4. Pernyataan Dasar

S. Indriani L, M.T

Jenis Pernyataan

Pernyataan berkondisi

- if → *single repetion structure*
- if_else → *double repetion structure*
- nested if, Switch → *multiple selection structure*

Pernyataan berulang

- while
- do_while
- for

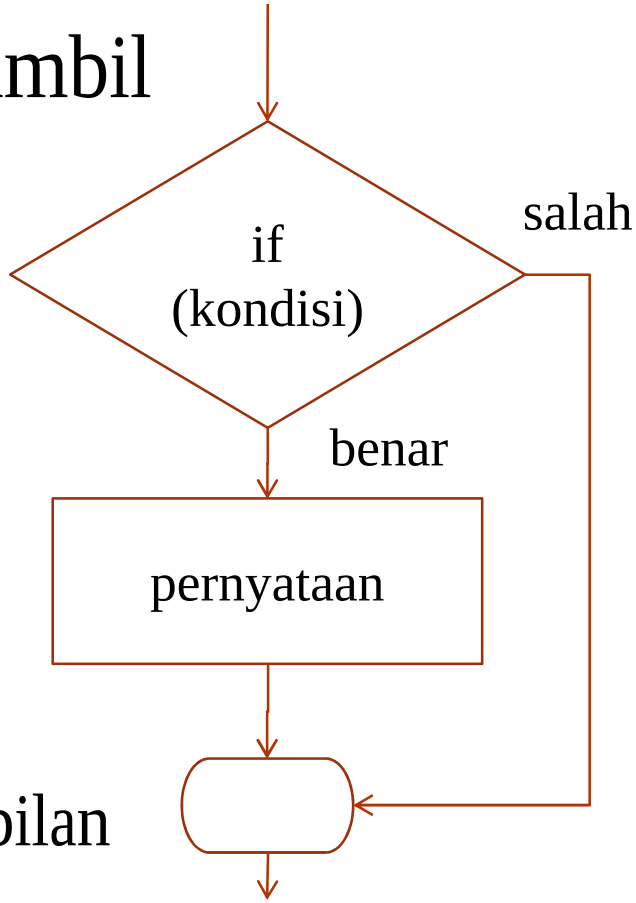
Pernyataan berkondisi

1. If
2. If_else
3. Nested_if, dan switch _case

1. If (Single Repetition Structure)

- Pernyataan *if* dapat dipakai untuk mengambil keputusan berdasarkan suatu kondisi.
- Pernyataan *if* yang paling sederhana berbentuk:

```
if (kondisi)
    pernyataan;
```



  - *Kondisi* digunakan untuk menentukan pengambilan keputusan
  - *Pernyataan* dapat berupa sebuah pernyataan ataupun pernyataan majemuk.
- Bagian ini dijalankan hanya kalau kondisi bernilai benar.

Penerapan if

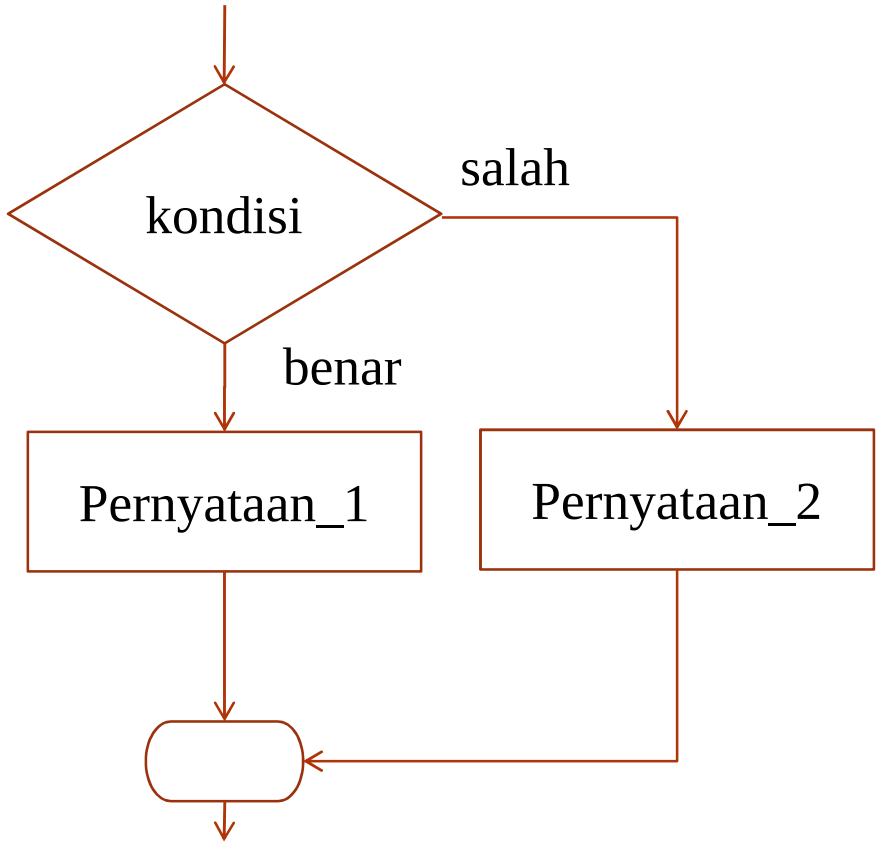
- *Penerapan if misalnya untuk menentukan seseorang boleh menonton pertunjukan bioskop atau tidak. Kondisi yang dipergunakan, seseorang boleh menonton jika sudah berusia 17 tahun.*
- Penulisan programnya sbb:

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int usia;
    clrscr();
    cout<<“Berapa usia anda”;
    cin>>usia;
    if (usia<17)
          cout<<“Anda tidak diperkenankan menonton”<<endl;
}
```

2. If-else (Double Repetion Sructure)

- Pernyataan *else* tidak pernah berdiri sendiri tanpa kehadiran *if*.

```
if (kondisi)
        pernyataan_1;
else
        pernyataan_2;
```



Kemungkinan lain...

- Kemungkinan 1

```
if (kondisi)
     pernyataan_1;
else
     pernyataan_2;

pernyataan_3;
```

- Kemungkinan 2

```
if (kondisi)
     pernyataan_1;
     pernyataan_2;
else
     pernyataan_3;
```

- Kemungkinan 3

```
if (kondisi)
     pernyataan_1;
     pernyataan_2;
else
     pernyataan_3;
     pernyataan_4;
```

Penerapan *If-else*

- Penulisan programnya sbb:

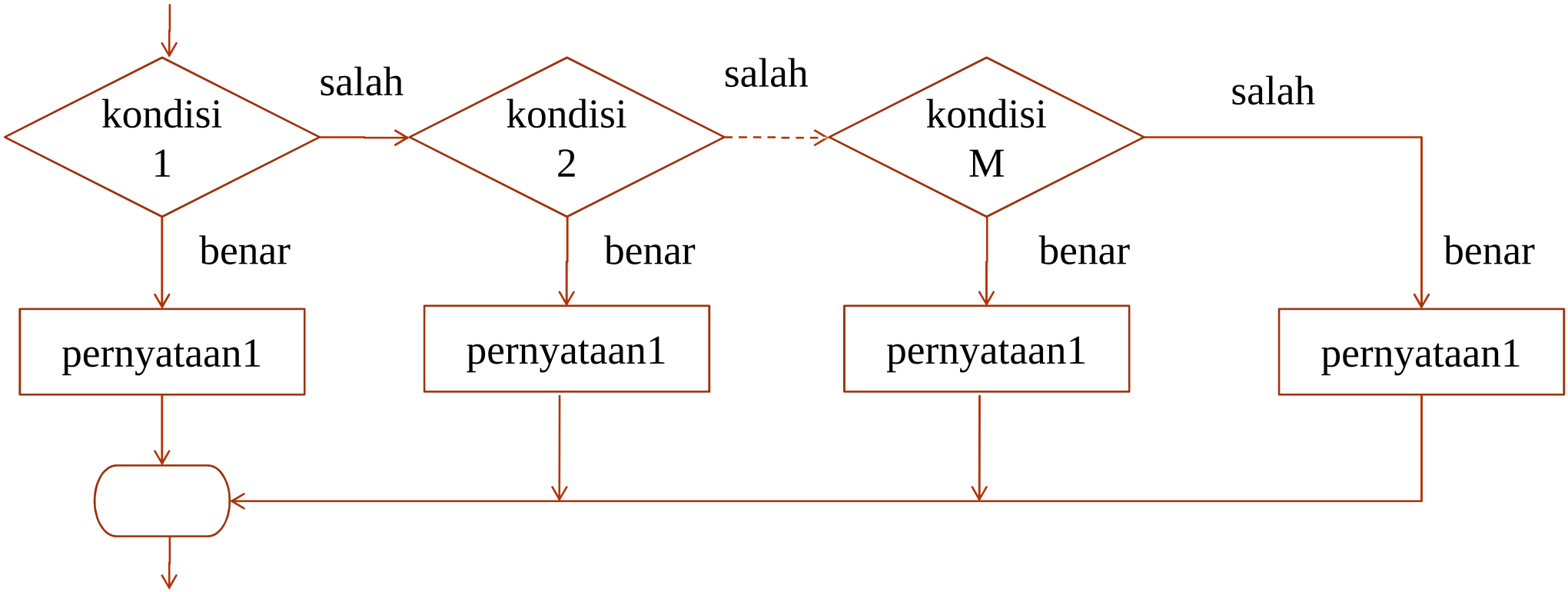
```
#include<iostream.h>
#include<conio.h>
void main()
{
    int usia;
    clrscr();
    cout<<“Berapa usia anda”;
    cin>>usia;
    if (usia<17)
         cout<<“Anda tidak diperkenankan menonton”<<endl;
    else
         cout<<“Selamat menonton”<<endl;
}
```

3. If-else if, switch-case (Multiple repetion structure)

a. if-else if

- Pernyataan if yang terletak didalam if sering disebut *nested if* atau *if bersarang*:

```
if (kondisi_1)
        pernyataan_1;
else if (kondisi_2)
        pernyataan_2;
else if (kondisi_3)
        pernyataan_3;
else if (kondisi_M)
        pernyataan_M;
else                            /*opsional*/
        pernyataan_N;           /*opsional*/
```


kondisi
1
salah
kondisi
2
salah
kondisi
M
salah
benar
benar
benar
benar
pernyataan1
pernyataan1
pernyataan1
pernyataan1

Contoh program

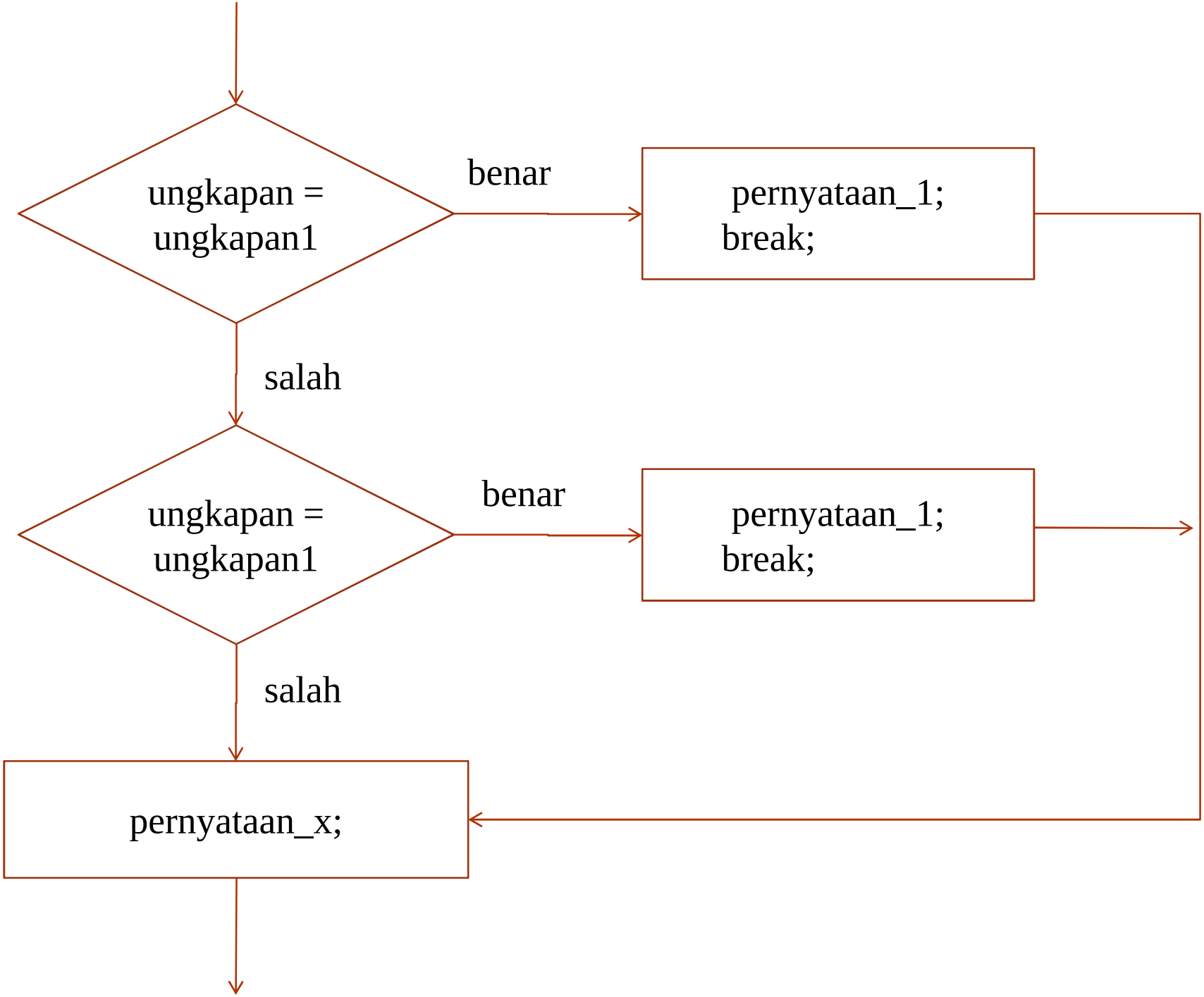
```
#include<iostream.h>
#include<conio.h>
void main()
{
  int usia;
  clrscr();
  cout<<"Berapa usia Anda : “;
  cin>>usia;
  if(usia<17)
      cout<<"Anda belum diperbolehkan untuk mengambil SIM";
  else if(usia>=17)
      cout<<"Anda diperbolehkan untuk mengambil SIM";
}
```

b. Switch_case

- Kaidah umum pernyataan switch:

```
switch (ungkapan)
{
    case ungkapan_1;
            pernyataan_1;
            break;
    case ungkapan_2;
            pernyataan_2;
            break;
    …
    default:                    /*opsional*/
            pernyataan_x;       /*opsional*/
}
```

Perlu diketahui!
Bagian default hanya akan dijalankan kalau ungkapan pada bagian case tidak ada yang cocok dengan ungkapan switch.


ungkapan =
ungkapan1
benar
pernyataan_1;
break;
salah
ungkapan =
ungkapan1
benar
pernyataan_1;
break;
salah
pernyataan_x;

```
//contoh program switch_case

#include<iostream.h>
#include<conio.h>
void main()
{
  int x;
  clrscr();
  cout<<"masukkan angka:”;
  cin>>x;
  switch (x)
  {
      case 1:
            cout<<"x adalah 1";
            break;
      case 2:
            cout<<"x adalah 2";
            break;
      default:
      cout<<"tidak terindentifikasi";
  }
}
```

Latihan

- Susun program untuk menginput 3 buah nilai integer (nilai ujian). Cetak perkataan “DITERIMA”, bila ketiga nilai tersebut berkategoti “lulus” (>=60). Selainnya bila ada nilai yang “tidak lulus”, maka cetak perkataan “TIDAK DITERIMA”.

Pekerjaan Rumah

1. Susun program untuk menginputkan sebuah nilai interger (nilai ujian mahasiswa)

   Anggap nilai yang diinput berada dalam batas (0-100), artinya tidak dibawah 0 dan tidak diatas 100

Cetak huruf	Bila nilai berada dalam batas:
A	80-100
B	70-79
C	60-69
D	40-59
E	0-39
Nilai yang anda masukkan salah	>100

2. Buatlah flowchart untuk soal diatas!

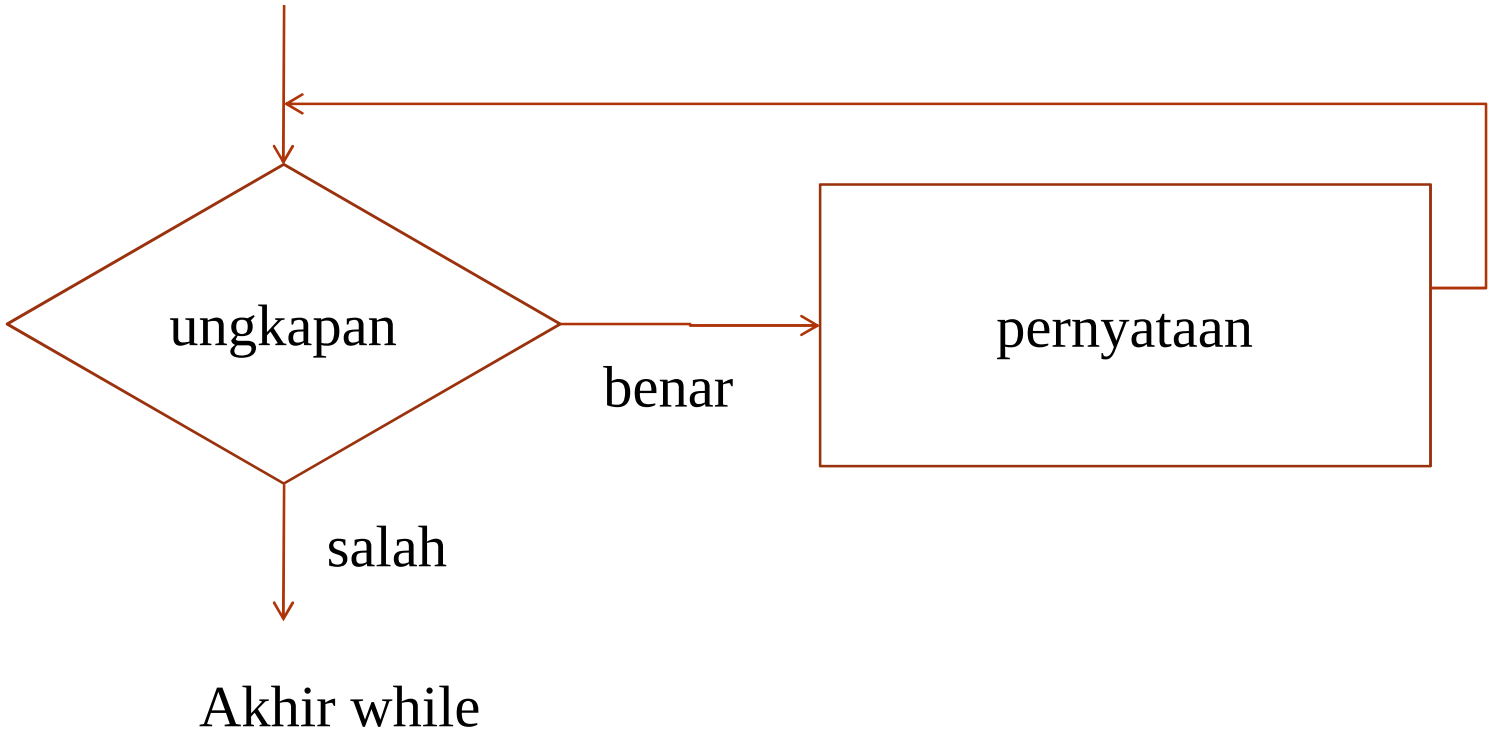
Pernyataan berulang

1. while
2. do_while
3. for

1. While

- Pernyataan while merupakan salah satu pernyataan yang berguna untuk memproses suatu pernyataan atau beberapa pernyataan beberapa kali
- Bentuk pernyataan ini:

```
while (ungkapan)
{
   pernyataan_1;
   pernyataan_2;
   …
   pernyataan_N;
}
```



- Kapan pernyataan while digunakan?

  pedomannya adalah jika anda bermaksud menjalankan suatu proses secara berulang.

Contoh program

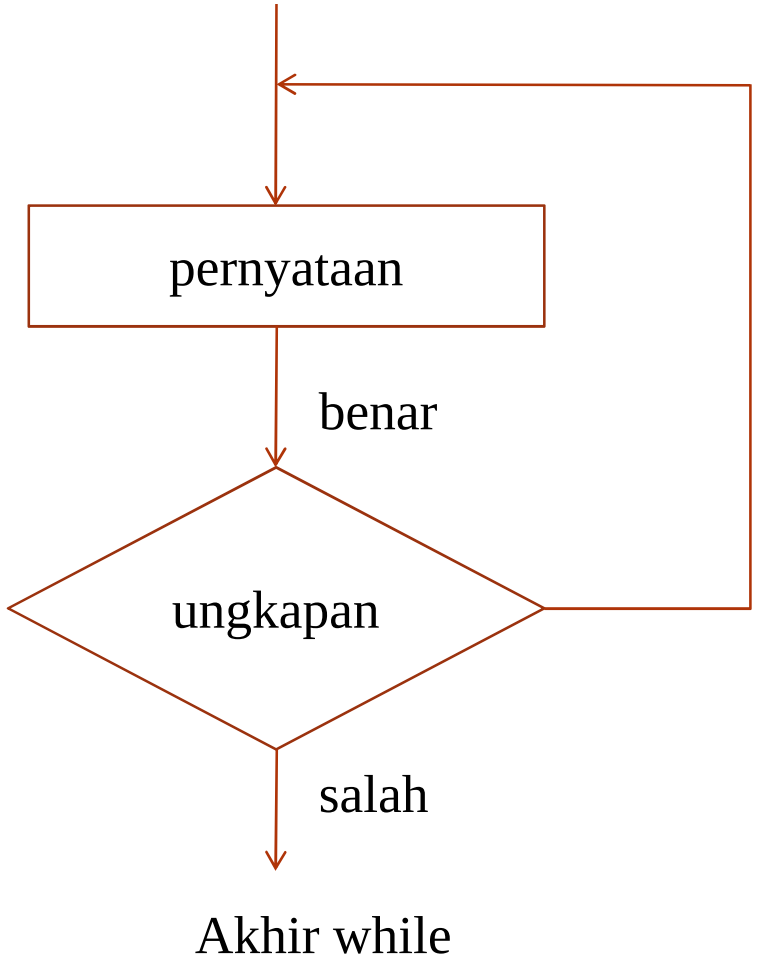
- *Sebuah gambaran pemakaian while untuk menampilkan tulisan C++ sebanyak 10 kali.*

```
#include<iostream.h>
#include<conio.h>
void main()
{
  int i;
  clrscr();
  i=0;
  while (i<10)
  {
    cout<<“C++”<<endl;
    i++;
  }
}
```

2. do-while

- Pernyataan do_while juga berguna untuk mengulang proses.
- Bentuk pernyataan:

```
do
{
    pernyataan_1;
    pernyataan_2;
    …
    pernyataan_N;
}
    while(ungkapan);
```



Contoh program

```
#include<iostream.h>
#include<conio.h>
void main()
{
  int i;
  clrscr();
  i=0;
  do
  {
     cout<<"C++"<<endl;
     i++;
  }
  while (i<10);

}
```

3. for

- Pernyataan for juga berguna untuk mengulang pengeksekusian terhadap satu atau sejumlah pernyataan.
- Bentuknya sebagai berikut:

```
for(inisialisasi; kondisi;counter)
    {    pernyataan;
    }
```

- Pernyataan akan diulangi jika kondisi bernilai true
- *Inisialisasi*: akan dieksekusi. Biasanya merupakan variabel yang akan dipakai sebagai counter atau pencacah. Bagian ini hanya dieksekusi hanya sekali.
- *Kondisi*: akan diperiksa, jika bernilai true maka perulangan akan dilanjutkan dan jika bernilai false maka perulangan akan dilewati.
- *Pernyataan:* akan dieksekusi.

Contoh program

```
#include<iostream.h>
#include<conio.h>
void main()
{
  int i;
  clrscr();
  for(i=0;i<10;i++)
     {
        cout<<"C++"<<endl;
     }
}
```

Perhatikan cara penulisan
for(*inisialisasi; kondisi;counter*)

Nested Loops (Perulangan bertumpuk)

- Perulangan bertumpuk secara sederhana dapat diartikan: terdapat satu atau lebih loop didalam sebuah loop.
- Banyaknya tingkatan perulangan, tergantung dari kebutuhan.
- Biasanya, nested loops digunakan untuk aplikasi matematika yang menggunakan baris dan kolom → loop luar biasanya digunakan digunakan untuk mendefinisikan baris, loop dalam digunakan untuk mendefinisikan kolom.

Contoh program

```
#include<iostream.h>
#include<conio.h>
void main()
{
  clrscr();
  for(int baris=1;baris<=4;baris++)
  {
     for (int kolom=1;kolom<=5;kolom++)
     {
          cout<<kolom<<"  ";
     }
     cout<<endl;
  }

}
```

Latihan

1. Buatlah program dengan tampilan sbb:

   PBO

   PBO

   PBO

   PBO

   PBO

2. Buatlah program untuk menampilkan:

   ==========

   Pemograman berorientasi objek

   ==========

Pekerjaan Rumah

- Buatlah program untuk membuat tampilan sbb:

  10 9 8 7 6 5 4 3 2 1 STOP!

  GUNAKAN PERINTAH WHILE, DO_WHILE dan FOR!

jawaban

```
#include<iostream.h>
#include<conio.h>
void main()
{
   int i;
   clrscr();
   i=10;
   while(i>0)
   {
        cout<<i<<" ";
        i--;
   }cout<<"STOP!";
}
```

```
#include<iostream.h>
#include<conio.h>
void main()
{
   int i;
   clrscr();
   i=10;
   do
        {
        cout<<i<<" ";
        i--;
   }
   while(i>0);
   cout<<"STOP!";
}
```

```
#include<iostream.h>
#include<conio.h>
void main()
{
   int i;
   clrscr();
   for (i=10;i>0;i--)
        {
        cout<<i<<"  ";
        }
        cout<<"STOP";
}
```