

Modul 1

Code

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

typedef long long LL;

struct walmart{
    LL price, stocks;
    char name[50], code[5];
} store[6];

void header(){
    printf("\n\n");
    printf("JOJO AIR FRESHENER SHOP CASHIER\n");
    printf("=====\n\n");
}

void warehouse(){
    //Air freshener codes
    strcpy(store[0].code, "AR004");
    strcpy(store[1].code, "AR005");
    strcpy(store[2].code, "AR007");
    strcpy(store[3].code, "AR011");
    strcpy(store[4].code, "AR016");
    strcpy(store[5].code, "AR019");
    strcpy(store[6].code, "AR022");

    //Air freshener names
    strcpy(store[0].name, "Classic Rose");
    strcpy(store[1].name, "Fresh Lemon");
    strcpy(store[2].name, "Wild Lavender");
    strcpy(store[3].name, "Elegant Vanilla");
    strcpy(store[4].name, "Freesia & Jasmine");
    strcpy(store[5].name, "Exotic Flower");
    strcpy(store[6].name, "Ocean Scape");

    //Stocks each air freshener
    store[0].stocks = 20;
    store[1].stocks = 40;
    store[2].stocks = 25;
    store[3].stocks = 20;
    store[4].stocks = 18;
    store[5].stocks = 21;
    store[6].stocks = 30;
```

```

    //Prices each air freshener
    store[0].price = 28000;
    store[1].price = 30000;
    store[2].price = 30000;
    store[3].price = 32000;
    store[4].price = 33000;
    store[5].price = 34000;
    store[6].price = 35000;
}

void shelf(){
    printf("No. | Code | Air Freshener Name\t| Available | Price\n");
    printf("-----\n");
    for(int i=0; i < 6; i++){
        printf("%02d. | %s | %s\t\t| \t %02lld | Rp. %lld,-\n", i+1,
store[i].code, store[i].name, store[i].stocks, store[i].price);
    }
    printf("-----\n");
}

void sell(){
    char codex[5]; LL qty;
    printf("Input Air Freshener Code [5 chars]: ");
    scanf("%[^\n]", &codex); fflush(stdin);

    int found_codex = -1;
    for(int i=0; i < 6; i++){
        if(strcmp(codex, store[i].code) == 0){
            found_codex = i; break;
        }
    }
    int founded = found_codex;

    if(founded != -1){
        printf("Input quantity [0..%lld]: ", store[founded].stocks);
        scanf("%lld", &qty); fflush(stdin);
        while(qty > store[founded].stocks){
            printf("\n...The quantity of air freshener is not enough...\n\n");
            printf("Input quantity [0..%lld]: ", store[founded].stocks);
            scanf("%lld", &qty); fflush(stdin);
        }
        store[founded].stocks -= qty;
        printf("\nTotal Price is : Rp %lld,- x %lld = Rp %lld,-\n",
store[founded].price, qty, qty*store[founded].price);
        printf("--- Thank You ---");
    } else{

```

```

        printf("--- The air freshener code doesn't exist ---\n");
        return sell();
    }
}

void addStock(){
    char codex[5]; LL qty;
    printf("Input Air Freshener Code [5 chars]: ");
    scanf("%[^\\n]", &codex); fflush(stdin);

    int found_codex = -1;
    for(int i=0; i < 6; i++){
        if(strcmp(codex, store[i].code) == 0){
            found_codex = i; break;
        }
    }
    int founded = found_codex;

    if(founded != -1){
        printf("Input quantity [1..100]: ");
        scanf("%lld", &qty); fflush(stdin);
        while(qty < 1 || qty > 100){
            printf("\n...The quantity of air freshener must be 1 to 100...\n\n");
            printf("Input quantity [1..100]: ");
            scanf("%lld", &qty); fflush(stdin);
        }
        store[founded].stocks += qty;
        printf("--- Adding Stock Success ---");
    } else{
        printf("--- The air freshener code doesn't exist ---\n");
        return addStock();
    }
}

int main(){
    int opt;
    printf("\n");
    printf("\t\t\tMade by\t\t\tPristian Budi Dharmawan - 2501983105\n");
    printf("\t\t\t\t\t\t\tVer. 2.03.10");
    warehouse();
    do{
        header();
        shelf();
        printf("\n\nMenu :\n");
        printf("1. Sell\n2. Add Stock\n3. Exit\n");
        scanf("%d", &opt, printf("Input choice : ")); getchar();
        switch (opt){
            case 1:
                sell(); break;

```

```

    case 2:
        addStock(); break;
    default:
        printf("\e[1;1H\e[2J"); //To Clear Console
        printf("\n\nThere's no menu no %d", opt);
        break;
    }
} while(opt != 3);
printf("\e[1;1H\e[2J"); //To Clear Console
printf("Thankyou for Coming....");
return 0;
}

```

ScreenShot Hasil

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

2. Add Stock
3. Exit
Input choice :
PS D:\SUNIB\B25\SEMESTER 2\COURSES\DATA STRUCTURE\LAB\SESSION 1\Exercises> cd "d:\SUNIB
PS D:\SUNIB\B25\SEMESTER 2\COURSES\DATA STRUCTURE\LAB\SESSION 1\Exercises> & .\DS Modu

Made by
Pristian Budi Dharmawan - 2501983105
Ver. 2.03.10

JOJO AIR FRESHENER SHOP CASHIER
=====

No. | Code | Air Freshener Name | Available | Price
-----
01. | AR004 | Classic Rose | 20 | Rp. 28000,-
02. | AR005 | Fresh Lemon | 40 | Rp. 30000,-
03. | AR007 | Wild Lavender | 25 | Rp. 30000,-
04. | AR011 | Elegant Vanilla | 20 | Rp. 32000,-
05. | AR016 | Freesia & Jasmine | 18 | Rp. 33000,-
06. | AR019 | Exotic Flower | 21 | Rp. 34000,-
-----

Menu :
1. Sell
2. Add Stock
3. Exit
Input choice : 1
Input Air Freshener Code [5 chars]: 

```

Made by
Pristian Budi Dharmawan - 2501983105
Ver. 2.03.10

JOJO AIR FRESHENER SHOP CASHIER

No.	Code	Air Freshener Name	Available	Price
01.	AR004	Classic Rose	20	Rp. 28000,-
02.	AR005	Fresh Lemon	40	Rp. 30000,-
03.	AR007	Wild Lavender	25	Rp. 30000,-
04.	AR011	Elegant Vanilla	20	Rp. 32000,-
05.	AR016	Freesia & Jasmine	18	Rp. 33000,-
06.	AR019	Exotic Flower	21	Rp. 34000,-

Menu :
1. Sell
2. Add Stock
3. Exit
Input choice : 1
Input Air Freshener Code [5 chars]: AR0
--- The air freshener code doesn't exist ---
Input Air Freshener Code [5 chars]: AR005
Input quantity [0..40]:

Menu :
1. Sell
2. Add Stock
3. Exit
Input choice : 1
Input Air Freshener Code [5 chars]: AR0
--- The air freshener code doesn't exist ---
Input Air Freshener Code [5 chars]: AR005
Input quantity [0..40]: 50

...The quantity of air freshener is not enough...

Input quantity [0..40]: 20

Total Price is : Rp 30000,- x 20 = Rp 600000,-
--- Thank You ---

JOJO AIR FRESHENER SHOP CASHIER

No.	Code	Air Freshener Name	Available	Price
01.	AR004	Classic Rose	20	Rp. 28000,-
02.	AR005	Fresh Lemon	20	Rp. 30000,-
03.	AR007	Wild Lavender	25	Rp. 30000,-
04.	AR011	Elegant Vanilla	20	Rp. 32000,-
05.	AR016	Freesia & Jasmine	18	Rp. 33000,-
06.	AR019	Exotic Flower	21	Rp. 34000,-

Menu :
1. Sell
2. Add Stock
3. Exit
Input choice :

```
Menu :
1. Sell
2. Add Stock
3. Exit
Input choice : 2
Input Air Freshener Code [5 chars]: AR007
Input quantity [1..100]: 0

...The quantity of air freshener must be 1 to 100...

Input quantity [1..100]: 15
--- Adding Stock Success ---

JOJO AIR FRESHENER SHOP CASHIER
-----
No. | Code | Air Freshener Name | Available | Price
-----
01. | AR004 | Classic Rose | 20 | Rp. 28000,-
02. | AR005 | Fresh Lemon | 20 | Rp. 30000,-
03. | AR007 | Wild Lavender | 40 | Rp. 30000,-
04. | AR011 | Elegant Vanilla | 20 | Rp. 32000,-
05. | AR016 | Freesia & Jasmine | 18 | Rp. 33000,-
06. | AR019 | Exotic Flower | 21 | Rp. 34000,-
-----

Menu :
1. Sell
2. Add Stock
3. Exit
Input choice : 
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Thankyou for Coming....
PS D:\SUNIB\B25\SEMESTER 2\COURSES\DATA STRUCTURE\LAB\SESSION 1\Exercises> 
```

Penjelasan Code

Seperti yang diminta pad soal module, disini saya membuat 3 menu utama, yaitu:

1. Sell

Menu Sell berfungsi untuk melakukan pembelian terhadap stock yang sudah tersedia. Apabila *user* ingin membeli barang dengan code barang yang telah diketik lebih dari stock, maka program akan melakukan printing "...The quantity of air freshener is not enough..." dan akan kembali lagi menanyakan berapa banyak produk yang ingin dibeli. Setelah melakukan pembelian, maka akan keluar sebuah rincian total price, dimana kita akan mengkalikan banyaknya produk yang dibeli dengan harga produk tersebut. Setelah itu, program akan kembali ke menu awal

2. Add Stock

Menu add stock berfungsi untuk menambahkan stock yang ada di dalam program. Misalkan di program sudah tersedia 20 barang AR004 kemudian kita ingin menambahkannya menjadi 40. Menu ini hampir sama dengan menu selling, hanya saja perlu adanya penyesuaian terhadap operator dan validasi data yang digunakan. Dalam menu ini diperlukan adanya sebuah validasi data di perulangan while dimana kita menggunakan syntax

```
while(qty < 1 || qty > 100)
```

Syntax diatas berfungsi ketika *user* mengisi berapa banyak produk yang ingin ditambahkan tidak lebih dari 100 dan tidak kurang dari 1.

3. Exit

Fungsi ini digunakan untuk membersihkan layar program dan keluar dari program

Pada akhirnya, saya menggunakan 6 buah fungsi, dimana fungsi-fungsi tersebut berkaitan dengan fungsi addStock(), sell(), main(). Adapun fungsi lain yang akan dijelaskan sbb:

- header()
Fungsi ini berfungsi untuk mencetak sebuah header dari program ini, yaitu "JOJO AIR FRESHENER SHOP CASHIER"
- warehouse()
Fungsi ini digunakan untuk melakukan storing values kedalam variables yang ada pada struct walmart
- shelf()
Fungsi shelf berfungsi untuk melakukan printing dari data stock yang ada pada fungsi warehouse(). Mengapa dipisah? Hal ini dilakukan agar ketika kita telah melakukan selling maupun adding stock, program kembali dengan nilai yang sudah dirubah bukan dengan nilai yang tetap seperti pada awal program berjalan.

Saya juga menambahkan syntax

```
printf("\e[1;1H\e[2J"); //To Clear Console
```

untuk membersihkan layar console. Perlu diingat bahwa syntax pembersihan console ini tidak bisa berjalan di semua system terminal karena syntax ini hanya dapat berfungsi di ANSI terminals atau terminal di dalam Visual Studio Code.