

RESPONSI

PRAKTIKUM KONSEP PEMROGRAMAN

2024

IDENTITAS

Nama : Muhammad Rafi Raihan Sonjaya

NIM : L0124065

Kelas : B

Judul Program : B_Responsi1KP_L0124065_Muhammad Rafi Raihan
Sonjaya_Toko.c

Deskripsi Program : membuat program toko dalam bahasa c, bisa melihat list barang yang dijual, bisa melihat harga, stok, dan memasukan barang ke dalam keranjang. Lalu bisa melakukan checkout.

Dokumentasi Program

Analisis Kode

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

#include adalah direktif preprosesor yang digunakan untuk mengimpor pustaka atau header file ke dalam program. Dalam kode ini, beberapa pustaka yang diimpor adalah:

- **#include <stdio.h>** Digunakan untuk menyertakan file header stdio.h yang berisi fungsi-fungsi standar untuk input dan output, seperti printf dan scanf. Fungsi-fungsi ini digunakan untuk menampilkan output di layar dan menerima input dari pengguna.
- **#include <stdlib.h>** Digunakan untuk menyertakan file header stdlib.h yang berisi fungsi-fungsi standar untuk manajemen memori, konversi angka, proses, dan fungsi utilitas umum lainnya. Pada program ini, salah satu fungsi dari stdlib.h yang digunakan adalah system("cls") untuk membersihkan layar konsol.
- **#include <string.h>** Digunakan untuk menyertakan file header string.h yang menyediakan fungsi-fungsi untuk manipulasi string, seperti strcpy yang digunakan dalam program untuk menyalin teks dari satu string ke string lainnya.

- **#include <conio.h>** Digunakan untuk menyertakan file header conio.h yang khusus untuk sistem operasi berbasis DOS atau Windows. Salah satu fungsi yang sering digunakan dari conio.h adalah getch(), yang memungkinkan program untuk menunggu pengguna menekan tombol sebelum melanjutkan.

```
//untuk setiap input yang dilakukan
int input;

// Counter untuk jumlah item di keranjang
int keranjangCount = 0;

// Membuat struct untuk menyimpan nama dan harga
struct keranjang {
    char nama[30];
    int harga;
};

// Menyimpan maksimal 100 item di keranjang
struct keranjang listKeranjang[100];
```

- **int input** digunakan untuk mendeklarasi variabel input dari pengguna
- **int keranjangCount** untuk memberikan tanda berapa banyak barang yang terdapat dalam keranjang
- **struct keranjang** berfungsi untuk menyimpan nama dan harga dari barang yang dimasukan ke dalam keranjang
- **struct keranjang listKeranjang** berfungsi untuk membatasi bahwa maksimal barang yang dimasukan ke dalam keranjang hanya bisa 100 barang

```

//menyimpan kategori barang yang dijual melalui array kategori
char kategori[3][20] = {"Handphone", "Laptop", "CPU"};

struct list {
    char nama[30];
    int harga;      // membuat struct, jadi tiap barang itu mempunyai nama harga dan
                    stock nya
    int stock;
};

//membuat array list Handphone dari struct list yang berisi nama harga dan stok
struct list listHandphone[3] = {
    {"IPHONE 15 PRO", 21, 5},
    {"SAMSUNG GALAXY S24 ULTRA", 20, 8},
    {"HUAWEI MATE 60 PRO", 23, 11}
};

//membuat array list Laptop dari struct list yang berisi nama harga dan stok
struct list listLaptop[3] = {
    {"ASUS VIVOBOK", 10, 3},
    {"HP PAVILION", 14, 7},
    {"THINKPAD", 7, 1}
};

//membuat array list CPU dari struct list yang berisi nama harga dan stok
struct list listCPU[3] = {
    {"INTEL CORE I9-7920X", 17, 6},
    {"INTEL CORE I7-14700K", 7, 3},
    {"INTEL CORE I5-10400F", 2, 1}
};

```

- **char kategori[3][20]** untuk menyimpan kategori barang yang dijual, disini ditulis 3 karna hanya terdapat 3 kategori. 20 merupakan huruf maksimal untuk char kategori.
- **struct list** digunakan untuk menyimpan bahwa setiap barang memiliki nama, harga, dan stock
- **struct list listHandphone[3]** membuat struct baru untuk menyimpan kategori handphone, yang di dalam list sudah dideklarasikan bahwa ada nama, harga, lalu stocknya. [3] digunakan untuk menandakan bahwa jumlah barang dari kategori handphone itu terdapat 3 barang
- **struct list listLaptop[3]** membuat struct baru untuk menyimpan kategori laptop, yang di dalam list sudah dideklarasikan bahwa ada nama, harga, lalu stocknya. [3] digunakan untuk menandakan bahwa jumlah barang dari kategori Laptop itu terdapat 3 barang

- **struct list listCPU[3]** membuat struct baru untuk menyimpan kategori CPU, yang di dalam list sudah dideklarasikan bahwa ada nama, harga, lalu stocknya. [3] digunakan untuk menandakan bahwa jumlah barang dari kategori CPU itu terdapat 3 barang

```
//fungsi print barang untuk kategori yang diminta
void printList(struct list *point) {
    // print barang dalam masing-masing kategori
    printf("%s\n", point->nama);
    printf("    Harga          : %d juta\n", point->harga);
    printf("    Stok tersedia : %d\n\n", point->stock);
}

//membuat fungsi untuk pengecekan keranjang
void printKeranjang() {
    //membuat pengkondisian jika keranjang masih kosong
    if (keranjangCount == 0) {
        printf("Keranjang Anda kosong.\n\n");
    } else {
        printf("Keranjang Anda:\n");
        //nge list setiap keranjang setelah user memasukan barang kedalam keranjang
        for (int i = 0; i < keranjangCount; i++) {
            printf("%d. %s - %d juta\n", i + 1, listKeranjang[i].nama, listKeranjang[i].harga);
        }
        printf("\n");
    }
}
```

- **void printList(struct list *point)** membuat fungsi untuk print setiap index array struct, seperti tadi list punya nama, harga, dan stock. maka akan ngeprint sesuai dengan index di dalamnya. Disini kita menggunakan pointer untuk menyimpan ke dalam alamatnya, supaya tersimpan ke dalam global, jadi saat dipanggil menggunakan variabel biasa, maka harus menggunakan &.
- **void printKeranjang()** membuat fungsi untuk user ketika ingin melihat keranjang belanja. disini kita menggunakan perkondisian jika keranjangnya masih kosong. Maka akan mencetak "Keranjang anda kosong". Tetapi jika user sudah memasukan barang ke dalam keranjang, maka counter tidak bernilai 1 lagi, dan akan mencetak semua barang yang telah user masukan ke dalam keranjang beserta harganya.

```

// Fungsi rekursif untuk menghitung total harga keranjang
int hitungTotal(int index) {
    if (index < 0) {
        return 0; // fungsinya akan berhenti jika index nya habis
    }
    //memanggil kembali fungsi, dan mengurangi parameter fungsi dengan 1
    return listKeranjang[index].harga + hitungTotal(index - 1);
}

//membuat fungsi untuk checkout keranjang
void checkoutKeranjang() {
    //membuat kondisi jika keranjangnya masih 0, maka print keranjang kosong
    if (keranjangCount == 0) {
        printf("Keranjang Anda kosong.\n\n");
    } else {
        printf("Checkout Keranjang:\n");
        for (int i = 0; i < keranjangCount; i++) {
            printf("%d. %s - %d juta\n", i + 1, listKeranjang[i].nama, listKeranjang[i].harga);
        }
        // Menghitung total menggunakan fungsi rekursif
        int total = hitungTotal(keranjangCount - 1);
        printf("=====\n");
        printf("Total pembayaran: %d juta\n\n", total);
        printf("Terimakasih telah berbelanja di toko kami!\n\n");
        printf("Keranjang akan dikosongkan kembali\n");
        keranjangCount = 0; // untuk mengosongkan keranjang setelah checkout
    }
}

```

- **int hitungTotal(int index)** membuat fungsi rekursi untuk menghitung total harga yang berada di dalam keranjang. disini kita membuat perkondisian jika index kurang dari 1 maka akan return 0, yang membuat fungsi dari menghitung total harga akan berhenti
- **void checkoutKeranjang()** membuat fungsi jika pengguna memilih ingin mencheckout barangnya, yang pertama itu akan diperiksa terlebih dahulu apakah keranjang nya sudah terisi atau belum. Jika keranjangnya masih 0, maka akan menceta "Keranjang Anda kosong". Tetapi ketika keranjang pengguna bukan 0, maka akan mencetak barang yang sudah pengguna masukan ke dalam keranjang menggunakan for, dan akan dicetak semua barang dan harga keranjang pengguna. Setelah mencetak barang pengguna, lalu akan menjumlahkan total harga dari keranjang menggunakan fungsi rekursif yang sudah dibuat tadi. Dan akan mencetak total harga dari keranjang pengguna. Setelah itu keranjangCount akan dibuat 0 karena keranjang pengguna akan dikosongkan kembali setelah melakukan checkout.

```

//fungsi setelah pengguna memasukan barang ke dalam keranjang
void menambahkanKeDalamKeranjang(struct list item, char *kategori) {
    // Batas maksimal 100 barang di dalam keranjang
    if (keranjangCount < 100) {
        system("cls");
        strcpy(listKeranjang[keranjangCount].nama, item.nama);
        listKeranjang[keranjangCount].harga = item.harga;
        keranjangCount++;
        printf("%s telah dimasukkan ke dalam keranjang.\n\n", item.nama);
        printf("Tekan apa saja untuk kembali\n");
        //berfungsi untuk menunggu pengguna menekan tombol, lalu kembali ke menu utama
        getch();
    } else {
        printf("Keranjang penuh, tidak dapat menambahkan item.\n\n");
    }
}

```

- **void menambahkanKeDalamKeranjang** fungsi ini bertujuan untuk melakukan apa yang terjadi setelah pengguna memilih untuk menambahkan barang ke dalam keranjang. yang pertama melakukan persyaratan bahwa total barang dalam keranjang tidak boleh lebih dari 100. lalu melakukan system("cls") untuk membersihkan terminal supaya terlihat lebih rapih. Setelah itu kita menyalin string menggunakan strcpy dan int harga yang berada di dalam list ke dalam keranjang. Dan menambahkan keranjangCount++, untuk menandakan bahwa total barang dalam keranjang bertambah 1. Lalu diberi notifikasi juga bahwa keranjang yang telah dipilih telah dimasukkan ke dalam keranjang. Dan menggunakan getch(). getch berfungsi supaya pengguna menekan tombol apa saja untuk kembali ke menu.

```

//fungsi untuk menampilkan barang sesuai dengan kategori yang dipilih
void displayKategori(struct list items[], int size, char *kategori) {
    system("cls");
    for (int i = 0; i < size; i++) {
        printf("%d. ", i + 1);
        printList(&items[i]);
    }
    printf("=====\n");
    printf("Masukkan angka (1-%d) jika ingin memasukkan ke dalam keranjang\n", size);
    printf("0. Kembali\n");
    //membuat pilihan pengguna ingin kembali, atau memasukan barang ke dalam keranjang
    scanf("%d", &input);

    if (input == 0) {
        system("cls");
    } else if (input >= 1 && input <= size) {
        menambahkanKeDalamKeranjang(items[input - 1], kategori);
    }
}

```

- **void displayKategori** fungsi ini bertujuan untuk menampilkan barang barang yang terdapat dalam masing-masing kategori. di dalam fungsi ini kita menggunakan pointer untuk mengambil index yang terdapat dalam struct kategori. Lalu kita juga menggunakan & untuk memanggil fungsi printList. karna fungsi printList parameteranya menggunakan pointer. disini juga dibuat scanf lagi untuk meminta pengguna apakah ingin kembali, atau memasukan barang ke dalam keranjang.

```

//fungsi untuk menampilkan kategori barang yang dijual
void tampilkanMenuKategori() {
    system("cls");
    printf("Kategori barang yang kami jual\n");
    for (int i = 0; i < 3; i++) {
        printf("%d. %s\n", i + 1, kategori[i]);
    }
    printf("0. Kembali\n");
    //meminta input untuk menanyakan ingin melihat barang apa
    scanf("%d", &input);

    if (input == 1) {
        displayKategori(listHandphone, 3, "Handphone");
    } else if (input == 2) {
        displayKategori(listLaptop, 3, "Laptop");
    } else if (input == 3) {
        displayKategori(listCPU, 3, "CPU");
    }
}

```

- **void tampilkanMenuKategori()** fungsi ini bertujuan untuk menampilkan kategori dari barang barang yang pengguna inginkan. disini di-print menggunakan for agar mencetak semua kategori yang ada di dalam toko rafi tersebut. Lalu menanyakan penggunaan ingin melihat barang dari kategori apa.


```

//fungsi untuk fitur melihat keranjang
void lihatKeranjang() {
    system("cls");
    printKeranjang();
    printf("tekan tombol apa saja untuk kembali\n");
    getch();
}

//fungsi untuk fitur checkout
void checkout() {
    system("cls");
    checkoutKeranjang();
    printf("\ntekan tombol apa saja untuk kembali\n");
    getch();
}

```

- **void lihatKeranjang()** fungsi ini berfungsi untuk melihat keranjang dari pengguna saat ini, dengan cara memanggil fungsi `printKeranjang()` yang sudah kita buat. lalu menggunakan `getch()` untuk meminta pengguna menekan tombol apa saja untuk kembali ke menu utama.
- **void checkout()** fungsi ini berfungsi untuk melakukan checkout setelah pengguna telah memasukan barang mereka ke dalam keranjang, dengan cara memanggil fungsi `checkoutKeranjang()` yang telah kita buat diatas. setelah fungsi `checkoutKeranjang()` telah dilaksanakan, maka kita menggunakan `getch()` untuk meminta pengguna untuk menekan tombol supaya kembali ke menu utama.

```
//fungsi untuk menu utama
void tampilMenuUtama() {
    system("cls");
    printf(" ##:::##::: '###::: '#####::: '###::: '##: '#####::: '###::: '#####: '###:\n");
    printf(" ##:::##::: '## ##::: ##... ##::: '## ##::: ##::: '##: ##... ##::: '## ##::: ##:::##::: '##::\n");
    printf(" ##:::##::: '##. ##: ##::: ##: '##. ##: ##: '##::: ##::: ##: '##. ##: ##:::##::: '##::\n");
    printf(" ##:::##::: '##::. ##: #####: '##::. ##: #####: #####: '##::. ##: #####: ##:##::\n");
    printf(" ##:::##::: #####: ##... ##: #####: ##. ##::: ##. ##: #####: ##... ##:##::\n");
    printf(" ##:::##::: ##... ##: ##:::##::: ##... ##: ##: ##: ##: ##... ##: ##:##::: '##::\n");
    printf(" #####: ##::: ##: ##:::##::: ##::: ##: ##: ##: ##: ##: ##: ##: ##:::##::: '#####:##::\n");
    printf("Selamat datang di lapak rafi\n");
    printf("1. Melihat list barang\n");
    printf("2. Exit\n");
    printf("3. Lihat keranjang\n");
    printf("4. Checkout keranjang\n");
    scanf("%d", &input);
}
```

- **void tampilkanMenuUtama()** fungsi ini berfungsi sebagai tampilan dari tokonya yaitu bernama lapak rafi, disini saya menggunakan tulisan ascii supaya textnya terlihat lebih bagus dan besar. Lalu menuliskan menu menu yang dapat diakses di dalam lapak rafi. dan meminta pengguna untuk memasukkan input yang mereka inginkan

```
//fungsi untuk memulai program
void jalankanProgram() {
    //dikasih tanda 1 untuk menandakan program masih berjalan
    int menjalankanProgram = 1;
    while (menjalankanProgram == 1) {
        tampilkanMenuUtama();

        if (input == 1) {
            tampilkanMenuKategori();
        } else if (input == 2) {
            system("cls");
            printf("TTTTTTTTTTTTTTTTTTTTT          iiiii          KKKKKKKKKK   KKKKKKKK\n");
            printf("T:.....:K K:.....:K K:.....:K          i:....i          K:.....:K K:.....:K\n");
            printf("T:.....:K K:.....:K K:.....:K          iiiii          K:.....:K K:.....:K\n");
            printf("T:.....:T:.....:T:.....:T          K:.....:K K:.....:K\n");
            printf("TTTTTT T:.....T TTTTTTeeeeeeeeeeee rrrrr rrrrrrrrrr iiiiinii mmmmmmm mmmmmmm aaaaaaaaaa KK:.....K K:.....:KKK aaaaaaaaaa ssssssssss\n");
            printf("T:.....T ee:.....:ee n:.....:rrrrr i:.....i m:.....:m m:.....:m a:.....:a K:.....:K K:.....:K a:.....:a s:.....s ssssssss\n");
            printf("T:.....T e:.....:eeeeee:.....:ee n:.....:rrrrr i:.....i m:.....:m m:.....:m aaaaaaa:.....a K:.....:K K:.....:K aaaaaaa:.....a s:.....s ssssssss\n");
            printf("T:.....T e:.....:e e:.....:ern:.....:rrrrr i:.....i m:.....:m m:.....:m a:.....:a K:.....:K K:.....:K a:.....:a s:.....s ssssssss\n");
            printf("T:.....T e:.....:eeeeee:.....:e n:.....:rrrrr i:.....i m:.....:m m:.....:m aaaaaa:.....a K:.....:K K:.....:K aaaaaa:.....a s:.....s ssssssss\n");
            printf("T:.....T e:.....:eeeeeeeeeeee n:.....:rrrrrrr i:.....i m:.....:m m:.....:m aa:.....:a K:.....:K K:.....:K aa:.....:a s:.....s ssssssss\n");
            printf("T:.....T e:.....:eeeeeeeeeeee n:.....:rrrrr i:.....i m:.....:m m:.....:m a:.....:aaaaa:.....a K:.....:K K:.....:K a:.....:aaaaa:.....a s:.....s ssssssss\n");
            printf("T:.....T e:.....:e n:.....:rrrrr i:.....i m:.....:m m:.....:m m:.....:ma:.....a a:.....:a KK:.....:K K:.....:KKK a:.....:a s:.....s ssssssss\n");
            printf("T:.....TT e:.....:e n:.....:rrrrr i:.....i m:.....:m m:.....:m m:.....:ma:.....a a:.....:a K:.....:K K:.....:K a:.....:a s:.....s ssssssss\n");
            printf("T:.....T e:.....:eeeeeeeeee n:.....:rrrrr i:.....i m:.....:m m:.....:m m:.....:ma:.....:aaaaa:.....a K:.....:K K:.....:K a:.....:a s:.....s ssssssss\n");
            printf("T:.....T ee:.....:e n:.....:rrrrr i:.....i m:.....:m m:.....:m a:.....:aaaa:.....a K:.....:K K:.....:K a:.....:a s:.....s ssssssss\n");
            printf("TTTTTTTTTTT eeeeeeeeeeeeeee rrrrrrrr iiiiiniimmmmm mmmmmmm mmmmmmm aaaaaa:.....a aaaaKKKKKKKKKK KKKKKKKK aaaaaa:.....a aaaa ssssssssssss\n");
            printf("\n");
            //dibah menjadi 0 agar program berhenti
            menjalankanProgram = 0;
        } else if (input == 3) {
            lihatKeranjang();
        } else if (input == 4) {
            checkout();
        } else {
            printf("Pilihan tidak valid\n");
        }
    }
}
```

- **void jalankanProgram()** fungsi ini bertujuan untuk menjalankan program utamanya, yang pertama mendeklarasi variabel menjalankanProgram = 1 yang berarti 1 itu program sedang dijalankan, sedangkan jika nilainya 0, itu berarti program telah berhenti. variabel menjalankanProgram akan bernilai 0 jika user memasukkan input 2 yaitu exit. karena setelah exit maka program akan berhenti dan menulis pesan TerimaKasih dalam bentuk ascii, supaya tulisannya menjadi lebih besar dan bagus. Di fungsi ini kita membuat perkondisian jika input 1,3,atau 4. Maka akan memanggil fungsi yang dibutuhkan, yang sudah dibuat di atas.

```
int main() {  
    //memanggil fungsi untuk menjalankan program  
    jalankanProgram();  
    return 0;  
}
```

- **int main()** Karena dalam program ini sudah membuat seluruh fungsi di atas, jadi didalam int main kita hanya perlu memanggil fungsi jalankanProgram(). Maka program diatas akan berjalan sesuai prosedurnya.

Analisis Program (saat dijalankan)

```
#####'#####:'#####:'#####:'#####:'#####'#####
##:'## ##:: ##... ##:'## ##:: ##:'##: ##... ##:'## ##:: ##..... ##:
##:'##. ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##:
##:'##::. ##: #####:'##::. ##: #####: ##: #####:'##::. ##: #####: ##:
##: #####: ##.....: #####: ##. ##: ##. ##: #####: ##.....: ##:
##: ##... ##: ##: ##: ##... ##: ##: ##: ##: ##... ##: ##: ##: ##:
#####: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##: ##:
Selamat datang di lapak rafi
1. Melihat list barang
2. Exit
3. Lihat keranjang
4. Checkout keranjang
```

- Saat kode dijalankan, ini adalah menu awal dari tampilan toko yang bernama lapak rafi. Di tampilan awal, terdapat pilihan seperti melihat list barang, exit, lihat keranjang, dan checkout keranjang.
- Disini user tidak bisa mengetik angka lain selain 1-4, karena jika user menginputkan selain 1-4 maka akan kembali ke tampilan menu seperti ini. Disini kita langsung masuk ke contoh saja. Semisal user memasukan input 2 yaitu exit, maka akan terjadi seperti ini.

[illegible]

- Screenshoot diatas adalah hal yang terjadi apabila user memilih input nomor 2 yaitu exit, maka program akan berhenti berjalan. Setelah itu akan menuliskan pesan “Terimakasih”.

```
Keranjang Anda kosong.
```

```
tekan tombol apa saja untuk kembali
```

```
█
```

- apabila user menginputkan angka 3 yaitu cek keranjang pada menu awal tadi. Maka akan bertuliskan “Keranjang Anda kosong.” karena user baru saja masuk ke dalam program, dan belum memasukan barang apa apa ke dalam keranjang.

```
Keranjang Anda kosong.
```

```
tekan tombol apa saja untuk kembali
```

```
█
```

- apabila user menginputkan angka 4 yaitu adalah checkout keranjang pada menu awal tadi. Maka tampilannya akan sama seperti saat cek keranjang. Hal ini terjadi dikarenakan user belum memasukan barang ke dalam keranjang, yang mengakibatkan keranjangnya masih kosong.

```
Kategori barang yang kami jual
```

```
1. Handphone
```

```
2. Laptop
```

```
3. CPU
```

```
0. Kembali
```

```
█
```

- apabila user menginputkan angka 1, user akan langsung diarahkan ke menu kategori. Di dalam menu kategori ini bertuliskan kategori dari barang yang dijual seperti Handphone, Laptop, dan CPU.
- Contoh semisal saya di sini memasukan angka 1,2, atau 3 maka akan mengeluarkan deskripsi deskripsi barang dari kategori yang dipilih. Berikut adalah contoh deskripsi produk dari kategori yang dipilih

```

1. IPHONE 15 PRO
  Harga      : 21 juta
  Stok tersedia : 5

2. SAMSUNG GALAXY S24 ULTRA
  Harga      : 20 juta
  Stok tersedia : 8

3. HUAWEI MATE 60 PRO
  Harga      : 23 juta
  Stok tersedia : 11

=====
Masukkan angka (1-3) jika ingin memasukkan ke dalam keranjang
0. Kembali

```

- ini adalah tampilan yang akan terjadi apabila user menginputkan angka 1. Handphone. Lalu program akan menampilkan deskripsi dari produk produk handphone yang tersedia, seperti nama dari barangnya, lalu ada harganya, dan ada informasi mengenai stok yang tersedia.
- Di dalam tampilan ini juga user bisa memasukan barang tersebut ke dalam keranjang. Semisal saya menginputkan angka 1 yaitu "IPHONE 15 PRO" maka akan menampilkan tampilan seperti ini

```

IPHONE 15 PRO telah dimasukkan ke dalam keranjang.

Tekan apa saja untuk kembali

```

- Ini adalah tampilan setelah user memasukan barang ke keranjang. Setelah ini user hanya perlu menekan tombol apapun, maka user akan dikembalikan ke menu utama.
- Sekarang kita lanjut ke menu apabila user memilih kategori Laptop.

```

1. ASUS VIVOBOK
  Harga      : 10 juta
  Stok tersedia : 3

2. HP PAVILION
  Harga      : 14 juta
  Stok tersedia : 7

3. THINKPAD
  Harga      : 7 juta
  Stok tersedia : 1

=====
Masukkan angka (1-3) jika ingin memasukkan ke dalam keranjang
0. Kembali

```

- ini adalah tampilan yang akan terjadi apabila user menginputkan angka 2. Handphone. Lalu program akan menampilkan deskripsi dari produk-produk laptop yang tersedia, seperti nama dari barangnya, lalu ada harganya, dan ada informasi mengenai stok yang tersedia.
- Di dalam tampilan ini juga user bisa memasukan barang tersebut ke dalam keranjang. Semisal saya menginputkan angka 2 yaitu "HP PAVILION" maka akan menampilkan tampilan seperti ini

```

HP PAVILION telah dimasukkan ke dalam keranjang.

Tekan apa saja untuk kembali

```

- Ini adalah tampilan setelah user memasukan barang ke keranjang. Setelah ini user hanya perlu menekan tombol apapun, maka user akan dikembalikan ke menu utama.
- Sekarang kita lanjut ke menu apabila user memilih kategori CPU.

```
1. INTEL CORE I9-7920X
  Harga      : 17 juta
  Stok tersedia : 6
```

```
2. INTEL CORE I7-14700K
  Harga      : 7 juta
  Stok tersedia : 3
```

```
3. INTEL CORE I5-10400F
  Harga      : 2 juta
  Stok tersedia : 1
```

```
=====
Masukkan angka (1-3) jika ingin memasukkan ke dalam keranjang
0. Kembali
█
```

- ini adalah tampilan yang akan terjadi apabila user menginputkan angka 3.CPU. Lalu program akan menampilkan deskripsi dari produk-produk CPU yang tersedia, seperti nama dari barangnya, lalu ada harganya, dan ada informasi mengenai stok yang tersedia.
- Di dalam tampilan ini juga user bisa memasukan barang tersebut ke dalam keranjang. Semisal saya menginputkan angka 3 yaitu “INTEL CORE I5-10400F” maka akan menampilkan tampilan seperti ini

```
INTEL CORE I5-10400F telah dimasukkan ke dalam keranjang.
```

```
Tekan apa saja untuk kembali
█
```

- Ini adalah tampilan setelah user memasukan barang ke keranjang. Setelah ini user hanya perlu menekan tombol apapun, maka user akan dikembalikan ke menu utama.
- Sekarang kita kembali ke menu cek keranjang setelah kita memasukan barang ke dalam keranjang

```
Keranjang Anda:
```

```
1. IPHONE 15 PRO - 21 juta
2. HP PAVILION - 14 juta
3. INTEL CORE I5-10400F - 2 juta
```

```
tekan tombol apa saja untuk kembali
█
```

- Ini tampilan setelah user memasukan barang ke dalam keranjang. Maka sekarang jika masuk ke menu keranjang tulisannya tidak kosong lagi, melainkan berisi barang barang yang user sudah masukan ke dalam keranjang.

- Sekarang kita melakukan menu checkout.

```
Checkout Keranjang:
1. IPHONE 15 PRO - 21 juta
2. HP PAVILION - 14 juta
3. INTEL CORE I5-10400F - 2 juta
=====
Total pembayaran: 37 juta

Terimakasih telah berbelanja di toko kami!

Keranjang akan dikosongkan

tekan tombol apa saja untuk kembali
█
```

- Ini adalah tampilan checkout, di tampilan ini program akan menampilkan barang barang yang user sudah masukan ke dalam keranjang. Dan menuliskan total harga dari barang barang user
- Setelah user melakukan checkout maka keranjang akan dikosongkan kembali.

Tabel pengimplementasian materi :

N o.	MATE RI	POI N	PERAN DALAM PROGRAM
1	Tipe Data	5	<pre>//untuk setiap input yang dilakukan int input; // Counter untuk jumlah item di keranjang int keranjangCount = 0; // Membuat struct untuk menyimpan nama dan harga struct keranjang { char nama[30]; int harga; }; struct list { char nama[30]; int harga; // membuat struct, jadi tiap barang itu mempunyai nam int stock; };</pre>

2	I/O	5	<pre>printf("=====\n"); printf("Masukkan angka (1-%d)\n", jumlahBarang); printf("0. Kembali\n"); //membuat pilihan pengguna input scanf("%d", &input); printf("0. Kembali\n"); //meminta input untuk menanyakan scanf("%d", &input); printf("Selamat datang di lapak rafi\n"); printf("1. Melihat list barang\n"); printf("2. Exit\n"); printf("3. Lihat keranjang\n"); printf("4. Checkout keranjang\n"); scanf("%d", &input);</pre>
3	Looping	15	<pre>for (int i = 0; i < size; i++) { printf("%d. ", i + 1); printList(&items[i]); } while (menjalankanProgram == 1) { tampilkanMenuUtama(); for (int i = 0; i < 3; i++) { printf("%d. %s\n", i + 1, kategori[i]); } //nge list setiap keranjang setelah user memasukan barang kedalam for (int i = 0; i < keranjangCount; i++) { printf("%d. %s - %d juta\n", i + 1, listKeranjang[i].nama, listKeranjang[i].harga); } }</pre>

4	Conditio nal	15	<pre> if (keranjangCount == 0) { printf("Keranjang Anda kosong.\n\n"); } else { printf("Keranjang Anda:\n"); //nge list setiap keranjang setelah use for (int i = 0; i < keranjangCount; i++) { printf("%d. %s - %d juta\n", i + 1, listKeranjang[i].nama, listKeranjang[i].harga); } printf("\n"); } if (index < 0) { return 0; // fungsinya akan berhenti jika index nya habi } if (keranjangCount == 0) { printf("Keranjang Anda kosong.\n\n"); } else { printf("Checkout Keranjang:\n"); if (input == 0) { system("cls"); } else if (input >= 1 && input <= size) { menambahkanKeDalamKeranjang(items[input - 1], kategori); } if (input == 1) { displayKategori(listHandphone, 3, "Handphone"); } else if (input == 2) { displayKategori(listLaptop, 3, "Laptop"); } else if (input == 3) { displayKategori(listCPU, 3, "CPU"); } } </pre>
5	Rekursi	5	<pre> // Fungsi rekursif untuk menghitung total harga keranjang int hitungTotal(int index) { if (index < 0) { // fungsinya akan berhenti jika index nya habis return 0; } //memanggil kembali fungsi, mengurangi parameter fungsi dengan 1 return listKeranjang[index].harga + hitungTotal(index - 1); } </pre>

6	Array	15	<pre>//menyimpan kategori barang yang dijual melalui array kategori char kategori[3][20] = {"Handphone", "Laptop", "CPU"}; //membuat array list Handphone dari struct list yang berisi r struct list listHandphone[3] = { {"IPHONE 15 PRO", 21, 5}, {"SAMSUNG GALAXY S24 ULTRA", 20, 8}, {"HUAWEI MATE 60 PRO", 23, 11} }; //membuat array list Laptop dari struct list yang berisi na struct list listLaptop[3] = { {"ASUS VIVOBOK", 10, 3}, {"HP PAVILION", 14, 7}, {"THINKPAD", 7, 1} }; //membuat array list CPU dari struct list yang berisi nama h struct list listCPU[3] = { {"INTEL CORE I9-7920X", 17, 6}, {"INTEL CORE I7-14700K", 7, 3}, {"INTEL CORE I5-10400F", 2, 1} };</pre>
7	Pointer	10	<pre>void printList(struct list *point) { // print barang dalam masing-masing kategori printf("%s\n", point->nama); printf(" Harga : %d juta\n", point->harga); printf(" Stok tersedia : %d\n\n", point->stock); } printList(&items[i]);</pre>
8	Function	10	<pre>//fungsi print barang untuk kategori yang diminta void printList(struct list *point) { // print barang dalam masing-masing kategori printf("%s\n", point->nama); printf(" Harga : %d juta\n", point->harga); printf(" Stok tersedia : %d\n\n", point->stock); }</pre>

```
//membuat fungsi untuk pengecekan keranjang
void printKeranjang() {
    //membuat pengkondisian jika keranjang masih kosong
    if (keranjangCount == 0) {
        printf("Keranjang Anda kosong.\n\n");
    } else {
        printf("Keranjang Anda:\n");
        //nge list setiap keranjang setelah user memasukkan barang kedalam
        for (int i = 0; i < keranjangCount; i++) {
            printf("%d. %s - %d juta\n", i + 1, listKeranjang[i].nama, listKeranjang[i].harga);
        }
        printf("\n");
    }
}
```

```
// Fungsi rekursif untuk menghitung total harga keranjang
int hitungTotal(int index) {
    if (index < 0) {
        // fungsinya akan berhenti jika index nya habis
        return 0;
    }
    //memanggil kembali fungsi, mengurangi parameter fungsi dengan 1
    return listKeranjang[index].harga + hitungTotal(index - 1);
}
```

```
//membuat fungsi untuk checkout keranjang
void checkoutKeranjang() {
    //membuat kondisi jika keranjangnya masih 0, maka print keranjang kosong
    if (keranjangCount == 0) {
        printf("Keranjang Anda kosong.\n\n");
    } else {
        printf("Checkout Keranjang:\n");
        for (int i = 0; i < keranjangCount; i++) {
            printf("%d. %s - %d juta\n", i + 1, listKeranjang[i].nama, listKeranjang[i].harga);
        }
        // Menghitung total menggunakan fungsi rekursif
        int total = hitungTotal(keranjangCount - 1);
        printf("=====\n");
        printf("Total pembayaran: %d juta\n\n", total);
        printf("Terimakasih telah berbelanja di toko kami!\n\n");
        printf("Keranjang akan dikosongkan\n");
        keranjangCount = 0; // untuk mengosongkan keranjang setelah checkout
    }
}
```

```
//fungsi setelah pengguna memasukan barang ke dalam keranjang
void menambahkanKeDalamKeranjang(struct list item, char *kategori)
{
    // Batas maksimal 100 barang di dalam keranjang
    if (keranjangCount < 100) {
        system("cls");
        strcpy(listKeranjang[keranjangCount].nama, item.nama);
        listKeranjang[keranjangCount].harga = item.harga;
        keranjangCount++;
        printf("%s telah dimasukkan ke dalam keranjang.\n\n", item.nama);
        printf("Tekan apa saja untuk kembali\n");
        //berfungsi untuk menunggu pengguna menekan tombol, lalu k
        getch();
    } else {
        printf("Keranjang penuh, tidak dapat menambahkan item.\n\n");
    }
}
}
```

```
//fungsi untuk menampilkan barang sesuai dengan kategori yang dip
void displayKategori(struct list items[], int size, char *kategori)
{
    system("cls");
    for (int i = 0; i < size; i++) {
        printf("%d. ", i + 1);
        printList(&items[i]);
    }
    printf("=====\n");
    printf("Masukkan angka (1-%d) jika ingin memasukkan ke dalam kategori\n");
    printf("0. Kembali\n");
    //membuat pilihan pengguna ingin kembali, atau memasukan barang
    scanf("%d", &input);

    if (input == 0) {
        system("cls");
    } else if (input >= 1 && input <= size) {
        menambahkanKeDalamKeranjang(items[input - 1], kategori);
    }
}
}
```

			<pre>//fungsi untuk menampilkan kategori barang yang dijual void tampilkanMenuKategori() { system("cls"); printf("Kategori barang yang kami jual\n"); for (int i = 0; i < 3; i++) { printf("%d. %s\n", i + 1, kategori[i]); } printf("0. Kembali\n"); //meminta input untuk menanyakan ingin melihat barang apa scanf("%d", &input); if (input == 1) { displayKategori(listHandphone, 3, "Handphone"); } else if (input == 2) { displayKategori(listLaptop, 3, "Laptop"); } else if (input == 3) { displayKategori(listCPU, 3, "CPU"); } }</pre>
--	--	--	--

			<pre>//fungsi untuk memulai program void jalankanProgram() { //dikasih tanda 1 untuk menandakan program masih berjalan int menjalankanProgram = 1; while (menjalankanProgram == 1) { tampilkanMenuUtama(); if (input == 1) { tampilkanMenuKategori(); } else if (input == 2) { system("cls"); printf("TTTTTTTTTTTTTTTTTTTTTTT\n"); printf("T::::::::::::::::::T\n"); printf("T::::::::::::::::::T\n"); printf("T:::::TT::::::::::TT:::::T\n"); printf("TTTTTT T:::::T TTTTTTeeeeeeeeeeeeee r\n"); printf(" T:::::T ee::::::::::::ee r\n"); printf(" T:::::T e::::::::eeeeeeeeer\n"); printf(" T:::::T e:::::e e:::::er\n"); printf(" T:::::T e::::::::eeeeeeeee i\n"); printf(" T:::::T e::::::::::::e i\n"); printf(" T:::::T e:::::::::eeeee\n");</pre>
Total : 80			

NB : +10 poin kerapian dan kompleksitas, +10 poin kelengkapan laporan