Latihan Kode Program C++: Membuat Aplikasi Kasir Sederhana

Dalam tutorial latihan algoritma bahasa C++ kali ini kita akan bahas cara membuat **aplikasi kasir sederhana**. Latihan ini akan menguji pemahaman seputar penggunaan variabel, tipe data, perulangan dan kondisi if-else.

Soal Membuat Aplikasi Kasir Sederhana

Buatlah kode program C++ untuk aplikasi kasir dengan ketentuan sebagai berikut:

1. User bisa menentukan jumlah jenis barang yang akan dibeli (maksimum 10 barang).
2. Untuk setiap jenis barang, user juga bisa menginput nama barang, harga satuan, dan jumlah barang.
3. Program akan menampilkan struk total belanjaan.
4. User akan menginput uang untuk membayar. Uang ini harus lebih besar dari total belanjaan.
5. Program menampilkan sisa uang kembalian (jika ada).

Berikut contoh hasil akhir yang diinginkan :

|  |
| --- |
| ##  Program C++ Aplikasi Kasir  ##  ==================================    # Selamat datang di Mini Market ILKOOM #    Masukkan jumlah barang yang ingin dibeli: 3    Barang ke-1  Nama barang  : Sabun mandi  Harga satuan : 3000  Jumlah Sabun mandi yang dibeli: 12    Barang ke-2  Nama barang  : Deterjen  Harga satuan : 25000  Jumlah Deterjen yang dibeli: 7    Barang ke-3  Nama barang  : Gula pasir  Harga satuan : 18000  Jumlah Gula pasir yang dibeli: 4      ----------------------------------------------------------  ####----         Struk Mini Market ILKOOM         ----####  ----------------------------------------------------------  No   Barang    Jumlah     Harga Satuan     Sub Total  1 Sabun mandi    12              3000          36000  2 Deterjen        7             25000         175000  3 Gula pasir      4             18000          72000  ----------------------------------------------------------  Total harga : Rp.283000    Jumlah Bayar: Rp.300000    Kembalian   : Rp.17000    ----------------------------------------------------------  ####   Terimakasih sudah berbelanja, datang kembali   ####  ---------------------------------------------------------- |
|  |

Tips Membuat Kode Program Aplikasi Kasir Sederhana

Soal program aplikasi kasir ini menguji banyak hal sekaligus, diantaranya pemahaman seputar penggunaan variabel, tipe data, perulangan dan kondisi if-else.

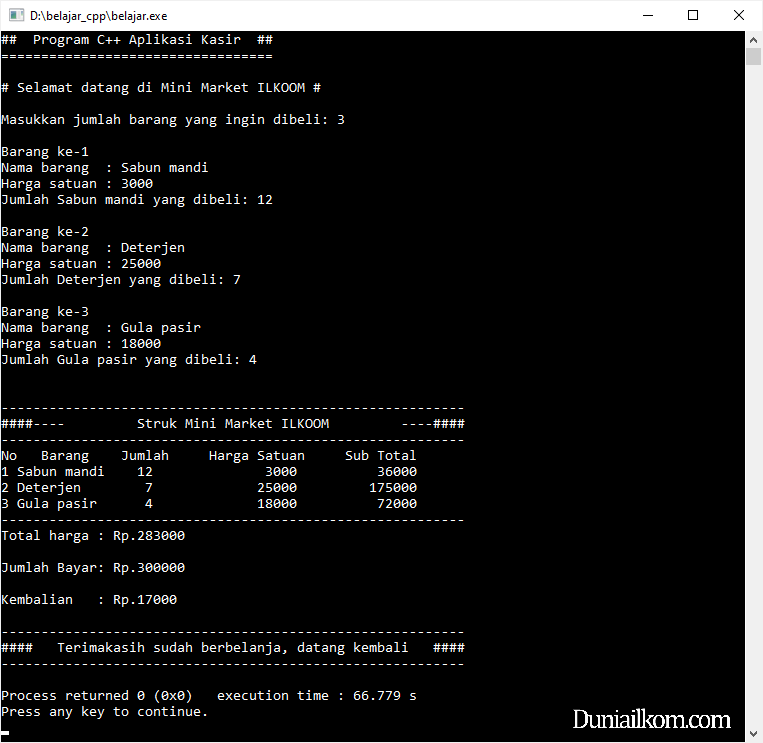
Berikut tutorial pendahuluan yang bisa diikuti:

Silahkan coba rancang sebentar kode program ini.

Kode Program C++ Membuat Aplikasi Kasir Sederhana

Berikut salah satu solusi dari soal membuat aplikasi kasir sederhana menggunakan bahasa pemrograman C++:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77 | #include <iostream>  #include <iomanip>  **using** **namespace** std;    **int** main() {    cout << "##  Program C++ Aplikasi Kasir  ##" << endl;    cout << "==================================" << endl;    cout << endl;    **const** **int** MAX\_BARANG = 10;    string namaBarang[MAX\_BARANG];  **long** hargaBarang[MAX\_BARANG];  **int** jumlahBarang[MAX\_BARANG];  **long** jumlahTotalBarang;  **long** totalHarga = 0;  **long** jumlahBayar = 0;      cout << "# Selamat datang di Mini Market ILKOOM #" << endl;    cout << endl;      cout << "Masukkan jumlah barang yang ingin dibeli: ";    cin >> jumlahTotalBarang;    cout << endl;    **if** (jumlahTotalBarang <= 0 || jumlahTotalBarang > MAX\_BARANG) {      cout << "Jumlah barang tidak valid!" << endl;  **return** 0;    }    **for** (**int** i = 0; i < jumlahTotalBarang; i++) {      cout << "Barang ke-" << i+1 << endl;      cout << "Nama barang  : ";      getline(cin >> ws,namaBarang[i]);      cout << "Harga satuan : ";      cin  >> hargaBarang[i];      cout << "Jumlah " << namaBarang[i] << " yang dibeli: ";      cin  >> jumlahBarang[i];      cout << endl;    }    cout << endl;      cout<<"----------------------------------------------------------"<<endl;    cout<<"####----         Struk Mini Market ILKOOM         ----####"<<endl;    cout<<"----------------------------------------------------------"<<endl;    cout<<"No   Barang    Jumlah     Harga Satuan     Sub Total"<<endl;  **for** (**int** i = 0; i < jumlahTotalBarang; i++){      cout << setw(1) << i+1 << " ";      cout << left << setw(12) << namaBarang[i];      cout << right << setw(5) << jumlahBarang[i];      cout << setw(18) << hargaBarang[i];      cout << setw(15) << jumlahBarang[i]\*hargaBarang[i];      cout << endl;      totalHarga = totalHarga + (jumlahBarang[i]\*hargaBarang[i]);    }    cout<<"----------------------------------------------------------"<<endl;      cout << "Total harga : Rp." << totalHarga << endl << endl;      cout << "Jumlah Bayar: Rp.";    cin  >> jumlahBayar;    cout << endl;    **while** (jumlahBayar - totalHarga < 0) {      cout << "Maaf, uang anda kurang. Mohon lakukan ulang pembayaran"<<endl;      cout << "Jumlah Bayar: Rp.";      cin  >> jumlahBayar;      cout << endl;    }      cout << "Kembalian   : Rp." << jumlahBayar - totalHarga << endl;    cout << endl;    cout<<"----------------------------------------------------------"<<endl;    cout<<"####   Terimakasih sudah berbelanja, datang kembali   ####"<<endl;    cout<<"----------------------------------------------------------"<<endl;    **return** 0;  } |

[](https://www.duniailkom.com/wp-content/uploads/2023/09/Latihan-Kode-Program-Cpp-Membuat-Aplikasi-Kasir-Sederhana.png)

Kode program kita cukup panjang karena aplikasi kasir ini butuh banyak penerapan konsep dasar programming.

Di awal kode program terdapat perintah #include <iostream> yang dipakai untuk mengimpor library **iostream** . Tujuannya agar kita bisa menggunakan perintah **cin** dan **cout** sepanjang kode program.

Lalu di baris 2 di ikuti perintah #include <iomanip> untuk mengimpor library **iomanip** agar bisa menggunakan manipulator **setw()** untuk mengatur lebar teks pada saat ditampilkan.

Di baris 3, perintah using namespace std berguna untuk menyederhanakan penulisan program. Tujuannya agar kita tidak perlu menuliskan **std::** di setiap perintah **cout** dan **cin**.

Masuk ke kode program utama, setelah menulis judul program di baris 6-8, di baris 10-16 terdapat deklarasi beberapa konstanta dan variabel. Berikut rinciannya:

* const int MAX\_BARANG = 10, dipakai untuk menentukan ukuran maksimum dari array **namaBarang**, **hargaBarang**, dan **jumlahBarang**. Sesuai dengan syarat soal, inilah batasan jumlah jenis barang yang bisa di input user.
* string namaBarang[MAX\_BARANG], adalah array untuk menyimpan nama barang yang dibeli.
* long hargaBarang[MAX\_BARANG], adalah array untuk menyimpan harga satuan barang yang dibeli.
* int jumlahBarang[MAX\_BARANG], adalah array untuk menyimpan jumlah barang yang dibeli.
* long jumlahTotalBarang, adalah variabel untuk menyimpan jumlah total barang yang ingin dibeli.
* long totalHarga = 0, adalah variabel untuk menyimpan total harga dari semua barang yang dibeli.
* long jumlahBayar = 0, adalah variabel untuk menyimpan jumlah pembayaran dari pelanggan.

Beberapa variabel di set dengan tipe data **long** agar bisa menampung angka yang lebih besar daripada int. Di beberapa compiler, tipe data int hanya bisa menampung hingga 32,767 sedangkan tipe data long sanggup menampung angka hingga 2,147,483,647.

Ketika dijalankan, user akan ditanya berapa total jenis barang yang akan di input. Ini dibaca oleh perintah cin >> jumlahTotalBarang di baris 22.

Kondisi **if** di baris 25 dipakai untuk memastikan jumlah barang yang di input tidak kurang dari 0 dan tidak lebih dari 10. Jika ini terjadi, kode program akan langsung berhenti dengan menjalankan perintah return 0.

Selanjutnya masuk ke perulangan **for** di baris 30-39. Disinilah pembacaan data dilakukan. Perulangan ini akan diulang sebanyak **jumlahTotalBarang**.

Dalam setiap iterasi, baca data nama barang, harga satuan, dan jumlah barang yang dibeli. Setiap data akan disimpan ke dalam array.

Perintah yang dipakai untuk membaca nama barang adalah getline(cin >> ws,namaBarang[i]) di baris 33. Kita memakai perintah **getline** dan bukan **cin** agar bisa membaca inputan nama barang yang mengandung spasi.

Setelah semua data selesai dibaca, tampilkan struk pembelian dengan perintah cout dan juga perulangan for di baris 42-55.

Tambahan perintah **setw()** dipakai untuk mengatur lebar output agar lebih rapi.

Di akhir perulangan for, tepatnya di baris 53, total harga di hitung dengan perintah totalHarga = totalHarga + (jumlahBarang[i]\*hargaBarang[i]), yang hasilnya ditampilkan dengan perintah cout di baris 59.

Setelah itu user diminta menginput jumlah uang yang dipakai untuk membayar dengan perintah cin >> jumlahBayar di baris 60.

Disini, ada kemungkinan user menginput jumlah bayar yang kurang dari total harga. Karena itulah ada perulangan while di baris 63-68. Selama uang yang di input user kurang dari total harga, tampilkan pesan "*Maaf, uang anda kurang. Mohon lakukan ulang pembayaran*".

Perulangan while ini hanya akan berhenti jika jumlah bayar sudah lebih dari total harga.

Terakhir, tampilkan jumlah uang kembalian di baris 70.

Demikian pembahasan tentang soal latihan membuat aplikasi kasir di bahasa pemrograman C++.

Soal ini masih bisa di tambah dengan berbagai fitur lain, misalnya jika total pembelian lebih dari nominal tertentu, bisa mendapat potongan diskon.

Kode programnya memang akan jadi lebih panjang, tapi sekaligus lebih menantang. Silahkan jika ingin dibuat untuk menguji pemahaman basic programming bahasa pemrograman C++.