 6.3.2 Test Case jika tidak berhasil login

6.4 Logout

Fitur yang di tes: LOGOUT

Hasil yang diharapkan:

>> **LOGOUT**

johndoe telah logout dari sistem PURRMART. Silakan REGISTER/LOGIN kembali untuk melanjutkan.

Hasil yang didapatkan:

```
Enter command: logout
Apakah anda sudah melakukan save? (Y/N)
y
Anda berhasil logout
Logging out...
```

Gambar 6.4.1 Test case jika berhasil logout

6.5 Register

Fitur yang di tes: REGISTER

Hasil yang diharapkan:

// Contoh register yang berhasil

>> **REGISTER**

Username: **johndoe**

Password: janedoe Akun dengan username johndoe telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
// Contoh register yang gagal karena username sudah ada >> REGISTER Username: johndoe Password: janedoe Akun dengan username johndoe gagal dibuat. Silakan lakukan REGISTER ulang.

Hasil yang didapatkan :

```

Enter command: register
Enter username: hanan2
Enter password: teshanan2
Enter initial money (or press Enter for default 0): 1000
Username: hanan2
Password: teshanan2
Registration successful! User 'hanan2' has been registered.

```

Gambar 6.5.1 Test Case jika berhasil register

```

Enter command: register
Enter username: admin
Username is already taken. Please try again.
Enter username: mingyu
Username is already taken. Please try again.
Enter username: jae
Enter password: 12
Enter initial money (or press Enter for default 0): 0
Username: jae
Password: 12
Registration successful! User 'jae' has been registered.

```

Gambar 6.5.2 Test case jika tidak berhasil register

6.6 Work

Fitur yang di tes: WORK

Hasil yang diharapkan:

>> WORK Daftar pekerjaan: 1. Evil Lab Assistant (pendapatan=100, durasi=14s) 2. OWCA Hiring Manager (pendapatan=4200, durasi=21s) 3. Cikapundunginator Caretaker (pendapatan=7000, durasi=30s) 4. Mewing Specialist (pendapatan=10000, durasi=22s) 5. Inator Connoisseur (pendapatan=997, durasi=15s) Masukkan pekerjaan yang dipilih: Mewing Specialist

Anda sedang bekerja sebagai Mewing Specialist... harap tunggu.

Pekerjaan selesai, +10000 rupiah telah ditambahkan ke akun Anda.

Hasil yang didapatkan:

Daftar pekerjaan:

1. Evil Lab Assistant (pendapatan=100, durasi=14s)
2. QWZA Hiring Manager (pendapatan=4200, durasi=24s)
3. Cikapundunginator Caretaker (pendapatan=7000, durasi=30s)
4. Mewing Specialist (pendapatan=10000, durasi=22s)
5. Inator Connoisseur (pendapatan=997, durasi=15s)

Masukkan pekerjaan yang dipilih: 1

Anda sedang bekerja sebagai Evil Lab Assistant... harap tunggu.

Pekerjaan selesai, +100 rupiah telah ditambahkan ke akun Anda.

Jumlah uang anda sekarang adalah 200.

Gambar 6.6.1 Test Case Pada work 1 (Evil Lab Assistant)

6.7 Tebak Angka (*work challenge*)

Fitur yang di tes: work challenge (TEBAK ANGKA)

Hasil yang diharapkan:

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1

Tebak angka: 50

Tebakanmu lebih kecil!

Tebak angka: 75

Tebakanmu lebih besar!

Tebak angka: 65

Tebakanmu lebih besar!

Tebak angka: 62

Tebakanmu benar! +350 rupiah telah ditambahkan ke akun anda.

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1

Tebak angka: 50

Tebakanmu benar! +500 rupiah telah ditambahkan ke akun anda.

Hasil yang didapatkan:

```
Enter command: tebak angka
200 koin telah dikurangi. Sisa uang Anda: 0.
Selamat datang di permainan Tebak Angka!
ATTENTION! :
Anda memiliki 10 kesempatan untuk menebak.
Apabila angka yang ditebak lebih kecil akan ada pemberitahuan.
Apabila angka yang ditebak lebih besar akan ada pemberitahuan.
Apabila angka yang ditebak benar akan ada pemberitahuan dan otomatis keluar dari permainan.
Tebak angka: 50
Tebakanmu lebih besar!
Tebak angka: 37
Tebakanmu lebih besar!
Tebak angka: 24
Tebakanmu lebih kecil!
Tebak angka: 29
Tebakanmu lebih kecil!
Tebak angka: 32
Tebakanmu lebih kecil!
Tebak angka: 35
Tebakanmu lebih besar!
Tebak angka: 34
Tebakanmu lebih besar!
Tebak angka: 33
Tebakanmu benar! Selamat!
Anda mendapatkan 150 koin!
Uang Anda saat ini: 150.
```

Gambar 6.7.1 Test case pada TEBAK ANGKA jika user berhasil menebak

```

Enter command: tebak angka
200 koin telah dikurangi. Sisa uang Anda: 150.
Selamat datang di permainan Tebak Angka!
ATTENTION! :
Anda memiliki 10 kesempatan untuk menebak.
Apabila angka yang ditebak lebih kecil akan ada pemberitahuan.
Apabila angka yang ditebak lebih besar akan ada pemberitahuan.
Apabila angka yang ditebak benar akan ada pemberitahuan dan otomatis keluar dari permainan.
Tebak angka: 9
Tebakanmu lebih kecil!
Tebak angka: 9
Tebakanmu lebih kecil!
Tebak angka: 8
Tebakanmu lebih kecil!
Tebak angka: 7
Tebakanmu lebih kecil!
Tebak angka: 6
Tebakanmu lebih kecil!
Tebak angka: 5
Tebakanmu lebih kecil!
Tebak angka: 4
Tebakanmu lebih kecil!
Tebak angka: 3
Tebakanmu lebih kecil!
Tebak angka: 2
Tebakanmu lebih kecil!
Tebak angka: 1
Tebakanmu lebih kecil!
Kesempatanmu sudah habis! Angka yang benar adalah 52
Uang Anda saat ini: 150.

```

Gambar 6.7.2 Test case pada TEBAK ANGKA jika user tidak berhasil menebak

6.8 WORDL3 (work challenge)

Fitur yang di tes: WORDL3

Hasil yang diharapkan:

```
>> WORK CHALLENGE
```

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: 2

WELCOME TO WORDL3, YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE!

```

-----
-----
-----
-----
-----

```

Masukan kata tebakan Anda: ADIEU

Hasil:

A% D% I% E% U*

```
-----
```


Masukan kata tebakan Anda: **OMBRE**

Hasil:

A% D% I% E% U*
O% M% B% R* E%

Masukan kata tebakan Anda: **PLUCK**

Hasil:

A% D% I% E% U*
O% M% B% R* E%
P% L* U C% K%

Masukan kata tebakan Anda: **SERIN**

Hasil:

A% D% I% E% U*
O% M% B% R* E%
P% L* U C% K%
S% E% R* I% N%

Masukan kata tebakan Anda: **TRULY**

Hasil:

A% D% I% E% U*
O% M% B% R* E%
P% L* U C% K%
S% E% R* I% N%
T R U L Y

Selamat, Anda menang!
+1500 rupiah telah ditambahkan ke akun Anda.

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. WORDL3 (biaya main=500)

Masukan challenge yang hendak dimainkan: **2**

WELCOME TO W0RDL3, YOU HAVE 5 CHANCES TO ANSWER BEFORE YOU LOSE!

Masukan kata tebakan Anda: **ADIEU**

Hasil:

A% D% I% E* U%

Masukan kata tebakan Anda: **TRULY**

Hasil:

A% D% I% E* U%

T% R% U% L% Y%

Masukan kata tebakan Anda: **LUCKY**

Hasil:

A% D% I% E% U*

T% R% U% L% Y%

L% U% C% K% Y%

Masukan kata tebakan Anda: **SLICK**

Hasil:

A% D% I% E% U*

T% R% U% L% Y%

L% U% C% K% Y%

S* L% I% C% K%

Masukan kata tebakan Anda: **LEAKY**

Hasil:

A% D% I% E% U*

T% R% U% L% Y%

L% U% C% K% Y%

S* L% I% C% K%

L% E A% K% Y%

Masukan kata tebakan Anda: **BUILD**

Hasil:

A% D% I% E% U*

T% R% U% L% Y%

L% U% C% K% Y%

S* L% I% C% K%

L% E A% K% Y%

B% U% I% L% D%

Boo! Anda kalah.

Hasil yang didapatkan :

```
Selamat datang di permainan W0RDL3!  
ATTENTION! :  
Apabila huruf yang ditebak benar dan posisinya benar, maka akan ditandai dengan huruf tersebut.  
ai dengan huruf tersebut diikuti dengan tanda '*'.  
Apabila huruf yang ditebak salah, maka akan ditandai dengan huruf tersebut diikuti dengan tanda '%'.  
Selamat datang di permainan W0RDL3!  
Tebak kata berjumlah 5 huruf dalam 6 kesempatan.  
  
Tebakan ke-1: BUCIN  
B% U* C* I% N%  
  
Tebakan ke-2: PUCUK  
P% U% C* U K%  
  
Tebakan ke-3: CUCUK  
C U% C% U K%  
  
Tebakan ke-4: CURUG  
C U% R% U G%  
  
Tebakan ke-5: CLODE  
C L O D* E%  
  
Tebakan ke-6: CLOUD  
C L O U D  
  
Selamat! Anda berhasil menebak kata: CLOUD  
Anda mendapatkan 1500 koin!  
Uang Anda saat ini: 11150.
```

Gambar 6.8.1 Tampilan pada WORDL3 ketika user berhasil menebak kata

```

Enter command: WORDL3
500 koin telah dikurangi. Sisa uang Anda: 10650.
Selamat datang di permainan WORDL3!
ATTENTION! :
Apabila huruf yang ditebak benar dan posisinya benar, maka akan ditandai dengan huruf tersebut.
Apabila huruf yang ditebak benar, tapi posisinya salah, maka akan ditandai dengan huruf tersebut diikuti de
ngan tanda '*'.
Apabila huruf yang ditebak salah, maka akan ditandai dengan huruf tersebut diikuti dengan tanda '%'.
Selamat datang di permainan WORDL3!
Tebak kata berjumlah 5 huruf dalam 6 kesempatan.

Tebakan ke-1: EMBER
E% M% B% E R

Tebakan ke-2: HARIS
H% A% R* I* S%

Tebakan ke-3: CUKUR
C% U% K% U% R

Tebakan ke-4: SPOON
S% P* O% O% N%

Tebakan ke-5: KIPAS
K% I P A% S%

Tebakan ke-6: PIPSS
P% I P S% S%

Kesempatan habis. Kata yang benar adalah: VIPER
Uang Anda saat ini: 10650.

```

Gambar 6.8.2 Tampilan pada WORDL3 ketika user gagal menebak kata

6.9 Store List

Fitur yang di tes: STORE LIST

Hasil yang diharapkan:

```

>> STORE LIST
List barang yang ada di toko :
- Platypus Laser
- Shrink Ray
- Net Shooter
- Camouflage Cloak
- Sleep Dart Gun
- Bubble Blaster

```

```

>> STORE LIST
TOKO KOSONG

```

Hasil yang didapatkan :

```

Enter command: STORE LIST
>> STORE LIST
- AK47
- Lalabu
- Ayam Goreng Crisbar

```

Gambar 6.9.1 Tampilan ketika user command store list

6.10 Store Request

Fitur yang di tes: STORE REQUEST

Hasil yang diharapkan:

>> STORE REQUEST Nama barang yang diminta: AK47
>> STORE REQUEST Nama barang yang diminta: Adaditoko Barang dengan nama yang sama sudah ada di toko!
>> STORE REQUEST Nama barang yang diminta: Adadiantrian Barang dengan nama yang sama sudah ada di antrian!

Hasil yang didapatkan:

```
Enter command: STORE REQUEST
Nama barang yang diminta: JEDAI
Barang berhasil dimasukkan ke antrian!
```

Gambar 6.10.1 Tampilan ketika user merequest barang yang belum ada di antrian shop

```
Enter command: STORE REQUEST
Nama barang yang diminta: AK47
Barang dengan nama yang sama sudah ada di toko!
```

Gambar 6.10.2 Tampilan ketika user merequest barang yang sudah berada dalam antrian di shop

6.11 Store Supply

Fitur yang di tes: STORE SUPPLY

Hasil yang diharapkan:

>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Terima Harga barang: 100 AK47 dengan harga 100 telah ditambahkan ke toko.
>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Tunda AK47 dikembalikan ke antrian.

<p>>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Tolak</p> <p>AK47 dihapuskan dari antrian.</p>
<p>>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Purry</p> <p>< Balik ke menu ></p>

Hasil yang didapatkan :

```
Enter command: STORE SUPPLY
Jika ingin menambahkan barang, silahkan ketik 'Terima'.
Jika ingin menolak, silahkan ketik 'Tolak'.
Jika ingin menunda, silahkan ketik 'Tunda'.
Pastikan Anda memasukkan jawaban dengan benar!
Apakah kamu ingin menambahkan barang JEDAI ?: Terima
Harga barang: 75
JEDAI dengan harga 75 telah ditambahkan ke toko.
```

Gambar 6.11.1 Tampilan ketika user ingin menambahkan barang pada antrian paling belakang akan diminta harga

```
Enter command: STORE SUPPLY
Tidak ada barang di antrian!
```

Gambar 6.11.2 Tampilan ketika user meminta menambah barang padahal tidak ada barang pada antrian

```
Enter command: STORE SUPPLY
Jika ingin menambahkan barang, silahkan ketik 'Terima'.
Jika ingin menolak, silahkan ketik 'Tolak'.
Jika ingin menunda, silahkan ketik 'Tunda'.
Pastikan Anda memasukkan jawaban dengan benar!
Apakah kamu ingin menambahkan barang LILIN AROMATERAPY ?: Tolak
LILIN AROMATERAPY dihapuskan dari antrian.
```

Gambar 6.11.3 Tampilan ketika user menolak menambahkan barang pada shop

```
Enter command: STORE SUPPLY
Jika ingin menambahkan barang, silahkan ketik 'Terima'.
Jika ingin menolak, silahkan ketik 'Tolak'.
Jika ingin menunda, silahkan ketik 'Tunda'.
Pastikan Anda memasukkan jawaban dengan benar!
Apakah kamu ingin menambahkan barang BASRENG HUHAH ?: Tunda
BASRENG HUHAH dikembalikan ke antrian.
```


Gambar 6.11.4 Tampilan ketika user menunda menambahkan barang ke shop

6.12 Store Remove

Fitur yang di tes: STORE REMOVE

Hasil yang diharapkan:

<p>>> STORE REMOVE Nama barang yang akan dihapus: Sticky Web Gun</p> <p>Sticky Web Gun telah berhasil dihapus.</p>
<p>>> STORE REMOVE Nama barang yang akan dihapus: Inator Neutralizer</p> <p>Toko tidak menjual Inator Neutralizer</p>

Hasil yang didapatkan :

```

Enter command: STORE REMOVE
>> STORE LIST
- AK47
- Lalabu
- Ayam Goreng Crisbar
- JEDAI
Nama barang yang akan dihapus: JEDAI
JEDAI telah berhasil dihapus.

```

Gambar 6.12.1 Tampilan ketika user menghapus barang yang sudah ada pada shop dengan “STORE REMOVE”

```

Enter command: STORE REMOVE
>> STORE LIST
- AK47
- Lalabu
- Ayam Goreng Crisbar
Nama barang yang akan dihapus: LILIN
Toko tidak menjual LILIN.

```

Gambar 6.12.2 Tampilan ketika user ingin menghapus barang yang tidak ada pada shop

6.13 Help

Fitur yang di tes: HELP

Hasil yang diharapkan:

<p>// Ketika perintah dipanggil pada <u>welcome menu</u> >> HELP ===== [Welcome Menu Help PURRMART] ===== 1. START -> Untuk masuk sesi baru 2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi 3. QUIT -> Untuk keluar dari program</p>
<p>// Ketika perintah dipanggil pada <u>login menu</u></p>

>> HELP

=====[Login Menu Help PURRMART]=====

1. REGISTER -> Untuk melakukan pendaftaran akun baru
2. LOGIN -> Untuk masuk ke dalam akun dan memulai sesi
3. QUIT -> Untuk keluar dari program

// Ketika perintah dipanggil pada main menu

>> HELP

=====[Menu Help PURRMART]=====

- WORK -> Untuk bekerja
- WORK CHALLENGE -> Untuk mengerjakan challenge
- STORE LIST -> Untuk melihat barang-barang di toko
- STORE REQUEST -> Untuk meminta penambahan barang
- STORE SUPPLY -> Untuk menambahkan barang dari permintaan
- STORE REMOVE -> Untuk menghapus barang
- LOGOUT -> Untuk keluar dari sesi
- SAVE -> Untuk menyimpan state ke dalam file
- QUIT -> Untuk keluar dari program

```
Enter command: help
=====[ Welcome Menu Help PURRMART ]=====
Ketik ';' untuk keluar dari menu help
PURRY MART HADIR UNTUK SELURUH WARGA
1. START -> Untuk masuk sesi baru
2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi
3. QUIT -> Untuk keluar dari program
```

Gambar 6.13.1 Tampilan menu HELP pada welcome menu

```
Enter command: help
=====[ Login Menu Help PURRMART ]=====
Ketik ';' untuk keluar dari menu help
PURRY MART HADIR UNTUK SELURUH WARGA
1. REGISTER -> Untuk melakukan pendaftaran akun baru
2. LOGIN -> Untuk masuk ke dalam akun dan memulai sesi
3. QUIT -> Untuk keluar dari program
4. MENU -> Untuk kembali ke menu awal
```

Gambar 6.13.2 Tampilan menu HELP pada login menu

```
Enter command: help
=====[ Menu Help PURRMART ]=====
Ketik ';' untuk keluar dari menu help
1. WORK -> Untuk pekerjaan biasa
2. WORK CHALLENGE -> Untuk pekerjaan spesial
3. STORE -> Untuk berbelanja
4. LOGOUT -> Untuk keluar dari sesi
5. SAVE -> Untuk menyimpan state ke dalam file
```

Gambar 6.13.3 Tampilan menu HELP pada main menu

6.14 Save

Fitur yang di tes: SAVE

Hasil yang diharapkan:

```
>> SAVE savefile.txt;

Save file berhasil disimpan.
// File disimpan pada /save/savefile.txt
```

Hasil yang didapatkan :

```
Enter command: save
Enter the filename to save the current state: mile1.txt
Data successfully saved to ../../save/mile1.txt.txt
```

Gambar 6.14.1 Test case jika berhasil save

6.15 Quit

Fitur yang di tes: QUIT

Hasil yang diharapkan:

```
>> QUIT

Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? N

Kamu keluar dari PURRMART.
Dadah ^ _ ^/

>> QUIT

Apakah kamu ingin menyimpan data sesi sekarang (Y/N)? Y
// memanggil fitur Save
```

Hasil yang didapatkan :

```
Enter command: quit
Apakah anda sudah melakukan save? (Y/N)
y
Exiting PURRMART. Goodbye!
```

Gambar 6.15 Test Case jika berhasil Quit

```
Enter command: quit
Apakah anda sudah melakukan save? (Y/N)
n
Anda belum melakukan save
Silahkan lakukan save file terlebih dahulu
Mengarahkan anda kembali...
```

Gambar 6.15.2 Test Case jika tidak berhasil Quit

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Fitur Start	Memeriksa apakah fitur start dapat membaca file.txt	Mengetikkan command START pada terminal	START	Fitur dapat membaca config.txt konfigurasi dan berhasil mengeluarkan output yang diharapkan	Sesuai yang diharapkan
2	Fitur Load	Memeriksa apakah fitur load dapat membaca file txt lain	Mengetikkan command LOAD dilanjut dengan nama file.txt pada terminal	LOAD	Fitur dapat membaca file txt eksternal dan berhasil mengeluarkan output yang diharapkan	Sesuai yang diharapkan
3	Fitur Login	Memeriksa apakah fitur login dapat memasukkan akun pengguna ke sistem yang sudah didaftarkan sebelumnya	Mengetikkan command LOGIN dilanjut dengan nama dan password seorang pengguna	LOGIN	Fitur dapat memasukkan pengguna ke sebuah sesi dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
4	Fitur Logout	Memeriksa apakah fitur logout dapat mengeluarkan akun pengguna dari sebuah sesi	Mengetikkan command LOGOUT pada terminal	LOGOUT	Fitur dapat mengeluarkan pengguna dari sebuah sesi dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
5	Fitur Register	Memeriksa apakah fitur register dapat mendaftarkan pengguna baru	Mengetikkan command REGISTER dilanjut dengan username dan password yang ingin didaftarkan	REGISTER	Fitur dapat mendaftarkan sebuah username dan password untuk pengguna baru dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
6	Fitur Work	Memeriksa apakah fitur work dapat menambahkan uang dari seorang pengguna	Mengetikkan command WORK dilanjut dengan mengetikkan daftar pekerjaan yang ingin dipilih	WORK	Fitur dapat menambahkan uang untuk pengguna sesuai dengan kerjaan yang dipilih dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
7	Fitur Work Challenge Tebak Angka	Memeriksa apakah fitur tebak angka dapat bermain tebak angka dan	Mengetikkan command WORK CHALLENGE dilanjut dengan TEBAK ANGKA	TEBAK ANGKA	Fitur dapat menjalankan challenge tebak angka, menambahkan	Sesuai yang diharapkan

		menambahkan uang seorang pengguna			nominal uang dengan tepat sesuai dengan kesempatan yang dimiliki user , dan mengeluarkan output yang diharapkan	
8	Fitur Work Challenge WORDL3	Memeriksa apakah fitur wordl3 dapat bermain tebak angka dan menambahkan uang seorang pengguna	Mengetikkan command WORK CHALLENGE	WORDL3	Fitur dapat menjalankan challenge wordl3, menambahkan nominal uang dengan tepat, dan mengeluarkan output yang tepat	Sesuai yang diharapkan
9	Fitur Store List	Memeriksa apakah fitur store list dapat menampilkan daftar barang	Mengetikkan command STORE LIST pada terminal	STORE LIST	Fitur dapat mengeluarkan menampilkan list barang dengan output yang diharapkan	Sesuai yang diharapkan
10	Fitur Store Request	Memeriksa apakah fitur store request dapat menambahkan barang baru ke dalam toko	Mengetikkan command STORE REQUEST pada terminal dilanjut dengan nama barang yang diminta	STORE REQUEST	Fitur dapat memasukkan sebuah barang ke dalam antrian, memastikan barang unik, dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
11	Fitur Store Supply	Memeriksa apakah fitur store supply dapat menambahkan barang baru ke dalam toko sesuai permintaan	Mengetikkan command STORE SUPPLY dilanjut dengan command terima, tunda, tolak, atau purry	STORE SUPPLY	Fitur dapat menambahkan, menunda, membatalkan barang yang sedang dalam antrian untuk masuk ke dalam toko dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
12	Fitur Store Remove	Memeriksa apakah fitur store remove dapat menghapus barang yang ada di toko	Mengetikkan command STORE REMOVE dilanjut dengan nama barang yang ingin dihapus	STORE REMOVE	Fitur dapat menghapus barang dari toko sesuai dengan permintaan dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
13	Fitur Help	Memeriksa apakah fitur help dapat menampilkan daftar command yang mungkin dieksekusi	Mengetikkan command HELP pada terminal	HELP	Fitur dapat menampilkan daftar command yang bisa dieksekusi sesuai	Sesuai yang diharapkan

					dengan state menu nya	
14	Fitur Save	Memeriksa apakah fitur save dapat menyimpan state ke dalam suatu file	Mengetikkan command SAVE diikuti dengan nama file	SAVE	Fitur dapat menyimpan state ke dalam suatu file tertentu dan mengeluarkan output yang diharapkan	Sesuai yang diharapkan
15	Fitur Quit	Memeriksa apakah fitur quit dapat mengeluarkan pengguna dari suatu sesi	Mengetikkan command QUIT dilanjut dengan konfirmasi menyimpan data, Y untuk menyimpan dan N untuk langsung keluar	QUIT	Fitur dapat mengeluarkan pengguna dari suatu sesi Purrmart	Sesuai yang diharapkan

8 Pembagian Kerja dalam Kelompok

Fitur	NIM Coder	NIM Tester	Catatan
START	18222133	18223125	
LOAD	18223125	18223125	
LOGIN	18223125	18223136	
LOGOUT	18223136	18223125	
REGISTER	18223125	18222133	
WORK	18223109	18223125	
TEBAK ANGKA	18223109	18223125	
WORDL3	18223109	18223136	
STORE LIST	18223110	18223136	
STORE REQUEST	18223110	18223136	
STORE SUPPLY	18223110	18223136	
STORE REMOVE	18223110	18223136	

HELP	18223136	18223136	
SAVE	18223125	18223125	
QUIT	18223136	18223125	
ADT CUSTOM	18222133, 18223110, 18223125		
ADT LIST	18223110		
ADT MESIN KARAKTER & MESIN KATA	18222133, 18223110, 18223125, 18223109		
ADT QUEUE	18223110		

9 Lampiran

9.1 Deskripsi Tugas Besar

Buatlah sebuah aplikasi simulasi 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. Daftar ADT yang wajib digunakan dapat dilihat pada bagian Daftar ADT. Library yang boleh digunakan hanya `stdio.h`, `stdlib.h`, `time.h`, dan `math.h`.

Tentang sistem, PURRMART adalah sebuah aplikasi yang dapat mensimulasikan aktivitas beli barang pada e-commerce. PURRMART memiliki beberapa fitur utama, yaitu:

- Menampilkan barang toko
- Meminta dan menyuplai barang baru ke toko
- Menyimpan dan membeli barang dalam keranjang
- Menampilkan barang yang sudah dibeli
- Membuat dan menghapus wishlist
- Bekerja untuk menghasilkan uang

Ketika program pertama kali dijalankan, PURRMART akan memperlihatkan main menu yang berisi welcome menu dan beberapa command yaitu START, LOAD, dan juga HELP. Setelah itu, program akan memasuki login menu yang memiliki command LOGIN, REGISTER, dan juga HELP. Jika pengguna berhasil memasuki kredensial suatu akun, maka mereka akan masuk ke menu selanjutnya. Main menu menerima masukan berupa command seperti WORK, WORK CHALLENGE, STORE LIST, STORE REQUEST, STORE SUPPLY, STORE REMOVE, LOGOUT, dan SAVE. Program akan terus menerima command sampai diberikan command QUIT yang berlaku pada seluruh menu.

9.2 Notulen Rapat

9.2.1 Rapat I

Tanggal : 14 November 2024

Waktu : 18.00 - 18.30

Pembahasan : Pembagian tugas, penyesuaian persepsi terkait spesifikasi yang ada

9.2.2 Rapat II

Tanggal : 18 November 2024

Pembahasan : progress keseluruhan sesuai pembagian tugas masing-masing dan penyesuaian adt

9.2.3 Rapat III

Tanggal : 23 November 2024

Waktu : 10.00 - 21.00






Pembahasan : Menyelesaikan program keseluruhan


9.3 Log Activity Anggota Kelompok

NIM	Nama	Keterangan
18223109	Fadhil Rifqi Rabbani Pane	20/11/2024 - 22/11/2024 : Mengerjakan Work 20/11/2024 - 23/11/2024 : Mengerjakan TebakAngka 20/11/2024 - 23/11/2024 : Mengerjakan Wordl3 23/11/2024 - 24/11/2024 : Mengerjakan laporan
18223110	Desati Dinda Saraswati	13/11/2024 - 20/11/2023 : Mengerjakan ADT 20/11/2024 : Mengerjakan Store List 21/11/2024 : Mengerjakan Store Request dan Store Remove 23/11/2024 : Mengerjakan Store Supply 22/11/2024 - 25/11/2024 : Mengerjakan Main
18223136	Geraldo Linggom Samuel T.	20/11/2024-21/11/2024 : Mengerjakan Help, Logout 23/11/2024 : Mencicil laporan bagian ADT, Ringkasan, Fungsi utama dan Data test 24/11/2024 : Testing sesuai pembagian fungsi dan mengerjakan laporan
18223125	Matilda Angelina Sumaryo	18/11/2024 - 23/11/2024 : Mengerjakan ADT, load, login, register, save, quit, driver, dan main. 24/11/2024 : Mengerjakan laporan
18222133	Hanan Fitra Salam	20/11/2024 : Mengerjakan Start 23/11/2024 : Mengerjakan Makefile 23/11/2024 : Mengerjakan laporan

9.4 Notulensi Asistensi

9.4.1 Notulensi Asistensi I

Tanggal : Jumat, 22 November 2024	<p>Catatan Asistensi:</p> <ul style="list-style-type: none"> - Register: hasil dari register tidak perlu langsung ditulis ke save file, hanya command save yang dapat digunakan untuk menyimpan data yang baru ditambahkan saat program berjalan ke dalam save file. Terdapat kemungkinan user yang melakukan register tidak melakukan save sehingga data user tersebut tidak tersimpan. - Store Supply: <ol style="list-style-type: none"> 1. Apabila user memilih untuk menunda, barang yang berada di antrian paling depan akan dikembalikan ke ujung antrian. 2. Apabila user memasukkan input yang tidak valid (selain Terima, Tunda, Tolak) program akan kembali ke menu dan antrian akan kembali seperti semula. - Work: dapat mengimplementasikan fungsi sleep untuk menerapkan waktu tunggu saat user sedang bekerja. - Input harus bisa membaca kalimat dan akan dibaca sampai user menekan enter (contoh: "Store List"), seragamkan terminatornya. - Usahakan file relative untuk savenya sama, kalau bisa di root. - Kalau run di linux, terminatornya itu beda, permasalahan umum yang terjadi biasanya terdapat di mesinkata dan mesinkarakter - Coba nya nanti run nya di wsl - Untuk interface di terminalnya dibebaskan, buat sekreatif mungkin saja.
Tempat : Zoom Meeting	
Kehadiran Anggota Kelompok:	
No	
NIM	
Tanda tangan	
1	<p>Tanda Tangan Asisten:</p>
18222133	
	
2	
18223109	
	
3	<p>Tanda Tangan Asisten:</p>
18223110	
	
4	
18223136	
	
5	<p>Tanda Tangan Asisten:</p>
18223125	
	

	 18221067 Fawwaz Abrial Saffa
--	---