Laporan Praktikum

Algoritma Pemograman



Disusun Oleh :

NIA RAMADHANI TANJUNG

**(2311531006)**

Dosen Pengampu : Dr. Wahyudi, S.T, M.T

Asisten Labor : Rahmad Dwirizki Olders

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2023

**Struktur Kondisional pada Bahasa Pemrograman Java**

1. **Tujuan Praktikum**
2. Memahami struktur kondisional pada bahasa pemograman Java dengan menggunakan pernyataan **if, else if** dan **else**.
3. **Pendahuluan**

Terdapat banyak bahasa pemrograman yang dapat digunakan untuk membuat aplikasi atau website. Masing-masing bahasa pemrograman memiliki fitur yang berbeda-beda yang berfungsi untuk pengembangan program. Fleksibilitas dan kemudahan pada penerapannya menjadi beberapa alasan pengembang dalam memilih bahasa pemrograman yang sesuai untuk permasalahan yang ada.

Salah satu yang memiliki keunggulan tersebut yaitu bahasa pemrograman Python. Struktur bahasa Python berfokus pada keterbacaan kode sehingga pengembang akan lebih fokus dalam membuat program daripada penyusunan kodenya. Fitur dan librarynya juga banyak dan mendukung pembuatan aplikasi, website, hingga proses data science seperti machine learning. Pengembang bisa menggunakan IDE atau editor kode untuk membuat program menggunakan Python seperti IDLE dan Visual Studio Code.

Dalam membuat program kita akan menuliskan perintah menggunakan fungsi-fungsi yang disediakan bahasa pemrograman Python. Misalnya untuk menampilkan hasil kita bisa menggunakan fungsi print atau untuk mengecek tipe data dengan fungsi type. Masih banyak lagi fungsi lainya dalam Python seperti fungsi yang digunakan untuk kondisi percabangan.

Kondisi ini sangat umum ada pada penulisan kode untuk suatu program. Oleh karena itu, pengembang harus memahami logika percabangan ini.

**1.If Statement**

Bentuk lengkap dari pernyataan **if** adalah:

if(*condition*) *statement;*

else *statement;*

dimana target dari **if** dan **else** adalah pernyataan tunggal. **else** bersifat opsional. Target dari **if** dan **else** dapat berupa blok pernyataan. Bentuk umum dari **if**, dengan menggunakan blok pernyataan adalah

if(*condition*)

{

*statement sequence*

}

else

{

*statement sequence*

}

Jika ekspresi kondisional benar, target if akan dieksekusi; jika tidak, jika ada, target **else** yang akan dieksekusi. Keduanya tidak akan pernah dieksekusi secara bersamaan. Ekspresi kondisional yang mengendalikan **if** harus menghasilkan hasil **boolean**.

### if-else-if Ladder

Konstruksi pemrograman umum yang didasarkan pada *nested i* adalah tangga **if-else-if**. Tangga **if-else-if** terlihat seperti ini:

if(*condition*)

*statement*;

else if(*condition*)

*statement*;

else if(*condition*)

*statement*;

*.*

*.*

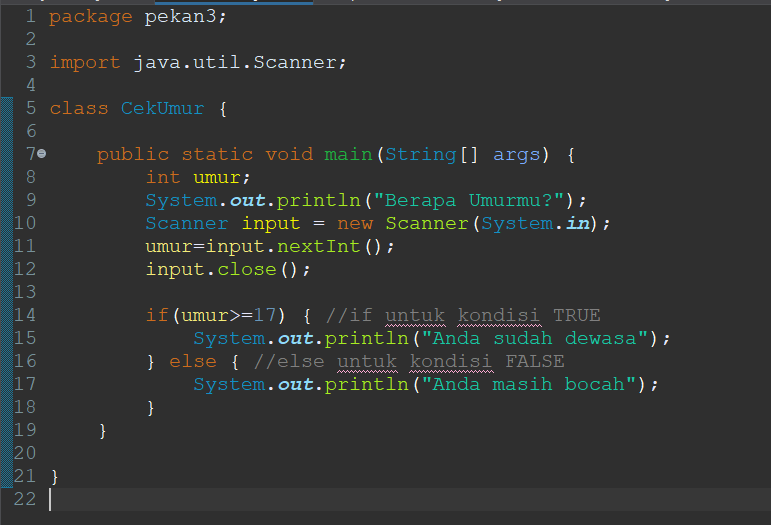
*.*

else

*statement*;

Ekspresi kondisional dievaluasi dari atas ke bawah. Segera setelah kondisi yang benar ditemukan, pernyataan yang terkait dengannya dieksekusi, dan sisa tangga dilewati. Jika tidak ada satupun kondisi yang benar, maka pernyataan **else** terakhir akan dieksekusi. Final **else** sering kali bertindak sebagai *kondisi default*; yaitu, jika semua pengujian kondisional lainnya gagal, pernyataan **else** terakhir akan dijalankan. Jika tidak ada final **else** dan semua kondisi lainnya salah, maka tidak ada tindakan yang akan dilakukan.

1. **Metode Praktikum**
2. **if-else statement**
3. Buat class program seperti biasa.
4. Tulis program berikut :

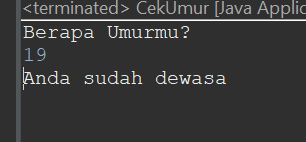


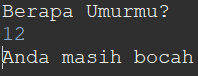
Pada program tersebut, nilai **umur** diinput setelah program dijalankan.

Program tersebut menggunakan pernyataan kondisional **if** dan **else**. Pada baris ke-14, ditulis pernyataan **if**, lalu kondisi diberi setelahnya dengan ditutup kurung biasa, yaitu (umur=>17) yang artinya kondisi **umur** yang diinput besar dari atau sama dengan 17. Selanjutnya, pernyataan yang akan dijalankan berdasarkan kondisi ditulis setelahnya dengan ditutup kurung kurawal. Pada baris ke-16, ditulis pernyataan **else**. pernyataan **else** tidak memberi kondisi khusus setelah dinyatakan, melainkan kondisi **else** dijalankan jika kondisi pada pernyataan **if** bernilai *false*.

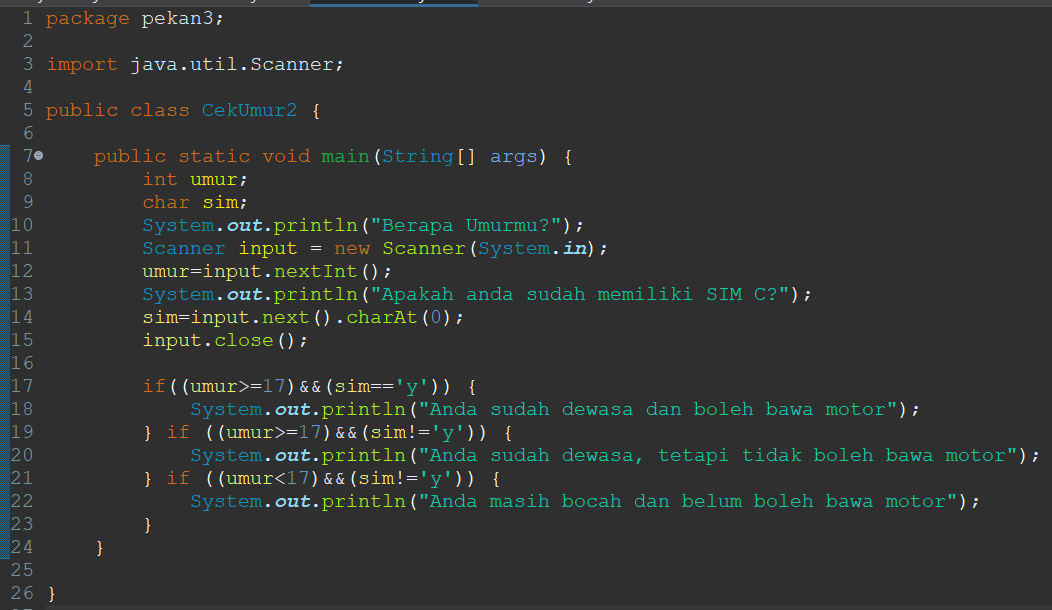
Jadi, jika **umur** yang diinputkan bernilai lebih besar atau sama dengan 17, maka kondisi **if** terpenuhi dan bernilai *true*. Maka, pernyataan-pernyataan di dalam **if** akan dijalankan. Sebaliknya, jika kondisi **if** bernilai *false* atau nilai **umur** yang diinputkan kecil dari 17, maka pernyataan-pernyataan di dalam **else** akan dijalankan.

1. Berikut contoh output dari program tersebut :



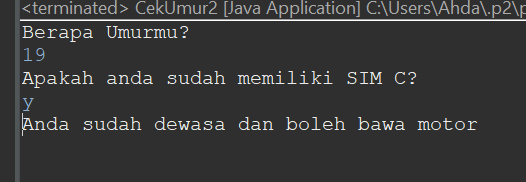


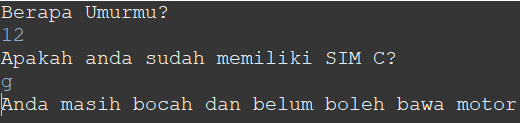
1. **if-if statement**
2. Buat class program seperti biasa.
3. Tulis program berikut :

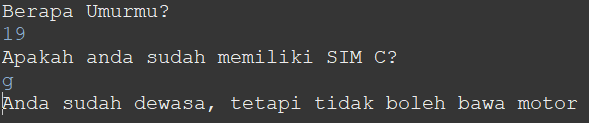


Program tersebut hampir sama dengan yang sebelumnya, hanya pada program ini diberi dua kondisi pada pernyataan **if** yang digabungkan dengan operator logika. Pernyataan **if** pada baris ke-17 memiliki kondisi dimana **umur** yang diinputkan bernilai besar dari atau sama dengan 17 DAN **sim** yang diinput adalah karakter y. Lalu, pernyataan **if** pada baris ke-19 memiliki kondisi dimana **umur** yang diinputkan bernilai besar dari atau sama dengan 17 DAN **sim** yang diinput bukan karakter y. Terakhir, pernyataan **if** pada baris ke-21 memiliki kondisi dimana **umur** yang diinputkan bernilai kecil dari 17 DAN **sim** yang diinput bukan karakter y. Pernyataan yang akan dijalankan bergantung pada **umur** dan **sim** yang diinputkan.

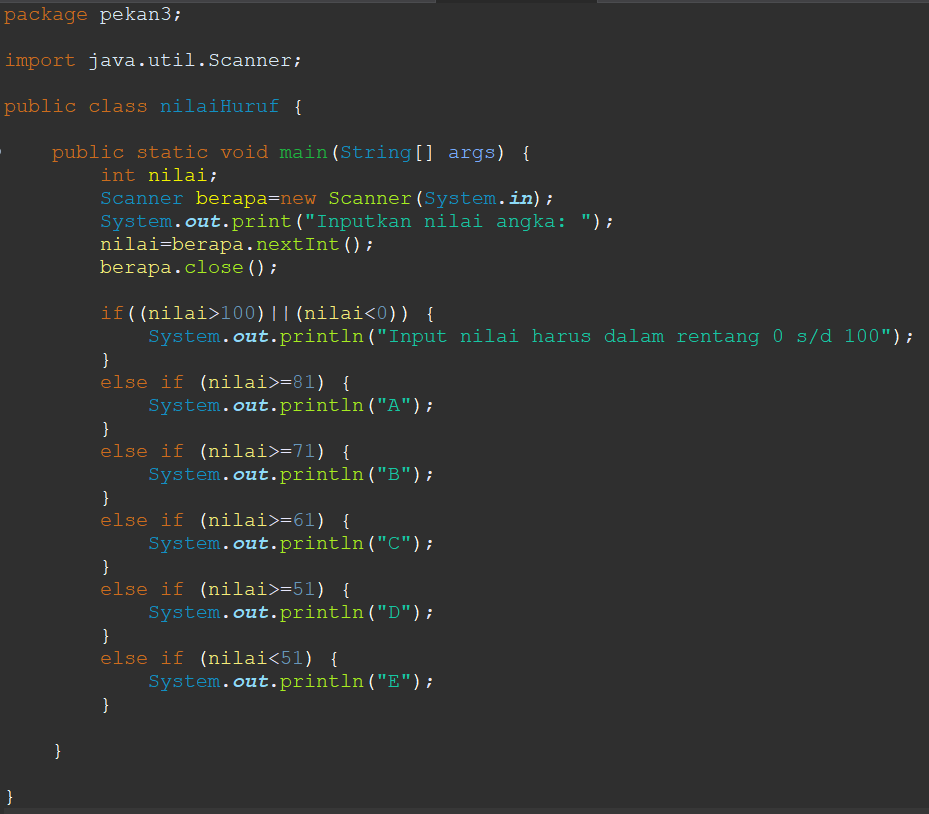
1. Berikut contoh output dari program tersebut:





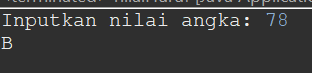


1. **if-else if** **statement**
2. Buat class program seperti biasa.
3. Tulis program berikut :



Program ini menggunakan pernyataan **else if**. Pernyataan **else if** memiliki kondisi, tidak seperti pernyataan **else**, dan kondisi tersebut berbeda dengan kondisi pernyataan **if** yang pertama. Pernyataan-pernyataan dalam **else if** akan dijalankan jika kondisinya terpenuhi, dan kondisi pernyataan **if** atau **else if** sebelumnya bernilai *false*.

1. Berikut contoh output dari program tersebut :



1. **Kesimpulan Praktikum**
2. Java dikenal sebagai bahasa pemrograman yang bersifat strongly, yang berarti diharuskan mendeklarasikan tipe data dari semua variable yang apabila lupa atau salah dalam mengikuti aturan pendeklarasian variable, maka akan terjadi error pada saat proses kompilasi.
3. Tipe data yang terdapat pada bahasa pemrograman Java pada umumnya tidak jauh berbeda dengan bahasa pemrograman lain seperti C, C++, Pascal, Basic, dan lainnya, yakni terdiri dari integer, floating point, char, dan boolean.
4. Aturan penulisan variabel pada bahasa pemrograman Java juga tidak jauh berbeda dengan bahasa pemrograman lain seperti C, C++, Pascal, Basic, dan lainnya, yakni harus diawali dengan huruf/abjad, karakter mata uang, dan underscore ( \_ ) dan terdiri dari huruf/abjad, angka, dan underscore ( \_ ), serta tidak boleh menggunakan kata-kata yang dikenal oleh bahasa pemrograman Java (keyword), seperti byte, case, int, dan lainnya.
5. Sama halnya dengan semua bahasa pemrograman, Java juga menyediakan operator-operator aritmatika untuk memanipulasi data numerik.
6. Selain operator aritmatika, Java juga mempunyai operator-operator lain, seperti operator increment dan decrement, operator assignment (penugasan), operator relasi, operator logical, operator bitwise (yang terdiri dari and, or, xor, dan complement), dan operator precedence.