

4

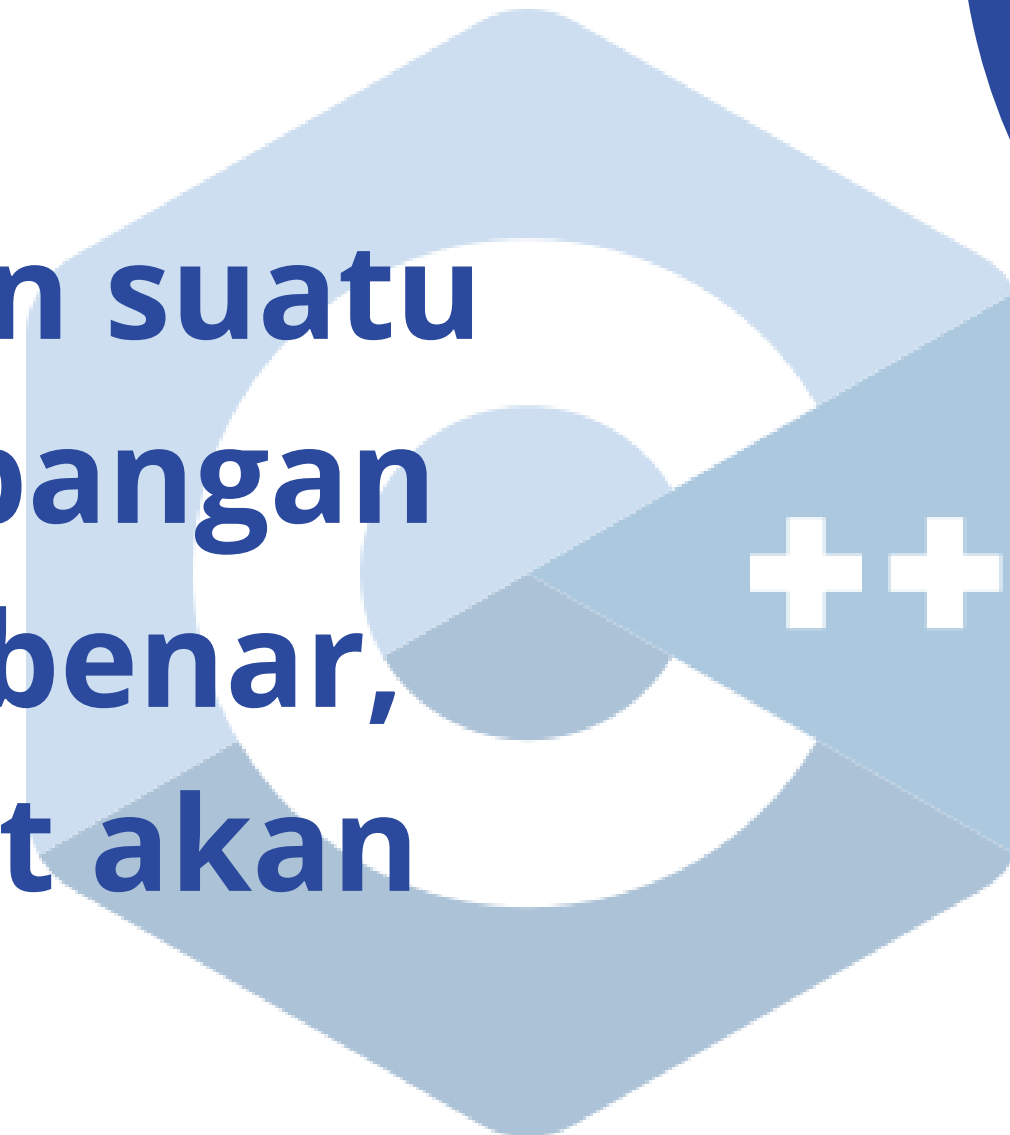


SWITCH CASE



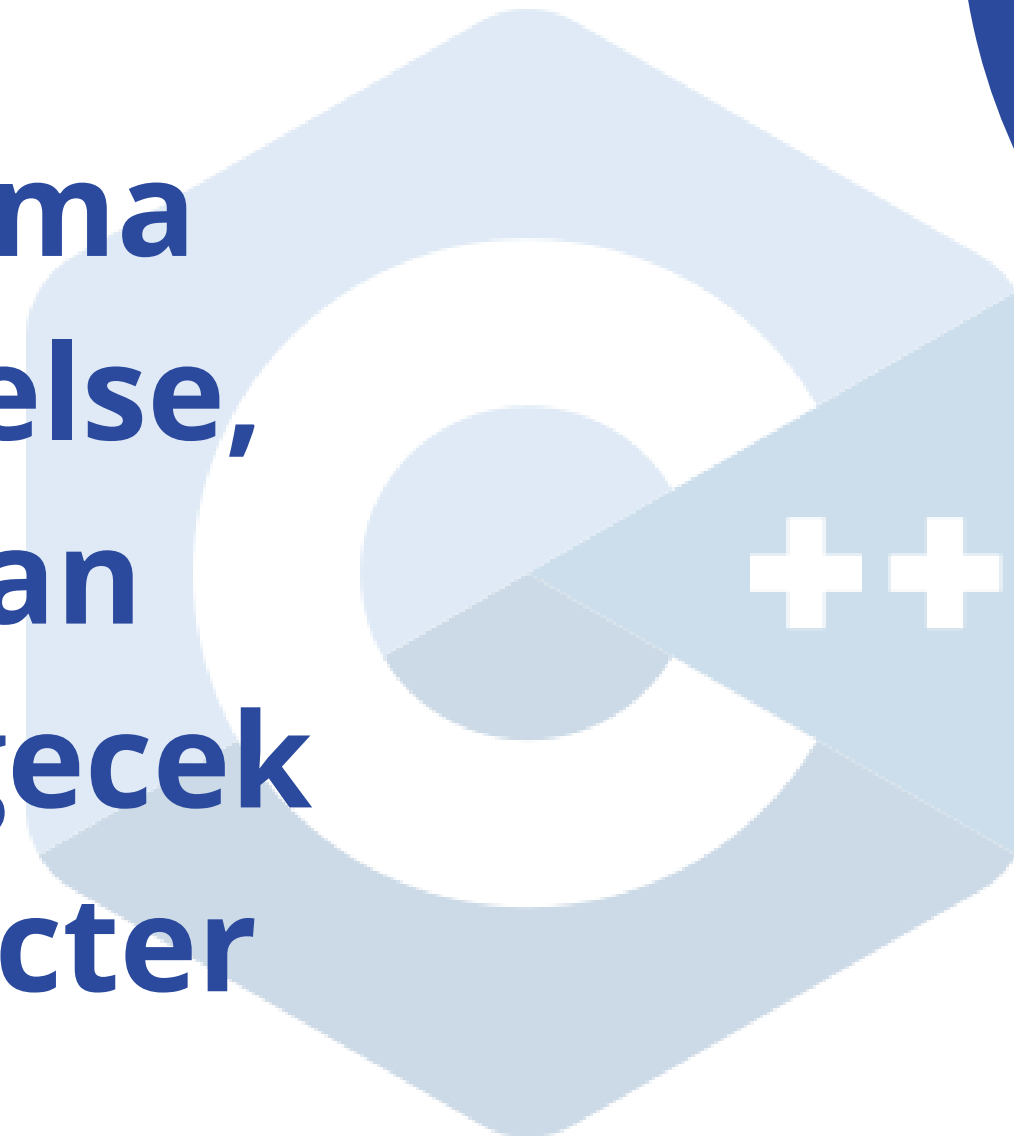
SWITCH CASE

Switch case merupakan suatu bentuk kondisi percabangan jika kondisi tersebut benar, maka kondisi tersebut akan di eksekusi.



SWITCH CASE

Switch case hampir sama dengan penggunaan if else, hanya saja penggunaan switch case untuk mengecek data dengan tipe character ataupun integer.



BENTUK UMUM SWITCH CASE

```
Switch (pernyataan)
{
    case nilai_konstan1:
        blok_pernyataan1;
        break;
    case nilai_konstan2:
        blok_pernyataan2;
        break;
    ...
    ...
    default:
        blok_pernyataan n;
}
```



CONTOH

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    cout<<"masukkan angka "; cin>>a;
    switch(a==5){
    case true:
    cout<<"angka benar";
    }
    return 0;
}
```



CONTOH

```
#INCLUDE<Iostream>
using namespace std;
int main()
{
    int A;
    cout<<"1. NASI GORENG "<<endl;
    cout<<"2. MIE GORENG "<<endl;
    cout<<"3. BIHUN GORENG "<<endl;
    cout<<"MASUKKAN PESANAN ANDA ";cin>>A;
    switch(A){
    case 1:
        cout<<"ANDA MEMESAN NASI GORENG";
        break;
    case 2:
        cout<<"ANDA MEMESAN MIE GORENG";
        break;
    case 3:
        cout<<"ANDA MEMESAN BIHUN GORENG";
    }
    return 0;
}
```

STRUKTUR PENGULANGAN(LOOPING)

Looping merupakan suatu proses yang dilakukan secara berulang sampai memenuhi kondisi yang diinginkan. Looping juga digunakan untuk menghasilkan perulangan sejumlah N kali yang dispesifikasikan.



LOOPING

Looping menyediakan 3 buah perintah

- **For**

For digunakan untuk mengulang suatu proses yang telah diketahui jumlahnya.

- **While**

Untuk mengulang suatu proses yang belum diketahui jumlahnya. Pengecekan kondisi akan dilakukan terlebih dahulu. Jika kondisi masih bernilai true, maka looping akan terus berlanjut. Statement WHILE juga digunakan untuk menyatakan perulangan.

LOOPING

- **Do-While**

Do-While sama seperti while, melakukan perulangan walaupun belum diketahui jumlahnya. Instruksi akan dijalankan lebih dahulu, kemudian dilakukan pengecekan kondisi apabila masih bernilai true maka looping akan terus berlanjut

BENTUK UMUM FOR

for (inisialisasi; kondisi; increment/decrement)

```
{  
aksi;  
}
```

contoh :

```
#include<iostream>  
using namespace std;  
int main(){  
    for(int i=0; i<10; i++){  
        cout<<"hello world"<<endl;  
    }  
}
```



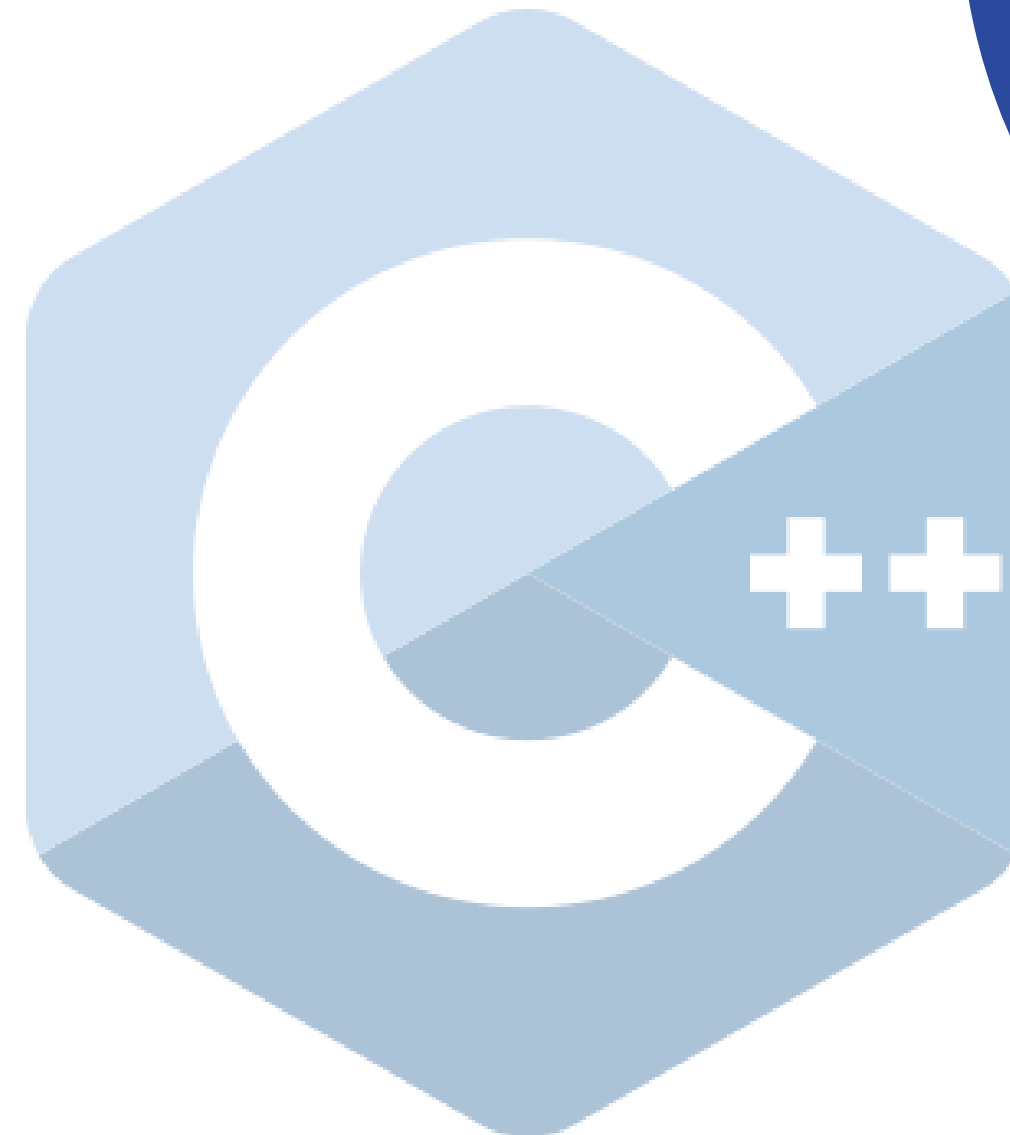
BENTUK UMUM WHILE

```
while(Kondisi)
```

```
{  
aksi;  
}
```

contoh :

```
#include<iostream>  
using namespace std;  
int main(){  
int i=0;  
while (i<9){  
cout<<(i+1)<<" ";  
i++;}  
return 0;  
}
```



BENTUK UMUM DO-WHILE

```
do{  
aksi;  
}  
while(kondisi);
```

contoh :

```
#include<iostream>  
using namespace std;  
int main(){  
int i=0;  
do{  
cout<<(i+1)<<" ";  
i++;  
}  
while (i<9) ;  
return 0;  
}
```



MENAMPILKAN BILANGAN GENAP

```
#include<iostream>
using namespace std;
int main(){
    for(int i=0; i<10; i++){
        if(i%2==1)continue;
        cout<<i<<endl;
    }
    return 0;
}
```

