

LAPORAN TUGAS BESAR
IF2111 Algoritma dan Struktur Data STI
Purrmart




Dipersiapkan oleh:

Kelompok 09 - Kelas 02

Karunia Mega Lestari	/ 18221126
Mahesa Satria Prayata	/ 18223082
Rafli Dwi Nugraha	/ 18223038
M Khalfani Shaquille Indrajaya	/ 18223104
Alghan Pridanusuta	/ 18223058
Michael Ballard Isaiah Silaen	/ 18223080

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		IF2111-TB-02-09		<jml hlm>
		Revisi	1	<Tgl release>

REVISI

Daftar Isi

1	Ringkasan.....	3
2	Penjelasan Tambahan Spesifikasi Tugas.....	4
	2.1 File Konfigurasi.....	4
	2.2 <Spesifikasi Fitur Tambahan 2>.....	4
3	Struktur Data (ADT).....	4
	3.1 ADT Boolean.....	4
	3.2 ADT User.....	5
	3.3 ADT Barang.....	6
	3.4 ADT List.....	7
	3.5 ADT Queue.....	8
	3.6 ADT Mesin Karakter dan Mesin Kata.....	10
4	Program Utama.....	13
5	Algoritma-Algoritma Menarik.....	18
	5.1 string_compare.....	18
	5.2 string_copy.....	18
	5.3 wordtoint.....	19
	5.4 strlent.....	19
6	Data Test.....	20
	6.1 Data Test 1.....	20
	6.2 Data Test 2.....	22
	6.3 Data Test 3.....	24
7	Test Script.....	26
8	Pembagian Kerja dalam Kelompok.....	28
9	Lampiran.....	29
	9.1 Deskripsi Tugas Besar.....	29
	9.2 Notulen Rapat.....	30
	9.3 Log Activity Anggota Kelompok.....	32

1 Ringkasan

Kami sebagai *team programmer* paling andal OWCA ditugaskan untuk menyelesaikan masalah *transport* dari Burma yang letaknya di Bojongsoang dengan membuat sistem bernama Purrmart. Purrmart adalah sistem jual beli barang-barang perang antar OWCA dengan Borma. Sistem ini dibutuhkan untuk menghentikan Dr. Asep Spakbor yang akan menggunakan mesin ‘Oppenheimer-inator’ untuk menghancurkan tiga negara bagian.

Sistem/aplikasi Purrmart ini akan berbasis CLI (*command-line interface*) dengan menggunakan beberapa ADT, diantaranya adalah ADT kustom (berisi adt dari masing-masing *command*), ADT *list*, ADT mesin karakter dan mesin kata, dan ADT *queue*.

Laporan ini secara khusus akan menjelaskan fitur-fitur dari Purrmart, penjelasan struktur data ADT yang kami gunakan, algoritma-algoritma menarik yang terdapat dalam sistem, dan hasil dari testing untuk memastikan sistem berjalan dengan lancar. Secara keseluruhan, tugas besar Purrmart berjalan dengan lancar dan hasil yang kami dapatkan cukup baik, semua fitur berjalan sesuai dengan spesifikasi yang diberikan serta penggunaanya.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 File Konfigurasi

File konfigurasi digunakan untuk menyimpan data awal dari barang apa saja yang digunakan di toko dan jumlah pengguna, sehingga diperlukan validasi data. Berikut adalah validasi-validasi tersebut,

- Hal yang berhubungan dengan uang harus dalam bilangan bulat positif.
- “Nama” tidak boleh mengandung spasi.

2.2 <Spesifikasi Fitur Tambahan 2>

3 Struktur Data (ADT)

3.1 ADT Boolean

```
// File boolean.h

#ifndef BOOLEAN_H
#define BOOLEAN_H

#define boolean unsigned char
```

```
#define true 1
#define false 0

#endif
```

ADT ini menyelesaikan masalah berupa tidak adanya tipe data boolean di C yang akan seringkali digunakan terutama untuk validasi.

3.2 ADT User

```
//file user.h

#define MAX_USERS 100
#define MAX_LEN 50

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

typedef struct {
    User buffer[MAX_USERS];
    int nEff;
} ListUser;

void CreateListUser(ListUser *list);

boolean IsUserExist(ListUser *list, char *username);

void AddUser(ListUser *list, User newUser);

User *FindUser(ListUser *list, char *username);
```

ADT *user* ini merepresentasikan *user* yang menggunakan aplikasi Purrmart. Terdapat struktur *User* yang memiliki atribut berupa:

- *name*, nama *user* bertipe *string* dengan maksimal 50 *character*.

- *password*, sandi dari *user* bertipe *string* dengan maksimal 50 *character*.
- *money*, jumlah uang yang dimiliki oleh *user* bertipe *integer*.

Selain *user* terdapat struktur lain yaitu *ListUser* untuk mempermudah dalam pengelolaan *user*, atribut-atributnya adalah sebagai berikut:

- *buffer*, menyimpan jumlah pengguna dengan maksimal 100 *user* bertipe *array of User* (struktur pertama).
- *nEff*, jumlah *user* yang terdaftar bertipe *integer*.

Ada juga berbagai fungsi yang digunakan dalam ADT ini, berikut adalah kegunaan dari masing-masing fungsi tersebut,

- *CreateUserList*, menginisialisasi *ListUser* dalam keadaan kosong.
- *IsUserExist*, mengecek apakah *username* tertentu sudah terdaftar.
- *AddUser*, menambah *user* baru bila *username* yang sama belum ada dalam pengguna yang terdaftar.
- *FindUser*, menampilkan data *user* sesuai dengan *username* yang dicari.

Persoalan yang diselesaikan dengan menggunakan ADT ini adalah segala hal yang menyangkut *user* seperti menambah *user* dan mengecek apakah *user* tertentu terdaftar dapat dilakukan dengan fungsi-fungsi yang terdapat di ADT ini. Struktur yang digunakan pada ADT ini juga mempermudah kami untuk mengakses sekaligus mengelola *user* yang terdaftar maupun yang akan daftar di aplikasi Purrmart.

3.3 ADT Barang

```
//file barang.h

#define MAX_LEN 100

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

ADT barang ini merepresentasikan barang satuan yang terdaftar di Purrmart. Terdapat struktur Barang yang memiliki atribut berupa:

- *name*, nama dari barang, bertipe *string* dengan maksimal 100 *character*.
- *price*, harga dari barang, bertipe *integer*.

Permasalahan yang diselesaikan dengan adanya ADT ini adalah barang di *store* Purrmart memiliki struktur sehingga mudah untuk diakses dan dikelola.

3.4 ADT List

```
//file listbarang.h

#include "boolean.h"
#include "barang.h"
#include "mesinkata.h"

typedef struct {
    Barang *buffer;
    int capacity;
    int neff;
} ListBarang;

void InitializeListBarang(ListBarang *list);

int LengthList(ListBarang list);

boolean IsListEmpty(ListBarang list);

Barang GetBarangAt(ListBarang list, int idx);

void AddBarang(ListBarang *list, Barang item);

int FindBarangByName(ListBarang list, char *name);

void RemoveBarangAt(ListBarang *list, int idx);

void DisplayListBarang(ListBarang list);

void FreeListBarang(ListBarang *list);
```

ADT list ini merepresentasikan semua barang yang terdaftar dalam *store* di aplikasi Purrmart. Struktur yang digunakan dalam ADT ini adalah *ListBarang* dengan atribut-atribut sebagai berikut:

- *buffer*, merupakan *list of* Barang (struktur pada ADT barang).
- *capacity*, kapasitas maksimum dari *buffer*.
- *nEff*, jumlah barang yang terdaftar, bertipe *integer*.

Ada juga berbagai fungsi yang digunakan untuk mengelola *ListBarang* ini, berikut adalah nama dari fungsi tersebut dan kegunaannya:

- *InitializeListBarang*, menginisialisasi *list* barang kosong dengan kapasitas awal 10 elemen.
- *LengthList*, mengecek jumlah barang yang terdaftar dalam *store* Purrmart.
- *IsListEmpty*, mengecek apakah *store* dalam Purrmart kosong.
- *GetBarangAt*, menampilkan data barang yang terdapat pada *index* tertentu.
- *AddBarang*, menambahkan barang baru dalam *ListBarang* yang diletakkan di urutan paling bawah. Saat *buffer* sudah penuh, kapasitas *buffer* akan dikali dua.
- *FindBarangByName*, menampilkan data barang sesuai dengan name dari barang tersebut.
- *RemoveBarangAt*, menghapus barang pada *index* tertentu.
- *DisplayListBarang*, menampilkan semua barang pada *store*. Bila *store* kosong maka akan menampilkan "TOKO KOSONG".
- *FreeListBarang*, membuat *list barang* menjadi kosong dan membebaskan memori.

Permasalahan yang diselesaikan dengan adanya ADT ini adalah pengelolaan barang (dari segi toko) dalam *store* Purrmart sudah terpenuhi, barang dapat disimpan dalam *list* sehingga mudah untuk diakses dan manipulasi data barang juga menjadi mudah dengan fungsi-fungsi yang ada.

3.5 ADT Queue

```
//file queue.h

#include "boolean.h"
#include "barang.h"
#include "mesinkata.h"

#define IDX_UNDEF -1
#define CAPACITY 100

typedef Barang ElemenQueue;

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

#define IDX_HEAD(q) (q).idxHead
#define IDX_TAIL(q) (q).idxTail
```



```

#define HEAD(q) (q).buffer[(q).idxHead]
#define TAIL(q) (q).buffer[(q).idxTail]

void CreateQueue(Queue *q);

boolean isEmpty(Queue q);

boolean isFull(Queue q);

int length(Queue q);

void enqueue(Queue *q, ElemenQueue val);

void dequeue(Queue *q, ElemenQueue *val);

boolean IsMemberQ(Queue q, char *name);

```

ADT ini merepresentasikan antrian barang yang menggunakan aturan dari *queue* yaitu *First in First Out* (FIFO). Dalam ADT ini ada struktur berupa *queue*, tersedia juga *IDX_UNDEF* sebagai konstanta untuk menandakan kondisi saat *queue* kosong. Struktur *queue* memiliki berbagai atribut di dalamnya berupa:

- *buffer*, menyimpan elemen barang dalam antrian dengan kapasitas tetap sebesar 100, bertipe elemen dari *queue* itu sendiri.
- *idxHead*, menunjukkan *index* elemen pertama dalam antrian, bertipe *integer*
- *idxTail*, menunjukkan *index* elemen terakhir dalam antrian, bertipe *integer*

Terdapat juga berbagai fungsi yang digunakan untuk memanipulasi *queue*, berikut adalah penjelasan mengenai fungsi-fungsi tersebut:

- *CreateQueue*, Menginisialisasi *queue* dengan *idxHead* dan *idxTail* menjadi *IDX_UNDEF* terlebih dahulu.
- *isEmpty*, mengecek apakah *queue* kosong.
- *isFull*, mengecek apakah antrian penuh.
- *length*, mengembalikan jumlah elemen dalam *queue*.
- *enqueue*, menambahkan elemen baru ke belakang *queue* (bagian *tail*). Jika antrian kosong maka elemen tersebut akan menjadi *head* sekaligus *tail*.
- *dequeue*, menghapus elemen *head* pada *queue*. Bila hanya satu elemen maka *queue* akan menjadi kosong.
- *isMemberQ*, mengecek apakah elemen dengan nama tertentu ada di dalam antrian.

Persoalan yang diselesaikan dalam penggunaan ADT ini adalah mempermudah pengelolaan antrian barang dengan prinsip FIFO, efisiensi pengelolaan yang baik karena struktur data *queue* menggunakan *circular buffer* untuk memanfaatkan ruang memori secara efisien, dan operasi

penambahan dan pengurangan elemen dilakukan dengan kompleksitas $O(1)$, sehingga cepat proses operasi tersebut dilaksanakan.

3.6 ADT Mesin Karakter dan Mesin Kata

```
//File mesinkarakter.c
#include <stdio.h>
#include "boolean.h"

#define MARK '.'

extern FILE *pita;
extern char currentChar;
extern boolean EOP;

void START();

void ADV();

char GetCC();

boolean IsEOP();

void StartFile(char *filename);

void CloseFile();

void StartWriteFile(char *filename);
```

ADT ini digunakan untuk membaca karakter demi karakter dari sebuah pita (bisa berupa *file* atau *standar input*). Di awal *file* akan ditetapkan beberapa variabel *extern* yang digunakan untuk menandakan mesin karakter ini, berikut adalah penjelasan masing-masing *file extern* yang digunakan:

- *pita*, sebagai petunjuk file yang dibaca.
- *currentChar*, karakter yang sedang dibaca oleh pita.
- *EOP*, berarti *End Of Processing* menandakan bahwa pita sudah mencapai akhir. Biasanya ditandai dengan *MARK*.

Fungsi-fungsi utama dalam ADT ini ada:

- *START*, untuk memulai pembacaan karakter dari *standar input* mulai dari karakter pertama dan mengatur *EOP*.
- *ADV*, memajukan pembacaan karakter dan menjalankan *EOP* jika karakter yang dibaca berupa *MARK*.
- *GetCC*, mengembalikan karakter yang sedang dibaca oleh mesin karakter.
- *isEOP*, mengecek apakah pita sudah mencapai *End Of Processing*.
- *StartFile*, membuka *file* untuk dibaca dan memulai pembicaraan karakter dari file tersebut.
- *CloseFile*, menutup *file* yang sedang dibaca.
- *StartWriteFile*, membuka *file* dalam *write mode*.

Persoalan yang diselesaikan dengan adanya file ADT ini adalah manipulasi sebuah data yang menggunakan tipe data *character* dapat dilaksanakan seperti pembentukan kata atau validasi. ADT ini juga sebagai dasar dari ADT mesin kata.

```
//File mesinkata.c

#ifndef MESINKATA_H
#define MESINKATA_H

#include "mesinkarakter.h"

#define MAX_WORD_LENGTH 256

typedef struct {
    char contents[MAX_WORD_LENGTH];
    int length;
} Word;

extern Word currentWord;

void IgnoreBlanks();

void StartWord();

void StartWords();

void AdvanceWord();
```

```

void CopyWord();

void CopyWords();

int IsWordEqual(Word word, char *string);

void CopyWordToString(char *destination);

int IsWordNumber(Word word);

int WordToInt(Word word);

int string_compare(char *s1, char *s2);

void string_copy(char *dest, char *src);

int strlen(char *str);

int wordContainsBlank(Word word);

boolean IsSameWord(Word word1, Word word2);

boolean IsSameFirstWord(char *s1, char *s2);

boolean remainderWord(char *s1, char *s2);

```

Beda dari mesin karakter yang hanya membaca satu kata, ADT ini membaca satu kata dalam sebuah pita. Dalam ADT ini terdapat struktur data berupa *Word* yang memiliki berbagai atribut berupa:

- *contents*, merupakan array of *character* dengan maksimal karakter berjumlah 256.
- *length*, panjang kata yang disimpan, bertipe data *integer*.

Fungsi-fungsi untuk menjalankan mesin kara ini beserta penjelasannya adalah sebagai berikut:

- *IgnoreBlanks*, mengabaikan karakter kosong seperti spasi, *tab*, atau *newline* hingga menemukan karakter yang valid.
- *StartWord*, menginisialisasi pembacaan kata.
- *StartWrods*, menginisialisasi pembacaan kata yang terdiri dari beberapa kata yang terpisah oleh karakter kosong.
- *AdvanceWord*, melanjutkan pembacaan pita ke kata selanjutnya.

- *CopyWord*, menyalin kata dari pita ke *currentWord*.
- *CopyWords*, menyalin kata yang terdiri dari beberapa kata yang terpisah oleh karakter kosong.
- *IsWordEqual*, mengecek apakah *currentWord* sama dengan *string* tertentu.
- *CopyWordToString*, menyalin isi *currentWord* ke *string* tujuan.
- *isWordNumber*, mengecek apakah *currentWord* merepresentasikan angka.
- *WordToInt*, mengubah *currentWord* menjadi integer (tentunya perlu *isWordNumber* sebagai validasi bahwa seluruh bagian dari pita berupa angka).
- *string_compare*, mengembalikan 0 jika kedua string sama, negatif ketika $s1 < s2$, dan positif ketika $s1 > s2$.
- *string_copy*, menyalin string dari source ke destination.
- *strlen*, menghitung panjang string.
- *wordContainsBlank*, mengembalikan 1 jika kata mengandung blank ' ' atau '\t', 0 jika tidak.
- *IsSameWord*, membandingkan antar seluruh kata dan mengembalikan true jika kata sama.
- *IsSameFirstWord*, Mengembalikan true jika **kata pertama** pada $s1$ sama dengan $s2$.
- *remainderWord*, mengembalikan true dan menghapus kata pertama pada $s1$ jika kata pertama pada $s1$ sama dengan $s2$.

Persoalan yang diselesaikan dengan adanya ADT ini adalah keterbatasan cara untuk memanipulasi tipe data *character*. Cara-cara yang dimaksud adalah parsing data, validasi kata, dan manipulasi *string* dapat diselesaikan dengan ADT ini.

4 Program Utama

Agar menjelaskan kode utama ini menjadi lebih mudah, kami akan membaginya jadi beberapa bagian penting. Berikut adalah bagian-bagian penting dari program utama yang kami buat. dalam setiap menu terdapat “*help*” yang dapat membantu user memahami program lebih lanjut.

- Sertakan file header yang akan digunakan dalam program utama

```
#include <stdio.h>
#include <stdlib.h>
#include "store.h"
#include "listbarang.h"
#include "queue.h"
#include "barang.h"
#include "mesinkata.h"
#include "mesinkarakter.h"
#include "boolean.h"
#include "help.h"
#include "load.h"
#include "login.h"
#include "logout.h"
```

```
#include "quit.h"
#include "register.h"
#include "save.h"
#include "start.h"
#include "user.h"
#include "wordl3.h"
#include "works.h"
#include "tebakangka.h"
```

- Inisialisasi struktur-struktur yang akan digunakan pada program utama seperti *ListBarang*, *Queue*, dll.

```
int main(){
    ListBarang listbarang;
    Queue queue;
    ListUser userlist;
    User user;
    Works work;
    WorksList worklist;

    CreateEmptyWorkList(&worklist);
    LoadWorkList(&worklist);
    CreateListUser(&userlist);
    user.name[0] = '\0';
    InitializeListBarang(&listbarang);
    CreateQueue(&queue);
```

- Kode awal saat baru masuk ke aplikasi Purrmart, khususnya untuk menjalankan fitur *START*, *LOAD*, dan *HELP*

```
printf("Selamat Datang di PURRMART, silahkan masukkan
command (START, LOAD, HELP): \n");
START();

char input[MAX_WORD_LENGTH];
int started = 1;
while (started == 1){
    StartWords();
    CopyWordToString(input);
```

```

        if (string_compare(input, "START") == 0){
            if (Starttr(&userlist, &listbarang) == 2){
                started = 0;
            }
        }
        else if (string_compare(input, "LOAD") == 0){
            if (LoadFile(&userlist, &listbarang) == 2){
                started = 0;
            }
        }
        else if (string_compare(input, "HELP") == 0){
            help_welcome();
        }
        else {
            printf("Command tidak dikenal. Masukkan HELP jika anda butuh bantuan. \n");
        }
    }
}

```

- Setelah melakukan *LOAD/START*, kita akan disambut dengan menu login yang menyediakan fitur *REGISTER*, *LOGIN*, dan *HELP*.

```

    printf("Silahkan masukkan command (REGISTER, LOGIN, HELP): \n");
    while(logged){
        START();
        StartWords();
        CopyWordToString(input);

        if (string_compare(input, "REGISTER") == 0){
            RegisterUser(&userlist);
        }

        else if (string_compare(input, "LOGIN") == 0){
            if (LoginUser(&userlist, &user) == 1){
                int menu = 1;
                help_main();
                while (menu){
                    printf("Silahkan masukkan command: ");
                    START();
                    StartWords();
                    char inputmain[MAX_WORD_LENGTH];

```

```

CopyWordToString(inputmain);

if (string_compare(inputmain, "WORK") ==
0){
    UserWorks(&worklist, &user);
}
else{
    printf("Gagal untuk login. \n");
}

else{
    printf("Command tidak dikenal. Masukkan HELP jika
anda butuh bantuan. \n");
}

```

- Setelah user selesai melewati proses *LOGIN/REGISTER*, *user* akan dapat berinteraksi dengan kode-kode inti program. Berikut adalah *work*, yaitu sebuah *command* agar *user* bisa mendapatkan uang dengan cara memilih antara beberapa pilihan pekerjaan, dan *work challenge* berjumlah dua yaitu *tebakangka* dan *wordl3*, sebuah game yang dimainkan *user* memerlukan uang untuk memainkannya. Namun bila *user* menang akan mendapatkan lebih banyak uang dari yang dibayar.

```

else if (string_compare(inputmain, "WORK CHALLENGE") ==
0){
    printf("Daftar challenge yang tersedia: \n");
    printf("1. Tebak Angka (biaya main = 50) \n");
    printf("2. W0RDL399 (biaya main = 50) \n");
    printf("3. Quit Challenge. \n");

    int inChallenge = 0;
    while (!inChallenge){
        printf("\nMasukan challenge yang hendak
dimainkan :");

        char inputChallenge[MAX_LEN];
        START();
        StartWords();
        CopyWordToString(inputChallenge);

        if (string_compare(inputChallenge, "1") == 0){
            ChallengeTebakAngka(&user);
            inChallenge = 1;
        }

        else if (string_compare(inputChallenge, "2") ==
0){
            PlayWordl3(&user);

```



```

                                inChallenge = 1;
                                }
                                else if (string_compare(inputChallenge, "3") ==
0){
                                printf("Anda keluar dari challenge. \n");
                                inChallenge = 1;
                                }
                                else {
                                printf("Command tidak dikenal. \n");
                                }
                                }

```

- Selain *work* dan *work challenges*, ada juga fitur untuk melihat toko yang ada di Purrmart. Berikut adalah kodenya dalam program utama,

```

                                else if (string_compare(inputmain, "STORE LIST") == 0){
                                display_store_list(listbarang);
                                }
                                else if (string_compare(inputmain, "STORE REQUEST") == 0){
                                store_request(listbarang, &queue);
                                }
                                else if (string_compare(inputmain, "STORE SUPPLY") == 0){
                                store_supply(&listbarang, &queue);
                                }
                                else if (string_compare(inputmain, "STORE REMOVE") == 0){
                                store_remove(&listbarang);
                                }

```

- Setelah user selesai dengan segala aktivitasnya di Purrmart, user dapat melakukan *save* keadaan *store* dan akun, *logout* dari akun, atau keluar dari program secara keseluruhan (kode untuk help terletak di akhir pada *menu* ini karena help tidak terlalu berhubungan dengan fungsi lainnya).

```

                                else if (string_compare(inputmain, "LOGOUT") == 0){
                                LogoutUser(&user);
                                menu = 0;
                                }
                                else if (remainderWord(inputmain, "SAVE")){
                                save(&userlist, &listbarang, inputmain);
                                }
                                else if (string_compare(inputmain, "QUIT") == 0){

```

```

        quit(&queue, &userlist, &listbarang);
        return 0;
    }

    else if (string_compare(inputmain, "HELP") == 0){
        help_main();
    }

    else if (string_compare(input, "HELP") == 0){
        help_login();
    }
    else{
        printf("Command tidak dikenal. \n");
    }
}

```

5 Algoritma-Algoritma Menarik

5.1 *string_compare*

```

int string_compare(char *s1, char *s2) {
    int i = 0;
    while (s1[i] != '\0' || s2[i] != '\0') {
        if (s1[i] != s2[i])
            return s1[i] - s2[i];
        if (s1[i] == '\0' || s2[i] == '\0')
            break;
        i++;
    }
    return 0;
}

```

Fungsi ini membandingkan dua buah *string* (s1 dan s2). Saat s1 = s2 maka program akan return 0. Sedangkan saat s1 tidak sama dengan s2, fungsi akan return hasil pengurangan *ascii value* s1[i] dengan s2[i]. Algoritma ini menarik karena berbeda dengan di python yang kami pelajari saat TPB, untuk membandingkan dua buah stream diperlukan perbandingan per karakter.

5.2 *string_copy*

```

void string_copy(char *dest, char *src) {
    int i = 0;
    while ((dest[i] = src[i]) != '\0')

```

```
        i++;  
    }
```

Fungsi *string_copy* digunakan untuk menyalin sebuah *string* ke *string* lain. Algoritma ini unik karena *copy* dilakukan tidak secara langsung, tapi dilakukan per karakter.

5.3 wordtoint

```
int WordToInt(Word word) {  
    int result = 0;  
    int i;  
    for (i = 0; i < word.length; i++) {  
        result = result * 10 + (word.contents[i] - '0');  
    }  
    return result;  
}
```

Fungsi ini mengubah *string* yang seluruh karakternya berupa angka menjadi integer (harus gunakan function *isWordNumber* terlebih dahulu). Algoritma ini menarik karena penggunaan `result = result * 10 + (word.contents[i] - '0')` untuk memastikan bahwa *string* dapat dengan benar dikonversi menjadi *integer*.

5.4 strlen

```
int strlen(char *str){  
    int len = 0;  
    while(str[len]!='\0'){  
        len++;  
    }  
    return len;  
}
```

Fungsi ini mengembalikan panjang dari *string* dengan cara menghitung per-karakter hingga mencapai `'\0'`. Algoritma ini menarik karena cara menghitung panjang string dilakukan secara manual per-karakter, bukan dengan fungsi seperti *length()* di python.

6 Data Test

6.1 Data Test 1

Pada data test pertama, dijalankan suatu pengecekan untuk fitur register, permintaan barang, penambahan barang, dan penghapusan barang. Output yang diharapkan adalah pengguna berhasil register dan masuk ke dalam game melalui username dan password yang didaftarkan. Kemudian, data tersimpan pada file konfigurasi.

- a. Berikut adalah tampilan file konfigurasi default sebelum penambahan user

```
saves > ≡ file_konfigurasi.txt
1      5
2      40 Raygun
3      15 DragonSlayer
4      80 Gurren
5      65 Aerial
6      5 Frying Pan
7      3
8      70 JaneDoe rat
9      250 FBK_fox japanesegoblin
10     0 ETH sibotak
```

- b. Kemudian dilakukan uji dengan melakukan command START

```
Selamat Datang di PURRMART, silahkan masukkan command (START, LOAD, HELP):
START
PURRMART berhasil di-start
```

- c. Selanjutnya dilakukan pendaftaran dengan command REGISTER dan memasukkan username serta password

```
Silahkan masukkan command (REGISTER, LOGIN, HELP):
REGISTER
>> REGISTER
Username: pocari
Password: sweat
```

- d. Dilakukan uji command LOGIN dengan memasukkan kredensial yang telah didaftarkan sebelumnya

```

Akun dengan username pocari telah berhasil dibuat. Silakan LOGIN untuk melanjutkan.
LOGIN
>> LOGIN
Username: pocari
Password: sweat

Anda telah login ke PURRMART sebagai pocari.
===== [ 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

```

- e. Setelah itu, dilakukan command QUIT dan menyimpan program

```

Silahkan masukkan command: QUIT
Apakah ingin melakukan save sebelum keluar? (Y/N): Y
Masukkan nama file: file_konfigurasi.txt
Save file berhasil disimpan
Kamu keluar dari PURRMART.
Dadah^_^/

```

- f. Berikut merupakan output yang diharapkan

```

1 5
2 Raygun
3 40
4 DragonSlayer
5 15
6 Gurren
7 80
8 Aerial
9 65
10 Frying Pan
11 5
12 4
13 70 JaneDoe rat
14 250 FBK_fox japanesegoblin
15 0 ETH sibotak
16 1000 pocari sweat
17

```

6.2 Data Test 2

Pada data test kedua, dijalankan suatu pengecekan untuk fitur melihat list barang, menambahkan barang ke dalam antrian, dan menambahkan barang baru ke dalam list. Output yang diharapkan adalah barang berhasil ditambahkan ke dalam list.

- a. Masuk ke dalam sistem dengan menjalankan command START kemudian LOGIN menggunakan existing user

```
(START, LOAD, HELP):  
START  
PURRMART berhasil di-start  
Silahkan masukkan command (REGISTER, LOGIN, HELP):  
LOGIN  
>> LOGIN  
Username: JaneDoe  
Password: rat
```

- b. Jalankan command STORE LIST dan lihat isinya

```
Silahkan masukkan command: STORE LIST  
List barang yang ada di toko:  
- Raygun  
- DragonSlayer  
- Gurren  
- Aerial  
- Frying Pan  
Silahkan masukkan command: █
```

- c. Jalankan command STORE REQUEST dan masukkan barang yang diinginkan

```
Nama barang yang diminta: Botol  
Permintaan barang 'Botol' telah ditambahkan ke antrian  
.  
Silahkan masukkan command: █
```

- d. Jalankan command yang sama dan masukkan barang yang sama untuk mengecek apakah barang bisa ditambahkan

```
Silahkan masukkan command: STORE REQUEST  
Nama barang yang diminta: Botol  
Barang dengan nama yang sama sudah ada di antrian!  
Silahkan masukkan command: █
```

- e. Ketik perintah “STORE SUPPLY” untuk menambahkan barang yang diminta, dan ketik “terima”

```
Silahkan masukkan command: STORE SUPPLY
Apakah kamu ingin menambahkan barang Botol (Terima/Turda/Tolak): Terima
Harga barang: 100
'Botol' dengan harga 100 telah ditambahkan ke toko.
Silahkan masukkan command: 
```

- f. Keluar dari sistem dengan melakukan command “QUIT” dan ketik “Y” untuk menyimpan file

```
Silahkan masukkan command: QUIT
Apakah ingin melakukan save sebelum keluar? (Y/N): Y
Masukkan nama file: barangbaru.txt
Save file berhasil disimpan
Kamu keluar dari PURRMART.
Dadah^_^/
```

- g. Periksa file konfigurasi

```
saves > ≡ barangbaru
1 6
2 Raygun
3 40
4 DragonSlayer
5 15
6 Gurren
7 80
8 Aerial
9 65
10 Frying Pan
11 5
12 Botol
13 100
14 3
15 70 JaneDoe rat
16 250 FBK_fox japanesegoblin
17 0 ETH sibotak
18
```

6.3 Data Test 3

- a. Ketik command “LOAD” di terminal

```
Selamat Datang di PURRMART, silahkan masukkan command  
(START, LOAD, HELP):  
LOAD  
>> LOAD: file_konfigurasi
```

- b. Ketik command “LOGIN” di terminal untuk masuk ke purrmart

```
>> LOGIN  
Username: JaneDoe  
Password: rat  
  
Anda telah login ke PURRMART sebagai JaneDoe.  
=====[ 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
```

- c. Ketik perintah “HELP”

```
Silahkan masukkan command: 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  
Silahkan masukkan command: █
```


- d. Ketik perintah “WORK”

```
Silahkan masukkan command: WORK
Daftar Pekerjaan:
1. Manager (pendapatan=100, durasi=10)
2. Sedot WC (pendapatan=50, durasi=5)
3. Kasir (pendapatan=70, durasi=7)
4. Supplier (pendapatan=80, durasi=8)
Masukkan pekerjaan yang dipilih : Sedot WC
Anda sedang bekerja sebagai Sedot WC... harap tunggu
pekerjaan selesai, +50 rupiah telah ditambahkan ke a
un anda
Silahkan masukkan command: █
```

- e. Ketik perintah “WORK CHALLENGE”

```

Silahkan masukkan command: WORK CHALLENGE
Daftar challenge yang tersedia:
1. Tebak Angka (biaya main = 50)
2. WORDL399 (biaya main = 50)
3. Quit Challenge.

Masukan challenge yang hendak dimainkan :1
Selamat datang di tebak angka silahkan masukan angka 1-100
Tebak Angka: 20
Tebakanmu lebih kecil
Tebak Angka: 50
Tebakanmu lebih kecil
Tebak Angka: 80
Tebakanmu lebih besar!
Tebak Angka: 70
Tebakanmu lebih besar!
Tebak Angka: 60
Tebakanmu lebih besar!
Tebak Angka: 65
Tebakanmu lebih besar!
Tebak Angka: 55
Tebakanmu lebih besar!
Tebak Angka: 52
Tebakanmu lebih kecil
Tebak Angka: 53
Selamat and mendapatkan 5 rupiah
Silahkan masukkan command: 5

```

f. Keluar dari program

```

Silahkan masukkan command: QUIT
Apakah ingin melakukan save sebelum keluar? (Y/N): N
Kamu keluar dari PURRMART.
Dadah^_^/

```

7 Test Script

Berikut skrip pengujian yang dilakukan.

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	START, REGISTER	Mengecek apakah file	1. Periksa file file konfigurasi.txt	Data Test 1	Username dan password baru	Sesuai ekspektasi

STEI- ITB	<nomor dokumen>	Halaman 26 dari 36 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.		

	, LOGIN, QUIT, SAVE	konfigurasi dapat dimuat pada ketika dilakukan START, REGISTER dapat menambahkan user baru, dibuktikan dengan berhasil LOGIN dengan kredensial yang diregister, kemudian terdapat perubahan pada file konfigurasi ketika dilakukan SAVE	2. Jalankan program dengan mengetik ./main 3. Ketik command "START" di terminal 4. Ketik command "REGISTER" di terminal 5. Masukkan username dan password sesuai keinginan 6. Ketik command "LOGIN" untuk masuk ke dalam purrmart 7. Keluar dari sistem dengan melakukan command "QUIT" dan ketik "Y" untuk menyimpan file 8. Periksa file_konfigurasi dan bandingkan hasilnya		berhasil ditambahkan melalui REGISTER pada file_konfigurasi	
2	START, LOGIN, REQUEST, SUPPLY, QUIT	Mengecek apakah program bisa menambahkan barang ke dalam permintaan, menambahkan permintaan, dan keluar lalu menyimpan	1. Jalankan program dengan mengetik ./main 2. Ketik command "START" di terminal 2. Ketik command "LOGIN" di terminal untuk masuk ke purrmart 3. Ketik perintah "STORE LIST" untuk melihat daftar barang pada toko 4. Ketik perintah "STORE REQUEST" untuk meminta barang ditambahkan 5. Ketik perintah yang sama, dan masukkan barang yang sama 6. Ketik perintah "STORE SUPPLY" untuk menambahkan barang yang diminta, dan ketik "terima" 7. Keluar dari program dan lakukan save, periksa file konfigurasi apakah	Data Test 2	Barang berhasil ditambahkan melalui command STORE SUPPLY dan terdapat di file konfigurasi	Sesuai ekspektasi

			barang telah disimpan atau belum			
3	LOAD, LOGIN, HELP, WORK, WORK CHALLENGE, QUIT	Mengecek apakah program bisa load file konfigurasi, menjalankan help, work, dan work challenge	1. Jalankan program dengan mengetik ./main 2. Ketik command "LOAD" di terminal 2. Ketik command "LOGIN" di terminal untuk masuk ke purrmart 3. Ketik perintah "HELP" untuk menjalankan tebak angka 4. Ketik perintah "WORK" 5. Ketik perintah "WORK CHALLENGE" 8. Keluar dari program dan lakukan save, periksa file konfigurasi apakah barang telah disimpan atau belum	Data Test 3	Berhasil LOAD, HELP berhasil dijalankan, WORK, WORK CHALLENGE berhasil dijalankan	Sesuai ekspektasi

8 Pembagian Kerja dalam Kelompok

Nama Lengkap - NIM	Deskripsi Tugas
Karunia Mega Lestari - 18221126	<ul style="list-style-type: none"> • Membuat quit, save • Membuat driver ADT • Membuat laporan
Mahesa Satria Prayata - 18223082	<ul style="list-style-type: none"> • Membuat help • Membuat file konfigurasi • Membuat laporan
Rafli Dwi Nugraha - 18223038	<ul style="list-style-type: none"> • Membuat work, workchallenge • Bug fixing • Menambahkan fungsi mesinkata • membantu pembuatan main
M Khalfani Shaquille Indrajaya - 18223104	<ul style="list-style-type: none"> • Membuat store list, request, supply, remove • Membuat ADT listbarang • Menambahkan dan menyempurnakan ADT mesinkata, queue • Memperbaiki bugs pada program lain • Menyusun main

Alghan Pridanusuta - 18223058	<ul style="list-style-type: none"> • Membuat menu register, login, logout • Membuat ADT User • Membantu membuat main • Membantu membuat laporan • Fix bugs
Michael Ballard Isaiah Silaen - 18223080	<ul style="list-style-type: none"> • Membuat Load, Start, Readme • Membantu pembuatan main • Membantu pembuatan laporan • Bug hunting

9 Lampiran

9.1 Deskripsi Tugas Besar

Agen Purry sedang menikmati tidur siangnya ketika dia tiba-tiba mendengar alarm dari belakang sofa. Suatu pintu rahasia terbuka di bawah dirinya dan ia jatuh ke ruang bawah tanah dan langsung disambut dengan misi terbarunya.



“Ah Agen Purry, maaf harus mengganggu waktu tidur kamu tapi kami mendapatkan laporan bahwa Dr. Asep Spakbor sedang membuat suatu mesin yang dinamakan ‘Oppenheimer-inator’ yang akan menghancurkan wilayah tiga negara bagian. Aku membutuhkan bantuanmu untuk menghentikan Dr. Asep Spakbor, ini merupakan ancaman terbesar yang pernah ia buat.”

And indeed it was. Setelah pertarungan sengit selama 3 bulan 13 hari 2 jam 47 menit dan 2 detik, suplai senjata dan suplai peralatan yang dimiliki OWCA mulai menipis. Harapan kemenangan OWCA mulai memudar...

Tanpa disangka, Agen Purry mengeluarkan senjata rahasia miliknya: menjadi orang Bojongsoang yang memiliki kenalan pegawai Borma. Toko Borma adalah komponen penting yang dapat membawakan kemenangan untuk OWCA pada waktu-waktu kritis ini. Sebab, meskipun Borma terlihat seperti supermarket pada umumnya, mereka sebenarnya merupakan pemasok barang-barang perang. Namun terdapat satu masalah kecil, Borma masih beroperasi secara tatap muka dan OWCA tidak memiliki transport untuk pergi ke Bojongsoang.



Untuk menyelesaikan permasalahan ini, OWCA mengontak tim programmer paling andalnya untuk merancang suatu sistem jual beli ke Borma dengan nama PURRMART! Benar, tim tersebut adalah kalian! Misi ini akan menantang dan menguji kalian. Namun, dengan kerja tim dan tekad yang kuat, kalian pasti dapat menghadapi tantangan ini.

Good luck, programmers! OWCA counts on you! (•̀ω•́) ✧

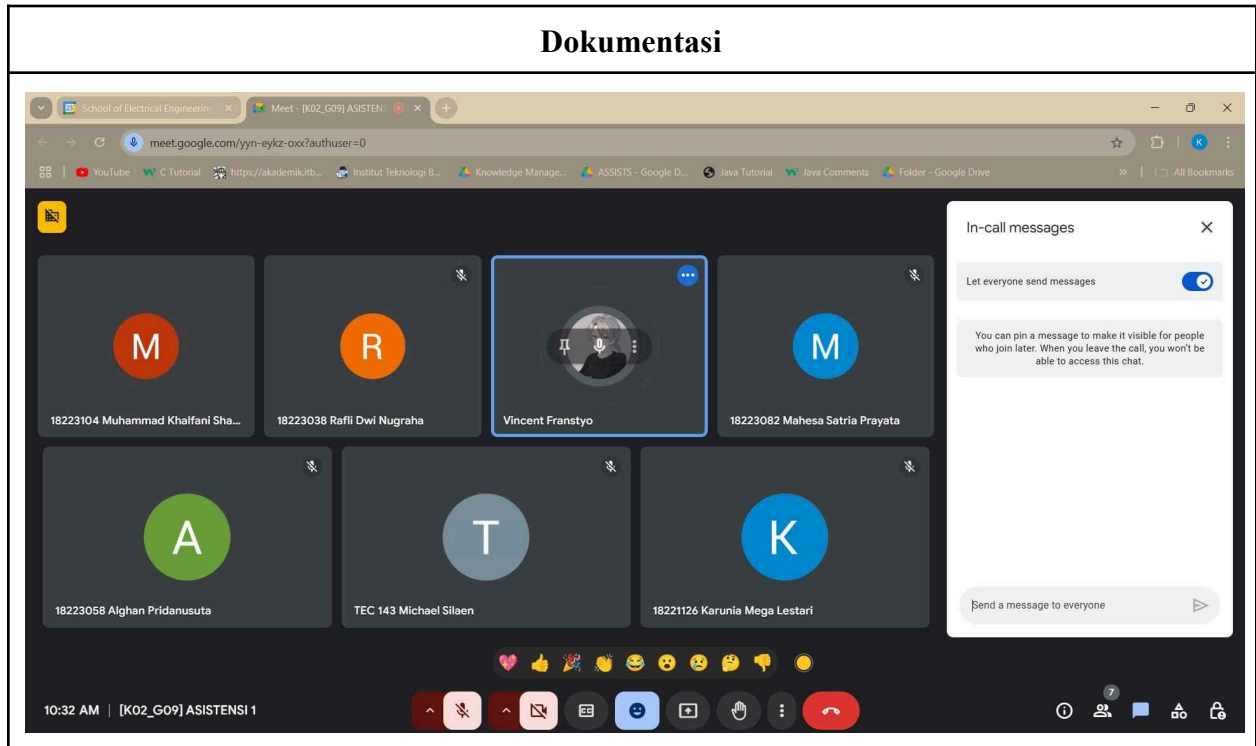
9.2 Notulen Rapat

Hari	Sabtu	Tanggal	23 November 2024
Kelas	02	Kelompok	03
Waktu	10.00 - 11.00	Tempat	Google Meet

Anggota Kelompok	NIM	Nama
	18221126	Karunia Mega Lestari
	18223082	Mahesa Satria Prayata
	18223038	Rafli Dwi Nugraha
	18223104	M Khalfani Shaquille Indrajaya
	18223058	Alghan Pridanusuta
	18223080	Michael Ballard Isaiah Silaen
Nama Asisten	Vincent Franstyo	

Hasil Asistensi	
No.	Catatan Asisten
1.	Coba cek lagi buat store di ADT yang handle 2 kata.
2.	Buat Bonus DNA, tetap harus input/bikin validasi nya dan untuk demo nya bebas asisten.
3.	Yang di save di file konfigurasi itu jumlah barang di sistem, harga, nama barang, bilangan bulat positif yang menunjuk berapa banyak pengguna.
4.	Kalau dari UI, minimal sesuaikan dengan yang ada di spesifikasi, tapi kalau bisa kreatif bebas kalian saja.
5.	Jangan lupa laporan
6.	sebenarnya input di masing-masing fungsi boleh saja, cuman itu bukan git practice yang baik, baiknya semua input di main. Sebaiknya di satu fungsi itu, cuman buat proses saja.
7.	Buat di register, lebih baik bikin validasi nya buat input 2 kata.
8.	Yang penting spesifikasi umum yang dikerjakan, kalau ada waktu boleh kerjain bonus, dan jangan lupa laporan.

9.	Harus bikin driver sama makefile.
10.	Driver berfungsi untuk memastikan fungsi dalam ADT kalian berjalan dengan lancar. Jadi bikin program/main buat ADT kalian.
11.	Readme di cicil.



9.3 Log Activity Anggota Kelompok

NAMA	LOG ACTIVITY
Alghan Pridanusuta	<p>16 November 2024: Pembagian tugas milestone 1.</p> <p>17-23 November 2024: Proses pembuatan Menu Login, Logout, Register.</p> <p>24 November 2024:</p>

	<p>Testing dan fix bugs</p> <p>25 November 2024: Membantu pembuatan main dan finalisasi Membantu membuat laporan.</p>
Mahesa Satria Prayata	<p>16 November 2024: Pembagian tugas milestone 1.</p> <p>17-22 November 2024 Pembuatan fitur help dan membuat laporan</p> <p>23-25 November 2024 Membantu membuat laporan</p>
Michael Ballard Isaiah Silaen	<p>16 November 2024: Pembagian tugas milestone 1.</p> <p>17-24 November 2024 Pembuatan Start, Load</p> <p>25 November Pembuatan readme Membantu main dan laporan</p>
Rafli Dwi Nugraha	<p>16 November 2024: Pembagian tugas milestone 1.</p> <p>20 November 2024: Pembuatan Repository Github.</p> <p>21-25 2024: Pembuatan Work, WorkChallenge Pembuatan compiler Penyesuaian struktur</p>
M Khalfani Shaquille Indrajaya	<p>16 November 2024: Pembagian tugas milestone 1.</p> <p>17-22 November 2024: Pembuatan ADT List Barang, store list, request, supply, move.</p> <p>23-25 November 2024:</p>

	Membuat Main dan fix debugging.
Karunia Mega Lestari	16 November 2024: Pembagian tugas milestone 1. 21 - 25 November 2024: Membuat quit, save, driver, dan laporan

DOKUMENTASI



9.4 <Lain-Lain>

main.c full code

```

printf("Selamat Datang di PURRMART, silahkan masukkan command (START, LOAD, HELP): \n");
START();

char input[MAX_WORD_LENGTH];
int started = 1;
while (started == 1){
    StartWords();
    CopyWordToString(input);
    if (string_compare(input, "START") == 0){
        if (Starttr(&userlist, &listbarang) == 2){
            started = 0;
        }
    }
    else if (string_compare(input, "LOAD") == 0){
        if (LoadFile(&userlist, &listbarang) == 2){
            started = 0;
        }
    }
    else if (string_compare(input, "HELP") == 0){
        help_welcome();
    }
    else {
        printf("command tidak dikenal. Masukkan HELP jika anda butuh bantuan. \n");
    }
}

```

```

int logged = 1;
printf("Silahkan masukkan command (REGISTER, LOGIN, HELP): \n");
while(logged){
    START();
    StartWords();
    copyWordToString(input);

    if (string_compare(input, "REGISTER") == 0){
        RegisterUser(&userlist);
    }

    else if (string_compare(input, "LOGIN") == 0){
        if (loginUser(&userlist, &user) == 1){
            int menu = 1;
            help_main();
            while (menu){
                printf("Silahkan masukkan command: ");
                START();
                StartWords();
                char inputmain[MAX_WORD_LENGTH];
                copyWordToString(inputmain);

                if (string_compare(inputmain, "WORK") == 0){
                    UserWorks(&worklist, &user);
                }

                else if (string_compare(inputmain, "WORK CHALLENGE") == 0){
                    printf("Daftar challenge yang tersedia: \n");
                    printf("1. Tebak Angka (biaya main = 50) \n");
                    printf("2. WORD1399 (biaya main = 50) \n");
                    printf("3. Quit Challenge. \n");

                    int inChallenge = 0;
                    while (!inChallenge){
                        printf("\nMasukan challenge yang hendak dimainkan :");

                        char inputChallenge[MAX_LEN];
                        START();
                        StartWords();
                        copyWordToString(inputChallenge);

                        if (string_compare(inputChallenge, "1") == 0){
                            challengeTebakAngka(&user);
                            inChallenge = 1;
                        }

                        else if (string_compare(inputChallenge, "2") == 0){
                            playWord13(&user);
                            inChallenge = 1;
                        }

                        else if (string_compare(inputChallenge, "3") == 0){
                            printf("Anda keluar dari challenge. \n");
                            inChallenge = 1;
                        }

                        else {
                            printf("Command tidak dikenal. \n");
                        }
                    }

                }

                else if (string_compare(inputmain, "STORE LIST") == 0){
                    display_store_list(listbarang);
                }

                else if (string_compare(inputmain, "STORE REQUEST") == 0){
                    store_request(listbarang, &queue);
                }

                else if (string_compare(inputmain, "STORE SUPPLY") == 0){
                    store_supply(&listbarang, &queue);
                }

                else if (string_compare(inputmain, "STORE REMOVE") == 0){
                    store_remove(&listbarang);
                }

                else if (string_compare(inputmain, "LOGOUT") == 0){
                    logoutUser(&user);
                    menu = 0;
                }

                else if (remainderWord(inputmain, "SAVE")){
                    save(&userlist, &listbarang, inputmain);
                }

                else if (string_compare(inputmain, "QUIT") == 0){
                    quit(&queue, &userlist, &listbarang);
                    return 0;
                }

                else if (string_compare(inputmain, "HELP") == 0){
                    help_main();
                }

                else{
                    printf("Command tidak dikenal. \n");
                }
            }
        }
    }
}

```

```
    }  
    }  
    else{  
        printf("Gagal untuk login. \n");  
    }  
}  
else if (string_compare(input, "HELP") == 0){  
    help_login();  
}  
else{  
    printf("Command tidak dikenal. Masukkan HELP jika anda butuh bantuan. \n");  
}  
}  
}
```