

# LAPORAN TUGAS BESAR


## IF2111 Algoritma dan Struktur Data

### Sistem E-commerce

Dipersiapkan oleh:

Rehuel Grace Marbun / 18218006

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

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen	Halaman
		IF2111-TB-01	31
			21 Juli 2020

# Daftar Isi

<b>Ringkasan</b>	<b>3</b>
<b>Penjelasan Tambahan Spesifikasi Tugas</b>	<b>3</b>
Program Utama	3
<b>Algoritma Menarik dan Fitur Program</b>	<b>6</b>
SIGN UP	6
LOGIN	8
ADD PRODUK	9
PRINT PRODUK	11
CART DAN UNDO	12
LIST PENGIRIMAN / CHECKOUT	18
TOP UP	18
<b>Tampilan Program</b>	<b>20</b>
Cara Compile dan Menjalankan Program	20
Tampilan Awal	20
Tampilan Fitur Sign Up	21
Tampilan Fitur Login	21
Tampilan Awal di Akun Penjual	22
Tampilan Fitur Add Produk oleh Penjual	22
Tampilan Fitur Print Produk oleh Penjual	23
Tampilan Awal di Akun Pembeli	24
Tampilan Fitur Beli oleh Pembeli	24
Tampilan Fitur Top Up oleh Pembeli	25
Tampilan Fitur Exit	27
Tampilan File Eksternal	27
<b>Lampiran</b>	<b>29</b>
Deskripsi Tugas Besar	29

# 1 Ringkasan

Program dalam tugas besar IF2111 Algoritma dan Struktur Data merupakan sistem *e-commerce* menggunakan bahasa pemrograman C. Program ini dijalankan dengan menggunakan command prompt. Secara garis besar, program ini diharapkan dapat melakukan beberapa hal yaitu *sign up*, *login*, menambahkan maupun membeli produk, *checkout* dan *top up* saldo. Pada program ini terdapat dua jenis *user* yaitu penjual dan pembeli yang akan saling bertransaksi seperti layaknya berbelanja secara online melalui *e-commerce*. Selain itu, program ini juga diharapkan memiliki data-data yang mencakup data *user*, produk dan juga transaksi.

## 2 Penjelasan Tambahan Spesifikasi Tugas

### 2.1 Program Utama

Berikut merupakan algoritma dari program utama yang dibuat.

```
void Menu(){
/* Menampilkan menu awal. */

printf("*****\n");
printf("Selamat datang!\n");
printf("Silahkan LOGIN atau DAFTAR terlebih dahulu.\n");
printf("1. Ketik LOGIN untuk login ke akun Anda.\n");
printf("2. Ketik DAFTAR jika Anda belum memiliki akun.\n");
printf("3. Ketik EXIT jika Anda ingin keluar dari program.\n");
printf("*****\n");
}

void PilihanTipeAkun(){
    printf("Mau login/daftar sebagai apa?\n");
    printf("1. Ketik PEMBELI untuk login / daftar sebagai pembeli.\n");
```

```

        printf("2. Ketik PENJUAL untuk login / daftar sebagai penjual.\n");
    }

void MenuPenjual(){
/* Menampilkan tampilan menu di penjual. */

printf("*****\n");
printf("Selamat datang di toko Anda!\n");
printf("Anda dapat melakukan hal dibawah ini.\n");
printf("1. ADD PRODUK untuk menambahkan produk ke akun Anda.\n");
printf("2. DAFTAR PRODUK untuk mengecek list produk Anda.\n");
printf("3. DAFTAR PESANAN untuk mengecek list pembayaran produk Anda.\n");
printf("4. EXIT jika Anda ingin logout dari akun Anda.\n");
printf("*****\n");
}

void MenuPembeli(){
/* Menampilkan tampilan menu di pembeli. */

printf("*****\n");
printf("Selamat datang di akun Anda!\n");
printf("Anda dapat melakukan hal dibawah ini.\n");
printf("1. ADD PRODUK untuk menambahkan produk ke keranjang Anda.\n");
printf("2. CHECKOUT untuk mengecek list produk yang akan dibayar Anda.\n");
printf("3. TOPUP jika Anda ingin topup saldo akun Anda.\n");
printf("4. EXIT jika Anda ingin logout dari akun Anda.\n");
printf("*****\n");
}

```

Program utama yang dibuat pada Tugas Besar ini berfokus pada menampilkan opsi inputan dengan membedakan dua jenis *user*. Saat program dijalankan, MenuTampilan() akan ditampilkan terlebih dahulu yang selanjutnya akan diminta input dari *user* dengan algoritma seperti berikut.

```
do {  
    printf(">>> DAFTAR / LOGIN / EXIT: ");  
    scanf("%s", &state);  
} while  
((strcmp(state,inputLOGIN)!=0)&&(strcmp(state,inputDAFTAR)!=0)&&(strcmp(state,inputEXIT)!=0));
```

Diberikan tiga opsi yaitu DAFTAR untuk melakukan sign up jika *user* belum terdaftar sebelumnya (*new user / account*). LOGIN untuk masuk ke akun *user*. Terakhir ada opsi EXIT untuk keluar dari program.

Untuk algoritma penyesuaian inputan, algoritma yang dipakai seperti berikut ini.

```
else if (strcmp(state,inputEXIT)==0){  
    printf("Terima kasih!");  
    exit = true;  
}
```

Dari cuplikan algoritma di atas, digunakan **strcmp(var1, var2)** untuk menyesuaikan inputan dengan variabel opsi yang ada. Untuk variabel opsinya sendiri, terdapat lima yang utama yaitu :

```
char inputLOGIN[] = "LOGIN";  
char inputDAFTAR[] = "DAFTAR";  
char inputEXIT[] = "EXIT";  
char inputPEMBELI[] = "PEMBELI";  
char inputPENJUAL[] = "PENJUAL";
```

### 3 Algoritma Menarik dan Fitur Program

Berikut merupakan algoritma-algoritma yang dirasa menarik pada program kali ini beserta penjelasannya.

#### 3.1 SIGN UP

Fitur Sign Up menggunakan ADT Array dan memakai konsep ADT Mesin Karakter. ADT Array digunakan untuk menyimpan inputan data dari *user* terlebih dahulu yang kemudian akan dimasukkan ke file eksternal menggunakan konsep ADT Mesin Karakter. Berikut algoritma untuk menangani fitur Sign Up.

```
void AddUser(Tab *T, char id[], char password[], char nama[], int saldo, char
tipeAkun[]){
    strcpy(T->id, id);
    strcpy(T->nama, nama);
    T->saldo = saldo;
    strcpy(T->tipeAkun, tipeAkun);
    strcpy(T->password, password);
}
```

Setelah inputan ditangani oleh array di atas, selanjutnya akan dimasukkan ke file eksternal. Berikut algoritma menangani pemasukan data array ke file eksternal.

```
void AddToListUser(Tab T){
    char id[3]; /* format 01-99*/
    char nama[5];
    char saldo[100];
    char password[9];
    char tipeAkun[2]; // tipe akun pembeli atau penjual, format T (penjual) dan B
    (pembeli)
```

```

        strcpy(id, T.id);
        strcpy(nama, T.nama);
        sprintf(saldo, "%d", T.saldo); // mengubah tipe integer saldo ke array of char.
        strcpy(password, T.password);
        strcpy(tipeAkun, T.tipeAkun);

        FILE *ListUser = fopen("ListUser.txt", "a+");
        fprintf(ListUser, "%s %s %s %s %s\n", id, password, nama, saldo, tipeAkun);
        fclose(ListUser);
    }

void AddToDataPribadi(Tab T){
    char id[3]; /* format 01-99*/
    char password[9];
    char tipeAkun[2];

    strcpy(id, T.id);
    strcpy(password, T.password);
    strcpy(tipeAkun, T.tipeAkun);

    FILE *filename = fopen("DataPribadi.txt", "a+");
    fprintf(filename, "%s %s %s\n", id, password, tipeAkun);
    fclose(filename);
}

```

Program ini memakai lima file eksternal, dua diantaranya untuk menangani *user*. File eksternal pertama yaitu **ListUser.txt**. Pada file eksternal pertama ini akan digunakan sebagai *backup* keseluruhan data *user* sehingga di dalamnya menampung inputan data berupa id, password, nama, saldo dan tipe akun. Sedangkan untuk file eksternal kedua

yaitu **DataPribadi.txt**. File ini hanya menyimpan id dan password *user*. Hal ini dilakukan untuk efisiensi dalam pengecekan *login* sehingga pada saat *user* ingin *login* ke akunnya, program tidak perlu melakukan *searching* di keseluruhan file *user* (ListUser.txt) melainkan dapat dilakukan di file DataPribadi.txt saja.

**Asumsi :** Dalam program ini id dipastikan unik sehingga tidak ada id yang sama meskipun berbeda atribut data lainnya seperti password, nama, saldo atau tipe akun.

## 3.2 LOGIN

Fitur login diharapkan mampu mendeteksi apakah inputan id dan password dari *user* sesuai dan sudah terdaftar sebelumnya pada file eksternal atau tidak. Karena hal tersebut, dibuatlah suatu fungsi yang mengembalikan boolean (true / false). Algoritma fungsi tersebut adalah sebagai berikut.

```
boolean isLoginValid(char id[], char pass[], char tipe[]){
    boolean valid;
    char id[3]; /* format 01-99*/
    char password[9];
    char tipeAkun[2];
    FILE *filename = fopen("DataPribadi.txt", "r");

    valid = false;
    while ((!feof(filename)) && (!valid)){
        fscanf(filename, "%s %s %s\n", id, password, tipeAkun);
        if((strcmp(id, id)==0) && (strcmp(pass, password)==0) &&
(strcmp(tipeAkun, tipe)==0)) {
            valid = true;
        }

        fflush(stdin);
        //printf("%s %s\n", id, password);
    }

    fclose(filename);
}
```



```
        return (valid);  
    }
```

Fungsi di atas masih merupakan bagian dari ADT Array dan memakai konsep ADT Mesin Karakter yang juga dimanfaatkan oleh fitur Sign Up. Fitur cek kevalidan ini terlebih dahulu melakukan searching untuk menemukan akhir dari data. Pada saat searching tersebut, dilakukan pembacaan file eksternal dengan menggunakan **fscanf(variabel\_file, “tipe\_data”, data\_yang\_dibaca)**. Saat dilakukan pembacaan file, data yang baru saja dibaca selanjutnya akan diproses dengan **strcmp(id\_input, id\_data)** dan juga **strcmp(pass\_input, pass\_data)** untuk memastikan apakah inputan dari *user* sesuai dengan data yang disimpan. Jika sesuai, maka akan mengembalikan nilai true yang berarti id dan password terdaftar pada file eksternal program dan *user* pun berhasil masuk ke akunnya. Jika tidak sesuai, maka akan dimasukkan ke **fflush(stdin)** untuk membersihkan output buffer dan memindahkan data buffer menuju console.

### 3.3 ADD PRODUK

Idenya sama dengan fitur Sign Up, untuk fitur menambahkan produk yang merupakan *privilege* dari penjual ini juga menggunakan ADT Array dan konsep ADT Mesin Karakter. Berikut algoritma untuk menyimpan data inputan dari *user* ke dalam array.

```
void AddProduk(Produk *P, char id_produk[], char id_penjual[], char  
nama_produk[], int harga, char kategori[]){  
    strcpy(P->id_produk, id_produk);  
    strcpy(P->id_penjual, id_penjual);  
    strcpy(P->nama_produk, nama_produk);  
    strcpy(P->kategori, kategori);  
    P->harga = harga;  
}
```

Selanjutnya akan dilakukan penyimpanan ke file eksternal ketiga yaitu

**DataProduk.txt**. Berikut algoritma untuk memasukkan array ke dalam file eksternal.

```
void AddToDataProduk(Produk P){
    char id_produk[3]; /* format P1-P9*/
    char id_penjual[3]; /* format 01-99*/
    char nama_produk[5];
    char kategori[3]; /* format K1-K9 */
    char harga[100];

    strcpy(id_produk, P.id_produk);
    strcpy(id_penjual, P.id_penjual);
    strcpy(nama_produk, P.nama_produk);
    strcpy(kategori, P.kategori);
    sprintf(harga, "%d", P.harga); // mengubah tipe integer harga ke array of char.

    FILE *filename = fopen("DataProduk.txt", "a+");
    fprintf(filename, "%s %s %s %s %s\n", id_produk, id_penjual, nama_produk,
harga, kategori);
    fclose(filename);
}
```

Dalam algoritma tersebut digunakan **strcpy(tujuan, asal)** untuk menyalin data yang ada di dalam array Produk ke dalam suatu variabel tujuan yang nantinya akan ditulis di file eksternal menggunakan **fprintf(nama\_file, "tipe\_data", data\_yang\_ingin\_disimpan)**.

### 3.4 PRINT PRODUK

Fitur Print Produk terbagi menjadi dua jenis yaitu:

1. Penjual

Print Produk pada penjual hanya menampilkan list produk yang dijual oleh penjual itu saja sehingga diperlukan identifikasi id penjual dalam penyaringan ini yang kemudian akan dibandingkan id yang diinput dengan id\_penjual yang terdaftar di file. Berikut ini adalah algoritmanya.

```
void PrintProduk(Produk P, char id[]){
    char id_produk[3]; /* format P1-P9*/
    char id_penjual[3]; /* format 01-99*/
    char nama_produk[5];
    char kategori[3]; /* format K1-K9 */
    char harga[100];

    FILE *filename = fopen("DataProduk.txt","a+");
    printf("id_produk id_penjual nama_produk harga kategori\n");
    while (!feof(filename)){
        fscanf(filename, "%s %s %s %s %s\n", id_produk, id_penjual, nama_produk,
harga, kategori);
        if(strcmp(id, id_penjual)==0){
            fflush(stdin);
            printf("%s %s %s %s %s\n", id_produk, id_penjual, nama_produk, harga,
kategori); } else { fflush(stdin); }
        }
    }
}
```

## 2. Pembeli

Berbeda dengan fitur Print Produk penjual, di pembeli fitur ini akan menampilkan semua produk yang ada di *e-commerce*. Berikut ini adalah algoritmanya.

```
void PrintAllProduk(Produk P){
    char id_produk[3]; /* format P1-P9*/
    char id_penjual[3]; /* format 01-99*/
    char nama_produk[5];
    char kategori[3]; /* format K1-K9 */
    char harga[100];

    FILE *filename = fopen("DataProduk.txt", "a+");
    printf("id_produk id_penjual nama_produk harga kategori\n");
    while (!feof(filename)){
        fscanf(filename, "%s %s %s %s %s\n", id_produk, id_penjual,
nama_produk, harga, kategori);
        fflush(stdin);
        printf("%s %s %s %s %s\n", id_produk, id_penjual, nama_produk,
harga, kategori);
    }
}
```

## 3.5 CART DAN UNDO

Fitur Cart dan Undo menggunakan ADT Stacklist yang juga digabungkan dengan konsep ADT Mesin Karakter. Fungsi dari fitur Cart sendiri yaitu untuk menampung pesanan dari pembeli sedangkan untuk fitur Undo digunakan untuk menangani apabila pembeli ingin membatalkan suatu pesanan, terlepas dari apakah pesanan itu sudah dibayar atau belum. Berikut ini adalah algoritmanya.

```

void Push (Stack * S, infotype X) {
    address P;
    AlokasiC(&P,X);
    if (P != Nil) {
        Next(P) = Top(*S);
        Top(*S) = P;  }
}
void Pop (Stack * S, infotype * X) {
    address P;

    P = Top(*S);
    Top(*S) = Next(Top(*S));
    Next(P) = Nil;
    (*X) = Info(P);
    DealokasiC(P);
}

```

Prosedur Push akan bertindak untuk menampung dan memasukkan pesan ke dalam suatu stack. Hal ini bertujuan agar fitur UNDO yang ditangani oleh prosedur Pop dapat dilakukan. Sebelum stack tersebut diberi inputan SIMPAN, maka stack itu tidak akan dituliskan ke dalam file eksternal yaitu **Pesanan.txt**. Berikut ini algoritma penyimpanan stack ke dalam file eksternal.

```

void AddPesanan(Stack S){
    address P;

    FILE *pesanan = fopen("Pesanan.txt","a+");
    P = Top(S);
    while (P != Nil){
        fprintf(pesanan, "%s %s %s %d %d %d %s\n", Info(P).id_produk,

```

```

Info(P).id_pembeli, Info(P).id_penjual, Info(P).harga, Info(P).jumlah_barang,
Info(P).total, Info(P).status);
    P = Next(P);
}
fclose(pesanan);
}

```

Sedangkan logaritma dari fitur untuk pembayaran maupun pembatalan transaksi adalah sebagai berikut.

```

void Transaksi(char id[], int transaksi){
    Tab T;
    char saldo[100];
    char nama[5];
    char password[9];
    char tipeAkun[2];
    char id[3];

    FILE *in=fopen("ListUser.txt","r+");
    FILE *temp=fopen("TempData.txt","a+");
    while(!feof(in)){
        fscanf(in,"%s %s %s %s %s\n", id, password, nama, saldo, tipeAkun);
        if (strcmp(id,id)!=0){
            fprintf(temp, "%s %s %s %s %s\n", id, password, nama, saldo,
tipeAkun); }
        else {
            T.saldo = atoi(saldo);
            if (T.saldo < transaksi){
                printf("Saldo tidak mencukupi, silahkan TOP UP terlebih dahulu\n");
                printf("ketik TUNDA untuk menunda pembayaran\n");
            }
        }
    }
}

```

```

        sprintf(saldo, "%d", T.saldo);
        fprintf(temp, "%s %s %s %s %s\n", id, password, nama, saldo,
tipeAkun);    }
        else if (T.saldo >= transaksi){
            T.saldo = T.saldo - transaksi;
            strcpy(T.id, id);
            strcpy(T.password, password);
            strcpy(T.nama, nama);
            strcpy(T.tipeAkun, tipeAkun);
            sprintf(saldo, "%d", T.saldo);
            fprintf(temp, "%s %s %s %s %s\n", T.id, T.password, T.nama,
saldo, T.tipeAkun);

            printf("PEMBAYARAN BERHASIL\n");
            printf("ketik BAYAR untuk menyimpan status pembayaran\n");
        }
    }
}

fclose(in);
fclose(temp);
remove("ListUser.txt");
rename("TempData.txt", "ListUser.txt");
}

void TransaksiBatal(char id[], int transaksi){
    Tab T;
    char saldo[100];
    char nama[5];
    char password[9];
    char tipeAkun[2];
    char id[3];

```

```

FILE *in=fopen("ListUser.txt","r+");
FILE *temp=fopen("TempData.txt","a+");
while(!feof(in)){
    fscanf(in,"%s %s %s %s %s\n", id, password, nama, saldo, tipeAkun);
    if (strcmp(idd,id)!=0){
        fprintf(temp, "%s %s %s %s %s\n", id, password, nama, saldo,
tipeAkun);
    } else {
        T.saldo = atoi(saldo);
        T.saldo = T.saldo + transaksi;
        strcpy(T.id, id);
        strcpy(T.password, password);
        strcpy(T.nama, nama);
        strcpy(T.tipeAkun, tipeAkun);
        sprintf(saldo, "%d", T.saldo);
        fprintf(temp, "%s %s %s %s %s\n", T.id, T.password, T.nama, saldo,
T.tipeAkun); }
    }
    fclose(in);
    fclose(temp);
    remove("ListUser.txt");
    rename("TempData.txt","ListUser.txt");
}

```

Berikut cuplikan kode dari penggunaan fitur Cart dan UNDO yang berada di program utama.

```

do{
    printf("Tambahkan ke keranjang / Hapus barang / Simpan pesanan?

```



(TAMBAH / HAPUS / SIMPAN): "); //JIKA MILIH HAPUS, ASUMSI LISTCART  
SUDAH TERISI

```
scanf("%s", &state);
if (strcmp(state,"TAMBAH")==0) {
    printf("Input ID Produk yang ingin ditambahkan ke keranjang: ");
    scanf("%s", &X.id_produk);
    printf("Input ID penjual: ");
    scanf("%s", &X.id_penjual);
    printf("Harga per satuan: ");
    scanf("%d", &X.harga);
    printf("Jumlah yang ingin dibeli: ");
    scanf("%d", &X.jumlah_barang);
    X.total = (X.harga * X.jumlah_barang);
    printf("Total Pesanan: %d\n", X.total);
    printf("Input ID Anda: ");
    scanf("%s", &X.id_pembeli);
    do{
        printf("Ketik BAYAR untuk melanjutkan: ");
        scanf("%s", &state);
    } while (strcmp(state, "BAYAR")!=0);
    if (strcmp(state, "BAYAR")==0){
        Transaksi(id, X.total);
        scanf("%s", &X.status);
    }
    Push(&C, X);
    printf("Belanjaan berhasil di tambahkan.\n");
} else if (strcmp(state,"HAPUS")==0){ //HAPUS ATAU UNDO BISA
JUGA MEMBATALKAN PESANAN
    Pop(&C,&X);
    printf("Input ID Anda: ");
```

```

scanf("%s", &id);
printf("Total transaksi: ");
scanf("%d", &nominal);
TransaksiBatal(id, nominal);
printf("Belanjaan berhasil di hapus.\n");
} else if (strcmp(state,"SIMPAN")==0){
    AddPesanan(C);
    printf("Daftar belanjaan berhasil di simpan. Silahkan tunggu
pengiriman apabila sudah membayar\n");
}
} while
((strcmp(state,"TAMBAH")!=0)&&(strcmp(state,"HAPUS")!=0)&&(strcmp(state,"S
IMPAN")!=0));

```

### 3.6 LIST PENGIRIMAN / CHECKOUT

Pada fitur ini diharapkan setelah pembeli melakukan transaksi, transaksi tersebut akan masuk ke penjual. Pada fitur *role* penjual / toko, akan ditampilkan list pemesanan produk yang dia jual. List tersebut diharapkan tersusun berdasarkan prioritas yaitu pesanan yang sudah dibayar dan masuk ke penjual terlebih dahulu berhak untuk didahulukan, sedangkan pesanan yang lainnya akan dipersiapkan oleh penjual selagi menunggu pembeli melakukan pembayaran.

### 3.7 TOP UP

Fitur Top Up merupakan fitur yang menangani kasus jika pembeli ingin melakukan penambahan saldo (top up). Fitur ini menerapkan prinsip ADT Array dan Mesin Karakter hal ini dikarenakan file eksternal yang dibaca (load) lebih mudah dilakukan dengan konsep ADT Mesin Karakter dan untuk menampung valuenya lebih memungkinkan jika memakai ADT Array yang diimplementasikan dalam header user. Dalam fitur ini, digunakan file eksternal sementara untuk menampung perubahan saldo pembeli yaitu **TempData.txt**. File ListUser.txt akan di *remove* sesudah memindahkan seluruh *tuple* ke TempData.txt tadi kecuali *tuple* yang ingin diperbaharui karena harus

terlebih dahulu diproses. Setelah itu, file TempData.txt akan di *rename* menjadi ListUser.txt. Berikut algoritma dari fitur ini.

```
void TopUp (char idd[], int transaksi){
    Tab T;
    char saldo[100];
    char nama[5];
    char password[9];
    char tipeAkun[2];
    char id[3];

    FILE *in=fopen("ListUser.txt","r+");
    FILE *temp=fopen("TempData.txt","a+");
    while(!feof(in)){
        fscanf(in,"%s %s %s %s %s\n", id, password, nama, saldo, tipeAkun);
        if (strcmp(idd,id)!=0){
            fprintf(temp, "%s %s %s %s %s\n", id, password, nama, saldo,
tipeAkun);
        } else {
            strcpy(T.id, id);
            strcpy(T.password, password);
            strcpy(T.nama, nama);
            strcpy(T.tipeAkun, tipeAkun);
            T.saldo = atoi(saldo);
            T.saldo = T.saldo + transaksi;
            sprintf(saldo, "%d", T.saldo);
            fprintf(temp, "%s %s %s %s %s\n", T.id, T.password, T.nama, saldo,
T.tipeAkun);
            printf("TOP UP SUCCESSFULLY\n");
        }
    }
}
```

```
fclose(in);  
fclose(temp);  
remove("ListUser.txt");  
rename("TempData.txt", "ListUser.txt");  
}
```

## 4 Tampilan Program

### 4.1 Cara Compile dan Menjalankan Program

Berikut cara compile program Tugas Besar ini di command prompt.

```
gcc -o driverTubes driverTubes.c cart.c produk.c user.c
```

Berikut cara menjalankan program Tugas Besar ini (apabila di Windows).

```
driverTubes
```

### 4.2 Tampilan Awal

Berikut tampilan awal program.

```
*****  
Selamat datang!  
Silahkan LOGIN atau DAFTAR terlebih dahulu.  
1. Ketik LOGIN untuk login ke akun Anda.  
2. Ketik DAFTAR jika Anda belum memiliki akun.  
3. Ketik EXIT jika Anda ingin keluar dari program.  
*****  
>>> DAFTAR / LOGIN / EXIT:
```

Pada gambar di atas, terdapat tiga pilihan input. *User* dapat menginput sesuai format yang diinstruksikan. Jika tidak sesuai, akan terjadi seperti berikut.

```
>>> DAFTAR / LOGIN / EXIT: daftar  
>>> DAFTAR / LOGIN / EXIT:
```

**\* Hal ini berlaku untuk semua instruksi pilihan.**

### 4.3 Tampilan Fitur Sign Up

Berikut tampilan fitur Sign Up untuk penjual.

```
>>> DAFTAR / LOGIN / EXIT: DAFTAR
Masukkan id (2 digit angka, format 01-99): 06
Masukkan password (8 karakter): PENJUAL1
Masukkan nama (max 4 karakter): TOK1
Masukkan saldo awal: 10000
Mendaftar sebagai apa? P (Pembeli) / T (Penjual): T
Pendaftaran akun sukses.
>>> DAFTAR / LOGIN / EXIT: _
```

Berikut tampilan fitur Sign Up untuk pembeli.

```
*****
Selamat datang!
Silahkan LOGIN atau DAFTAR terlebih dahulu.
1. Ketik LOGIN untuk login ke akun Anda.
2. Ketik DAFTAR jika Anda belum memiliki akun.
3. Ketik EXIT jika Anda ingin keluar dari program.
*****
>>> DAFTAR / LOGIN / EXIT: DAFTAR
Masukkan id (2 digit angka, format 01-99): 01
Masukkan password (8 karakter): PEMBELI1
Masukkan nama (max 4 karakter): RERE
Masukkan saldo awal: 100000
Mendaftar sebagai apa? P (Pembeli) / T (Penjual): P
Pendaftaran akun sukses.
>>> DAFTAR / LOGIN / EXIT: _
```

### 4.4 Tampilan Fitur Login

Berikut tampilan fitur Login.

```
>>> DAFTAR / LOGIN / EXIT: LOGIN
Mau login/daftar sebagai apa?
1. Ketik PEMBELI untuk login / daftar sebagai pembeli.
2. Ketik PENJUAL untuk login / daftar sebagai penjual.
>>> PEMBELI / PENJUAL: _
```

Setelah memilih *role login* dan ternyata *user* salah menginput data maka akan tampil seperti gambar di bawah ini.

```

Mau login/daftar sebagai apa?
1. Ketik PEMBELI untuk login / daftar sebagai pembeli.
2. Ketik PENJUAL untuk login / daftar sebagai penjual.
>>> PEMBELI / PENJUAL: PENJUAL
id: 07
password: PENJUAL3
Cek ulang id dan password Anda.
>>> DAFTAR / LOGIN / EXIT: _

```

\* *User dapat mengulang proses login.*

## 4.5 Tampilan Awal di Akun Penjual

Setelah proses login dan memilih role login sebagai penjual, maka akan tampil menu awal seperti di bawah ini.

```

>>> DAFTAR / LOGIN / EXIT: LOGIN
Mau login/daftar sebagai apa?
1. Ketik PEMBELI untuk login / daftar sebagai pembeli.
2. Ketik PENJUAL untuk login / daftar sebagai penjual.
>>> PEMBELI / PENJUAL: PENJUAL
id: 06
password: PENJUAL1
Login sebagai penjual berhasil.
*****
Selamat datang di toko Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke akun Anda.
2. DAFTAR PRODUK untuk mengecek list produk Anda.
3. DAFTAR PESANAN untuk mengecek list pembayaran produk Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4 :

```

## 4.6 Tampilan Fitur Add Produk oleh Penjual

Penjual dapat menambahkan produk sesuai dengan instruksi yang diberikan.

Berikut ini merupakan tampilan fitur Add Produk yang dapat dilakukan oleh penjual jika menginput “1”.

```

>>> INPUT 1 / 2 / 3 / 4 : 1
Masukkan id produk (2 digit, format P1-P9): P1
Masukkan id penjual (2 digit angka, format 01-99): 06
Masukkan nama produk (max 4 karakter): GULA
Masukkan kategori produk (2 digit, format K1-K9): K1
Masukkan harga produk: 10000
Produk berhasil ditambahkan.

```

## 4.7 Tampilan Fitur Print Produk oleh Penjual

Print Produk hanya akan menampilkan produk yang dijual oleh penjual itu saja, seperti tampilan di atas. Berikut ini tampilan fitur Print Produk di akun penjual.

```
*****
>>> DAFTAR / LOGIN / EXIT: LOGIN
Mau login/daftar sebagai apa?
1. Ketik PEMBELI untuk login / daftar sebagai pembeli.
2. Ketik PENJUAL untuk login / daftar sebagai penjual.
>>> PEMBELI / PENJUAL: PENJUAL
id: 06
password: PENJUAL1
Login sebagai penjual berhasil.
*****
Selamat datang di toko Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke akun Anda.
2. DAFTAR PRODUK untuk mengecek list produk Anda.
3. DAFTAR PESANAN untuk mengecek list pembayaran produk Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4 : 2
id_produk id_penjual nama_produk harga kategori
P1 06 GULA 10000 K1
P2 06 KACA 20000 K2
P3 06 PAKU 900 K3
P4 06 TEH 10000 K4
P5 06 BAJU 50000 K5
>>> DAFTAR / LOGIN / EXIT: LOGIN
```

```
*****
Selamat datang di toko Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke akun Anda.
2. DAFTAR PRODUK untuk mengecek list produk Anda.
3. DAFTAR PESANAN untuk mengecek list pembayaran produk Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4 : 2
id_produk id_penjual nama_produk harga kategori
P1 07 TEH 9900 K1
P2 07 LAMP 20000 K2
P3 07 500000 K10
P4 07 HP 1980000 K2
P5 07 JAM 80000 K5
>>> DAFTAR / LOGIN / EXIT: LOGIN
```

```
*****
Selamat datang di toko Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke akun Anda.
2. DAFTAR PRODUK untuk mengecek list produk Anda.
3. DAFTAR PESANAN untuk mengecek list pembayaran produk Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4 : 2
id_produk id_penjual nama_produk harga kategori
P1 08 BAJU 100000 K1
P2 08 KAYU 10000 K2
P3 08 KAYU 10000 K2
P4 08 LOGO 1000 K3
P5 08 HP 2000000 K2
>>> DAFTAR / LOGIN / EXIT: LOGIN
```



## 4.8 Tampilan Awal di Akun Pembeli

Setelah proses login dan memilih role login sebagai pembeli, maka akan tampil menu awal seperti di bawah ini.

```
>>> PEMBELI / PENJUAL: PEMBELI
id: 01
password: PEMBELI1
Login sebagai pembeli berhasil.
*****
Selamat datang di akun Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke keranjang Anda.
2. CHECKOUT untuk mengecek list produk yang akan dibayar Anda.
3. TOPUP jika Anda ingin topup saldo akun Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4: 1
```

## 4.9 Tampilan Fitur Beli oleh Pembeli

Fitur Beli oleh pembeli dapat diakses dengan menginput “1” pada perintah >>> INPUT 1 / 2 / 3 / 4. Fitur ini akan menampilkan semua produk yang dijual pada *e-commerce* ini.

```
>>> PEMBELI / PENJUAL: PEMBELI
id: 01
password: PEMBELI1
Login sebagai pembeli berhasil.
*****
Selamat datang di akun Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke keranjang Anda.
2. CHECKOUT untuk mengecek list produk yang akan dibayar Anda.
3. TOPUP jika Anda ingin topup saldo akun Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4: 1
id_produk id_penjual nama_produk harga kategori
P1 06 GULA 10000 K1
P2 06 KACA 20000 K2
P3 06 PAKU 900 K3
P4 06 TEH 10000 K4
P5 06 BAJU 50000 K5
P1 07 TEH 9900 K1
P1 08 BAJU 100000 K1
P2 08 KAYU 10000 K2
Tambahkan ke keranjang / Hapus barang / Simpan pesanan? (TAMBAH / HAPUS / SIMPAN):
```

Kemudian pembeli dapat menginput TAMBAH jika ingin membeli produk, HAPUS jika ingin membatalkan produk / pesanan yang telah dipilih sebelumnya, maupun SIMPAN jika merasa sudah yakin dengan pesannya. Pembeli dilarang untuk menginput “SIMPAN” jika masih mau mengubah pesannya.



```

*****
>>> INPUT 1 / 2 / 3 / 4: 1
id_produk id_penjual nama_produk harga kategori
P1 06 GULA 10000 K1
P2 06 KACA 20000 K2
P3 06 PAKU 900 K3
P4 06 TEH 10000 K4
P5 06 BAJU 50000 K5
P1 07 TEH 9900 K1
P1 08 BAJU 100000 K1
P2 08 KAYU 10000 K2
Tambahkan ke keranjang / Hapus barang / Simpan pesanan? (TAMBAH / HAPUS / SIMPAN): TAMBAH
Input ID Produk yang ingin ditambahkan ke keranjang: P3
Input ID penjual: 06
Harga per satuan: 900
Jumlah yang ingin dibeli: 1
Total Pesanan: 900
Input ID Anda: 01
-----
id_produk id_pembeli id_penjual harga jumlah_barang total
P3 01 06 900 1 900
-----
Ketik BAYAR untuk melanjutkan: BAYAR
PEMBAYARAN BERHASIL
ketik BAYAR untuk menyimpan status pembayaran
BAYAR
Belanjaan berhasil di tambahkan.

Belanjaan berhasil di tambahkan.
Tambahkan ke keranjang / Hapus barang / Simpan pesanan? (TAMBAH / HAPUS / SIMPAN): HAPUS
Input ID Anda: 01
Total transaksi: 900
Belanjaan berhasil di hapus.

Tambahkan ke keranjang / Hapus barang / Simpan pesanan? (TAMBAH / HAPUS / SIMPAN): SIMPAN
Daftar belanjaan berhasil di simpan. Silahkan tunggu pengiriman apabila sudah membayar

```

Perlu diketahui, saat memilih TAMBAH dan lalu membayarnya akan terjadi pembaharuan data saldo di file ListUser.txt jika transaksi berhasil (kategori berhasil yaitu saldo mencukupi atau terjadi pembatalan pesanan atas permintaan pembeli).

#### 4.10 Tampilan Fitur Top Up oleh Pembeli

Pada fitur ini, pembeli diwajibkan untuk memasukkan id dan nominal *top up*. Berikut tampilan dari fitur Top Up.

```

*****
Selamat datang!
Silahkan LOGIN atau DAFTAR terlebih dahulu.
1. Ketik LOGIN untuk login ke akun Anda.
2. Ketik DAFTAR jika Anda belum memiliki akun.
3. Ketik EXIT jika Anda ingin keluar dari program.
*****
>>> DAFTAR / LOGIN / EXIT: LOGIN
Mau login/daftar sebagai apa?
1. Ketik PEMBELI untuk login / daftar sebagai pembeli.
2. Ketik PENJUAL untuk login / daftar sebagai penjual.
>>> PEMBELI / PENJUAL: PEMBELI
id: 01
password: PEMBELI1
Login sebagai pembeli berhasil.
*****
Selamat datang di akun Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke keranjang Anda.
2. CHECKOUT untuk mengecek list produk yang akan dibayar Anda.
3. TOPUP jika Anda ingin topup saldo akun Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4: 3
ID Anda: 01
Nominal: 500
TOP UP SUCCESSFULLY

```

Berikut tampilan file eksternal ListUser.txt sebelum pembeli melakukan *top up*.

```

ListUser - Notepad
File Edit Format View Help
01 PEMBELI1 RERE 100000 P
02 PEMBELI2 REY 10000 P
03 PEMBELI3 ANNA 50000 P
04 PEMBELI4 ROY 120000 P
05 PEMBELI5 NINO 30000 P
06 PENJUAL1 TOK1 10000 T
07 PENJUAL2 TOK2 0 T
08 PENJUAL3 TOK3 150000 T

```

Di bawah ini adalah tampilan file eksternal ListUser.txt setelah user melakukan *top up*.

```

ListUser - Notepad
File Edit Format View Help
01 PEMBELI1 RERE 100500 P
02 PEMBELI2 REY 10000 P
03 PEMBELI3 ANNA 50000 P
04 PEMBELI4 ROY 120000 P
05 PEMBELI5 NINO 30000 P
06 PENJUAL1 TOK1 10000 T
07 PENJUAL2 TOK2 0 T
08 PENJUAL3 TOK3 150000 T

```

Bisa dilihat dari kedua tampilan file eksternal di atas, pembeli ber-ID 1 berhasil melakukan top up sebesar 500.

#### 4.11 Tampilan Fitur Exit

Berikut ini adalah tampilan fitur Exit jika *user* menginput “EXIT” di pilihan awal.

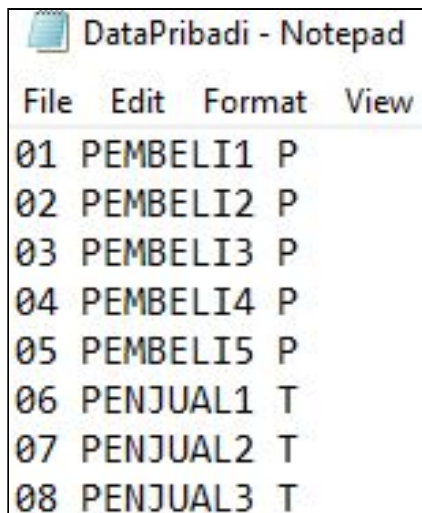
```
>>> DAFTAR / LOGIN / EXIT: EXIT
Terima kasih!
```

Berikut ini adalah tampilan fitur Exit jika *user* menginput “4” saat sudah menjadi salah satu *role login* (penjual / pembeli).

```
Login sebagai penjual berhasil.
*****
Selamat datang di toko Anda!
Anda dapat melakukan hal dibawah ini.
1. ADD PRODUK untuk menambahkan produk ke akun Anda.
2. DAFTAR PRODUK untuk mengecek list produk Anda.
3. DAFTAR PESANAN untuk mengecek list pembayaran produk Anda.
4. EXIT jika Anda ingin logout dari akun Anda.
*****
>>> INPUT 1 / 2 / 3 / 4 : 4
Terima kasih!
```

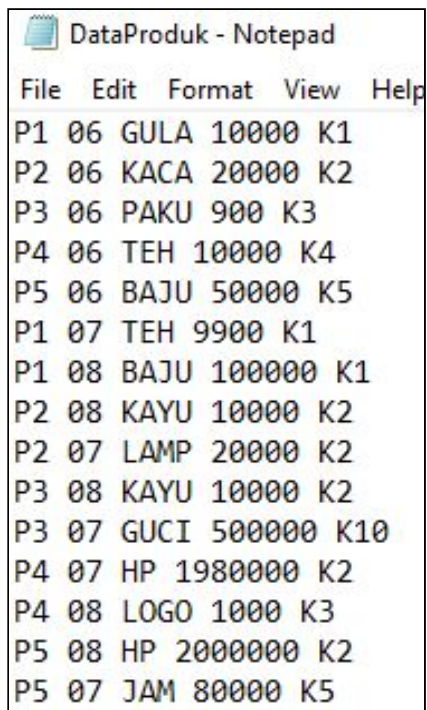
#### 4.12 Tampilan File Eksternal

Berikut file eksternal DataPribadi.txt dari *user*.



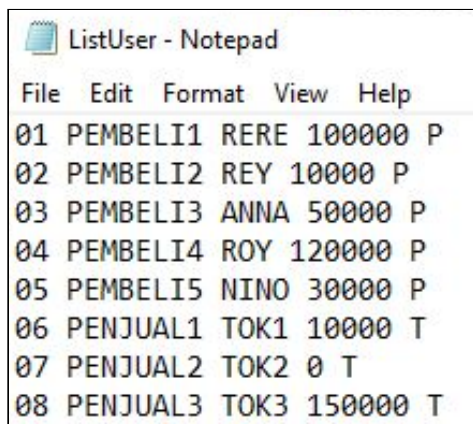
	File	Edit	Format	View
01	PEMBELI1	P		
02	PEMBELI2	P		
03	PEMBELI3	P		
04	PEMBELI4	P		
05	PEMBELI5	P		
06	PENJUAL1	T		
07	PENJUAL2	T		
08	PENJUAL3	T		

Berikut file eksternal DataProduk.txt.



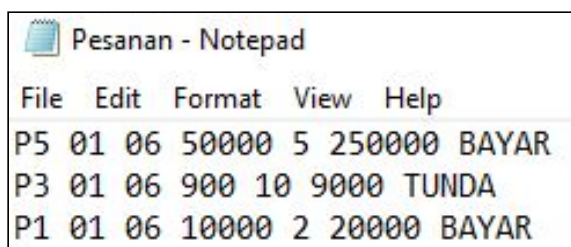
```
DataProduk - Notepad
File Edit Format View Help
P1 06 GULA 10000 K1
P2 06 KACA 20000 K2
P3 06 PAKU 900 K3
P4 06 TEH 10000 K4
P5 06 BAJU 50000 K5
P1 07 TEH 9900 K1
P1 08 BAJU 100000 K1
P2 08 KAYU 10000 K2
P2 07 LAMP 20000 K2
P3 08 KAYU 10000 K2
P3 07 GUCI 500000 K10
P4 07 HP 1980000 K2
P4 08 LOGO 1000 K3
P5 08 HP 2000000 K2
P5 07 JAM 80000 K5
```

Berikut file eksternal ListUser.txt.



```
ListUser - Notepad
File Edit Format View Help
01 PEMBELI1 RERE 100000 P
02 PEMBELI2 REY 10000 P
03 PEMBELI3 ANNA 50000 P
04 PEMBELI4 ROY 120000 P
05 PEMBELI5 NINO 30000 P
06 PENJUAL1 TOK1 10000 T
07 PENJUAL2 TOK2 0 T
08 PENJUAL3 TOK3 150000 T
```

Berikut file eksternal Pesanan.txt



```
Pesanan - Notepad
File Edit Format View Help
P5 01 06 50000 5 250000 BAYAR
P3 01 06 900 10 9000 TUNDA
P1 01 06 10000 2 20000 BAYAR
```

## 5 Lampiran

### 5.1 Deskripsi Tugas Besar

Pada Tugas Besar IF2111 Algoritma dan Struktur Data ini akan dibuat suatu sistem *e-commerce*. Program ini memakai *command line* sebagai media untuk menjalankannya. Sistem *E-Commerce* ini memiliki spesifikasi sebagai berikut.

1. User

Pada program ini, akan digunakan dua jenis *user* yaitu penjual dan pembeli. Kedua *user* ini dimungkinkan untuk melakukan transaksi jual-beli seperti layaknya belanja online di *e-commerce* pada umumnya.

2. File Eksternal

Program ini memakai beberapa file eksternal dengan berformat “xx.txt”. Pemakaian file eksternal ini sangat penting bagi keberjalanan sistem *e-commerce* karena jika tidak memakainya, data yang telah diinput akan hilang begitu program selesai. Pada spesifikasi Tugas Besar yang diberikan, file eksternal setidaknya harus menyimpan :

1. Data User (id, nama, saldo, jenis\_akun); jenis\_akun terbagi menjadi dua kategori yaitu pembeli dan penjual.
2. Data Produk (id\_produk, id\_penjual, nama, kategori, harga).

3. Sign Up dan Login

Sebelum *login*, *user* diwajibkan untuk sign up jika belum pernah terdaftar. Data yang dimasukkan pada proses *sign up* akan tersimpan pada file eksternal. Dari file eksternal, sistem akan melakukan pengecekan terhadap inputan *login* dari *user* nantinya. *User* akan dapat melakukan aktivitas selanjutnya jika dinyatakan valid saat pengecekan id dan password yang diinput *user*.

4. Menambahkan Produk oleh Penjual

Penjual yang telah dinyatakan valid pada proses pengecekan *login* akan mendapatkan beberapa *privilege* diantaranya yaitu dapat menambahkan produk di akunnya.

Informasi produk yang disimpan yaitu id produk, id penjual, nama produk, kategori,

harga. Informasi-informasi tersebut akan disimpan juga di dalam file eksternal nantinya.

5. Menampilkan List Produk

Spesifikasi ini dapat terpenuhi jika *user* telah dinyatakan valid pada saat proses pengecekan kevalidan login. Secara garis besar, poin ini akan menampilkan list produk yang ada di file eksternal ke command prompt.

6. Tambah Produk ke Cart dan Undo

Seperti *e-commerce* pada umumnya, pada program ini juga diharapkan pembeli dapat menambahkan / membeli produk yang ada di program *e-commerce*. Pembeli juga dapat membatalkan input pesanan yang belum dibayar maupun yang sudah dibayar dengan memanfaatkan fitur UNDO.

7. List Pembayaran

List pembayaran disini maksudnya adalah data yang hanya dapat dilihat oleh penjual dengan tujuan mengecek pesanan yang sudah dibayar dan yang masih ditunda pembayarannya. Pada poin ini juga akan diproses pengirimannya jika pesanan sudah dibayar oleh pembeli.

8. Top Up

Pada poin ini, pembeli maupun penjual dapat melakukan pengisian saldo. Saldo awal yang sudah tersimpan di file eksternal juga akan otomatis tergantikan oleh saldo sesuai top up.

Selain itu, proses bisnis dari sistem *e-commerce* ini antara lain sebagai berikut.

1. Setiap penjual dapat menambahkan data produk yang ingin di jual di *e-commerce*.
2. Pembeli dapat membeli produk dari penjual manapun dengan cara memilih produk lalu menambahkannya ke cart / keranjang. Pembeli juga dapat mengurangi produk yang ada di cart dengan membatalkan menggunakan UNDO.
3. Setelah mengisi cart, pembeli dapat melihat tampilan checkout (berisi jumlah barang, total harga serta data terkait seperti id penjual, id produk dan id pembeli).
4. Setelah itu, pembeli dapat langsung membayar maupun menunda pembayarannya. Pembayaran akan berhasil bila pembeli memiliki saldo yang cukup untuk melakukan suatu transaksi pembelian.

5. List pesanan ini, baik yang sudah dibayar maupun yang belum dibayar akan masuk ke penjual dan diurutkan serta ditampilkan berdasarkan prioritas yaitu waktu input pesanan dan status pesanan. Pesanan yang lebih dahulu masuk dan dengan status yang sudah dibayar akan diprioritaskan pengirimannya.
6. Ada 2 klasifikasi status pesanan yaitu jika belum melakukan pembayaran akan berstatus “menunggu pembayaran”. Jika sudah melakukan pembayaran akan berstatus “dikirim”.
7. Pembeli dapat melakukan top up untuk menambahkan saldo di rekening akunnya.
8. Apabila saldo tidak mencukupi untuk melakukan pembayaran, pesanan akan masuk ke berstatus “menunggu pembayaran” dan dapat melakukan pembayaran di lain waktu.

Adapun spesifikasi akan ADT yang harus dipenuhi dalam pengerjaan ini yaitu :

1. ADT Stack dalam list linear
2. ADT Queue dalam list linear
3. ADT lainnya yang diperlukan