MODUL 8 PROSEDUR 2

PRAKTIKUM DASAR PEMROGRAMAN UNIVERSITAS ATMA JAYA YOGYAKARTA

TUJUAN

- 1. Memahami konsep parameter secara lebih mendalam.
- 2. Memahami konsep parameter input, output, dan input/output.
- 3. Memahami konsep passing parameter, by value dan by reference.

Note: Jangan lupa pelajari kembali materi Prosedur 1.

TEORI

Prosedur adalah modul program yang mengerjakan tugas atau aktivitas yang spesifik dan menghasilkan suatu efek netto.

Berdasarkan maksud penggunaanya, terdapat 3 jenis parameter, yaitu :

1. Parameter Input

adalah parameter dimana isi (*value*)nya sudah jelas terdefinisi ketika dilakukan pemanggilan prosedur. Isi (*value*) dari parameter ini digunakan sebagai masukan untuk prosedur. Penggunaan parameter ini sering disebut dengan passing parameter *by value*. Isi (*value*) dari parameter ini harus dijamin tidak mengalami perubahan selama pemrosesan prosedur (tidak akan berubah baik saat memasukan kedalam prosedur maupun mengeluarkan hasil keluar dari prosedur). Dalam suatu assignment di dalam body prosedur, parameter ini biasanya terletak disebelah **kanan** 'sama dengan'. Perhatikan contoh berikut (*Gambar 1*);

```
#include <stdio.h>
                                              Parameter Input
   2
       #include <stdlib.h>
   3
   4
       void LuasPersegiPanjang(int *Luas, int p, int l
                                                             Pendeklarasian Prosedur
   5
   6 	☐ int main(int argc, char *argv[]) {
           int L:
Pemanggilan
                                              → Inisialisasi
           int panjang = 3, lebar = 5;
   Prosedur
  10
                                                    Isi (value)
           LuasPersegiPanjang(&L,
  11
  12
  13
           LuasPersegiPanjang(&L, panjang,
  14
  15
           return 0;
                                              Parameter Input
  16
  17
  19
                                                            Pendefinisian Prosedur
  20
  21
                              Terletak disebelah kanan '='
  22
                                                                              1
```

2. Parameter Output

adalah parameter target yang hendak diubah atau parameter yang menampung hasil keluaran dari prosedur. Parameter ini mengalami perubahan dan efek netto dari prosedur (menjadi tujuan pengubahan). Parameter ini memiliki isi (value) yang sudah jelas ataupun belum jelas, yang pasti parameter ini akan diubah di dalam prosedur. Dalam suatu assignment di dalam body prosedur, parameter ini tertelak disebelah **kiri** 'sama dengan'. Perhatikan contoh berikut (*Gambar 2*);

```
#include <stdio.h>
                                     Parameter Output
     2
         #include <stdlib.h>
     3
     4
         void LuasPersegiPanjang(int *Luas
                                               int p, int 1); → Pendeklarasian Prosedur
     5
     6 	☐ int main(int argc, char *argv[]) {
              int L:
Pemanggilan
                                        Inisialisasi
              int luas = 50; -
   Prosedur
    10
              LuasPersegiPanjang(&L,
                                       3, 5);
              printf("Luas = %d", L);
    11
                                       //hasil L ya akan tercetak = 15
    12
                                                  Isi (value)
    13
              LuasPersegiPanjang(&luas,
              printf("Luas = %d", luas); //hasil luas yg akan tercetak = 15
    14
    15
    16
              return 0;
                                      Parameter Output
    17
    18
    19 -
         void LuasPersegiPanjang(int *Luas
                                                int p, int 1){-
    20
    21
                                       → Assignment
                                                                  Pendefinisian Prosedur
    22
                        Terletak disebelah kiri '='
```

3. Parameter Input/Output

Adalah parameter output yang juga sekaligus menjadi parameter input (sebagai masukan sekaligus keluaran bagi prosedur tersebut). Parameter ini pastilah sudah jelas isinya ketika memasuki prosedur, karena dia juga berperan sebagai parameter input. Penggunaan parameter ini sering disebut dengan passing parameter *by reference*. Isi (value) dari parameter aktual (ketika prosedur dipanggil) akan diisikan ke parameter formal (di prosedur) yang urutannya bersesuaian dan hasil dari pengolahan prosedur akan dikembalikan ke parameter aktual. Perhatikan contoh berikut (*Gambar 3*);

```
1
         #include <stdio.h>
     2
         #include <stdlib.h>
                                              Parameter Formal
     3
         void tukar(int *n1, int *n2);
     4
                                              Pendeklarasian Prosedur
     5
     6 ☐ int main(int argc, char *argv[]){
                                                       → Inisialisasi
     7
              int nilai1 = 5, nilai2 = 10; -
     8
                                             Parameter Aktual
              tukar(&nilai1
                               &nilai2)
     9
                                  setelah ditukar : %d", nilai1);
   Pemanggilan printf("
                       \nN<mark>ilai 2 setelah ditukar : %d", nilai2);</mark>
      Prosedur
              //Silakan cek hasilnya sendiri..
    13
5
                                                                                   6
    14
              return 0;
    15
                                              Parameter Formal
    16
    void tukar(int *n1, int
                                    *n2){
    18
              int temp;
    19
                                              Pendefinisian Prosedur
    20
              temp=*n1;
    21
              *n1 = *n2;
    22
              *n2 = temp:
    23
```

Jenis Prosedur

Pada dasarnya prosedur membutuhkan input dan output. Input yang standar berasal dari piranti input standar (*standard input device*), yaitu **keyboard**. Output yang standar berasal dari piranti output standar (*standard output device*), yaitu **monitor**. Kedua piranti I/O standar ini dijembatani penggunaannya dalam C oleh library **stdio.h** (standard I/O), yang memperkenalkan printf untuk keperluan output dan scanf untuk keperluan input.

Berdasarkan dari mana datangnya input dan output, prosedur dikategorikan ke dalam 4 macam:

Input	Output	Prosedur
Std dev	Std dev	Nai've
Parameter	Std dev	Semi Nai've
Std dev	Parameter	Semi Nai've
Parameter	Parameter	Nett Effect

Hanya ada satu jenis prosedur yang direkomendasikan, yaitu prosedur yang menghasilkan efek netto (nett effect procedure). Dimana prosedur akan menghasilkan efek netto yang berarti tidak menggunakan standard device untuk input maupun output alias tidak terdapat printf dan scanf di dalam prosedur. Salah satu prosedur yang masih memungkinkan untuk dipakai adalah prosedur CETAK, yaitu menggunakan printf untuk mencetak informasi ke monitor. Sedangkan prosedur INPUTAN, yaitu menggunakan scanf atau buffer read lebih baik digunakan pada main.

Gambar 4 Contoh prosedur yang tidak direkomendasikan

```
#include <stdio.h>
#include <stdib.h>

void menghitung();
int main(int argc, char *argv[]){
    menghitung();
    return 0;
}

void menghitung(){
    int a,b,hasil;

    printf("\nMasukkan Angka 1 : ");
    printf("\nMasukkan Angka 2 : ");
    hasil=a+b;
    printf("Hasil : %d",hasil);
}
Terdapat Inputan
}
```

Gambar 5 Contoh prosedur yang direkomendasikan (nett effect procedure)

a) Gambar 5.1 Prosedur yang menghasilkan efek netto

```
#include <stdio.h>
               #include <stdlib.h>
               void menghitung(int *hasil, int a, int b);
               int main(int argc, char *argv[]){
                   int a, b, hasil;
                   printf("\nMasukkan Angka 1 : "); scanf("%d", &a);
printf dan scanf
                   printf("\nMasukkan Angka 2 : "); scanf("%d", &b);
berada di MAIN
                   menghitung(&hasil, a, b);
                   printf("Hasil : %d",hasil);
                    return 0;
               void menghitung(int *hasil, int a, int b){
                                                                Tidak ada printf
                    *hasil = a + b;
                                                                maupun scanf
```

b) Gambar 5.2 Prosedur yang menghasilkan efek netto dengan parameter I/O

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

void kuadrat(int *angka);

int main(int argc, char *argv[]){
   int a;

   printf("\nMasukkan Angka yang ingin dikuadratkan : ");
   scanf("%d", &a);
   kuadrat(&a);
   printf("Hasil Kuadrat : %d",a);

   return 0;
}

void kuadrat(int *angka){

   *angka = (*angka) * (*angka);
}

Menjadi inputan
sekaligus hasil akhir
(Parameter I/O)
```

Gambar 6 Contoh prosedur yang masih memungkinkan untuk dipakai

```
#include <stdio.h>
#include <stdlib.h>
void tampilData(int hasil, int a, int b);
int main(int argc, char *argv[]){
    int a, b, hasil;
   printf("\nMasukkan Angka 1 : "); scanf("%d", &a);
   printf("\nMasukkan Angka 2 : "); scanf("%d", &b);
   hasil = a + b;
   printf("Hasil : %d",hasil);
   tampilData(hasil, a, b);
    return 0;
void tampilData(int hasil, int a, int b){
   printf("\nAngka 1 : %d", a);
                                             Keseluruhan hanya
   printf("\nAngka 2 : %d", b);
                                             berisi printf
   printf("\nHasil : %d",hasil);
```

Pemanggilan prosedur oleh prosedur lain

Suatu prosedur bukan merupakan program yang berdiri sendiri sehingga tidak dapat dieksekusi secara langsung ketika program dijalankan (berbeda dengan main program). Untuk itu suatu prosedur memerlukan sebuah pemanggil. Prosedur dapat dipanggil darimana saja, dari main () function, dari fungsi lain, maupun dari prosedur lain. Pemanggilan dari main () function telah dipelajari pada modul sebelumnya. Materi fungsi akan didapatkan pada modul selanjutnya. Kali ini, pembahasan akan berfokus pada pemanggilan prosedur oleh prosedur lain.

Perhatikan contoh kasus berikut, diberikan prosedur akumulasi dan kenaikan gaji:

Gambar 7

Ket: hasil sudah ada isi/nilainya, hendak diakumulasikan dengan delta. delta sudah terdefinisi dan ada isi/nilainya juga. Kemudian, hasil akan ketambahan sebesar delta. Isi/nilai dari delta tetap.

Gambar 8

```
void kenaikanGaji(double *gaji, double persenKenaikan){
   double kenaikan;

kenaikan = (persenKenaikan / 100) * (*gaji);

*gaji = (*gaji) + kenaikan;
}
```

Ket: gaji sudah ada isi/nilainya, hendak diakumulasikan dengan persenKenaikan dikalikan gaji. persenKenaikan sudah terdefinisi dan ada isi/nilainya juga. Kemudian, gaji akan ketambahan sebesar persenKenaikan dikalikan gaji. Isi/nilai dari persenKenaikan tetap.

Apakah kalian melihat adanya struktur yang sama pada isi prosedur akumulasi dan kenaikan gaji? Jika kalian melihat dan mencermati, *gaji = (*gaji) + kenaikan; pada isi prosedur kenaikan gaji memiliki struktur yang sama dengan isi pada prosedur akumulasi, bukan?

Ini menunjukkan adanya ketidakefektifan karena adanya code yang ditulis berulang-ulang. Lalu bagaimana mengatasinya? Adalah dengan memanggil prosedur akumulasi yang sudah dibuat sebelumnya. Sehingga, prosedur berubah menjadi:

```
void akumulasi(double *hasil, double delta){
         *hasil = (*hasil) + delta;
18
19
20
21 - void kenaikanGaji(double *gaji, double persenKenaikan){
22
         double kenaikan;
23
         kenaikan = (persenKenaikan / 100) * (*gaji);
24
25
                                                         Pemanggilan prosedur
26
         akumulasi(&(*gaji), kenaikan);
                                                         di dalam prosedur
27
```

Perlu diingat!

Ada beberapa aturan yang harus diperhatikan, yakni:

1. Pada parameter formal, di depan variable output maupun input/output diberi tanda "*".

```
void tukar(int *n1, int *n2);
int main(int argc, char *argv[]) {
    ...
}
void tukar(int *n1, int *n2) {
    int temp;
    temp = *n1;
    *n1 = *n2;
    *n2 = temp;
}
Variabel n1 dan n2 adalah
variabel input/output,
sehingga diberi "*"

Variabel n1 dan n2 adalah
variabel input/output,
sehingga diberi "*"

Variabel n1 dan n2 adalah
variabel input/output,
sehingga diberi "*"

Variabel n1 dan n2 adalah
variabel input/output,
sehingga diberi "*"

Variabel n1 dan n2 adalah
variabel input/output,
sehingga diberi "*"

*n2 = temp;
}
```

2. Apabila variabel pada parameter formal memiliki tambahan '*' di depannya (variabel output atau variabel input/output), maka di dalam prosedur tersebut, jika ingin menggunakan/memanggil variabel itu harus juga ditulis lengkap dengan '*'.

3. Apabila variabel pada parameter formal memiliki tambahan '*' di depannya (variabel output atau variabel input/output), maka variabel di dalam pemanggilan atau variabel pada parameter aktual harus didahului dengan '&'.

```
void tukar(int *n1, int *n2);
int main(int argc, char *argv[]) {
    int nilai1 = 5, nilai2 = 10;

    tukar(&nilai1, &nilai2);
    return 0;
}

void tukar(int *n1, int *n2) {
        ...
}
Karena pada parameter formal
        memiliki tambahan '*', maka di
        parameter aktualnya menggunakan '&'
        ...
}
```

```
void akumulasi(double *hasil, double delta) {
    *hasil = (*hasil) + delta;
}

void kenaikanGaji(double *gaji, double persenKenaikan) {
    double kenaikan;

kenaikan = (persenKenaikan / 100) * (*gaji);
    akumulasi(&(*gaji), kenaikan);
}

Jika variabel yang akan digunakan memiliki tambahan '*', maka '*' tetap dituliskan
```

4. Pemanggilan prosedur harus berdiri sendiri sebagai sebuah statement merdeka, tidak boleh ditempelkan pada statement lain.

```
void LuasPersegiPanjang(int *Luas, int p, int l);
int main(int argc, char *argv[]) {
                                             (BENAR)
      LuasPersegiPanjang(&L, 3, 5);
                                             Berdiri sendiri sebagai
                                             statement terpisah
      printf("Luas = %d", L);
      printf("Luas = %d", LuasPersegiPanjang(&L, 3,
                                                        5));
      return 0;
                                                        (SALAH)
                                                        Merupakan
                                                        pemanggilan fungsi,
void LuasPersegiPanjang(int *Luas, int p, int l) {
                                                        bukan prosedur
      *Luas = p * 1;
```

5. **Nama prosedur** baik disaat pemanggilan, pendeklarasian, maupun disaat pendefinisian **harus sama persis** (*Case Sensitive*).

```
void LuasPersegiPanjang (int *Luas, int p, int 1);
int main(int argc, char *argv[]) {
    int L;
    LuasPersegiPanjang (&L, 3, 5);
    printf("Luas = %d", L);
    return 0;
}

void LuasPersegiPanjang (int *Luas, int p, int 1) {
    *Luas = p * 1;
}
```

```
void LuasPersegiPanjang(int *Luas, int p, int 1);
int main(int argc, char *argv[]) {
    int L;
    Luaspersegipanjang(&L, 3, 5);
    printf("Luas = %d", L);
    return 0;
}

void LuasPersegiPanjang(int *Luas, int p, int 1) {
    *Luas = p * 1;
}
*Luas = p * 1;
```

6. **Nama variabel** di parameter aktual dan di parameter formal **dapat berbeda/sama**. Namun, untuk **tipe data nya harus sama.**

```
void kuadrat(int *angka);
int main(int argc, char *argv[]) {
    int a;
    printf("\nMasukkan Angka yang ingin dikuadratkan : ");
    scanf("%d", &a);
    kuadrat(&a);
    printf("Hasil Kuadrat : %d",a);
    return 0;
}
void kuadrat(int *angka) {
    *angka = (*angka) * (*angka);
}

a merupakan variable
local yang berfungsi di
    main, sedangkan angka
    merupakan variable local
    yang berfungsi di
    prosedur.
}
```

7. **CACAH** (jumlah) parameter, **TIPE DATA** parameter, dan **URUTAN** parameter **harus sama** baik di parameter aktual maupun di parameter formal.

```
void penjumlahan(float *hasil, int nilai1, float nilai2);
int main(int argc, char *argv[]){
    float jawaban;
    penjumlahan(&jawaban) 5, 4.3);
    printf("\nJawaban = %.2f", jawaban);
    return 0;
}

void penjumlahan(float *hasil, int nilai1, float nilai2){
    *hasil = (float)nilai1 + nilai2;
}
```

```
void penjumlahan(float *hasil, int nilai1, float nilai2);
int main(int argc, char *argv[]){
    float jawaban;
    penjumlahan(sjawaban, 4.3, 5, 10);
    printf("\nJawaban = %.2f", jawaban);
    return 0;
}
void penjumlahan(float *hasil, int nilai1, float nilai2){
    *hasil = (float)nilai1 + nilai2;
}
```

GUIDED

1. Contoh Penerapan parameter Input/Output.

```
#include <stdio.h>
#include <stdlib.h>
//deklarasi prosedur
void tukarNilai(int *N1, int *N2);
void kuadrat(int *nilai);
//main program
int main(int argc, char *argv[]) {
   int nilai1, nilai2;
    printf("\t\t GUIDED 1 \n");
    printf("\n Masukkan nilai 1 : "); scanf("%d", &nilai1); //inputan berada di main
    printf("\n Masukkan nilai 2 : "); scanf("%d", &nilai2);
    printf("\n Input Berhasil [!]");
    getch();
    system("cls");
    printf("\t\t GUIDED 1 \n");
    printf("\n Nilai awal");
    printf("\n Nilai 1 = %d", nilai1);
    printf("\n Nilai 2 = %d", nilai2);
    kuadrat(&nilai1); //pemanggilan prosedur kuadrat
    kuadrat(&nilai2); //pemanggilan prosedur kuadrat
    printf("\n\n Setelah dikuadratkan");
    printf("\n Nilai 1 = %d", nilai1);
    printf("\n Nilai 2 = %d", nilai2);
    tukarNilai(&nilai1, &nilai2); //pemanggilan prosedur tukarNilai
   printf("\n\n Setelah Nilai1 & Nilai2 ditukar");
    printf("\n Nilai 1 = %d", nilai1);
    printf("\n Nilai 2 = %d", nilai2);
    return 0;
}
//pendefinisian prosedur
void tukarNilai(int *N1, int *N2){ //prosedur net effect
   int temp; //temp variabel pembantu saja
    temp=*N1;
    *N1=*N2;
    *N2=temp;
}
void kuadrat(int *nilai){ //prosedur net effect
    *nilai=(*nilai) * (*nilai);
}
```

Output:

```
GUIDED 1

Masukkan nilai 1 : 5

Masukkan nilai 2 : 2

Input Berhasil [!]
```

2. Buatlah program seperti dibawah ini dan pahami kasus contoh soal! (Berguna sekali untuk UGD)

Siska memiliki kegemaran untuk membuat program. Diapun ingin membuat program untuk pencatatan penjualan miliknya. Ia memiliki usaha penjualan tas. Tas yang dijualnya memiliki 3 tipe dengan harga yang berbeda-beda. Tipe A dengan harga Rp 500.000, Tipe B dengan harga Rp 650.000, dan Tipe C dengan harga Rp 700.000. Siska memiliki persediaan awal untuk tas A, B, dan C masing masing 10 tas. Ia ingin memastikan bahwa programnya aman dan tidak bisa sembarang orang gunakan. Maka dari itu, Siska menerapkan fitur login di awal programnya. Buatlah program seperti yang Siska buat tersebut dengan kriteria, sbb:

LOGIN

Saat pertama kali program dijalankan, program akan menampilkan tampilan login dan meminta user untuk memasukkan username beserta password. Username dan password yang digunakan, yaitu:

Username : Siska → case sensitive

Password: 12345

Program akan menampilkan error/peringatan apabila username dan password yang dimasukan salah. Toleransi kesalahan yang diberikan sebanyak 5 kali, jika user salah memasukkan username dan password sampai 5 kali, maka program akan keluar. Apabila user berhasil login, maka program akan masuk ke pilihan menu.

Di atas pilihan menu, tampilkan sisa persediaan untuk masing-masing tas.

1. Input Data Pembeli

Menu ini hanya dapat diakses sekali.

Menu ini akan meminta data pembeli berupa nama, dan tanggal pembelian. Untuk nama tidak boleh kosong dan tanggal pembelian wajib berjumlah 10 karakter (DD-MM-YYYY).

2. Input Pembelian

Jika belum input data pembeli, menu ini tidak dapat diakses.

Menu ini akan meminta inputan pembelian apa yang ingin dibeli. Program akan meminta inputan untuk tipe tas yang dibeli, lalu menampilkan harganya dan meminta inputan jumlah pembelian. Jika berhasil, maka sisa persediaan akan

otomatis berkurang sebanyak jumlah pembelian. Jumlah pembelian tidak boleh kurang dari atau sama dengan nol.

3. Rincian

Jika belum input data pembeli dan input pembelian, menu ini tidak dapat diakses. Menu ini akan menampilkan rincian pembelian. Rincian yang ditampilkan adalah nama, tanggal pembelian, jumlah pembelian untuk masing-masing tas, dan di akhir akan menampilkan total yg harus dibayar. Tampilkan juga harga masingmasing tas.

4. Logout

Menu ini akan mengembalikkan tampilan ke bagian login kembali dan sekaligus menginisialisasi kembali seluruh data.

Sourch Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define hargaA 500000 //set harga tas A secara default (konstanta)
#define hargaB 650000 //set harga tas B secara default (konstanta)
#define hargaC 700000 //set harga tas C secara default (konstanta)
typedef char string[50];
void inisialisasi(int *toleransi, int *tipeA, int *tipeB, int *tipeC, int *pembeli, int *pembelian, int *jmlA,
                  int *jmlB, int *jmlC, int *totA, int *totB, int *totC, int *total); //untuk menginisialisasi nilai awal
void cekLogin(int *cek, string user, int pass); // untuk mengecek kondisi login benar atau tidak
void penjumlahan(int *angka, int penjumlah); //untuk operasi penjumlahan
void pengurangan(int *angka, int pengurang); //untuk operasi pengurangan
void perkalian(int *hasil, int angka, int pengali); //untuk operasi perkalian
void cekHarga(int *harga, string tipe); //untuk mengecek harga sesuai tipe tas inputan
void cekJumlahTipe(int *temp, int tasA, int tasB, int tasC, string tipe); //untuk cek Ketersediaan tas dengan tipe tertentu
void pembaruanKetersediaanTas(int *tasA, int *tasB, int *tasC, int *jmlA, int *jmlB, int *jmlC, int jml, string tipe);
void printRincian(string nama, string tgl, int jmlA, int jmlB, int jmlC, int totA, int totB, int totC, int total);
void hitungTotal(int *tot, int jmlA, int jmlB, int jmlC); //untuk menghitung total pembelian keseluruhan
void cekLogout(int *logout, string konfirm); //untuk mengecek kondisi logout benar atau tidak
int main(int argc, char *argv[]) {
    string user, konfirm, nama, tgl, tipe;
    int pass, menu, cek, logout, harga, jumlah, tempJumlah;
    int toleransi, tipeA, tipeB, tipeC, pembeli, pembelian, jmlA, jmlB, jmlC, totA, totB, totC, total;
    inisialisasi(&toleransi, &tipeA, &tipeB, &tipeC, &pembeli, &pembelian, &jmlA, &jmlB, &jmlC, &totA, &totB, &totC,
&total);
        logout=0; // untuk inisialisasi variabel logout
        system("cls"); // untuk clear screen
        printf("\n\t\t=== LOGIN SYSTEM ===");
        printf("\n\t\t\tToleransi : %d\n", toleransi);
        printf("\tUsername : "); fflush(stdin); gets(user);
        printf("\tPassword : "); scanf("%d", &pass);
        cekLogin(&cek, user, pass); //pengecekan login dengan prosedur nett effect
        if(cek==1){
           do{
                system("cls");
                printf("\n\t\tTOKO TAS SISKA");
                printf("\n\t\t==== MENU ====");
                printf("\n\t %d Tipe A - %d Tipe B - %d Tipe C", tipeA, tipeB, tipeC);
                printf("\n\t[3] Rincian");
                printf("\n\t[0] Logout");
                printf("\n\t>>> "); scanf("%d", &menu);
                switch(menu){
                    case 1:
                        if(pembeli==1){
                           printf("\n\t[!] Data Pembeli sudah terinput!");
                            printf("\n\t\t==INPUT DATA PEMBELI==\n");
                            printf("\tNama : "); fflush(stdin); gets(nama);
                            while(strlen(nama)==0 || strcmpi(nama,"-")==0){
                                printf("\n\t\t[!] Nama tidak boleh kosong!\n");
                                printf("\n\tNama : "); fflush(stdin); gets(nama);
                            printf("\tTanggal Pembelian : "); fflush(stdin); gets(tgl);
                            while(strlen(tgl)!=10){ //format yang diinginkan DD-MM-YYYY (10 karakter)
                                printf("\n\t\t[!] Tanggal salah!\n");
                                printf("\tTanggal Pembelian : "); fflush(stdin); gets(tgl);
                            printf("\n\t\tBerhasil Input Data Pembeli..");
                            pembeli=1:
                        break;
```

```
case 2:
                    if(pembeli==0){
                        printf("\n\t[!] Data Pembeli belum ada!");
                        printf("\n\t\t==INPUT PEMBELIAN==\n");
                        printf("\tTipe Tas (A/B/C) : "); fflush(stdin); gets(tipe);
                        while(strcmpi(tipe, "A")!=0 && strcmpi(tipe, "B")!=0 && strcmpi(tipe, "C")!=0){
                            printf("\n\t\t[!] Tipe tidak diketahui!\n");
                            printf("\tTipe Tas (A/B/C) : "); fflush(stdin); gets(tipe);
                        cekHarga(&harga, tipe); //ini akan mengecek harga dari tipe tas yang diinputkan user
                                                     : %d", harga);
                        printf("\n\tHarga
                        printf("\n\tJumlah Pembelian : "); scanf("%d", &jumlah);
                        cekJumlahTipe(&tempJumlah, tipeA, tipeB, tipeC, tipe); /*cek ketersediaan dari
                        if(jumlah<=0){</pre>
                            printf("\n\t[!] Jumlah Pembelian tidak boleh kurang dari 1!");
                        }else if(jumlah>tempJumlah){
                            printf("\n\t[!] Jumlah Pembelian melebihi ketersediaan!");
                            pembaruanKetersediaanTas(&tipeA, &tipeB, &tipeC, &jmlA, &jmlB, &jmlC, jumlah, tipe);
                            printf("\n\tBerhasil Input Pembelian..");
                            pembelian=1;
                    break;
                case 3:
                    if(pembeli==0){
                        printf("\n\t[!] Data Pembeli belum ada!");
                    }else if(pembelian==0){
                        printf("\n\t[!] Belum ada Pembelian!");
                    }else{
                        printf("\n\t\t==RINCIAN==\n");
                        perkalian(&totA, jmlA, hargaA); //menghitung total harga untuk tipe tas A
                        perkalian(&totB, jmlB, hargaB); //menghitung total harga untuk tipe tas B
                        perkalian(&totC, jmlC, hargaC); //menghitung total harga untuk tipe tas C
                        hitungTotal(&total, totA, totB, totC); //menghitung total harga keseluruhan
                        printRincian(nama, tgl, jmlA, jmlB, jmlC, totA, totB, totC, total);
                    break;
                case 0:
                    printf("\n\tYakin ingin logout? (ya/tidak)\n\t"); fflush(stdin); gets(konfirm);
                    cekLogout(&logout, konfirm);
                    if(logout==1){
                        printf("\n\tTerima kasih");
                        inisialisasi(&toleransi, &tipeA, &tipeB, &tipeC, &pembeli, &pembelian,
                                     &jmlA, &jmlB, &jmlC, &totA, &totB, &totC, &total);
                    }
                    break;
                    printf("\n\t[!] Menu tidak ada!");
            }getch();
        }while(logout!=1);
    }else{
        printf("\n\t\t[!] Username / Password Salah!");
        pengurangan(&toleransi, 1); //toleransi dikurangi 1 setiap kesalahan login
    getch();
}while(toleransi!=0);
printf("\n\n\t\tKeluar");
printf("\n\t\tTerima kasih...");
```

```
void inisialisasi(int *toleransi, int *tipeA, int *tipeB, int *tipeC, int *pembeli, int *pembelian, int *jmlA,
                   int *jmlB, int *jmlC, int *totA, int *totB, int *totC, int *total){
    *tipeB=10;
    *pembeli=0;
    *pembelian=0;
    *jmlA=0;
    *jmlB=0;
    *jmlC=0;
    *totA=0;
    *totB=0;
    *totC=0;
    *total=0;
void cekLogin(int *cek, string user, int pass){
   if(strcmp(user,"Siska")==0 && pass==12345){
        *cek=1;
void penjumlahan(int *angka, int penjumlah){
    (*angka) = (*angka) + penjumlah;
void pengurangan(int *angka, int pengurang){
    (*angka) = (*angka) - pengurang;
void perkalian(int *hasil, int angka, int pengali){
    (*hasil) = angka * pengali;
void cekHarga(int *harga, string tipe){
    if(strcmpi(tipe,"A")==0){
        *harga=hargaA;
    }else if(strcmpi(tipe, "B") == 0){
        *harga=hargaB;
        *harga=hargaC;
void cekJumlahTipe(int *temp, int tasA, int tasB, int tasC, string tipe){
        *temp=tasA;
    }else if(strcmpi(tipe,"B")==0){
        *temp=tasB;
        *temp=tasC;
void pembaruanKetersediaanTas(int *tasA, int *tasB, int *tasC, int *jmlA, int *jmlB, int *jmlC, int jml, string tipe)
{ if(strcmpi(tipe,"A")==0){
    pengurangan(&(*tasA), jml); //tas A akan berkurang sebanyak jml
penjumlahan(&(*jmlA), jml); //untuk mendapatkan total jumlah pembelian untuk tipe tas A
}else if(strcmpi(tipe,"B")==0){
        pengurangan(&(*tasB), jml); //tas B akan berkurang sebanyak jml
        penjumlahan(\&(\star jmlB), jml); //untuk mendapatkan total jumlah pembelian untuk tipe tas B
        pengurangan(&(*tasC), jml); //tas C akan berkurang sebanyak jml
penjumlahan(&(*jmlC), jml); //untuk mendapatkan total jumlah pembelian untuk tipe tas C
void printRincian(string nama, string tgl, int jmlA, int jmlB, int jmlC, int totA, int totB, int totC, int total){
    void hitungTotal(int *tot, int jmlA, int jmlB, int jmlC){
void cekLogout(int *logout, string konfirm){
    if(strcmpi(konfirm,"ya")==0){
        *logout=1;
        *logout=0;
```

Output:

Berhasil Input Pembelian.._

```
=== LOGIN SYSTEM ===
                    Toleransi : 5
 Username : Siska
 Password: 12345
        TOKO TAS SISKA
        ==== MENU ====
                                                                TOKO TAS SISKA
 10 Tipe A - 10 Tipe B - 10 Tipe C
                                                                ==== MENU ====
[1] Input Data Pembeli
                                                         10 Tipe A - 10 Tipe B - 10 Tipe C
[2] Input Pembelian
                                                       [1] Input Data Pembeli
[3] Rincian
                                                       [2] Input Pembelian
[0] Logout
                                                       [3] Rincian
>>> 1
                                                       [0] Logout
        ==INPUT DATA PEMBELI==
Nama : Saya
Tanggal Pembelian : 11-11-2022
                                                       [!] Data Pembeli sudah terinput!_
        Berhasil Input Data Pembeli..
                                                                 TOKO TAS SISKA
         TOKO TAS SISKA
                                                                 ==== MENU ====
         ==== MENU ====
                                                           10 Tipe A - 10 Tipe B - 10 Tipe C
  10 Tipe A - 10 Tipe B - 10 Tipe C
                                                         [1] Input Data Pembeli
 [1] Input Data Pembeli
                                                         [2] Input Pembelian
 [2] Input Pembelian
[3] Rincian
                                                         [3] Rincian
[0] Logout
                                                         >>> 2
 [0] Logout
 >>> 2
                                                                 ==INPUT PEMBELIAN==
                                                         Tipe Tas (A/B/C) : W
         ==INPUT PEMBELIAN==
 Tipe Tas (A/B/C) : W
                                                         [!] Tipe tidak diketahui!
Tipe Tas (A/B/C) : S
         [!] Tipe tidak diketahui!
 Tipe Tas (A/B/C) : S
                                                                 [!] Tipe tidak diketahui!
                                                         Tipe Tas (A/B/C) : A
        [!] Tipe tidak diketahui!
 Tipe Tas (A/B/C) : A
                                                                        : 500000
                                                         Jumlah Pembelian : 15
                  : 500000
 Jumlah Pembelian : _
                                                         [!] Jumlah Pembelian melebihi ketersediaan!
         TOKO TAS SISKA
         ==== MENU ====
  10 Tipe A - 10 Tipe B - 10 Tipe C
[1] Input Data Pembeli
                                                                  TOKO TAS SISKA
[2] Input Pembelian
                                                                 ____ MENU ====
                                                        0 Tipe A - 10 Tipe B - 10 Tipe C

[1] Input Data Pembeli

[2] Input Pembelian
[3] Rincian
[0] Logout
>>> 2
                                                        [3] Rincian
         ==INPUT PEMBELIAN==
                                                        [0] Logout
Tipe Tas (A/B/C) : A
                                                        >>>
                  : 500000
Harga
Jumlah Pembelian : 10
```

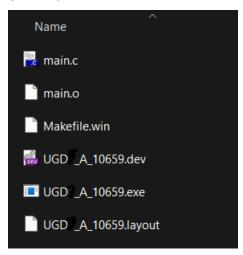
```
TOKO TAS SISKA
       ==== MENU ====
 0 Tipe A - 10 Tipe B - 10 Tipe C
[1] Input Data Pembeli
[2] Input Pembelian
[3] Rincian
[0] Logout
       ==RINCIAN==
               : Saya
Tanggal Pemelian : 11-11-2022
 [Tas A]
[Tas B]
[Tas C]
             | 10 |
                         @Rp 500000 |
                                      ---> Total : Rp 5000000
                         @Rp 650000
              0 |
                                       ---> Total : Rp 0
            0
                                      ---> Total : Rp 0
                       @Rp 700000
Total yang harus dibayar: Rp 5000000_
            TOKO TAS SISKA
                                                                           TOKO TAS SISKA
                                                                           ==== MENU ====
            ==== MENU ====
                                                                     0 Tipe A - 10 Tipe B - 10 Tipe C
     0 Tipe A - 10 Tipe B - 10 Tipe C
                                                                   [1] Input Data Pembeli
[2] Input Pembelian
   [1] Input Data Pembeli
  [2] Input Pembelian
                                                                   [3] Rincian
  [3] Rincian
                                                                   [0] Logout
                                                                   >>> 0
  [0] Logout
  >>> 0
                                                                   Yakin ingin logout? (ya/tidak)
  Yakin ingin logout? (ya/tidak)
                                                                   Terima kasih
                                                                            === LOGIN SYSTEM ===
                                                                                      Toleransi : 5
                                                                 Username : _
```

Guided dikerjakan dalam folder terpisah (**GD8_X_YYYYY_NoSoal**) lalu dimasukkan kedalam 1 folder dengan format (**GD8_X_YYYYY**).

Ket: X adalah kelasYYYYY adalah 5 digit NPM terakhir

Tambahan untuk pengumpulan:

1. Dalam folder terdapat beberapa file (.c, .o, .win, .dev, .exe, .layout) seperti berikut (contoh)



2. Jangan hanya mengumpulkan file .c nya saja!