**LAPORAN AKHIR PRAKTIKUM**

**MATA KULIAH**

**MODUL 2**

**JUDUL MODUL : PEMILIHAN KONDISI**



**Nama : Kurnia Fajar Rahyudi Putra**

**No. BP : 2211512013**

**Hari/Tanggal : Senin/ 15 Mei 2023**

**Shift : 1**

**Dosen : Dodon Yendri, M. Kom.**

**LABORATORIUM KOMPUTER DAN JARINGAN**

**DEPARTEMEN TEKNIK KOMPUTER**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG**

**2023**

**BAB I**

**PENDAHULUAN**

* 1. Tujuan

1. Mahasiswa dapat memahami dan dapat mempraktekan struktur dasar pemilihan kondisi IF THEN.
2. Mahasiswa dapat memahami dan dapat mempraktekan struktur dasar pemilihan kondisi IF THEN ELSE.
3. Mahasiswa dapat memahami dan dapat mempraktekan struktur dasar pemilihan kondisi IF THEN IF ELSE.
4. Mahasiswa dapat memahami dan dapat mempraktekan struktur dasar pemilihan kondisi NESTED IF.
5. Mahasiswa dapat memahami dan dapat mempraktekan struktur dasar pemilihan kondisi SWITCH CASE.
   1. Landasan Teori
6. Pemilihan atau Seleksi Kondisi

Dalam kehidupan sehari-hari, kadang kala kita disudutkan pada beberapa pilihan atau harus memilih salah satu dari opsi yang ada dimana pilihan-pilihan tersebut didasarkan atas kondisi tertentu. Dengan kata lain, pilihan tersebut hanya dapat dilakukan apabila kondisi telah terpenuhi. Sebagai contoh, perhatikan statemen di bawah ini.

*“Jika Gunawan memiliki banyak uang, maka ia akan membeli mobil mewah”*

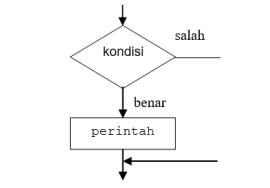
Pada statemen di atas, Gunawan akan dapat membeli mobil mewah hanya apabila ia memiliki banyak uang. Hal ini berarti apabila ternyata Gunawan tidak memiliki banyak uang (kondisi tidak terpenuhi), maka Gunawan pun tidak akan pernah membeli mobil mewah. Begitupun di dalam bahasa pemrograman, kita juga dapat melakukan pemilihan statemen yang akan dieksekusi, yaitu dengan melakukan pengecekan terhadap kondisi tertentu yang didefinisikan. Adapun kondisi yang dimaksud di dalam program tidak lain adalah suatu ekspresi. Dalam bahasa C, pemilihan statemen dapat dilakukan melalui dua buah cara, yaitu dengan menggunakan statemen if dan statemen switch.

1. Fungsi Pemilihan Kondisi

Penyeleksian kondisi digunakan untuk mengarahkan perjalanan suatu proses. Penyeleksian kondisi dapat diibaratkan sebagai katup atau kran yang mengatur jalannya air. Bila katup terbuka maka air akan mengalir dan sebaliknya bila katup tertutup air tidak akan mengalir atau akan mengalir melalui tempat lain. Fungsi penyeleksian kondisi penting artinya dalam penyusunan bahasa C, terutama untuk program yang kompleks.

1. Statement IF THEN

Pernyataan IF mempunyai pengertian, “Jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan”. Dari pengertian tersebut dapat dilihat dari diagram alir berikut:



Bentuk umum dari statement IF THEN sebagai berikut:

if (*kondisi*) {

pernyataan;

……

}

Pernyataan dilaksanakan jika dan hanya jika kondisi yang diinginkan terpenuhi, jika tidak program tidak memberikan hasil apa-apa.

1. Statement IF THEN ELSE

Dalam struktur kondisi IF THEN ELSE minimal terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan. Bentuk umumnya adalah sebagai berikut :

if (*kondisi*) {

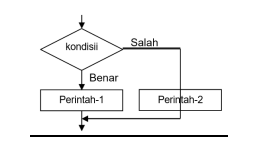
pernyataan-1;

} else {

pernyataan-2;

}

Pernyataan-1 dilaksanakan jika dan hanya jika kondisi yang diinginkan terpenuhi, jika tidak, lakukan pernyataan-2. Jika tidak mempergunakan pernyataan ELSE program tidak akan error, namun jika mempergunakan pernyataan ELSE tanpa didahului pernyataan IF, maka program akan error. Bentuk diagram flowchartnya dapat digambarkan sebagai berikut :



1. Statement IF THEN IF ELSE

Struktur percabangan IF lebih dari dua kondisi / IF-ELSE majemuk sebenarnya merupakan pengembangan dari struktur IF dua kondisi, karena percabangan ini akan menambahkan (menyisipkan) satu atau lebih kondisi ke dalamnya. Artinya dalam pernyataan IF lebih dari dua kondisi: *“Jika kondisi1 bernilai benar, maka perintah-1 akan dikerjakan, jika kondisi1 salah maka akan mengecek kondisi2 dan jika kondisi2 bernilai benar, maka perintah-2 akan dikerjakan, jika kondisi2 juga salah maka akan mengecek kondisi berikutnya dan akan mengerjakan perintah pada struktur blok yang memiliki kondisi bernilai benar”*. Seandainya tidak ada kondisi yang bernilai benar, maka akan mengerjakan perintah yang berada pada struktur blok else. Bentuk umum dari statement IF THEN IF ELSE sebagai berikut:

if (*kondisi*) {

pernyataan;

} else if (*kondisi*) {

pernyataan;

} else {

pernyataan;

}

1. Statement NESTED IF (IF Bersarang)

Dalam struktur IF kondisi, dikenalkan juga apa yang disebut dengan bentuk IF bersarang atau sering disebut *NESTED IF* . Bentuk ini diguanakan untuk pemilihan pernyataan dari sejumlah kemungkinan tindakan. Penyeleksian dilakukan secara bertingkat. Misalnya untuk menentukan kode hari, atau kode bulan dan sebagainya, bisa memanfaatkan pernyataan bentuk IF di dalam IF. Bentuk umunya sebagai berikut :

if (*kondisi*) {

if (*kondisi*)

pernyataan;

else

pernyataan;

else {

if (*kondisi*)

pernyataan;

else

pernyataan;

}

1. Statement SWITCH CASE

Bentuk dari *SWITCH-CASE* merupakan pernyataan yang dirancangan khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternatif penyelesaian. Pernyataan *SWITCH-CASE* ini memiliki kegunaan sama seperti IF-ELSE bertingkat, tetapi penggunaannya untuk memeriksa data yang bertipe karakter atau integer.

Struktur ini akan melaksanakan salah satu dari beberapa pernyataan ‘case’ tergantung nilai kondisi yang ada di dalam switch. Selanjutnya proses diteruskan hingga ditemukan pernyataan ‘break’. Jika tidak ada nilai pada case yang sesuai dengan nilai kondisi, maka proses akan diteruskan kepada pernyataan yang ada di bawah ‘default’. Bentuk penulisan perintah ini sebagai berikut:

switch (*variabel*) {

case 1: pernyataan-1;

break;

case 2: pernyataan-2;

break;

…….

case n: pernyataan-n;

break;

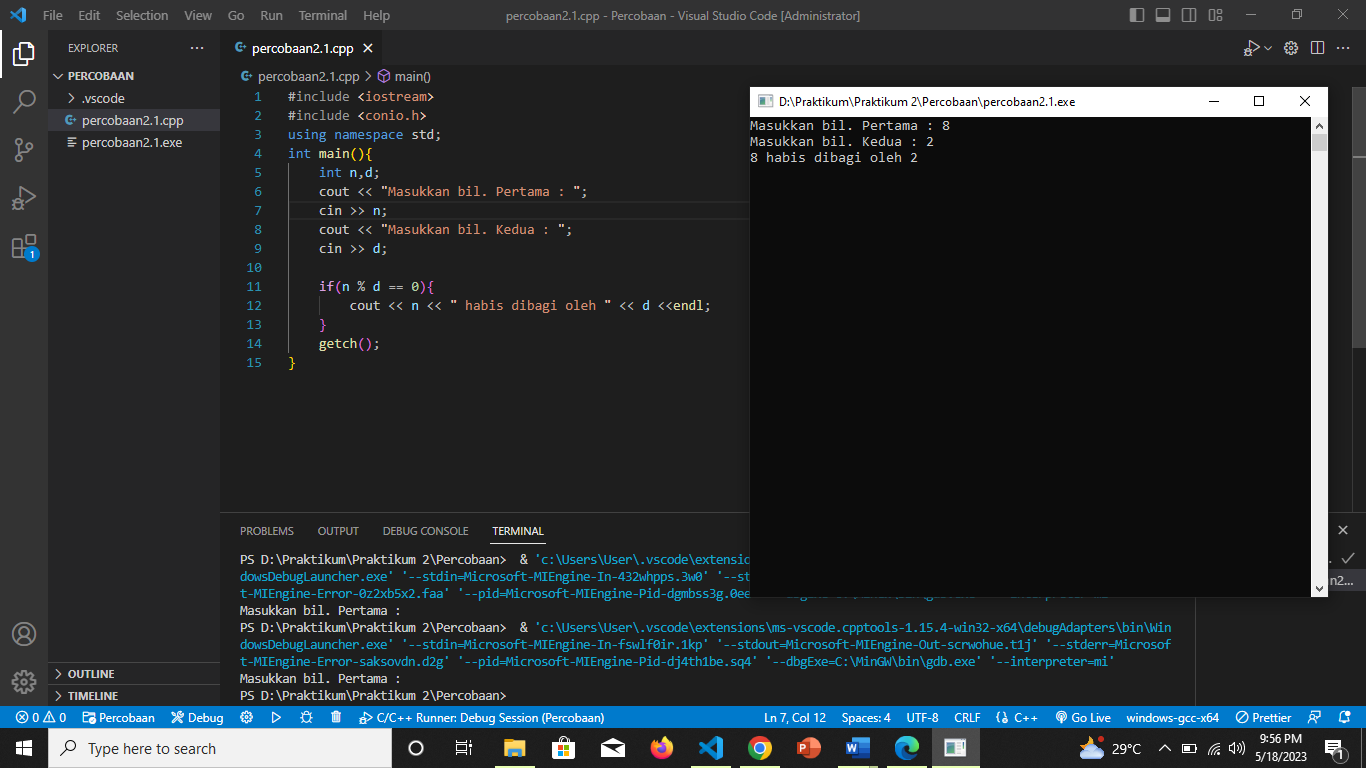
default: pernyataan-m;

break;

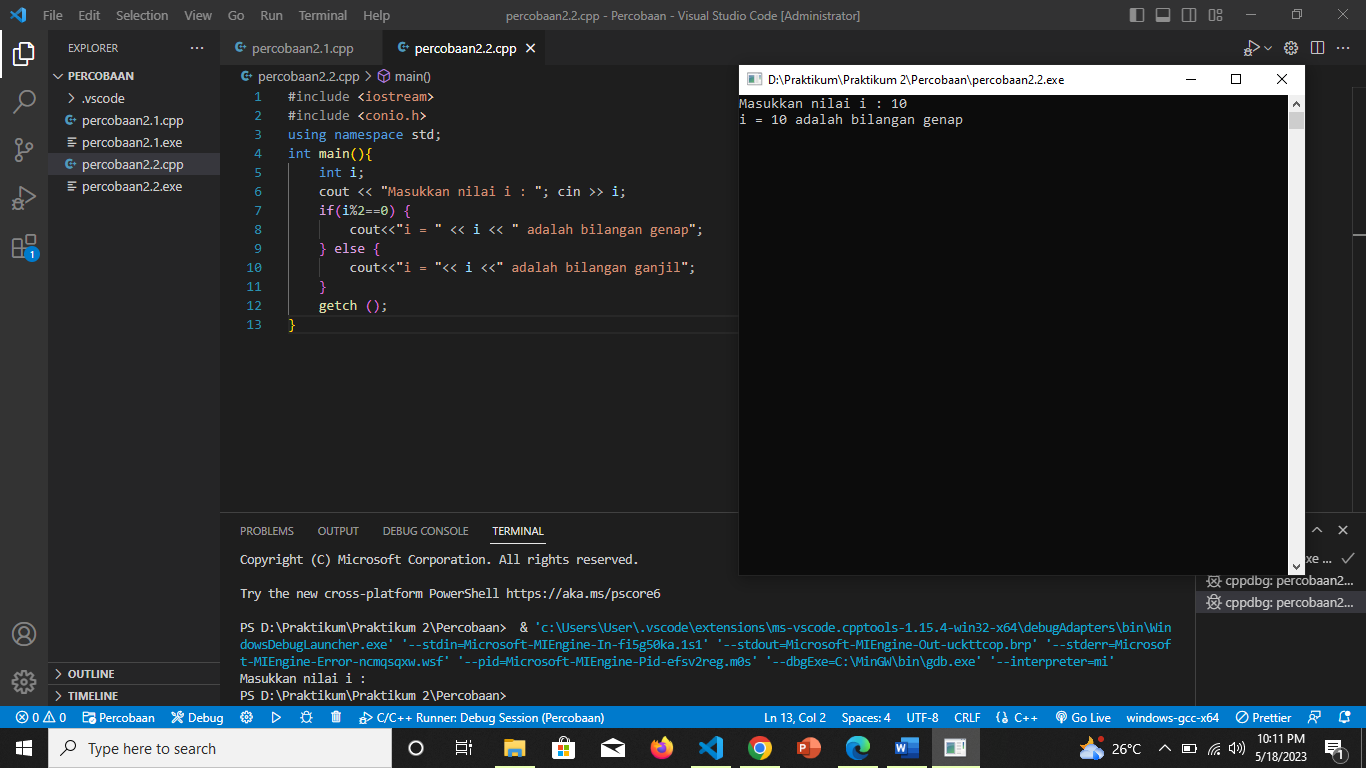
**BAB II**

**PERCOBAAN**

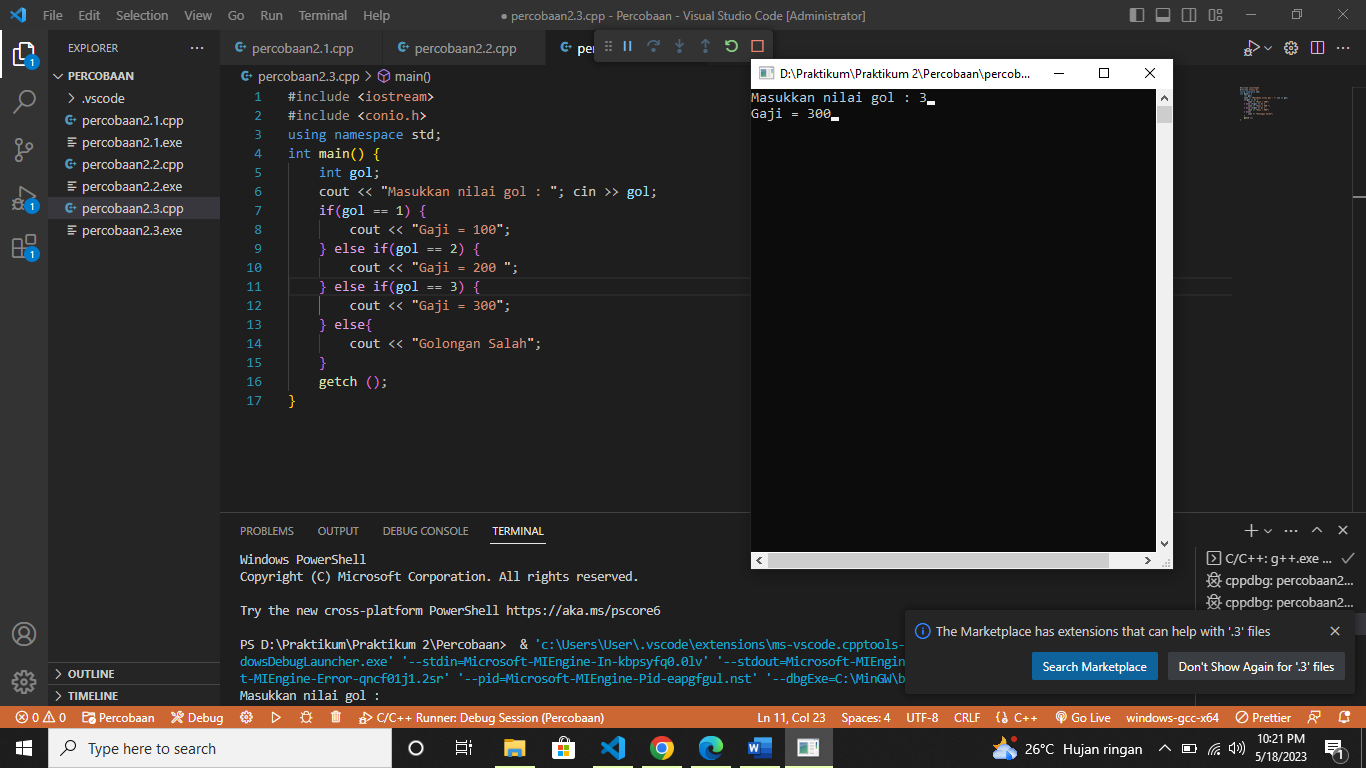
* 1. Percobaan 2.1



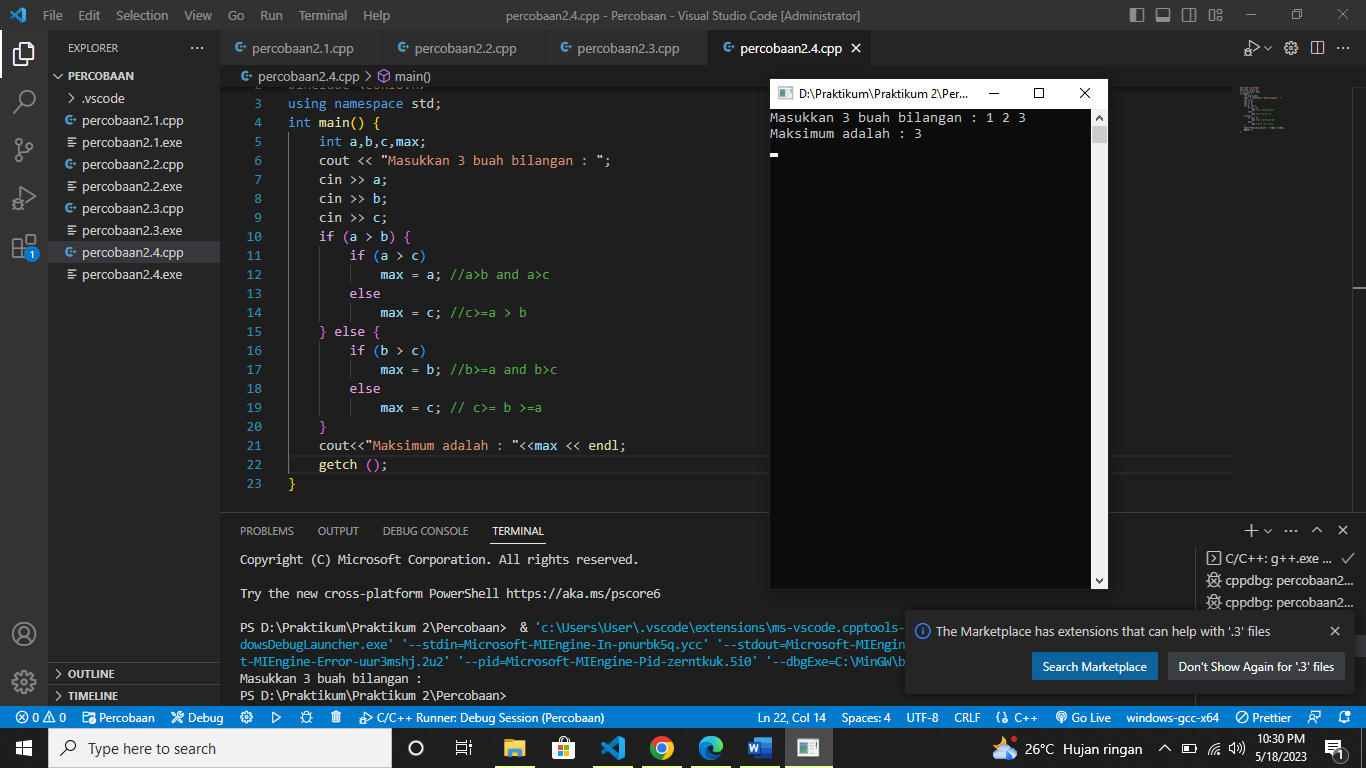
* int n, d;: Mendeklarasikan dua variabel bertipe int, yaitu n dan d, yang akan digunakan untuk menyimpan bilangan yang dimasukkan oleh pengguna.
* cout << "Masukkan bil. Pertama: ";: Menampilkan pesan untuk meminta pengguna memasukkan bilangan pertama.
* cin >> n;: Menggunakan cin untuk mengambil input dari pengguna dan menyimpannya ke dalam variabel n.
* cout << "Masukkan bil. Kedua: ";: Menampilkan pesan untuk meminta pengguna memasukkan bilangan kedua.
* cin >> d;: Menggunakan cin untuk mengambil input dari pengguna dan menyimpannya ke dalam variabel d.
* if(n % d == 0) {: Memulai blok if. Pernyataan dalam kurung akan dievaluasi dan jika hasilnya true (benar), maka blok if akan dijalankan. Pernyataan tersebut memeriksa apakah n habis dibagi oleh d dengan memeriksa apakah hasil modulo (%) dari n dibagi oleh d adalah 0.
* cout << n << " habis dibagi oleh " << d << endl;: Jika bilangan pertama (n) habis dibagi oleh bilangan kedua (d), maka pesan ini akan dicetak ke layar, memberitahu pengguna bahwa n habis dibagi oleh d.
* }: Menutup blok if.
  1. Percobaan 2.2



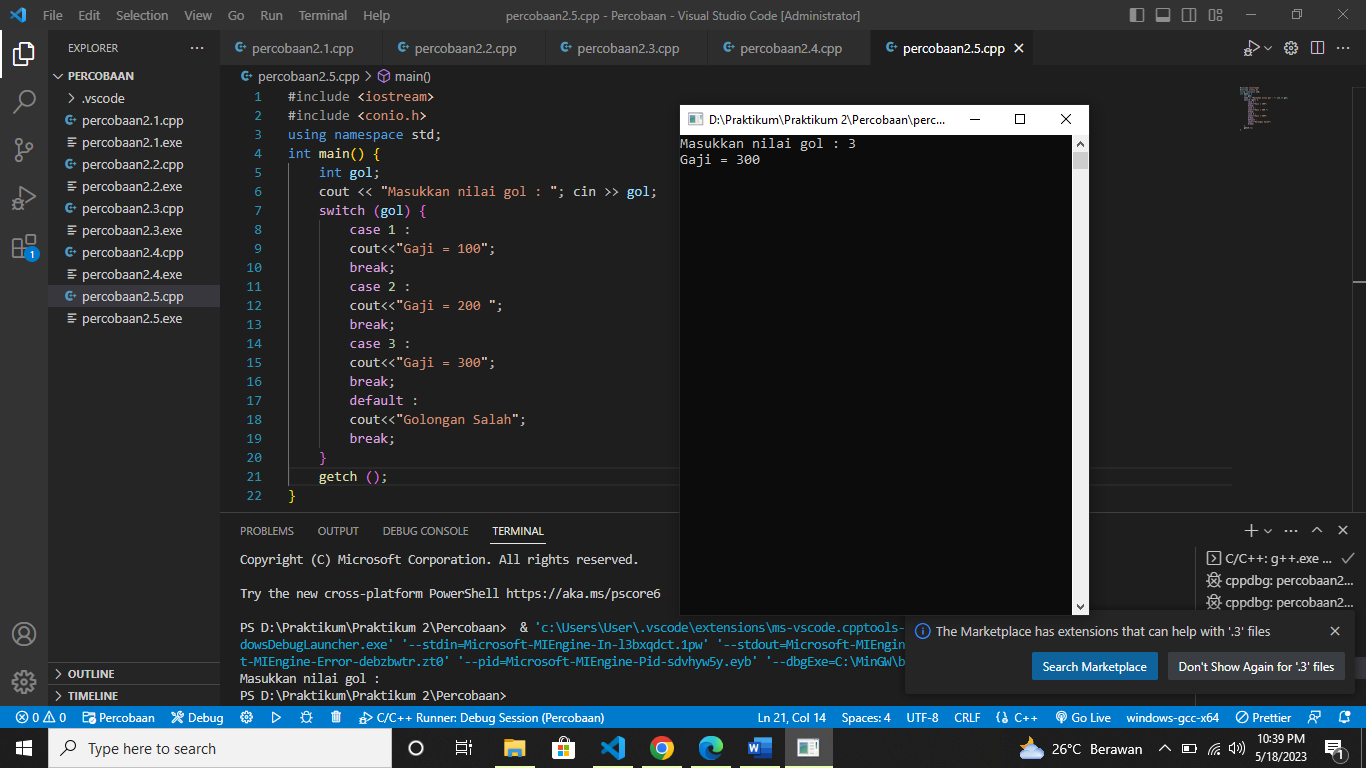
* int i;: Mendeklarasikan sebuah variabel bertipe int dengan nama i, yang akan digunakan untuk menyimpan bilangan yang dimasukkan oleh pengguna.
* cout << "Masukkan nilai i : "; cin >> i;: Menampilkan pesan untuk meminta pengguna memasukkan sebuah bilangan. Kemudian, menggunakan cin untuk mengambil input dari pengguna dan menyimpannya ke dalam variabel i.
* if(i%2==0) {: Memulai blok if. Pernyataan dalam kurung akan dievaluasi dan jika hasilnya true (benar), maka blok if akan dijalankan. Pernyataan tersebut memeriksa apakah i habis dibagi oleh 2 dengan memeriksa apakah hasil modulo (%) dari i dibagi oleh 2 adalah 0. Jika ya, maka i adalah bilangan genap.
* cout << "i = " << i << " adalah bilangan genap";: Jika bilangan i adalah genap, maka pesan ini akan dicetak ke layar, memberitahu pengguna bahwa i adalah bilangan genap.
* else {: Jika pernyataan dalam if pada baris 7 tidak benar (yaitu i bukan bilangan genap), maka blok else akan dijalankan.
* cout << "i = " << i << " adalah bilangan ganjil";: Jika i adalah bilangan ganjil, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa i adalah bilangan ganjil.
  1. Percobaan 2.3



* int gol;: Mendeklarasikan sebuah variabel bertipe int dengan nama gol, yang akan digunakan untuk menyimpan nilai golongan yang dimasukkan oleh pengguna.
* cout << "Masukkan nilai gol : "; cin >> gol;: Menampilkan pesan untuk meminta pengguna memasukkan nilai golongan. Kemudian, menggunakan cin untuk mengambil input dari pengguna dan menyimpannya ke dalam variabel gol.
* if(gol == 1) {: Memulai blok if. Pernyataan dalam kurung akan dievaluasi dan jika hasilnya true (benar), maka blok if akan dijalankan. Pernyataan tersebut memeriksa apakah nilai gol (gol) sama dengan 1. Jika ya, maka gaji yang ditampilkan adalah 100.
* cout << "Gaji = 100";: Jika nilai gol (gol) adalah 1, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa gaji yang diterima adalah 100.
* else if(gol == 2) {: Jika pernyataan dalam if pada baris 7 tidak benar, maka blok else if akan dievaluasi. Pernyataan ini memeriksa apakah nilai gol (gol) sama dengan 2.
* cout << "Gaji = 200";: Jika nilai gol (gol) adalah 2, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa gaji yang diterima adalah 200.
* else if(gol == 3) {: Jika pernyataan dalam else if pada baris 9 tidak benar, maka blok else if berikutnya akan dievaluasi. Pernyataan ini memeriksa apakah nilai gol (gol) sama dengan 3.
* cout << "Gaji = 300";: Jika nilai gol (gol) adalah 3, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa gaji yang diterima adalah 300.
* else {: Jika semua pernyataan sebelumnya dalam if dan else if tidak benar, maka blok else akan dijalankan.
* cout << "Golongan Salah";: Jika nilai gol (gol) tidak sama dengan 1, 2, atau 3, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa nilai golongan yang dimasukkan salah.
  1. Percobaan 2.4



* int a, b, c, max;: Mendeklarasikan empat variabel bertipe int, yaitu a, b, c, dan max, yang akan digunakan untuk menyimpan bilangan-bilangan yang dimasukkan oleh pengguna dan nilai maksimum.
* cout << "Masukkan 3 buah bilangan : ";: Menampilkan pesan untuk meminta pengguna memasukkan tiga buah bilangan.
* cin >> a; cin >> b; cin >> c;: Menggunakan cin untuk mengambil input dari pengguna dan menyimpannya di dalam variabel a, b, dan c secara berurutan.
* if (a > b) {: Memulai blok if. Pernyataan dalam kurung akan dievaluasi dan jika hasilnya true (benar), maka blok if akan dijalankan. Pernyataan tersebut memeriksa apakah a lebih besar dari b.
* if (a > c) max = a;: Jika a lebih besar dari c, maka nilai maksimum adalah a. Nilai a akan disimpan di dalam variabel max.
* else max = c;: Jika a tidak lebih besar dari c, maka nilai maksimum adalah c. Nilai c akan disimpan di dalam variabel max.
* else {: Jika pernyataan pada baris 8 tidak benar, maka blok else akan dijalankan.
* if (b > c) max = b;: Jika b lebih besar dari c, maka nilai maksimum adalah b. Nilai b akan disimpan di dalam variabel max.
* else max = c;: Jika b tidak lebih besar dari c, maka nilai maksimum adalah c. Nilai c akan disimpan di dalam variabel max.
* cout << "Maksimum adalah : " << max << endl;: Menampilkan pesan beserta nilai maksimum ke layar.
  1. Percobaan 2.5

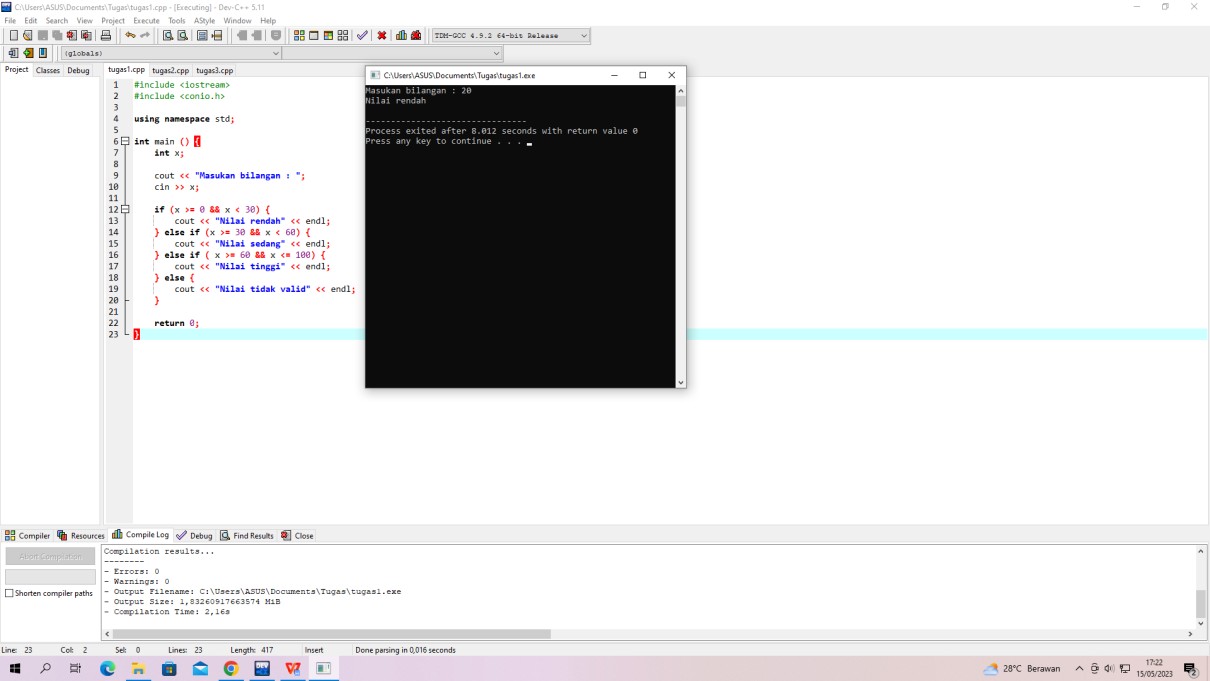


* int gol;: Mendeklarasikan sebuah variabel bertipe int dengan nama gol, yang akan digunakan untuk menyimpan nilai golongan yang dimasukkan oleh pengguna.
* cout << "Masukkan nilai gol : "; cin >> gol;: Menampilkan pesan untuk meminta pengguna memasukkan nilai golongan. Kemudian, menggunakan cin untuk mengambil input dari pengguna dan menyimpannya ke dalam variabel gol.
* switch (gol) {: Memulai konstruksi switch dengan gol sebagai ekspresi yang akan dievaluasi.
* case 1 :: Jika nilai gol sama dengan 1, maka blok kode di dalam case ini akan dijalankan.
* cout << "Gaji = 100";: Jika gol adalah 1, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa gaji yang diterima adalah 100.
* break;: Menghentikan eksekusi switch dan keluar dari switch.
* case 2 :: Jika nilai gol sama dengan 2, maka blok kode di dalam case ini akan dijalankan.
* cout << "Gaji = 200";: Jika gol adalah 2, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa gaji yang diterima adalah 200.
* break;: Menghentikan eksekusi switch dan keluar dari switch.
* case 3 :: Jika nilai gol sama dengan 3, maka blok kode di dalam case ini akan dijalankan.
* cout << "Gaji = 300";: Jika gol adalah 3, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa gaji yang diterima adalah 300.
* break;: Menghentikan eksekusi switch dan keluar dari switch.
* default :: Jika nilai gol tidak sama dengan 1, 2, atau 3, maka blok kode di dalam default akan dijalankan.
* cout << "Golongan Salah";: Jika nilai gol tidak sesuai dengan kasus yang ada, pesan ini akan dicetak ke layar, memberitahu pengguna bahwa golongan yang dimasukkan salah.
* break;: Menghentikan eksekusi switch dan keluar dari switch

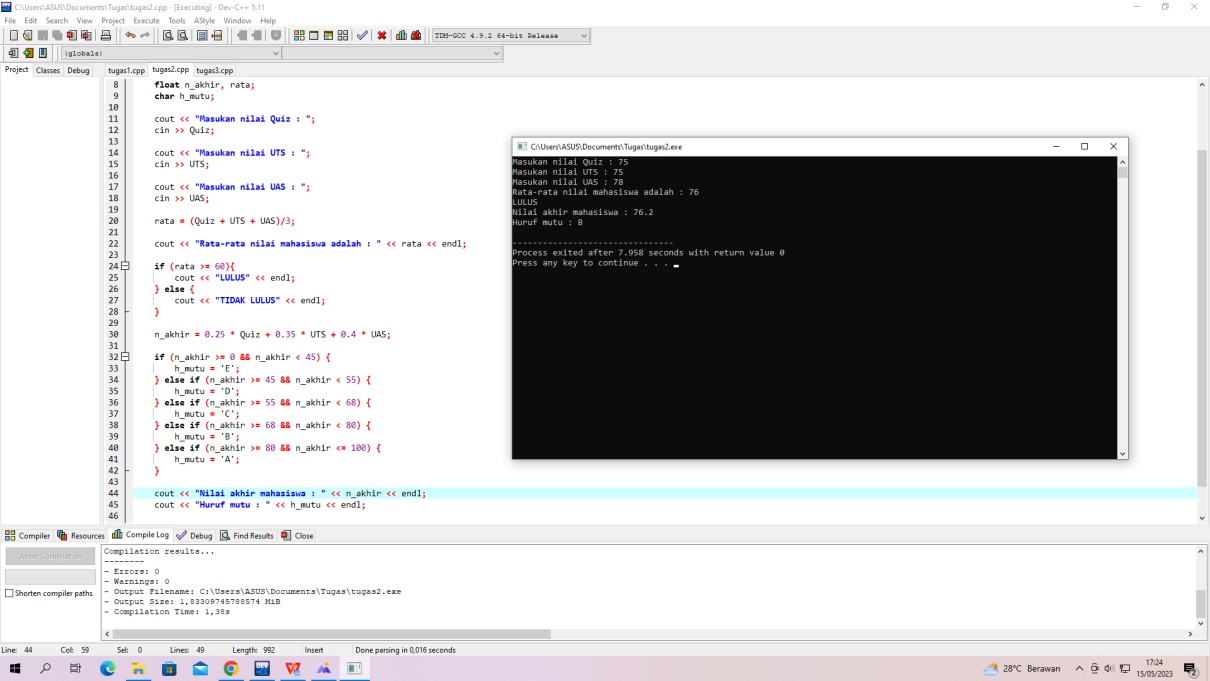
**BAB III**

**TUGAS PRAKTIKUM**

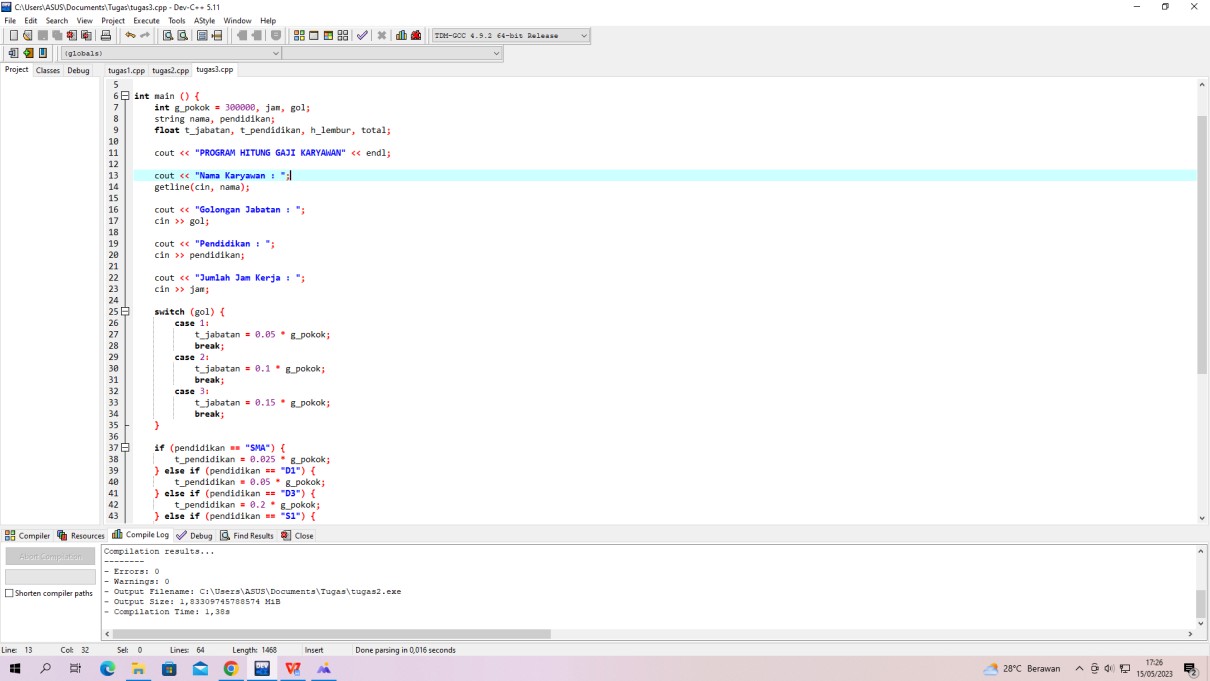
* 1. Tugas 1

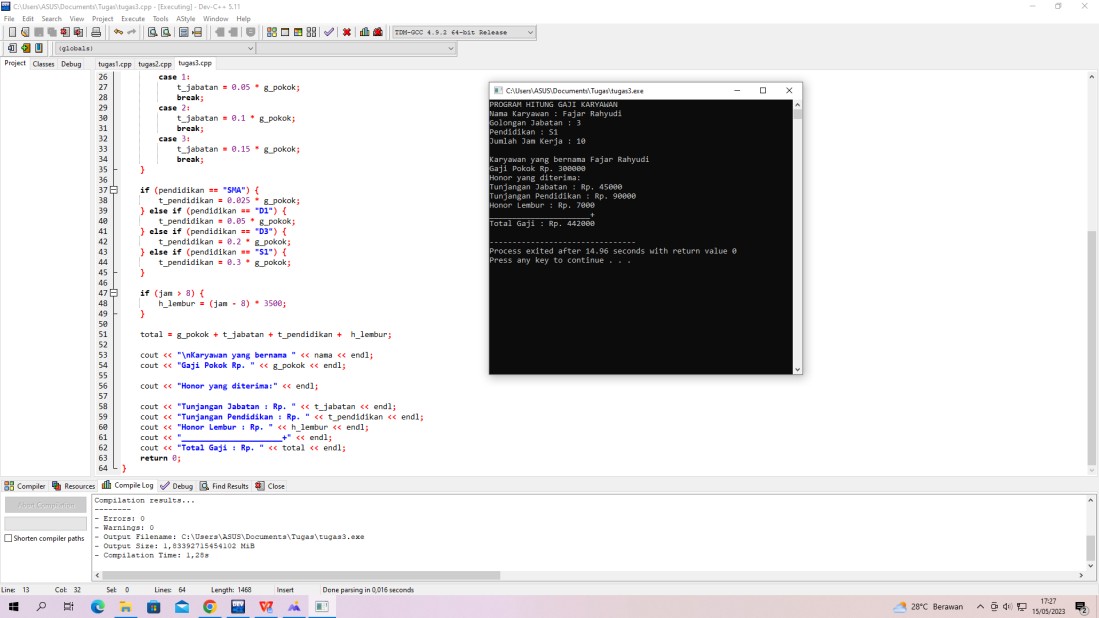


* 1. Tugas 2



* 1. Tugas 3





**BAB IV**

**ANALISA**

* 1. Tugas 1
     1. Poin A

Pada program tersebut user diminta memasukan sebuah nilai dari keyboard, kemudian program akan membaca input tersebut sebagai value dari x.

* + 1. Poin B

Setelah user memasukan sebuah bilangan integer, program akan memeriksa nilai tersebut, apakah memenuhi salah satu kondisi yang telah dibuat. Jika salah satu kondisi telah terpenuhi, maka program akan menjalankan pernyataan sesuai kondisi tersebut.

* 1. Tugas 2

1. Poin A

Untuk dapat menghitung dan mengeluarkan rata-rata dari nilai mahasiswa,kita dapat menggunakan operator aritmatika pada C++. Dan untuk mencari rata-rata dari nilai mahasiswa, dapat menggunakan rumus : (quiz + uts + uas)/3 dengan output yang dihasilkan “Rata-rata = (hasil penjumlahan semua nilai dibagi 3)”. Kemudian program akan mengolah semua nilai tersebut untuk mendapatkan nilai akhir dari nilai mahasiswa tersebut menggunakan rumus : 25% \* Quiz + 35% \* UTS + 40% \* UAS, dan program akan memeriksa nilai akhir untuk menentukan huruf mutu.

1. Poin B

Setelah semua nilai didapatkan maka program akan menghasilkan output berupa nilai rata-rata, nilai akhir, huruf mutu, dan kata LULUS jika nilai rata-rata sama dengan atau lebih besar dari 60.

1. Tugas 3
   1. Poin A

Untuk dapat menghitung total gaji dan mengeluarkan hasil dari total gaji, kita harus mengetahui nilai yang akan diinput yaitu golongan, Pendidikan, jam kerja

* 1. Poin B

Kita harus membuat rumus yang dapat menghitung tunjangan setiap aspek, diantaranya tunjangan golongan berdasarkan golongan, tunjangan Pendidikan berdasarkan Pendidikan, dan tunjangan lembur jika jam kerja lebih dari 8 jam.

* 1. Poin C

Dengan menginputkan golongan dengan rentang 1 s/d 3, Pendidikan dari SMA s/d S1, serta jam kerja, maka kita bisa mendapatkan total gaji yang kemudian akan ditampilkan pada konsol.

**BAB V**

**PENUTUP**

5.1 Kesimpulan

Percabangan (Branching) atau bisa juga disebut dengan *dicision-making* adalah suatu control flow pada algoritma pemrograman yang cara kerjanya melakukan pemilihan pernyataan pada kondisi dua pilihan atau lebih.Pernyataan If..Then akan dikerjakan jika kondisi bernilai benar dan jika tidak memenuhi syarat maka akan diabaikan.

* + - Pernyataan If..Then..Else, yaitu jika kondisi benar maka akan mengerjakan perintah 1, dan jika salah makaakaan mengerjakan perintah.
    - Pernyataan If..Then..Else If merupakan if..then..else if majemuk karena memuat lebih dari dua kondisi.
    - Pernyataan Nested If yaitu pernyataan if yang berada di dalam pernyataan if lainnya.
    - Pernyataan Switch case mirip dengan if..else majemuk tetapi penggunaannya untuk data yang bertipe integer atau karakter.
  1. Saran

Sebelum memulai membuat program, hendaknya mempelajari dna memahami proses dan cara kerja dari Bahasa pemrograman yang dipakai agar tidak terjadi error, sehinggga program dapat dijalankan dengan lancar.

**DAFTAR PUSTAKA**

Anonim. (2023). *MODUL 2 PRAKTIKUM ALGORITMA DAN PEMOGRAMAN.*

Handoko, P. (2019). *MODUL PRAKTIKUM DASAR-DASAR PEMOGRAMAN.*

Suyanto. (2009). BAB 4 KONDISI/PEMILIHAN.

YOGYAKARTA, U. N. (2023). Retrieved from Pembelajaran 4 : Percabangan dalam C++: http://staffnew.uny.ac.id/upload/131872515/pendidikan/Bab+II+Keg+Pemb+4\_+Percb.pdf