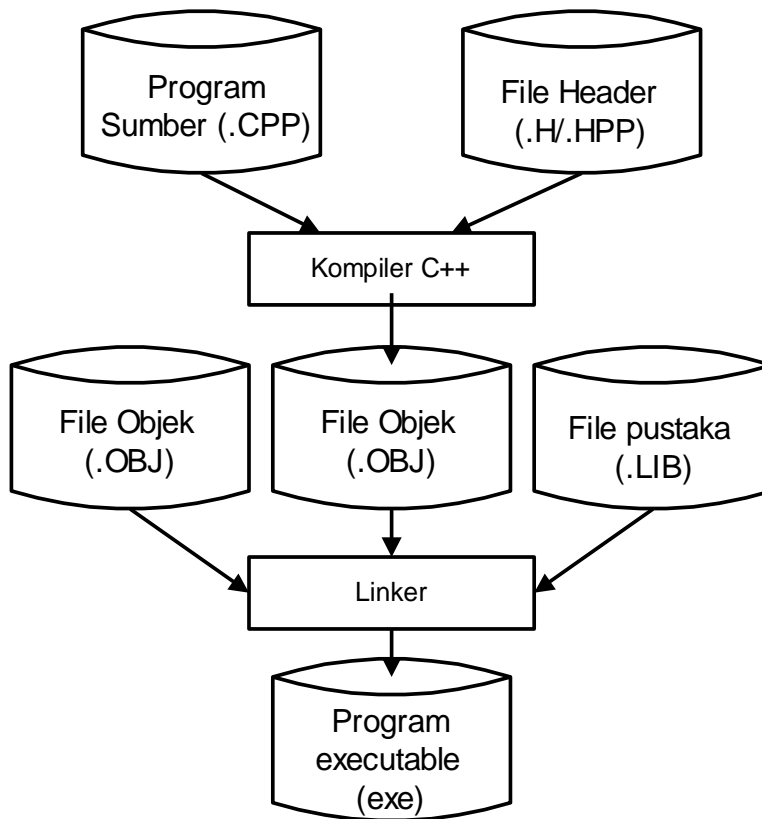


PENDAHULUAHAN DASAR-DASAR PEMROGRAMAN C++

Sebenarnya program C++ dapat ditulis menggunakan sembarang editor teks, seperti EDIT (milik DOS), Wordstar, SideKick ataupun menggunakan editor bawaan dari compiler.

Program C++ biasa ditulis dengan nama CPP (dari kata C Plus Plus) Agar program ini bias dijalankan (eksekusi), program harus dikompilasi terlebih dahulu dengan menggunakan compiler C++. Pada saat kompilasi, program sumber (CPP) bersama file-file header (berekstensi .H atau .HPP) akan diterjemahkan oleh compiler C++ menjadi kode obyek (.OBJ) file ini berupa file dalam format biner (berkode 0 dan 1).

Selanjutnya file obyek ini bersama-sama dengan file obyek lain serta file pustaka (.LIB) dikaitkan menjadi satu oleh linker. Hasilnya berupa file yang bersifat *executable*. File inilah yang bias dijalankan dari system operasi secara langsung. Proses pengkompilasian file sumber hingga menjadi file eksekutabel dapat digambarkan seperti berikut ini:

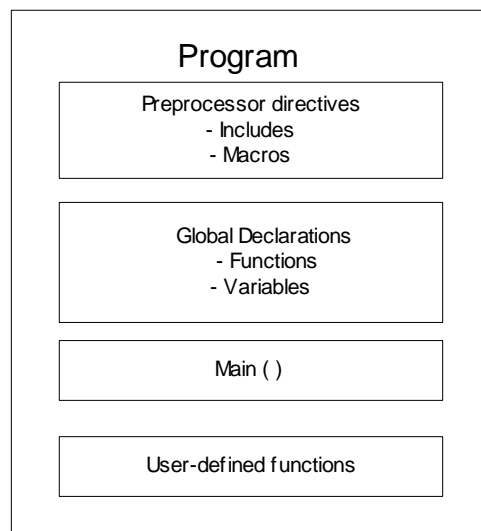


Saat ini banyak compiler C++ yang beredar di pasaran. Sebagai contoh Microsoft Corporation, sebagai perusahaan pembuat perangkat lunak yang sangat termashur, mengeluarkan produk compiler C++ berupa Microsoft C/C++ serta visual C++ Adapun Borland International yang sangat populer dengan turbo Pascal-nya, mengeluarkan Turbo C++ dan Borlan C++, compiler yang terakhir inilah yang kita gunakan praktikum ini. Dibawah ini adalah tampilan program editor Borland C++ versi 5.22.

Gambar tampilan

Sekilas tentang Struktur program C++

Pada dasarnya struktur program C++ adalah terdiri dari beberapa komponen pokok, diantaranya: pengarah preprocessor, deklarasi global, fungsi main (), dan fungsi user-defined.



Pengarah Preprocessor

Dalam program C atau C++ bagian pertama yang menyusun sebuah tugas, terbentuk di pengarah preprocessor. Pengarah preprocessor menginstruksikan compiler untuk membentuk beberapa tugas sebelum komplisasi dimulai. Pengarah preprocessor #include, salah satu yang sering terlihat, contoh

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

menginstruksikan compiler untuk menyertakan berkas C++ sumber yang lain sebelum kompilasi dimulai. Pengarah preprocessor yang lain adalah #define, yang merupakan definisi sebuah macro.

Deklarasi Global

Bagian utama yang kedua dari sebuah program C++ berisi deklarasi global. Deklarasi global memberitahu compiler fungsi user-defined dan variable (atau konstanta) yang dipakai untuk semua fungsi dalam program sumber. Contoh:

```
Int sum(int a,int b); // deklarasi fungsi user-defined sum
```

```
Int X; // deklarasi variable X
```

Program main ()

Bagian utama ketiga dari program C++ adalah fungsi main () (yang merupakan tubuh utama program), setiap program harus memiliki fungsi main (). Eksekusi program dimulai dari fungsi main () dan dalam struktur program yang baik, eksekusi berujung pada fungsi ini.

Contoh :

```
Main()
```

```
{
```

```
  Pernyataaan;
```

```
}
```

Fungsi User-defined

Bagian terakhir dari program C ++ berisi user-define yang merupakan kelompok dari perintah yang dibuat untuk menyempurnakan perintah-perintah program. Fungsi user-define dapat digunakan untuk mengerjakan perintah-perintah yang dikehendaki,sesuai dengan daya imajinasi dari pemrogram.

Tipe data

Sewaktu anda mendeklarasikan variable, konstanta atau fungsi (yang menghasilkan nilai balik) dalam program, anda harus memberi nama variable, konstanta, atau fungsi tersebut beserta

tipe datanya kepada compiler C++. Tipe mendefinisikan nilai yang dapat disimpan pada variable, atau konstanta, atau nilai yang dapat dihasilkan oleh fungsi dan juga sekumpulan operasi yang dapat dilakukan terhadap data tersebut.

Tipe data dasar pada C++ meliputi

- ⇒ Char
- ⇒ Int
- ⇒ Short
- ⇒ Long
- ⇒ Float
- ⇒ Double
- ⇒ Long double

Tipe data yang berhubungan dengan bilangan bulat adalah char,int,shor,dan long. Sedangkan hanya berhubungan dengan bilangan pecahan.

Ukuran memori yang diperlukan untuk masing-masing tipe data sangat bergantung pada perangkat keras dari computer yang digunakan. Karena itu jangkauan bilangan dari masing-masing tipe data juga bias berlainanantara satu jenis mesin dengan mesin lainnya. Nilai yang tercakup pada tipe data bilangan (bulat maupun pecahan) pada arsitektur PC berbasis DOS dapat dilihat pada table di bawah ini:

Tipe Data	Ukuran Memori	Jangkauan Nilai	Jumlah Digit Presisi
Char	1 byte	-128 hingga + 127	-
Int	2 byte	-32768 hingga +32767	-
Long	4 byte	-2.147.438.648 hingga 2.147.438.647	-
Float	4 byte	3.4×10^{-38} $3.4 \times 10^{+308}$	6 - 7
Double	8 byte	1.7×10^{-308} hingga $1.7 \times 10^{+308}$	15 -16
Long double	10 byte	3.4×10^{-4932} hingga $3.4 \times 10^{+4932}$	19

UNIT I

FUNGSI INPUT / OUTPUT

Agar program yang ditulis dapat berguna, baik sebagai pengolah data maupun pemberi informasi, maka harus dapat menerima masukan dari pemakai dan mampu menampilkan informasi hasil pengolahan masukan. Pada unit ini akan menjelaskan fungsi input (masukan) dan fungsi output (keluaran) C++.

a. **Fungsi print f()** → Fungsi output

Fungsi untuk character dan string baik untuk menghasilkan hanya informasi yang berbasis character. Suatu saat jika dibutuhkan untuk menampilkan semua jenis data. Tidak hanya data character, dapat digunakan fungsi `f()`. Dengan menggunakan fungsi `printf ()` dapat dihasilkan output data terformat. Sebab fungsi ini memberikan keleluasan untuk mengatur tampilan hasil output.

Bentuk : `printf (cont char *format [,argument,...]);`

Tampak pada contoh diatas, argument terdiri dari dua bagian. Bagian pertama adalah format string, yang untuk mengatur tampilan data output. Bagian kedua berisi data yang akan dihasilkan oleh `printf()`

b. **Fungsi scan ()** → Fungsi Input

Fungsi `scan { }` membentuk input terformat. Dengan demikian `scan ()` memasukan data yang sudah ditentukan sebelumnya, atau sudah ditentukan bentuknya. Pemrogram yang menentukan bentuk input mana yang akan dipilih.

Bentuk: `scanf(const char*format,&variable)`

Contoh 1:

Tulislah program ini pada program Editor :

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

```
main( )
```

```

{
printf("Praktikum");
printf("Pemrograman C ++ \n\n");
printf("Praktikum\n");
printf("pemrograman C++\n")
}

```

Simpanlah program tersebut dengan nama contoh1.epp

Contoh2:

Tulislah program ini pada Editor:

```

#include<stdio.h>
#include<conio.h>
main ( )
{
printf("Praktikum");
printf("Pemrograman C++\n\n");
printf("Praktikum\n");
printf("Pemrograman C++\n");
getche( );
}

```

Lakukanlah hal-hal berikut dan jawaban/hasilnya tulislah sebagai **resume** kerja anda hari ini:

1. Compile-lah program anda sampai berhasil kemudian jalankan, tulislah hasilnya!
2. Apa pengaruh format “\n” pada fungsi printf (), jika melihat efek dari hasil tulisan yang dibentuk.
3. Hapuslah semua format “\n” pada tiap fungsi printf(), jalankan program..dari hasilnya dan dengan dibandingkan hasil saat belum dihapus,apa fungsi dari format”/n” pada fungsi print () tersebut
4. Tambahkan perintah getch(); di baris paling bawah pada program utama (diatas kurung kurawal tutup []). Jalankan program,perhatikan apa yang terjadi!

5. Lengkapi program anda tersebut, hingga menjadi seperti ini:

```
#include<stdio.h>
#include<conio.h>
main ( )
{
    printf("Praktikum");
    printf("Pemrograman C++\n\n");
    clrscr( );
    printf("Praktikum\n");
    printf("Pemrograman C++\n");
    printf("Masukkan Nilai A:");
    int a; int b; int c;
    scanf("%d",&b);
    printf(" Masukkan Nilai B : ");
    scanf("%d",&b);
    c = a + b;
    printf("Hasil A + B = %d\n",c);
    Getche( );
}
```

6. Apa fungsi dari **clrscr ()** ?

(Untuk memahaminya jalankan program anda secara berulang-ulang dengan clrscr(), dan tanpa clrscr())

7. Apa fungsi dari **getche()** ?

8. Apa fungsi format "%d" dan tanda "&" pada fungsi scanf()? (Untuk memahaminya jalankan program anda secara berulang-ulang dengan getche(), dan tanpa getche())

9. Hapuslah tulisan #include<conio.h>; pada program anda catatlah pesan kesalahan yang terjadi?

10. Buatlah tulisan lain menurut kehendak anda.

11. Buatlah kesimpulan untuk unit ini!

12. Simpanlah program dalam disket dengan nama lat 1.

Tugas rumah

1. Sebutkan 3 (tiga) macam percabangan dalam C++ yang anda ketahui,dan beri penjelasan secukupnya!
2. gambarlah aliran program (flowchart) untuk masing-masing fungsi percabangan tersebut pada soal nomor 1, dan beri penjelasan secukupnya!
3. Buatlah contoh program sederhana dengan menggunakan fungsi percabangan **if-else**

UNIT II

PERCABANGAN

1. Pengertian percabangan (kondisional)

Biasanya program akan dieksekusi baris perbaris setiap saat secara berurutan. Eksekusi baris perbaris sangat baik, sebab tidak akan terdapat saat vacuum pada program. Jika ingin perubahan, dalam hal ini program melompat (tidak berurutan) baik ke baris perintah jauh dibawahnya maupun melompat ke baris perintah diatasnya. Maka kita membutuhkan perintah percabangan, yaitu fungsi yang dapat mengatur program agar melompat ke suatu baris perintah yang ditentukan, sesuai dengan kondisi yang sedang diterima. Fungsi percabangan yang biasa digunakan dalam C++ adalah :if,if-else,dan switch.

2. Percabangan bersyarat if

Percabangan if akan menangani suatu ekspresi apabila kondisi terpenuhi. Bentuk dasar pernyataan if adalah:

If (ekspresi kondisional)

 Statemen ;

Atau

If (ekspresi kondisional)

{

 Statemen 1

 Statemen _2

 Statemen _3

 Statemen _n

} Grup statemen

}

Bentuk diatas menunjukkan,jika ekspresi kondisional di dalam tanda kurung benar, maka statemen atau grup statemen atau grup statemen dibawahnya akan dieksekusi,sebaliknya jika ekspresi kondisionalnya salah maka statemen/grup statemen di bawahnya tidak akan dieksekusi (fungsi if tidak menghasilkan apa-apa). Ekpresi kondional adalah ekspresi yang menjadi dasar bagi pernyataan berkondisi (selalu dipakai pada fungsi if),yang memberikan nilai benar atau salah. Hasil ekspresi berupa :

- 0 kalau ekspresi bernilai salah
- 1 kalau ekspresi bernilai benar

Adapaun elemen yang membentuk ekspresi ini adalah operator:

- Relasi dan
- Logika

Operator relasi

Operator relasi biasa digunakan untuk membandingkan dua buah nilai. Keseluruhan operator relasi pada C++ dapat dilihat pada table dibawah ini:

Operator	Keterangan	Contoh
==	Sama dengan	4=4
!=	Tidak sama dengan	3!=7
>	Lebih besar	5>2
<	Lebih kecil	2<5
>=	Lebih besar atau sama dengan	3>=x
<=	Lebih kecil atau sama dengan	X<=3

Contoh:

If(x==5)

Hasil ekspresi diatas bernilai:

- 1 atau benar jika x sama dengan 5
- 0 atau salah jika x tidak sama dengan 5

Operator logika

Operator logika biasa digunakan untuk menghubungkan dua buah ekspresi kondisional atau lebih menjadi satu ekspresi kondisional. Operator-operator ini berupa:

- `&&` → Operator logika dan (AND)
- `||` → operator logika (OR)
- `!` → operator logika bukan (NOT)

Operator `||` dan `&&`

Bentuk pemakain operator `|` dan `&&` adalah sebagai berikut

Ekspresi1 `|` ekspresi2

Ekspresi1 `&&` ekspresi2

Pada bentuk pertama, hasil ekspresi hanya bernilai salah apabila kedua ekspresi bernilai salah, sedangkan pada bentuk kedua bernilai benar apabila kedua ekspresi bernilai benar.

Table operator logika `|` (atau) serta `&&` (dan)

Ekspresi1	Ekspresi2	Hasil <code> </code>	Hasil <code>&&</code>
Salah	Salah	Salah	Salah
Salah	Benar	Benar	Salah
Benar	Salah	Benar	Salah
Benar	Benar	Benar	Benar

Operator logika `!` (bukan / NOT)

Operator `!` digunakan dengan bentuk:

`!ekspresi`

Hasilnya berupa

- benar kalau ekspresi bernilai salah
- salah kalau ekspresi bernilai benar

3. Percabangan if-else

Percabangan if-else akan menangani salah satu dari dua ekspresi / grup ekspresi sesuai nilai kondisi yang dihasilkan. Bentuk percabangan if else adalah sebagai berikut:

If(ekspresi kondisional)

 Statemen_1;

Else

 Statemen_2;

Atau

If (ekspresi kondisional)

```
{  
    Statemen_1_1  
    Statemen_1_2  
    .....  
    Statemen_1_n  
}
```

Grup statemen_1

```
else  
{  
    Statemen_2_1  
    Statemen_2_2  
    .....  
    Statemen_2_n  
}
```

Grup statemen_2

Bentuk diatas mempunyai arti,apabila ekspresi kondisional menghasilkan nilai benar maka program akan mengeksekusi statemen_1 / grup statemen_1,namun sebaliknya apabila ekspresi kondisional bernilai salah maka yang akan dieksekusi adalah statemen_2 / grup statemen_2

4. Percabangan Switct

Percabangan switch adalah pernyataan yang digunakan untuk menjalankan salah satu pernyataan dari beberapa kemungkinan pernyataan,berdasarkan nilai dari sebuah ekspresi dan nilai penyeleksi. Secara singkatnya,pernyataan ini merupakan pernyataan pilihan berganda.

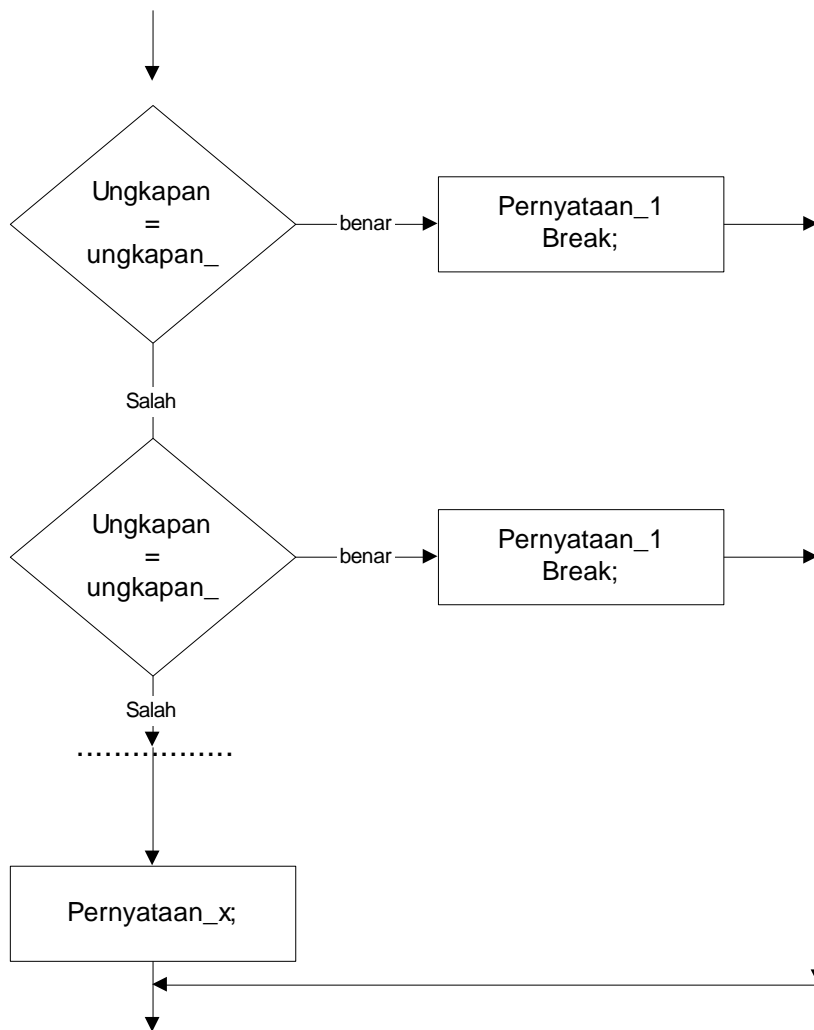
Bentuk umum pernyataan switch :

```
Switch (ekspresi)
{
    case ekspresi_1
        pernyataan_1;
        break;
    case ekspresi_2
        break;
    case ekspresi_n;
        break;
    default;
        pernyataan _x
}
```

Pada pernyataan switch,ekspresi dapat berupa ekspresi,konstanta ataupun variable.Adapaun ekspresi1,ekspresi2,dan seterusnya dapat berupa sembarang konstanta bertipe int atau char.

Pencocokan ekspresi dengan ekspresi1,ekpresi2,dan sebagainya dilakukan secara berurutan,dimulai yang pertama. Sekiranya cocok,pernyataan yang mengikuti case bersangkutan dijalankan. Kalau selanjutnya pernyataan break ditemukan,eksekusi dari pernyataan switch berakhir. Perlu diketahui,bagian default hanya akan dijalankan kalau ungkapan pada bagian case tidak ada yang cocok dengan ungkapan switch.

Aliran program (flowchart) dari fungsi switch digambarkan sebagai berikut:



Tulislah program dibawah ini pada program editor:

```
#include<stdio.h>
#include<conio.h>
main( )
{
int x;
printf("Masukan Bilangan Bulat:");
scanf("%d",&x);
if (x.>=30)
printf("bilangan yang anda masukan lebih besar atau sama dengan 30");
else
```

```

printf("bilangan yang anda masukan lebih kecil dari 30");
getche();
int kode_hari;
clrscr();
print("Menentukan hari\n");
printf("1.SENIN 3.RABU 5. JUMAT 7. MINGGU\n");
printf("2. SELASA 4.KAMIS 6.SABTU\n\n");
printf(Kode hari (1..7):");
scanf("%d",&kode_hari);
switch(kode_hari)
{
case 1: printf("senin");break;
case 2 : printf("selasa");break;
case 3 : printf("Rabu");break;
case 4: printf("Kamis");break;
case 5 : printf("Jum'at");break;
case 6 : printf("Sabtu");break;
case 7 : printf("Minggu");break;
}
getche();
}

```

Lakukan langkah-langkah dibawah ini:

1. Jalankalah program diatas dan catatlah hasilnya!
2. Hapuslah compiler "&" pada baris perintah: scanf("%d"&x): dan berisi perintah : jelaskan mengapa demikian?
3. Gantila type int pada baris deklarasi : int kode_hari; menjadi float (lengkapnya: float kode_hari), jalankan program ! catatlah pesan error yang terjadi,jelaskan maksud dari pesan error tersebut!
4. Sebutkan fungsi pernyataan default pada argument switch?

5. Hapuslah semua argument break yang ada di dalam argument switch, dan jalankan program, apakah yang terjadi, jelaskan mengapa demikian!
6. Buatlah program singkat dengan menggunakan argument switch (sertakan pada resume anda)!

Tugas Rumah

1. Sebutkan macam-macam fungsi perulangan!
2. jelaskan masing-masing fungsi perulangan tersebut pada nomor 1!
3. Jelaskan kapan kita memakai fungsi perulangan for, dan kapan kita memakai perulangan while / do while!
4. Buatlah program untuk menampilkan angka 100 samapai 200 dengan menggunakan fungsi perulangan for!

UNTI 111 PERULANGAN

Pada pembuatan suatu program kita sering melakukan proses yang sama diulang-ulang. Namun kita perlu menulis proses tersebut secara berulang-ulang, karena C++ menyediakan fungsi perulangan. Macam-macam fungsi perulangan tersebut adalah:

- For
- While
- Do While

1. for

Fungsi perulangan for digunakan untuk pengulangan suatu pernyataan atau beberapa pernyataan (yang tergabung dalam satu blok pernyataan), dimana jumlah perulangan yang akan dilakukan dapat ditentukan.

Bentuk perulangan for adalah sebagai berikut:

```
For (ungkapan_1,ungkapan_2,ungkapan_3)  
    Pernyataan;
```

Atau

```
For (ungkapan_1,ungkapan_2,ungkapan_3)  
{  
    pernyataan_1;  
    pernyataan_2;  
    .....  
    pernyataan_2;  
}
```

} blok pernyataan

Dimana:

- ungkapan_1 merupakan pernyataan inisialisasi nilai awal keadaan (sebagai batas awal).
- Ungkapan_2 berlaku sebagai kondisi yang menentukan pengulangan (batas akhir) terhadap pernyataan atau blok pernyataan.

- Ungkapan_3 digunakan sebagai pengatur (step counter) variable yang digunakan di dalam ungkapan_1.

Contoh:

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

main()
{
    char abjad;
    clrscr();
    for(abjad = 'A'; abjad <='Z'; abjad++)
        printf("%c", abjad);
}
```

2. While

Pernyataan while merupakan salah satu pernyataan yang berguna untuk memproses suatu pernyataan atau pernyataan beberapa kali (jumlah perulangan tidak ditentukan). Bentuk pernyataan while sebagai berikut:

```
While(ungkapan)
    Pernyataan;
```

Atau:

```
While(ungkapan )
{
    pernyataan_1;
    pernyataan_2;
    .....
    pernyataan_n;
}
```

blok pernyataan

Pada bagian pernyataan / blok pernyataan yang mengikuti (di bawah) while akan dieksekusi selama ungkapan pada while bernilai benar (tidak sama dengan nol). Perlu diketahui, pengujian terhadap ungkapan pada while dilakukan sebelum bagian pernyataan / blok pernyataan. Oleh karena itu ada kemungkinan pernyataan pada while tidak dijalankan sama sekali, yaitu bila kondisi yg pertama kali bernilai salah.

3. do while

Pernyataan do-while juga berguna untuk mengulang proses. Bentuk perulangan do while sebagai berikut:

Perulangan do-while sebagai berikut:

```
do
{
    pernyataan_1
    pernyataan_2;
    .....;
    pernyataan_n;
}while(ungkapan)
```

Bagian pernyataan_1 hingga pernyataan_n dijalankan secara berulang sampai ungkapan bernilai salah (sama dengan nol). Namun berbeda dengan while, pengujian ungkapan dilakukan di belakang (setelah bagian pernyataan).

Tulislah program di bawah ini pada program editor:

```
#include<stio.h>
#include<conio.h>

main()
{
    int i;
    clrscr();
```

```

for(I = 0;I <=12;i++)
    printf("praktikum\n");
i = 0;
do
{
    printf("C++");
    i++;
}while(i<10);
i =0;
while(i.-18)
{
    printf('Dengan Mudah')
    i--;
}
getche();
}

```

Lakukanlah langkah-langkah dibawah ini sebagai resume anda:

1. Jalankan program,catatlah hasilnya di resume anda!
2. buatlah aliran program (flowchart) dari program diatas!
3. Jelaskan cara kerja program diatas!
4. Buatlah program lain dengan menggunakan ketiga fungsi perulangan for, while, do while!
5. Buatlah flowchart dari program yang anda telah buat!
6. Simpanlah program tersebut dengan naman lat3.ccp!

Tugas rumah

1. Apakah yang dimaksud dengan array pada C++?
2. jelaskan cara deklarasi variable tipe array,berilah salah satu contohnya?
3. Bagaimana cara memberi penugasan/nilai pada sebuah elemen array berdimensi tunggal?
4. jelaskan cara mendeklarasikan tipe array 2 dimensi,dan cara memberi penugasan, tulislah salah satu contohnya!

UNIT IV

ARRAY

A. Pengertian array

Array (larik), dalam bahasa c++ termasuk dalam kategori tipe agrerat (tipe turunan khusus), adalah sekumpulan object data yang disebut elemen array, yang memiliki tipe dasar sama dan memerlukan area memori contiguous.

B. Deklarasi array

Bentuk deklarasi array sebagai berikut:

```
Typename identifier[size_constant_constant_expression];
```

Dimana:

Typename	: jenis data elemen array
Identifier	: Nama array
Size_constant_expression	: ukuran / panjang elemen array

Contoh:

```
Int array_data[ 100 ];
```

C. Array dua dimensi (Multidimensi)

Menggunakan array multidimensi sama mudahnya dengan menggunakan array satu dimensi, hanya memerlukan sedikit perubahan dalam deklarasinya.

Deklarasi multidimensi (n dimensi) adalah sebagai berikut:

```
Typename identifier[a1]a2]....[an];
```

Dimana:

a1,a2,.....an : ukuran/ panjang element pada masing-masing dimensi

Contoh : array 2 dimensi

```
Int matrix_3x2[3][2];
```

D. Cara penugasan dan mengakses elemen array

Penugasan adalah suatu perintah / langkah untuk memberikan suatu nilai pada sebuah variable, cara penugasan pada elemen array adalah sebagai berikut;

identifierArray[indeks]= const; → → array 2 dimensi

identifierArray[indeks1][indeks2]=const → → array 2 dimensi

dimana

indeks : nomor indeks / urut dari elemen yang akan diberi nilai

Const : nilai yang akan diberikan.

Cara mengakses elemen array:

Bentuk Var = identifierArray[indeks]

Dimana var adalah variable yang akan mengakses atau mengambil nilai dari suatu elemen array.

Tulislah program dibawah ini pada program editor:

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

```
#include<conio.h>
```

```
main()
```

```
{
```

```
int i ; int my_data[12];
```

```
for (I = 0;I ,=11;i++)
```

```
my_data[ i ]= 3*i + 1;
```

```
for ( i = 0; i <=11;i + +;
```

```
printf('my_data[%d] = %d\n',i +1,my_data[i]);
```

```
printf('\n\n\n');
```

```
{
```

```
printf("| ");
```

```
for( j = 0;j <=2;j++)
```

```
printf("%d ",matrix_2x3[i][j]);
```

```
printf("\n");
```



```
}  
getche();  
}
```

lakukanlah hal-hal ini dan jawaban / hasilnya tulislah sebagai resume kerja anda hari ini:

1. Compile-lah program anda sampai anda berhasil kemudian jalankan,tulislah hasilnya!
2. Hapuslah semua tulisan (format)%d pada baris perintah :printf(“my_data[%d\n”,i+1,my_data[1]); dan pada baris perintah : printf(“%d”,matrix_2x3[i][i]);,tulislah apa yang terjadi dan jelaskan mengapa demikian?
3. Hapuslah baris perintah : printf(“\n\n\n\n”); dan hapuslah tulisan (format)\n pada baris perintah: printf(“\n”);, tulislah apa yang terjadi dan jelaskan mengapa demikian?
4. Gantilah baris perintah : int j; dengan int k; catatlah apa yang terjadi?
5. lalau (dengan tetap memakai perintah int k;) ubahlah program anda sampai benar dan dapat dijalankan dengan hasil eksekusi (tampilan) sama seperti saat program belum di ubah,tulislah program yang telah diubah tersebut?
6. buatlah program untuk menampilkan matriks berordo 3x4 dengan nilai-nilai elemen sembarang(menggunakan array)?
7. simpanlah program anda dengan nama Lat4!

Tugas rumah

1. Apakah yang disebut dengan fungsi dalam bahasa pemrograman C++?
2. Sebutkan dua macam fungsi menurut pengembalian nilainya!
3. di bagian manakah suatu fungsi dideklarasikan?
4. dan di bagian manakah suatu fungsi didefinisikan?
5. buatlah contoh program sederhana dengan menggunakan fungsi!

UNIT V

FUNGSI

Sebuah fungsi sejumlah pernyataan yang dikemas dalam sebuah nama . nama ini selanjutnya dapat dipanggil beberapa kali di beberapa tempat di dalam program.

Tujuan pembuatan fungsi adalah:

- Memudahkan dalam mengembangkan program. Hal ini merupakan kunci dalam pembuatan program yang terstruktural. Program dibagi menjadi beberapa modul yang kecil.
- Menghemat ukuran program. Manfaat ini terasakan kalau ada beberapa deretan instruksi yang sama digunakan pada beberapa tempat di dalam program.

Umumnya fungsi menerima masukan (disebut argument atau parameter). Masukan ini selanjutnya diproses oleh fungsi. Hasil akhir berupa sebuah nilai yang disebut nilai balik. Namun adapula fungsi yang tidak menghasilkan nilai balik.

Sebuah fungsi tidak dapat dipanggil kecuali sudah dideklarasikan. Setelah fungsi dideklarasikan selanjutnya fungsi didefinisikan, letak pendefinifian dapat dilakukan dimana saja. Namun biasanya bagian deklarasi dilakukan pada bagian definisi fungsi (di bawah bagian program utama) bagian deklarasi dan definisi fungsi dapat dilihat dari contoh bentuk program dibawah ini:

Contoh:

```
//bagian deklarasi;
```

```
float pangkat(float a,float b);
```

```
//program utama
```

```
main( )
```

```
{
```

```
.....
```

```
.....
```

```
.....
```

```
}
```

```
//bagian definisi
float pangkat(float a,float b)
(
.....
.....
.....
}
```

Seperti telah dikatakan di atas fungsi dibagi menjadi jenis, yaitu:

- Fungsi yang tidak mengembalikan nilai
- Fungsi yang mengembalikan nilai

1. Fungsi tanpa nilai balik

Adakalanya suatu fungsi tidak perlu memiliki nilai balik. Misalnya fungsi yang hanya dimaksudkan untuk menampilkan suatu keterangan saja.pada fungsi seperti ini,type nilai balik fungsi yang diperlukan adalah void.

Sintaks : void nama_fungsi(daftar_parameter):

Catatan :daftar parameter bersifat opsional,artinya bisa ada bisa tidak mempunyai parameter

Contoh :

```
Void tampilan_judul( )
{
printf("PT Angin Ribut Ribet\n");
printf("Jl. HOS cokrominoto 23 a\n");
printf("Yogyakarta\n");
}
```

Pada contoh fungsi diatas,tidak ada pernyataan return, mengingat fungsi tidak memiliki nilai balik. Namun penggunaan pernyataan return secara eksplisit juga diperkenankan.,sehingga fungsi tampilan_judul diatas dapat ditulis sebagai berikut:

```

Void tampilkan_judul( )
{
    printf("PT Angin Ribut Ribet\n");
    printf("Jl. HOS Cokroaminoto 23 A\n");
    print("Yogyakarta\n");
    return;
}

```

2. Fungsi dengan nilai

Pada dasarnya fungsi dengan nilai balik sama dengan fungsi tanpa nilai balik. Beberapa hal yang perlu diperhatikan adalah sebagai berikut:

- Nama fungsi didahului dengan tipe nilai yang dihasilkan oleh fungsi. Keadaan ini dipakai oleh compiler untuk menentukan jenis nilai yang akan dihasilkan oleh fungsi. Nilai yang dihasilkan dapat berupa integer, karakter atau tipe –tipe data yang lain. Hal inilah yang membedakan dengan fungsi tanpa nilai balik.
- Nama fungsi diikuti dengan beberapa argument/ parameter yang terletak dalam tanda kurung setelah nama fungsi, yang digunakan compiler untuk mengetahui berapa banyak dan macam argument yang dibutuhkan fungsi.
- Tubuh dari fungsi diapit dengan tanda kurung kurawal

Sintaks :tipe_data_fungsi nama_fungsi(daftar_parameter);

Pada umumnya suatu fungsi dideklarasikan di atas fungsi utama (main), dan didefinisikan di bawah program utama.

Contoh:

Tulislah program di bawah ini pada program editor:

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

```
//deklarasi fungsi(tanpa nilai balik)
```

```
void cetak_copyRight(void);
```

```
//deklarasi fungsi (tanpa dengan balik)
```

```

int kali(int a,int b);
//program utama;
main( )
{
    int x,y,z;
    clrscr()
    cetak_copyright();
    printf("Masukan nilai x");
    scant(x)
    printf("masukan nilai y");
    scanf(y);
    z = kali(x,y);
    printf("Z = X + Y = %d",z);
    getche();
}
//definisi fungsi;
void Cetak_CopyRight(void)
{
    gotoxy(50,25);
    printf("copyright by : Lab TE");
    gotoxy(1,1);
}
int c;
c = a*b;
retrun c;
retrun c;
}

```

Kerjakanlah langkah-langkah dibawah ini sebagai resume:

1. Jalankanlah program yang telah anda tulis,catatlah hasilnya,dan jelaskan mengapa demikian?
2. Sempurnakanlah program diatas sehingga program dapat dijalankan dengan benar!

3. tuliskan perubahan yang diperlukan (bagian apa yg perlu ditambahkan /bagian mana yang perlu dihapus [jika]) dari program diatas ?
4. Kembangkan program anda sehingga selain dapat menghitung perkalian antar variable a dan variable b, tetapi juga dapat menghitung / menampilkan hasil penjumlahan, pengurangan,dan pembagian antara variable a dengan variable b,dengan menggunakan fungsi!
5. tuliskan pada resume fungsi-fungsi yang telah anda tambahkan (pada soal nomor 4),baik bagian pendeklerasian maupun bagian defenisinya!
6. simpanlah program akhir anda dengan nama Lat5.cpp!

Tugas Rumah

1. Apakah yang dimaksud dengan fungsi rekursif?
2. pelajari proses perhitungan nilai factorial untuk bilangan bulat positif, dan tuliskan secara singkat proses perhitungan nilai factorial !
3. pelajari kembali perulangan for, tuliskan sintaks / format fungsi perulangan for tersebut!

UNIT VI FUNGSI REKURSIF

Pada praktikum unit yang lalu, kita telah memahami bahwa suatu program dapat dibagi menjadi beberapa sub program (fungsi) yang mempunyai maksud tertentu, misal agar program mudah untuk dianalisa, dibaca, atau dimengerti, ataupun memudahkan dalam pembuatan program itu sendiri, selain itu juga untuk menghindari penulisan beberapa argument yang sama berulang-ulang.

Fungsi dalam c++ dapat dipakai secara rekursi. Artinya suatu fungsi dapat memanggil fungsi yang merupakan dirinya sendiri (memanggil dirinya sendiri). Fungsi seperti ini dapat kita terapkan, contohnya untuk menghitung nilai factorial dari suatu bilangan (n):

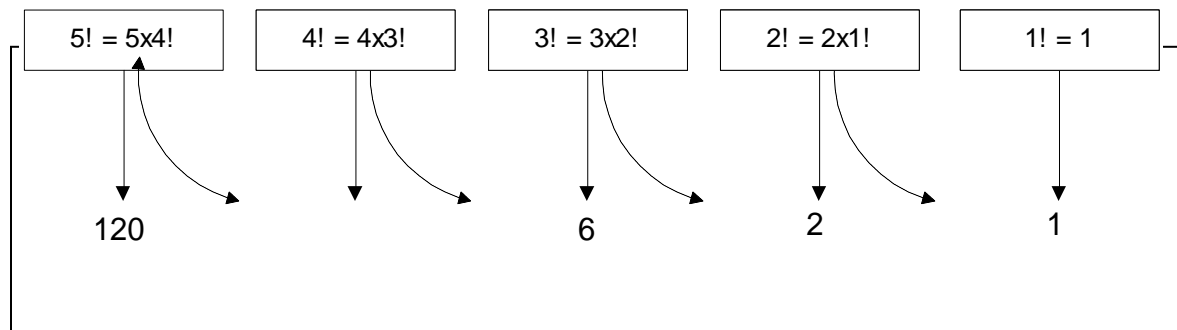
$$n! = n \times (n-1) \times (n-2) \times \dots \times 1$$

dengan n berupa bilangan bulat positif ($n > 0$, $n \in \text{bulat}$). Solusi dari persoalan ini berupa:

$$\text{JIKA } n = 1 \text{ MAKA } n! = n$$

$$\text{SELAIN ITU } n! = n \times (n-1)$$

Dari solusi diatas jika kita ambil contoh $n = 5$, maka dapat digambarkan sebagai berikut:



Dari gambar di atas dapat kita lihat bahwa nilai $5!$ adalah:

- $1! = 1$
- $2! = 2 \times 1! = 2$
- $3! = 3 \times 2! = 3 \times 2 = 6$
- $4! = 4 \times 3! = 4 \times 6 = 24$
- $5! = 5 \times 4! = 5 \times 24 = 120$

Sehingga dapat ditulis : $5 \times (4 \times (3 \times (2 \times 1))) = 120$

Penerapan Program

Tulislah program di bawah ini pada program Editor :

```

//program factorial;

#include<stdio.h>
#include<conio.h>

long int Faktorial (int n);

main( )
{
    Clrscr();
    Gotoxy(25,6);
    Printf("MENGHITUNG NILAI FAKTORIAL n!\n\n");
    Printf("Masukkan Suatu Bilangan Positif:");
    Int bil;scanf("%d",&bila);
    Printf("n\n\n");
    Printf("Maka Faktorial %d(%d!) adalah %d",bil,bil,Faktorial(bil));
    Getche();
}

long int Faktorial(int n)
{
    if (n == 1) return(n*Faktorial(n-1));
}

```

Lakukanlah langkah-langkah berikut dan catatlah hasilnya sebagai Resume anda pada hari ini:

1. Gambarkanlah aliran program (flow chart) program di atas?
2. Kalau kita pikirkan / amati dalam program tersebut terdapat proses looping,sebutkan bagian mana yang mengalami proses looping,dan apa sebabnya sampai bias terjadi proses looping!

3. (Dari jawaban soal No. 2) Sebutkan komponen manakah yang mengatur proses looping tersebut, sehingga proses looping dapat berhenti sesuai dengan yang kita targetkan!

4. Ubahlah isi fungsi Faktorial:

```
Long int Faktorial(int n)
{
    if(n == 1) return(n)
    else return(n*Faktorial(n-1));
}
```

menjadi

```
long int Faktorial(int n)
{
    return(n*Faktorial(n-1))
}
```

lalu jalankan program, catatlah apa yang terjadi, dan jelaskan mengapa bias demikian?

5. Rubahlah program diatas menggunakan fungsi looping for (fungsi rekursi diganti fungsi looping for), dengan menampilkan hasil perhitungan pada tiap proses looping!

6. Dengan menggunakan fungsi rekursi buatlah program menampilkan nilai perpangkatan antara dua bilangan bulat positif A^b !

Tugas Rumah

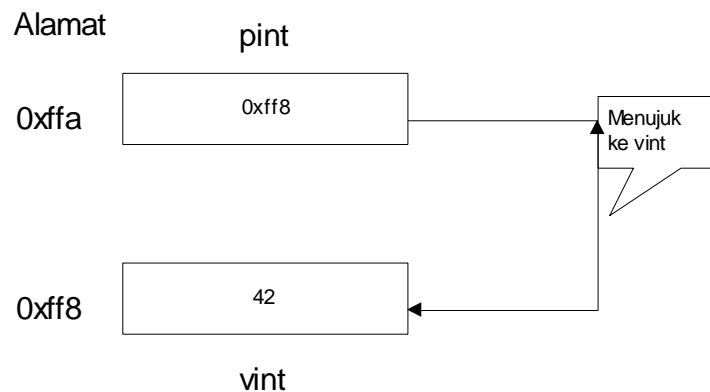
1. Sebutkan perbedaan variable dan pointer!
2. Bagaimana cara mengetahui alamat dari suatu variable?
3. Bagaimana cara mengetahui alamat dari suatu variable?
4. Berilah contoh bagaimana mengisi variable pointer!
5. Bagaimanakah cara mengakses nilai yang ditunjuk pointer?

BAB VII

POINTER

Konsep dari pointer sebenarnya tidak sekompleks yang diduga oleh beberapa orang. Pointer sesungguhnya berisi alamat dari suatu data, bukan data sebagaimana pada variable yang telah anda kenal.

Seperti diketahui, setiap byte di dalam memori computer memiliki sebuah alamat. Alamat memori dimulai dari 0. Pada computer yang memiliki memori 640 Kbyte, alamat memori tertinggi adalah 655.359. Di dalam memori inilah variable disimpan. Tetapi tentu saja pemrogram tidak perlu menyebutkan alamat dari suatu variable secara eksplisit. Pada saat program dimuat didalam memori, variable akan diletakkan dengan sendirinya pada alamat tertentu. Suatu pointer di dalam program dimaksudkan untuk menunjuk ke suatu alamat memori:



Mengetahui Alamat Suatu Variabel

Alamat dari suatu variable dapat diketahui dengan mudah. Caranya tambahkan operator alamat, berupa symbol &, di depan nama variable. Dengan mengirimkan ke count, alamat dari suatu variable akan di tampilkan ke layar.

Mengedefinisikan Variabel Pointer

Suatu variable pointer didefinisikan dengan bentuk sebagai berikut:

Sintaks : tipe_data *nama_variabel;

- Tipe_data dapat berupa sembarang tipe seperti halnya pada pendefinisian variable bukan pointer.
- Nama_variabel adalah nama variable pointer

Contoh

```
Int*pint;    //pointer ke int
Chart*pch;  //pointer ke char;    float*pfl    //pointer ke float
```

Beberapa variable pointer dapat didefinisikan dengan sebuah pernyataan. Misalnya:

```
Char*pch1,*pch2;
```

Adalah mendefinisikan:

- Pch1 : pointer yang menunjuk ke data bertipe char
- Pch2: pointer yang menunjuk ke data bertipe char

Mengisi variable pointer

Supaya suatu variable pointer menunjuk ke variable lain, mula-mula harus di isi dengan alamat dari variable yang hendak ditunjuk. Misalnya terdapat pendefinisian sebagai berikut:

```
Int vint = 55;
```

```
Int *pint;
```

Variable pointer pint dapat diatur agar menunjuk ke vint dengan cara sebagai berikut:

```
Pint = &vint;
```

Pernyataan diatas berarti : pint diisi dengan alamat dari vint”. Operator & menyatakan “alamat dari”.

Mengakses nilai yang ditunjuk pointer

Cara mengakses nilai yang ditunjuk oleh pointer adalah dengan melibatkan operator “tidak langsung”. Operator ini berupa symbol* dan diletakkan di depan nama variable pointer.

Contoh:

```
*pint
```

kode diatas berarti “nilai yang ditunjuk oleh pint”.

Contoh program:

Tulislah program dibawah ini pada program editor

```
#include<conio.h>
```

```
main()
```

```
{
```

```
int bi11 = 123
```

```
float bil12 = 12.34;
```

```

    double bi13 = 12.345678;
clrscr();
cout<<"isi variable : "<< endl;
cout<<"bi11 = "<<bi11 << endl;
cout<<"bi12 = "<<bi12 << endl;
cout<<"bi13 = "<<bi13 << endl;
cout<< "\nAlamat Variabel: "<<endl
cout<<"bi11 = "<<bi11 << endl;
cout<<"bi12 = "<<bi12 << endl;
cout<<"bi13 = "<<bi13 << endl;
cout<<"\nMasukkan Sebuah bilangan pecahan : ";
float vpecahan;
ppecahan = &vpecahan;
cout << "\nNilai variable (vpecahan)yang anda isikan : "<< vpecahan<<
endl;
cout << "Alamat memori variable vpecahan:"<<ppecahan<<endl;
getche()
}

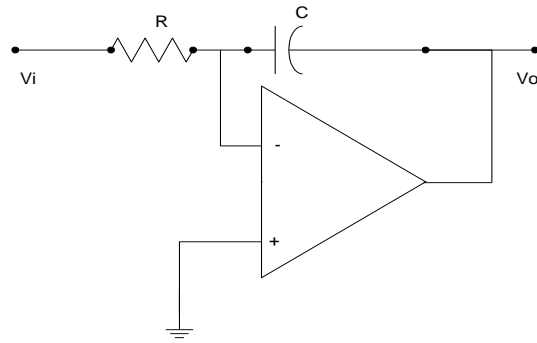
```

Lakukanlah langkah-langkah dibawah ini dan catatlah hasil pada resume anda:

1. jalankan program,catatlah pesan error yang terjadi,lalu betulkan program sampai bias dijalankan dengan benar,dan catatlah bagian yang perlu ditambahkan!
2. jelaskan fungsi cout() pada program diatas,apakah perbedaannya dengan fungsi printf()?
3. jelaskan fungsi cin() pada program di atas,apakah perbedaannya dengan fungsi scanf()?
4. dengan melihat hasil program,sebut fungsi perintah endl?
5. simpanlah program anda dengan nama Lat7!

Tugas Rumah

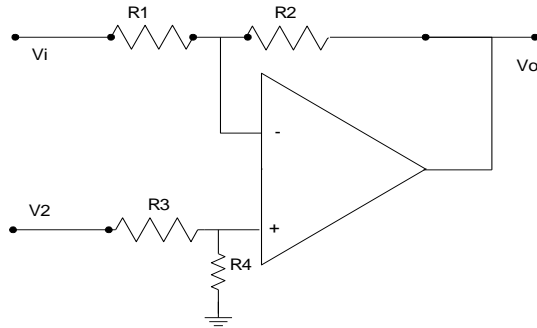
Di bawah ini adalah sebuah rangkaian OP-Amp:



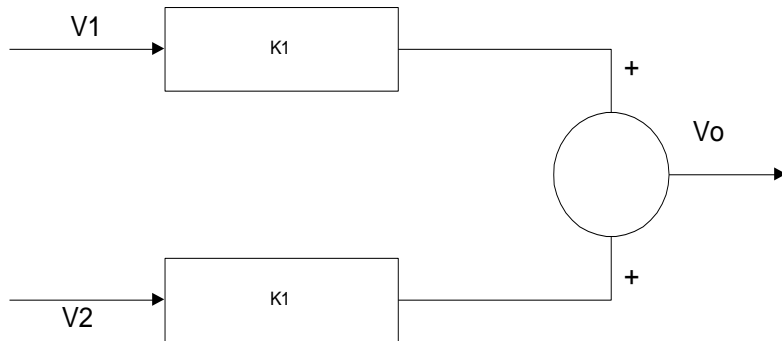
1. Sebutkan jenis OP AMP dari gambar rangkain di atas!
2. Carilah rumus penguatan (gain) dari rangkaian di atas!
3. Hitunglah nilai tegangan keluaran (v_o) jika diketahui nilai $R = 12 \text{ K ohm}$, dan nilai $C = 7,4 \text{ nFarad}$, serta tegangan masukan $v_i = 12 \text{ Volt}$!

UNIT VIII
 APLIKASI UNTUK APLIKASI
 OP AMP

Di bawah ini adalah salah satu gambar rangkain adder Op Amp. Dimana nilai tegangan keluaran V_o adalah merupakan penjumlahan V_1 dan V_2



Gambar rangkaian di atas dapat diubah menjadi block diagram berikut:



Dari blok diagram di atas maka dapat dituliskan rumus-rumus sebagai berikut:

Nilai-nilai penguatan(gain) K_1 dan K_2 adalah sebagai berikut:

$$K_1 = \frac{R_2}{R_1} \text{ dan } K_2 = \left(\frac{R_1 + R_2}{R_1} \right) \left(\frac{R_4}{R_3 + R_4} \right)$$

sehingga setelah penguatan diketahui maka nilai tegangan keluaran V_o dapat dihitung dengan rumus:

$$V_o = K_1 \cdot V_1 + K_2 \cdot V_2$$

Tulislah program untuk menghitung penguatan (gain) serta nilai tegangan keluaran di bawah ini pada program editor:

```
#include<stdio.h>
#include<conio.h>
#include<ctype.h>

main()
{
    char ulang = 'Y';
    while(toupper(ulang)=='Y')
    {
        float R1,R2,R3,R4,K1,K2,Vo,V1,V2;
        clrscr();
        printf("Masukan Nilai R1 (Ohm)=");
        scanf("%f",&R1);
        printf("Masukan Nilai R2 (Ohm)=");
        scanf("%f",&R2);
        printf("Masukan Nilai R3 (Ohm)=");
        scanf("%f",&R3);
        printf("Masukan Nilai R4 (Ohm)=");
        scanf("%f",&R4);

        K1 = R2/R1
        K2 = ((R1+R2)/R1)*(R4/(R3+R4));
        Printf("\n\n\nNilai Penguatan K1 adalah %f\n",K1);
        Printf("Dan Nilai Penguatan K2 adalah %f\n\n\n",K2);
        Printf("Masukan Nilai Tegangan Masukan V1 = ");
        Scanf("%f",&V1);
        Printf("Masukan Nilai Tegangan Masukan V2 = ");
        Scanf("%f",&V2);
```

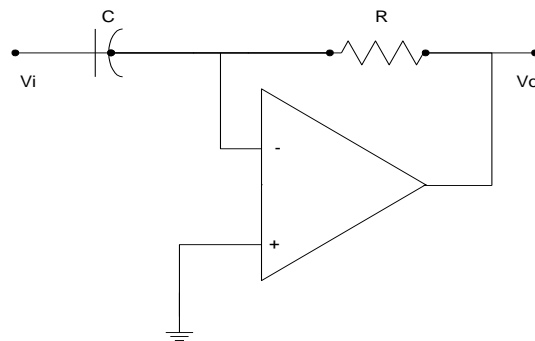
```

    Vo = K1*V1+K2*V2;
    Printf("\nMaka Nilai Tegangan Keluaran Vo adalah %f",Vo);
    Gotoxy(1,20);
    Printf("Apakah akan Mengulang lagi (Y/y):");
    Scanf("%s",&ulang);
}
}

```

Kerjakanlah Perintah di bawah ini :

1. jelaskan program di atas secara rinci,variable yang digunakan beserta tipe datanya, fungsi-fungsi (pernyataan-pernyataan) yang digunakan beserta argument-argumen yang terlibat.
2. Buatlah flowchart dari program di atas!
3. Dari gambar di bawah ini,buatlah program untuk menghitung penguatannya (K) dan nilai tegangan keluaran (Vo), dimana nilai R,C dan V1 sebagai variable masukan!



UNIT IX

PROYEK

(Prasyarat Mengikuti Ujian Akhir)

Pada unit-unit yang lalu anda telah mempelajari beberapa fungsi-fungsi, perintah, atau variable dasar Bahasa Pemrograman Borlan C++

Pada unit ini anda ditugaskan untuk membuat pemrograman (tidak terlalu sederhana, tapi juga tidak terlalu rumit) yang berkaitan dengan dunia keteknikan, yang merupakan aplikasi/ terapan dari semua komponen yang telah anda pelajari pada unit yang lalu.

Ketentuan tentang Pembuatan Program :

Akan disampaikan oleh Supervisor/ asisten pada akhir pertemuan UNIT VIII