

LAPORAN TUGAS BESAR

IF2111 Algoritma dan Struktur Data STI


PURRMART

Dipersiapkan oleh:

Kelompok 6 - K02

Brandon Theodore Ferrinov	(18223020)
Ratukhansa Salsabila	(18223034)
Darryl Rayhananta Adenan	(18223042)
Raditya Zaki Athaya	(18223086)
Matthew Sebastian Kurniawan	(18223096)

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2111-TB-K02-06</i>		38
		<i>Revisi</i>	0	25/11/2024

Daftar Isi

1	Ringkasan	3
2	Penjelasan Tambahan Spesifikasi Tugas	5
2.1	Spesifikasi Fitur Quantum WORDL3	5
3	Struktur Data (ADT)	5
3.1	ADT List	5
3.2	ADT Queue	5
3.3	ADT Mesin Karakter	6
3.4	ADT Mesin Kata	6
3.5	ADT Custom : User	6
3.6	ADT Custom : Barang	7
3.7	ADT List Dinamis	7
4	Program Utama	8
5	Algoritma-Algoritma Menarik	14
5.1	Algoritma Random Word pada WORDL3	14
5.2	Algoritma Case Insensitive pada WORDL3	14
6	Data Test	14
6.1	Data Test START	14
6.2	Data Test LOAD	15
6.3	Data Test LOGIN	16
6.4	Data Test LOGOUT	17
6.5	Data Test REGISTER	17
6.6	Data Test WORK	18
6.7	Data Test WORK CHALLENGE TEBAK ANGKA	19
6.8	Data Test WORK CHALLENGE WORDL3	20
6.9	Data Test STORE LIST	23
6.10	Data Test STORE REQUEST	23
6.11	Data Test STORE SUPPLY	24
6.12	Data Test STORE REMOVE	25
6.13	Data Test HELP	26
6.14	Data Test SAVE	27
6.15	Data Test QUIT	28
6.16	Data Test BONUS QUANTUM WORDL3	28
7	Test Script	30
8	Pembagian Kerja dalam Kelompok	33
9	Lampiran	35
9.1	Deskripsi Tugas Besar	35
9.2	Notulen Rapat	35

9.3 Log Activity Anggota Kelompok	37
9.4 Lampiran Link Repository	38

1 Ringkasan

Agen Purry membantu OWCA mengatasi permasalahan yang disebabkan oleh Dr. Asep Spakbor. Dr. Asep Spakbor menciptakan suatu mesin penghancur bernama ‘Oppenheimer-inator’ yang akan menghancurkan wilayah tiga negara bagian, yang mengakibatkan pasokan barang semakin menipis. Untuk menghentikan ini, Agen Purry membuka tokonya untuk membantu OWCA dalam menghadapi kendala pasokan barang perang. Namun, OWCA tidak memiliki akses transportasi untuk mencapai lokasi toko tersebut. Sehingga diciptakan aplikasi bernama Purrmart yang memungkinkan OWCA mengakses barang-barang perang untuk bisa memesan dan mengelola kebutuhan perang secara *virtual*.

Purrmart merupakan sebuah aplikasi supermarket yang mensimulasikan aktivitas beli barang. Aplikasi ini berbasis CLI (*Command-line Interface*) dan diprogram menggunakan bahasa C. Purrmart menggunakan struktur data list, mesin karakter, mesin kata, queue, dan custom. Pengguna dapat melakukan berbagai operasi seperti melihat barang toko, meminta dan menyuplai barang ke toko, menyimpan dan membeli barang dalam keranjang, menampilkan barang yang sudah dibeli, hingga bekerja untuk menghasilkan uang. *Command* akan di input oleh pengguna.

Laporan ini berisikan penjelasan mengenai aplikasi PurrMart yang telah kami buat. Laporan ini mencakup penjelasan singkat spesifikasi PurrMart, Struktur Data dan Algoritma yang digunakan, Test Data dan Script yang dilakukan, Pembagian kerja, serta lampiran terkait aplikasi PurrMart.

Tugas besar ini disusun untuk menambah wawasan kami sebagai mahasiswa STI yang mengikuti mata kuliah Algoritma Struktur Data STI IF2111 mengenai penggunaan struktur data (ADT) dan bahasa C yang telah kami pelajari di kelas.

2 Penjelasan Tambahan Spesifikasi Tugas

2.1 Spesifikasi Fitur Quantum WORDL3

Challenge Quantum WORDL3 merupakan permainan tebak kata berjumlah lima karakter. Pemain memiliki 9 (sembilan) kesempatan untuk menebak 4 (empat) kata sekaligus dalam satu waktu (keempat kata harus dijamin berbeda). Kata harus berupa kata valid, tidak boleh sekadar *string* acak, bahasa dibebaskan (disarankan bahasa Indonesia/Inggris). Pada setiap giliran, program akan mencetak ulang kata yang dimasukkan, tetapi dengan penanda tertentu. Huruf yang benar dan berada pada tempat yang tepat dicetak biasa. Huruf yang benar, tetapi berada di tempat yang salah diberi tanda “*” setelah hurufnya. Huruf yang tidak ada sama sekali pada kata diberi tanda “%” setelah hurufnya.

3 Struktur Data (ADT)

3.1 ADT List

```
typedef struct {  
    EType A[MaxEl];  
} List;
```

ADT list bertipe integer. ADT List digunakan untuk melakukan penyimpanan data statis, melakukan operasi manipulasi list sederhana dan manajemen kumpulan elemen integer dengan batasan ukuran. ADT list dipilih karena dapat menyimpan data username, mencatat transaksi, dan *logging* sederhana, serta mudah diimplementasikan. ADT List diimplementasikan dengan nama file “list.c” dan header “list.h”.

3.2 ADT Queue

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

ADT Queue ini bertipe integer. ADT Queue digunakan untuk mengatur sistem antrian sesuai aturan FIFO, pencatatan data yang berurut, dan buffer data sementara. ADT Queue dipilih karena sederhana dan efisien. ADT Queue diimplementasikan dengan nama file “queue.c” dan header “queue.h”.

3.3 ADT Mesin Karakter

ADT Mesin Karakter bertipe character. ADT ini digunakan untuk melakukan *parsing* teks, Membaca file karakter per karakter, *Preprocessing* input sebelum diproses lebih lanjut, implementasi *scanner* sederhana. ADT mesin karakter juga digunakan sebagai dasar untuk membuat mesin kata. Mesin karakter digunakan untuk membaca karakter yang ada, input *command* dari pengguna, membaca isi file, dan diimplementasikan di mesin kata. Mesin karakter diimplementasikan sebagai ADT Mesin Karakter dengan nama file “mesinkarakter.c” dan header “mesinkarakter.h”.

3.4 ADT Mesin Kata

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

ADT Mesin Kata bertipe character, merupakan salah satu implementasi dari ADT Mesin Karakter. ADT ini Membaca perintah dari input pengguna Mengurai konfigurasi file, Mengekstrak informasi dari teks, Validasi input, Implementasi parser sederhana. Mesin kata digunakan untuk membaca kata yang ada, input *command* dari pengguna, dan membaca isi file. Mesin kata diimplementasikan sebagai ADT Mesin Kata dengan nama file “mesinkata.c” dan header “mesinkata.h”.

3.5 ADT Custom : User

```
typedef struct {
    char name[MAX_LEN];
    char password[MAX_LEN];
```

```
integer money;  
} User;
```

ADT User mendefinisikan struktur data *User* yang memiliki atribut *name* dan *password* masing-masing memiliki maksimal 100 karakter, serta *money* yang berupa integer. ADT *Custom User* dipilih karena sederhana dan efisien dalam menyimpan data *User*. ADT *User* diimplementasikan dalam file header bernama “custom.h”.

3.6 ADT *Custom* : Barang

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

ADT Barang mendefinisikan struktur data Barang yang memiliki atribut *name* yang memiliki maksimal 100 karakter dan atribut *price* yang bertipe integer. ADT Custom Barang dipilih karena sederhana dan efisien dalam menyimpan data Barang. ADT Barang diimplementasikan dalam file header bernama “custom.h”.

3.7 ADT *List* Dinamis

```
typedef struct {  
    Barang *A;  
    int Capacity;  
    int Neff;  
} ArrayDin;
```

ADT List Dinamis bertipe integer. ADT List Dinamis digunakan untuk melakukan penyimpanan data dinamis, melakukan operasi manipulasi list dinamis dan manajemen kumpulan elemen integer tanpa batasan ukuran. ADT list dinamis dipilih karena dapat menyimpan data secara fleksibel dan ukuran yang dinamis, serta mudah diimplementasikan. ADT List dinamis diimplementasikan dengan nama file “dynamiclist.c” dan header “dynamiclist.h”.

4 Program Utama

```
// Program "main.c"
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include "main.h"

boolean start;
boolean login;
boolean load;
User user;

int main(){
    while(true){
        if(!start && !login){
            printf("\n");
            printf("  _____
\n");
            printf("  \ \ \ \      / /  _ _ | |  / _ / _ \ \ | \ \ / | _ _ | | _ / _
\n");
            printf("  \ \ \ \ / \ / / | _ | | | | | | | | \ \ | | _ | | | |
| | \n");
            printf("  \ \ v v / | | _ | | | | | | | | | | _ | | | |
| \n");
            printf("  \ \ _ \ \ _ / | _ _ | _ _ \ \ _ \ \ _ / | | | | _ |
\n");
            printf("\n");
            printf("
_____ \n");
            printf("
| _ \ \ | | | | _ \ \ | _ \ \ \ \ / | / \ \
| _ \ \ _ _ | \n");
            printf("
| _ | | | | | | | | | | | | \ \ / | / _ \ \ |
| _ | | | | \n");
```



```

printf("          | _/| |_| | _ <| _ <| | | / _ _ \\| _
< | | \\n");
printf("          |_|   \\_/_/|_| \\_\\_| \\_\\_| |/_/
\\_\\_| \\_\\_| \\n");

printf("-----\\n");

printf("-----\\n");

printf("\\n");

printf("1. START\\n");
printf("2. LOAD\\n");
printf("3. QUIT\\n");
printf("4. HELP\\n");
printf(">>> ");
STARTWORD2();
char command[50];
char txt[50];
int i = 0;
while(currentWord.TabWord[i] != ' ' && currentWord.TabWord[i] != '\\0'){
    command[i] = currentWord.TabWord[i];
    (&currentWord)->TabWord[i] = '\\0';
    i++;
}
command[i] = '\\0';
(&currentWord)->TabWord[i] = '\\0';
if(isEqual(command, "START")){
    MULAI();
} else if (isEqual(command, "LOAD")) {
    i++;
    int j = i;
    while(currentWord.TabWord[i] != '\\0'){
        txt[i - j] = currentWord.TabWord[i];
        (&currentWord)->TabWord[i] = '\\0';
    }
}

```

```

        i++;
    }
    txt[i-j] = '\0';
    LOAD(txt);
} else if(isEqual(command, "QUIT")){
    printf("Aplikasi akan ditutup");
    exit(EXIT_SUCCESS);
} else if(isEqual(command, "HELP")){
    HELP(start, login);
} else {
    printf("Command tidak valid, silakan ulangi kembali!\n");
}
} else if (!login && start){
    printf("1. REGISTER\n");
    printf("2. LOGIN\n");
    printf("3. QUIT\n");
    printf("4. HELP\n");
    printf(">>> ");
    STARTWORD2();
    char command[50];
    char txt[50];
    int i = 0;
    while(currentWord.TabWord[i] != ' ' && currentWord.TabWord[i] != '\0'){
        command[i] = currentWord.TabWord[i];
        (&currentWord)->TabWord[i] = '\0';
        i++;
    }
    command[i] = '\0';
    (&currentWord)->TabWord[i] = '\0';
    if(isEqual(command, "REGISTER")){
        REGISTER();
    } else if (isEqual(command, "LOGIN")) {
        LOGIN();
    } else if(isEqual(command, "QUIT")){
        printf("Aplikasi akan ditutup");
        exit(EXIT_SUCCESS);
    }
}

```

```

    } else if(isEqual(command, "HELP")){
        HELP(start, login);
    } else {
        printf("Command tidak valid, silakan ulangi kembali!\n");
        MULAI();
    }
} else {
    printf("1. WORK\n");
    printf("2. WORK CHALLENGE\n");
    printf("3. STORE LIST\n");
    printf("4. STORE REQUEST\n");
    printf("5. STORE SUPPLY\n");
    printf("6. STORE REMOVE\n");
    printf("7. LOGOUT\n");
    printf("8. SAVE\n");
    printf("9. QUIT\n");
    printf("10. HELP\n");
    printf(">>> ");
    STARTWORD2();
    char command[50];
    char txt[50];
    int i = 0;
    while(currentWord.TabWord[i] != ' ' && currentWord.TabWord[i] != '\0'){
        command[i] = currentWord.TabWord[i];
        (&currentWord)->TabWord[i] = '\0';
        i++;
    }
    command[i] = '\0';
    (&currentWord)->TabWord[i] = '\0';
    if (isEqual(command, "WORK")) {
        i++;
        int j = i;
        while(currentWord.TabWord[i] != '\0'){
            txt[i - j] = currentWord.TabWord[i];
            (&currentWord)->TabWord[i] = '\0';
            i++;
        }
    }
}

```

```

    }
    txt[i-j] = '\0';

    if(txt[j] != '\0'){
        printf("\n");
        printf("saldo anda sekarang adalah: %d rupiah\n",user.money);
        selectChallenge();
        printf("saldo anda sekarang adalah: %d rupiah\n",user.money);
    } else {
        main_work();
        printf("saldo anda sekarang adalah: %d rupiah\n",user.money);
    }
}
else if(isEqual(command, "STORE")){
    i++;
    int j = i;
    while(currentWord.TabWord[i] != '\0'){
        txt[i - j] = currentWord.TabWord[i];
        (&currentWord)->TabWord[i] = '\0';
        i++;
    }
    txt[i-j] = '\0';
    if(isEqual(txt, "LIST")){
        STORE_LIST();
    } else if(isEqual(txt, "REQUEST")){
        STORE_REQUEST();
    } else if(isEqual(txt, "SUPPLY")){
        STORE_SUPPLY();
    } else if(isEqual(txt, "REMOVE")){
        STORE_REMOVE();
    }
} else if(isEqual(command, "LOGOUT")){
    LOGOUT();
} else if(isEqual(command, "HELP")){
    HELP(start, login);
}

```

```

else if(isEqual(command, "SAVE")){
    i++;
    int j = i;
    while(currentWord.TabWord[i] != '\0'){
        txt[i - j] = currentWord.TabWord[i];
        (&currentWord)->TabWord[i] = '\0';
        i++;
    }
    txt[i-j] = '\0';
    SAVE(txt);
}
else if(isEqual(command, "QUIT")){
    printf("Apakah kamu ingin menyimpan file?(Y/N)\n");
    printf(">>> ");
    STARTWORD2();
    char command[50];
    int i = 0;
    while(currentWord.TabWord[i] != ' ' && currentWord.TabWord[i] !=
'\0'){
        command[i] = currentWord.TabWord[i];
        (&currentWord)->TabWord[i] = '\0';
        i++;
    }
    command[i] = '\0';
    (&currentWord)->TabWord[i] = '\0';
    if(isEqual(command, "Y")){
        SAVE("default.txt");
    }
    printf("Aplikasi akan ditutup");
    exit(EXIT_SUCCESS);
} else {
    printf("Command tidak valid, silakan ulangi kembali!\n");
}
}
}

```

```
}
```

5 Algoritma-Algoritma Menarik

5.1 Algoritma *Random Word* pada WORDL3

Algoritma ini digunakan untuk menentukan jawaban pada fitur WORDL3 dan Quantum WORDL3. Algoritma ini memanfaatkan fungsi `rand()` yang sudah tersedia di dalam `stdlib.h`. Fungsi `rand()` bekerja dengan memilih secara acak dari array kata yang sudah tersedia. Algoritma ini menarik karena dapat menghasilkan jawaban yang random dan tidak mungkin sama (dalam quantum WORDL3).

5.2 Algoritma *Case Insensitive* pada WORDL3

Algoritma ini digunakan untuk mengubah kata yang di-input pemain yang memiliki karakter tidak kapital akan diubah menjadi kapital dan ditampilkan kembali dalam huruf kapital sehingga program WORDL3 akan tetap berjalan walaupun ada huruf kecil.

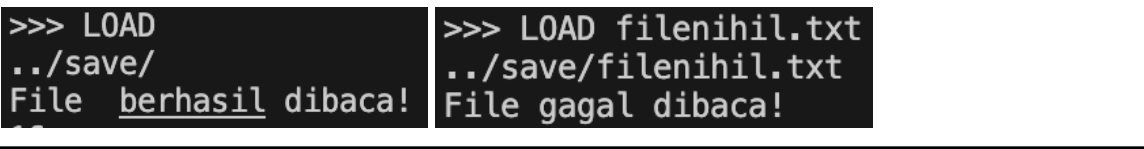
6 Data Test

6.1 Data Test *START*

Data tes:


<p>Hasil yang diharapkan:</p>
<pre>>> START File konfigurasi aplikasi berhasil dibaca. PURRMART berhasil dijalankan.</pre>

6.2 Data Test *LOAD*

<p>Data tes:</p>

<p>Hasil yang diharapkan:</p>
<pre>>> LOAD filenihil.txt // Meload /save/filenihil.txt* Save file tidak ditemukan. PURRMART gagal dijalankan.</pre>
<pre>>> LOAD savefile.txt // Meload /save/savefile.txt Save file berhasil dibaca. PURRMART berhasil dijalankan.</pre>

6.3 Data Test *LOGIN*

Data tes:

```
>>> LOGIN
Username: jojo
Password: gacor12345
LOGIN BERHASIL!
```

```
>>> LOGIN
Username: jojo
Password: 12345
Password Salah!
```

```
>>> LOGIN
Username: juna
Password: 12345
Username tidak ditemukan!
```

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.

6.4 Data Test *LOGOUT*

Data tes:

```
>>> LOGOUT  
1. REGISTER  
2. LOGIN  
3. QUIT  
4. HELP
```

(kembali ke menu awal)

Hasil yang diharapkan:

```
>> LOGOUT
```

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

6.5 Data Test *REGISTER*

Data tes:

```
>>> REGISTER  
Username: jojo  
Password: gacor12345
```

```
>>> REGISTER  
Username: praktikan  
Username sudah ada!
```

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.

6.6 Data Test *WORK*

Data tes:

Daftar pekerjaan:

1. Evil Lab Assistan (pendapatan=100, durasi=14s)
2. OWCA Hiring Manag (pendapatan=4200, durasi=21s)
3. Cikapundunginator Caretak (pendapatan=7000, durasi=30s)
4. Mewing Specialis (pendapatan=10000, durasi=22s)
5. Inator Connoisseur (pendapatan=997, durasi=15s)

Catatan: Masukkan nama pekerjaan persis seperti yang tertulis di atas

Masukkan pekerjaan yang dipilih: evil lab assitan

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

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

saldo anda sekarang adalah: 100 rupiah

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.

6.7 Data Test *WORK CHALLENGE* TEBAK ANGKA

Data tes:

```
saldo anda sekarang adalah: 10100 rupiah
```

```
>>> WORK CHALLENGE
```

```
Daftar challenge yang tersedia:
```

1. Tebak Angka (biaya main=200)
2. WORDL3 (biaya main=500)
3. Quantum WORDL3 (biaya main =1000)

```
Masukan challenge yang hendak dimainkan: 1
```

```
|*****!Selamat datang di permainan Tebak Angka!*****|
|*****!Anda memiliki 10 kesempatan untuk menebak!*****|
|Hadiah awal: 500 rupiah (berkurang 50 setiap tebakan salah)|
```

```
Kesempatan tersisa: 10
```

```
Hadiah saat ini: 500
```

```
Tebak angka (1-100): 99
```

```
!Tebakanmu lebih besar!
```

```
Kesempatan tersisa: 9
```

```
Hadiah saat ini: 450
```

```
Tebak angka (1-100): 100
```

```
Kesempatan tersisa: 1
```

```
Hadiah saat ini: 50
```

```
Tebak angka (1-100): 75
```

```
-----!Selamat! Tebakanmu benar!-----
+50 rupiah telah ditambahkan ke akun anda.
saldo anda sekarang adalah: 9950 rupiah
```

Hasil yang diharapkan:

```
>> WORK CHALLENGE
```

```
Daftar challenge yang tersedia:
```

1. Tebak Angka (biaya main=200)
2. W0RDL399 (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. W0RDL399 (biaya main=500)

Masukan challenge yang hendak dimainkan: 1

Tebak angka: 50

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

6.8 Data Test *WORK CHALLENGE W0RDL3*

Data tes:

>>> WORK CHALLENGE

Daftar challenge yang tersedia:

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)

1. Tebak Angka (biaya main=200)

2. WORDL3 (biaya main=500)

3. Quantum WORDL3 (biaya main =1000)

Masukan challenge yang hendak dimainkan: 2

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

Masukan kata tebakan Anda: PLUCK

Hasil:

A % D % I % E % U *

T % R % U L * Y %

P L U C K

Selamat, Anda menang!

+1500 rupiah telah ditambahkan ke akun Anda.

Hasil yang diharapkan:

>> WORK CHALLENGE

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)

2. W0RDL3 (biaya main=500)

3. Quantum W0RDL3 (biaya main=1000)

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.

```

6.9 Data Test *STORE LIST*

Data tes:

```

>>> STORE LIST
List barang yang ada di toko:
AK47
Lalabu
Ayam_Goreng_Crisbar

```

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

```

6.10 Data Test *STORE REQUEST*

Data tes:

<pre>>>> STORE REQUEST Nama barang yang diminta: kol_goreng Barang kol_goreng berhasil dimasukkan ke dalam antrian.</pre> <pre>>>> STORE REQUEST Nama barang yang diminta: Lalabu Barang dengan nama yang sama sudah ada di toko!</pre>
Hasil yang diharapkan:
<pre>>> STORE REQUEST Nama barang yang diminta: AK47</pre>
<pre>>> STORE REQUEST Nama barang yang diminta: Adaditoko Barang dengan nama yang sama sudah ada di toko!</pre>
<pre>>> STORE REQUEST Nama barang yang diminta: Adadiantrian Barang dengan nama yang sama sudah ada di antrian!</pre>

6.11 Data Test *STORE SUPPLY*

Data tes:
<pre>>>> STORE SUPPLY Apakah kamu ingin menambahkan barang kol_goreng: Tunda kol_goreng dikembalikan ke antrian.</pre> <pre>>>> STORE SUPPLY Apakah kamu ingin menambahkan barang kol_goreng: Tolak kol_goreng dihapuskan dari antrian.</pre> <pre>>>> STORE SUPPLY Apakah kamu ingin menambahkan barang kol_goreng: Tidaktidak < Balik ke menu ></pre> <pre>>>> STORE SUPPLY Apakah kamu ingin menambahkan barang kol_goreng: Terima Harga barang: 154 kol_goreng dengan harga 154 telah ditambahkan ke toko.</pre>

Hasil yang diharapkan:
<p>>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Terima Harga barang: 100</p> <p>AK47 dengan harga 100 telah ditambahkan ke toko.</p>
<p>>> STORE SUPPLY Apakah kamu ingin menambahkan barang AK47: Tunda</p> <p>AK47 dikembalikan ke antrian.</p>
<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>

6.12 Data Test *STORE REMOVE*

Data tes:
<pre>>>> STORE REMOVE Nama barang yang akan dihapus: anak_ayam Toko tidak menjual anak_ayam >>> STORE REMOVE Nama barang yang akan dihapus: buku_kuning buku_kuning telah berhasil dihapus.</pre>
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>

Toko tidak menjual Inator Neutralizer

6.13 Data Test *HELP*

Data tes:

```
>>> HELP
===SELAMAT DATANG DI BANTUAN WELCOME MENU===
1. START -> Untuk masuk sesi baru
2. LOAD -> Untuk memulai sesi berdasarkan file konfigurasi
3. QUIT -> Untuk keluar dari program
=====

===SELAMAT DATANG DI BANTUAN LOGIN MENU===
1. REGISTER -> Untuk mendaftarkan akun baru
2. LOGIN -> Untuk masuk ke dalam akun yang sudah terdaftar dan memulai sesi
3. QUIT -> Untuk keluar dari program
=====
1. REGISTER

===SELAMAT DATANG DI BANTUAN PURRMART===
1. WORK -> Untuk bekerja
2. WORK CHALLENGE -> Untuk mengerjakan challenge
3. STORE LIST -> Untuk melihat barang - barang di toko
4. STORE REQUEST -> Untuk meminta penambahan barang
5. STORE SUPPLY -> Untuk menambahkan barang dari permintaan
6. STORE REMOVE -> Untuk menghapus barang
7. LOGOUT -> Untuk keluar dari sesi
8. SAVE -> Untuk menyimpan state ke dalam file
9. QUIT -> Untuk keluar dari program
=====
```

Hasil yang diharapkan:

```
// Ketika perintah dipanggil pada welcome menu
>> 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

// Ketika perintah dipanggil pada login menu
>> 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]=====
```

1. WORK → Untuk bekerja
2. WORK CHALLENGE → Untuk mengerjakan challenge
3. STORE LIST → Untuk melihat barang-barang di toko
4. STORE REQUEST → Untuk meminta penambahan barang
5. STORE SUPPLY → Untuk menambahkan barang dari permintaan
6. STORE REMOVE → Untuk menghapus barang
7. LOGOUT → Untuk keluar dari sesi
8. SAVE → Untuk menyimpan state ke dalam file
9. QUIT → Untuk keluar dari program

6.14 Data Test SAVE

Data tes:

```
>>> SAVE  
  
../save/  
File berhasil di simpan!
```

Hasil yang diharapkan:

```
>> SAVE savefile.txt
```

Save file berhasil disimpan.

```
// File disimpan pada /save/savefile.txt
```

6.15 Data Test QUIT

Data tes:

```
>>> QUIT
Aplikasi akan ditutup%
```

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
```

6.16 Data Test BONUS QUANTUM WORDL3

Data tes:

```
>>> WORK CHALLENGE
```

```
saldo anda sekarang adalah: 10100 rupiah
```

```
>>> WORK CHALLENGE
```

```
Daftar challenge yang tersedia:
```

```
1. Tebak Angka (biaya main=200)
```

```
2. WORDL3 (biaya main=500)
```

```
3. Quantum WORDL3 (biaya main =1000)
```

```
Masukan challenge yang hendak dimainkan: 3
```

```
WELCOME TO QUANTUM WORDL3, YOU HAVE 9 CHANCES TO ANSWER 4 DIFFERENT WORDS!
```

```
-----
-----
-----
-----
-----
-----
-----
-----
-----
```

```
Masukkan kata ke-1: hallo hallo hallo hallo hallo
```

Hasil:

H % A % L * L * O % H % A % L * L O % H % A % L * L O % H % A * L * L O %

```
-----
-----
-----
-----
-----
-----
-----
-----
```

Hasil yang diharapkan:

>> **WORK CHALLENGE**

Daftar challenge yang tersedia:

1. Tebak Angka (biaya main=200)
2. W0RDL3 (biaya main=500)
3. Quantum W0RDL3 (biaya main=1000)

Masukan challenge yang hendak dimainkan: **3**

WELCOME TO QUANTUM W0RDL3, YOU HAVE 9 CHANCES TO ANSWER BEFORE YOU LOSE!

```
-----
-----
-----
-----
-----
-----
-----
-----
-----
```

7 Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Ditampilkan (Sesuai/Tidak)
1	START	Memeriksa file config.txt berhasil dibaca	<ul style="list-style-type: none"> - Melakukan <i>compile</i> semua file yang berkaitan dengan main.c - Memasukkan <i>command</i> START 	Data Test START	Data Test START	SESUAI
2	LOAD	Memeriksa fitur Load	Memasukkan <i>command</i> LOAD <filename> dua case : <ol style="list-style-type: none"> 1. file kosong 2. file tidak kosong 	Data Test LOAD	Data Test LOAD	SESUAI
3	LOGIN	Memeriksa fitur Login	Memasukkan <i>command</i> LOGIN dilakukan 3 case : <ol style="list-style-type: none"> 1. Login berhasil 2. Login gagal karena password salah 3. Login gagal karena username tidak ada 	Data Test LOGIN	Data Test LOGIN	SESUAI
4	LOGOUT	Memeriksa fitur Logout	Memasukkan <i>command</i> LOGOUT	Data Test LOGOUT	Data Test LOGOUT	SESUAI
5	REGISTER	Memeriksa fitur Register	Memasukkan <i>command</i> REGISTER dengan 2 case : <ol style="list-style-type: none"> 1. Register berhasil 2. Register gagal karena username sudah ada 	Data Test REGISTER	Data Test REGISTER	SESUAI
6	WORK	Memeriksa fitur Work	<ul style="list-style-type: none"> - Memasukkan <i>command</i> WORK - Jalankan dengan memilih salah satu WORK 	Data Test WORK	Data Test WORK	SESUAI
7	TEBAK ANGKA	Memeriksa fitur Tebak Angka	<ul style="list-style-type: none"> - Masukkan command 	Data Test WORK CHALLENGE	ata Test WORK CHALLENGE	SESUAI

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Ditampilkan (Sesuai/Tidak)
1	START	Memeriksa file config.txt berhasil dibaca	<ul style="list-style-type: none"> - Melakukan <i>compile</i> semua file yang berkaitan dengan main.c - Memasukkan <i>command</i> START 	Data Test START	Data Test START	SESUAI
			WORK CHALLENGE <ul style="list-style-type: none"> - Pilih Tebak angka - Mulai bermain 	E TEBAK ANGKA	E TEBAK ANGKA	
8	WORDL3	Memeriksa fitur WORDL3	<ul style="list-style-type: none"> - Masukkan command WORK CHALLENGE - Pilih WORDL3 - Mulai bermain 	Data Test WORK CHALLENGE WORDL3	Data Test WORK CHALLENGE WORDL3	SESUAI
9	STORE LIST	Memeriksa fitur STORE LIST	Memasukkan <i>command</i> STORE LIST	Data Test STORE LIST	Data Test STORE LIST	SESUAI
10	STORE REQUEST	Memeriksa fitur STORE REQUEST	Memasukkan command STORE REQUEST dengan 2 case <ol style="list-style-type: none"> 1. Jika barang belum ada, maka nama barang akan berhasil ditambahkan 2. Jika barang sudah ada, maka <i>request</i> gagal 	Data Test STORE REQUEST	Data Test STORE REQUEST	SESUAI
11	STORE SUPPLY	Memeriksa fitur STORE SUPPLY	Memasukkan command STORE SUPPLY dengan 4 case : <ol style="list-style-type: none"> 1. Supply diterima 2. Supply ditunda 3. Supply ditolak 4. Kembali ke menu 	Data Test STORE SUPPLY	Data Test STORE SUPPLY	SESUAI

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Ditampilkan (Sesuai/Tidak)
1	START	Memeriksa file config.txt berhasil dibaca	<ul style="list-style-type: none"> - Melakukan <i>compile</i> semua file yang berkaitan dengan main.c - Memasukkan <i>command</i> START 	Data Test START	Data Test START	SESUAI
12	STORE REMOVE	Memeriksa fitur STORE REMOVE	Memasukkan command STORE REMOVE dengan 2 case : <ol style="list-style-type: none"> 1. Nama barang berhasil di <i>remove</i> 2. Nama barang tidak berhasil di <i>remove</i> karena barang tidak dijual di toko 	Data Test STORE REMOVE	Data Test STORE REMOVE	SESUAI
13	HELP	Memeriksa fitur Help	Memasukkan command HELP	Data Test HELP	Data Test HELP	SESUAI
14	SAVE	Memeriksa fitur Save	Memasukkan command SAVE	Data Test SAVE	Data Test SAVE	SESUAI
15	QUIT	Memeriksa fitur Quit	Memasukkan command QUIT	Data Test QUIT	Data Test QUIT	SESUAI

8 Pembagian Kerja dalam Kelompok

Nama Lengkap (NIM)	Deskripsi Tugas
Brandon Theodore Ferrinov (18223020)	<ul style="list-style-type: none"> - Memperbaiki ADT Queue - Memperbaiki ADT Kustom - Memperbaiki ADT MesinKata (isEqual, len) - Memperbaiki ADT List - Membuat fungsi START - Membuat fungsi LOAD - Membuat fungsi QUIT - Membuat fungsi LOGIN

	<ul style="list-style-type: none"> - Membuat fungsi REGISTER - Membuat fungsi HELP - Memperbaiki fungsi STORE - Membuat fungsi LOGOUT - Membuat fungsi SAVE - Membuat MAIN - Memperbaiki Fungsi WORK
Ratukhansa Salsabila (18223034)	<ul style="list-style-type: none"> - Membuat fungsi store list - Membuat fungsi store request - Membuat fungsi store supply - Membuat fungsi store remove - Mencatat notulensi selama asistensi - Mengisi dokumen form asistensi - Melengkapi dokumen bagian ringkasan - Membuat tabel data test - Mengerjakan laporan bagian data test - Mengerjakan laporan bagian Test Script
Darryl Rayhananta Adenan (18223042)	<ul style="list-style-type: none"> - Menambahkan ADT - ADT yang diperlukan - Mengerjakan code WORDL3 - Mengerjakan bonus Quantum WORDL3 - Membuat README.md - Melakukan formatting laporan - Mengerjakan laporan bagian Ringkasan - Mengerjakan laporan bagian Struktur Data (ADT) - Mengerjakan laporan bagian Algoritma Menarik - Membuat Tabel Kontribusi - Mengisi Lampiran - Mengerjakan Test Script - Merapikan interface
Raditya Zaki Athaya (18223086)	<ul style="list-style-type: none"> - Membuat fungsi store list - Membuat fungsi store request - Membuat fungsi store supply - Membuat fungsi store remove - Mengerjakan laporan bagian data test
Matthew Sebastian Kurniawan (18223096)	<ul style="list-style-type: none"> - Membuat dan merapikan kode <i>command</i> work - Membuat dan merapikan kode <i>command</i> bagian work - Merapikan github - Membuat dan merapikan kode <i>command</i> bagian work challenge (tebak angka dan WORDL3) - Merapikan read me

	<ul style="list-style-type: none"> - Membuat MakeFile - Mengisi laporan bagian data testing - Merapikan interface coding - Mengerjakan dan merapikan bonus Quantum WORDL3 - Menggabungkan kode khususnya bagian work challenge dan menampilkan saldonya - Membuat driver untuk setiap ADT - Sumber penghubung dengan asisten dan memberi informasi
--	---

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**.

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*, dan Bekerja untuk menghasilkan uang.

9.2 Notulen Rapat

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

No. Kelompok/Kelas : Kelompok 6 / K-02
Nama Kelompok : Daemon
Anggota Kelompok (Nama/NIM) : 1. Darryl Rayhananta Adenan (18223042)
2. Matthew Sebastian Kurniawan (18223096)
3. Brandon Theodore Ferrinov (18223020)
4. Ratukhansa Salsabila (18223034)





STEI- ITB	<nomor dokumen>	Halaman 34 dari 37 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.		



5. Raditya Zaki Athaya (18223086)

Asisten Pembimbing

: Rayhan Maheswara Pramanda

Asistensi I

Tanggal : 21 November 2024	Catatan Asistensi: <ol style="list-style-type: none">1. Daftar ADT yang ada di spesifikasi wajib dipakai semuanya2. Boleh pakai fungsi lain tapi di laporan dijelaskan alasan penggunaannya3. Usahain program ada validasi input4. Pastikan semua spesifikasi wajib dulu yang diselesaikan sebelum lanjut ke spesifikasi bonus karena poinnya kecil5. Bagian laporan terdapat spesifikasi fitur tambahan, berisi penjelasan terkait spesifikasi bonus6. Jika terdapat algoritma yang menarik maka di bagian laporan dijelaskan, jika tidak ada maka boleh dihapus7. Penggunaan fgetc boleh8. Interface boleh diganti asalkan fungsionalitasnya masih sama dengan spesifikasi9. Untuk penggunaan getc nanti akan dikonfirmasi lagi
Tempat : Zoom Meeting	
Kehadiran Anggota Kelompok:	
<div>No</div> <div>NIM</div> <div>Tanda tangan</div> <div>1</div> <div>18223042</div>  <div>2</div> <div>18223096</div>  <div>3</div> <div>18223020</div>  <div>4</div> <div>18223034</div> 	

<p>5 18223086</p> 	
	<p>Tanda Tangan Asisten:</p> 

9.3 Log Activity Anggota Kelompok

Tanggal	Nama (NIM)	Aktivitas
13 November 2024	<ul style="list-style-type: none"> Brandon Theodore Ferrinov (18223020) Ratukhansa Salsabila (18223034) Darryl Rayhananta Adenan (18223042) Raditya Zaki Athaya (18223086) Matthew Sebastian Kurniawan (18223096) 	Pembagian Tugas berdasarkan branch <ul style="list-style-type: none"> - Work (18223042 & 18223096) - Store (18223034 & 18223086) - Main (18223020)
14 - 21 November 2024	<ul style="list-style-type: none"> Brandon Theodore Ferrinov (18223020) Ratukhansa Salsabila (18223034) Darryl Rayhananta Adenan (18223042) Raditya Zaki Athaya (18223086) Matthew Sebastian Kurniawan (18223096) 	Pengerjaan program Work
21 November	<ul style="list-style-type: none"> Brandon Theodore Ferrinov (18223020) 	Asistensi I

2024	<ul style="list-style-type: none"> • Ratukhansa Salsabila (18223034) • Darryl Rayhananta Adenan (18223042) • Raditya Zaki Athaya (18223086) • Matthew Sebastian Kurniawan (18223096) 	
22 November 2024	<ul style="list-style-type: none"> • Brandon Theodore Ferrinov (18223020) • Ratukhansa Salsabila (18223034) • Darryl Rayhananta Adenan (18223042) • Matthew Sebastian Kurniawan (18223096) 	Melakukan Rapat Kelompok
23 - 25 November 2024	<ul style="list-style-type: none"> • Brandon Theodore Ferrinov (18223020) • Ratukhansa Salsabila (18223034) • Darryl Rayhananta Adenan (18223042) • Raditya Zaki Athaya (18223086) • Matthew Sebastian Kurniawan (18223096) 	Finishing Program dan pembuatan laporan

9.4 Lampiran Link Repository

<https://github.com/BrandonTheodore/TubesAlstrukdat.git>