# BAB III

# PENGKONDISIAN

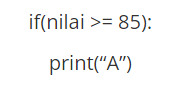
## 3.1. Tujuan

1. Praktikan dapat memahami konsep pengkondisian.
2. Praktikan dapat mengetahui penggunaan pengkondisian dengan bahasa pemrograman Python, C++, Java, dan PHP.
3. Praktikan mampu mengimplementasikan pengkondisian dalam bahasa pemrograman Python, C++, Java, dan PHP.
4. Praktikan mampu memahami fungsi pengkondisian dalam bahasa pemrograman Python, C++, Java, dan PHP.
5. Praktikan mampu membedakan pengkondisian dalam bahasa pemrograman Python, C++, Java, dan PHP.
6. Praktikan mampu memahami berbagai statement untuk pengkondisian dalam bahasa pemrograman Python, C++, Java, dan PHP.
7. Praktikan mampu menentukan statement pengkondisian untuk membuat program tertentu dalam bahasa pemrograman Python, C++, Java, dan PHP.

## 3.2. Dasar Teori

### 3.2.1. If

*If* adalah *logical block* yang digunakan dalam pemrograman. Dengan kata lain *if* adalah pernyataan bersyarat yang menjalankan program secara fleksibel berdasarkan *conditional expression* yang berupa tipe data *boolean.* *Statement if* digunakan dalam percabangan pada pengkondisian. Berikut adalah contoh *statement if* pada bahasa pemrograman Python:



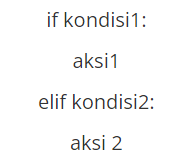
Gambar 3. 1 *Statement if* dalam Python

Sumber: <https://www.dqlab.id/python-percabangan-if-if-else-dan-else-if>

Premis dari statement ini adalah “Jika-Maka”. Sebagai contoh, jika nilai siswa lebih dari atau sama dengan 85, maka predikatnya adalah A.

### 3.2.2. If Else If

Pernyataan *if else* dalam pemrograman adalah pernyataan bersyarat yang menjalankan serangkaian pernyataan yang berbeda tergantung pada apakah suatu ekspresi benar atau salah. Pernyataan *if else* yang khas akan tampak mirip dengan yang di bawah ini (contoh ini adalah JavaScript, dan akan sangat mirip dalam bahasa gaya-C lainnya). Jika pada *statement else* program mampu memilih antara dua aksi, maka pada statement *elif* tidak hanya mengeksekusi jika kondisi *if* di atasnya bernilai salah atau *false*, melainkan akan melakukan pengecekan tambahan pada kondisi kedua yang berada setelah *elif*. Jika kondisi kedua bernilai *true* maka *elif* akan dijalankan. Sintaks umumnya yang paling sederhana adalah sebagai berikut:



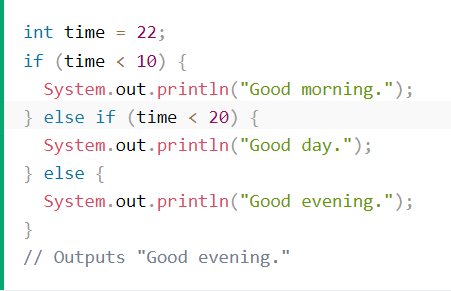
Gambar 3. 2 Statement if else if dalam Python

Sumber: <https://www.dqlab.id/python-percabangan-if-if-else-dan-else-if>

### 3.2.3. If Else If Else

Pada urutan *statement if-else if-else* mirip seperti *if-else if* namun memiliki *else* sebagai kondisi terakhir jika kondisi *else-if* masih belum memenuhi aksi yang ditentukan. Dengan kata lain, *if* sebagai syarat awal, *else* *if* sebagai kondisi syarat kedua dan selanjutnya, lalu *else* sebagai kondisi terakhir di luar kondisi *else if*.

Contohnya sebagai berikut:



Gambar 3. 3 Statement if-else if-else dalam Java

Sumber: <https://www.w3schools.com/java/java_conditions.asp>

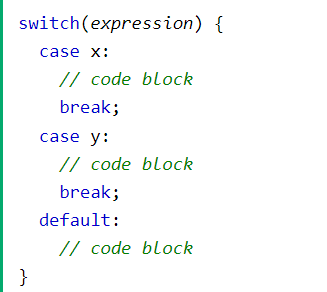
Pada contoh di atas, waktu (22) lebih besar dari 10, sehingga kondisi pertama salah. Kondisi berikutnya, dalam pernyataan if else, juga salah, jadi kita beralih ke kondisi lain karena kondisi1 dan kondisi2 keduanya salah - dan membuat *output* "*Good* *evening*". Namun, jika waktunya 14, karena di bawah 20 maka program akan membuat *output* "*Good* *day*."

### 3.2.4. Switch Case

*Switch case* adalah suatu bentuk kondisi percabangan dimana jika suatu kondisi benar(*true*), maka blok kode bernilai true akan dijalankan. *Switch case* ini hampir sama dengan percabangan yang lain yaitu *if else*, hanya saja penggunaan *switch case* untuk mengecek data yang bertipe karakter dan integer.

Pada percabangan *switch case*, *switch* digunakan untuk menyeleksi suatu kondisi biasanya memasukkan variable misal *switch*(nama\_variabel){}. *Case* digunakan sebagai kondisi-kondisi atau kasus-kasusnya. Jika suatu kondisi bernilai *true* maka blok yang *true* akan ditampilkan, misalnya seperti ini saya mempunyai *switch*(angka){} dan sebelumnya saya suruh *user* untuk memasukkan angka bebas.

Contoh *switch case*:



Gambar 3. 4 Statement switch-case dalam C++

Sumber: <https://www.w3schools.com/cpp/cpp_switch.asp>

Dalam program di atas:

* Ekspresi *switch* dievaluasi sekali.
* Nilai ekspresi dibandingkan dengan nilai setiap *case.*
* Jika ada cocok, maka program akan dieksekusi.
* Kata kunci *break* dan *default* opsional, break untuk menghentikan eksekusi pada *case* tertentu jika cocok, sedangkan *default* menentukan beberapa kode untuk dijalankan jika tidak ada *case* yang cocok.