LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

PERNYATAAN KONDISIONAL



Oleh :

Tiara Azizah

2411533001

MATA KULIAH ALGORITMA PEMROGRAMAN

DOSEN PENGAMPU : DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, DESEMBER 2024

1. Pendahuluan
2. Pernyataan Kondisional

Pernyataan kondisional atau dalam Bahasa Inggrisinya, conditional execution merupakan suatau kondisi tertentu yang bernilai benar atau salah, atau lebih sering disebut dengan ekspresi Boolean. Pernyataan kondisional adalah salah satu dari struktur control di java. Pernyataan ini didasari pada suatu kondisi dan menghasilkan Keputusan yang sesuai.

1. Jenis Pernyataan Kondisional di Java
2. If (Jika)

“Jika” adalah eksekusi pernyataan yang bergantung pada ekspresi boolean tertentu dan menghasilkan nilai Boolean. Kondisi ini hanya mengikuti satu kata kunci, yaiut “If”. Ketika nilai kondisi dinyatakan “Benar”, maka blok pada saat ini akan dieksekusi. Namun, ketika pernyataan bernilai “Salah” akan beralih ke pernyataan berikutnya.

1. If… Else

If… Else adalah suatu kondisi yang bergantung pada pemilihan serangkaian pernyataan yang dipilih. Kondisi ini mengikuti dua jenis kondisi yaitu, “If” dan “Else”. Disaat pernyataan menghasilkan nilai benar, maka pernyataan ini akan dijalankan sesuai dengan blok if. Sebaliknya, ketika nilai pernyataan menghasilkan nilai salah, pernyataan blok ini akan dijalankan pada kondisi “Else” yang bergantung dengan kondisi selanjutnya.

1. Nested If

Pernyataan nested-if atau if  bersarang merupakan pernyataan kondisional yang memiliki pilihan lebih dari satu. Ketika sebuah parameter yang dimasukkan bernilai “salah”, maka pernyataan selanjutnya yang akan diperiksa. Jika semua pernyataan tidak memenuhi syarat, pilihan yang paling akhir “else” akan dijalankan.

1. Switch-Case

Switch-case digunakan untuk mengeksekusi pernyataan kondisional untuk mengeksekusi sebuah kondisi dalam bentuk perbandingan. Biasanya perintah ini digunakan dalam game.

1. Tujuan
2. Mengontrol Alur Program

Menentukan sebuah program berjalan sesuai dengan data yang diinputkan dan sesuai dengan situasi.

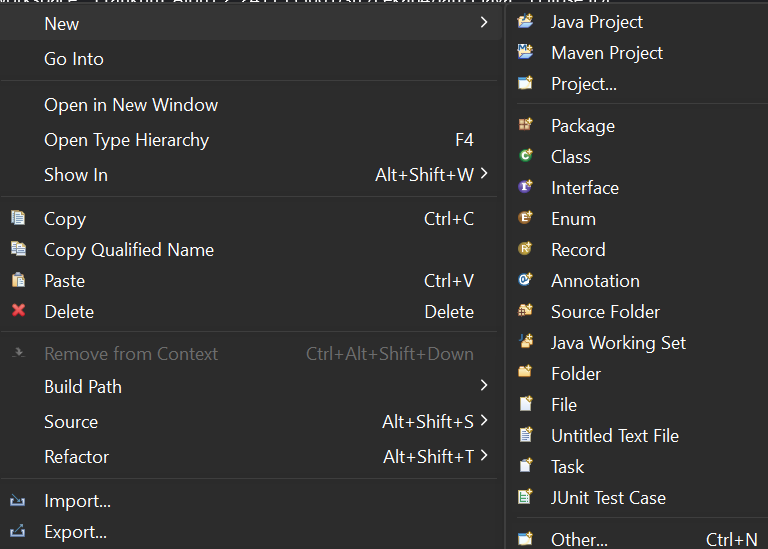
1. Meningkatkan Fleksibelitas Program

Menghasilakan program yang dapat menangani berbagai kasus tanpa harus membuat banyak kode.

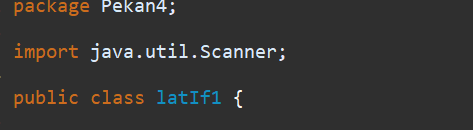
1. Mempermudah memperbaiki kode yang eror.
2. Alur Pengerjaan
3. If

Pada class pertama ini, akan menjelaskan bagaimana cara membuat program “if”. Program ini berisikan tentang bagaimana cara menjalankan kata kunci “if” untuk menilai kondisi IPK yang berada di atas ketentuan yang telah ditetapkan.

1. Pertama buatalah package dan claas baru. Beri nama class sesuai kebutuhan.



1. Import syntax scanner untuk membaca data inputan IPK dari konsol.



1. Buat nilai IPKnya menjadi nilai double, bilangan desimal.



1. Buat perintah cetak kalimat “Input IPK Anda”, untuk memasukan nilai IPK yang akan diperiksa oleh system.



1. Buat syntax seperti di bawah ini, untuk memerintahkan program membaca nilai double yang telah diinputkan pengguna.



1. Selanjutnya tambahkan kata kunci “If” untuk menilai data inputan IPK dari konsol apakah memenuhi syarat atau tidak.



1. Buat perintah mencetak kalimat “Anda Lulus Sangat Memuaskan Dengan IPK “. Kalimat ini akan dicetak jika kondisi data yang diinputkan memenuhi syarat “If”



1. Jika inputan data yang diberikan oleh konsol memenuhi syarat dari batas “If”, program akan mencetak seperti gambar di bawah ini.



1. Jika inputan data yang diberikan konsol atau pengguna tidak memenuhi, program ajan mencetak sebagai berikut.

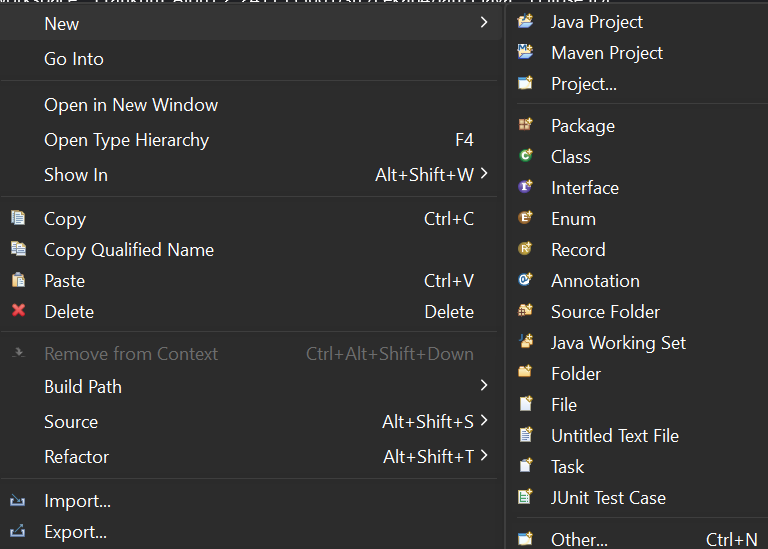


Hal ini terjadi karena tidak ada perintah selanjutnya jika data yang inputkan pengguna atau konsol tidak memenuhi syarat.

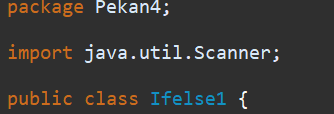
1. If Else (1)

Class ke dua ini menggunakan program If Else yang berfungsi untuk menentukan dua kemungkinan. Jika pernyataan kemungkinan pertama salah, pernyataan akan dilanjutkan ke kemungkinan ke dua dan akan dicetak.

1. Langkah pertama buatlah class baru dengan cara klik kanan di bagian package yang sudah ada. Beri nama class sesuai dengan kebutuhan.



1. Import syntax scanner untuk membaca data inputan IPK dari pengguna.



1. Buat nilai IPKnya menjadi nilai double, bilangan desimal.



1. Berikan perintah untuk mencetak kalimat “Input IPK Anda “. Perintah ini bertujuan untuk memberikan input data nilai IPK yang akan diperiksa sytem.



1. Buat syntax untuk memerintahkan program membaca bilangan double yang telah diberikan pengguna.



1. Berikutnya tambahkan kata kunci “If” untuk memeriksa pernyataan pertama.



Jika pernyataan pertama benar, system akan mencetak hasil seperti,



1. Setalah menambahkan kata kunci “If” tambahkan kata kunci “else”, untuk memeriksa pernyataan ke dua.



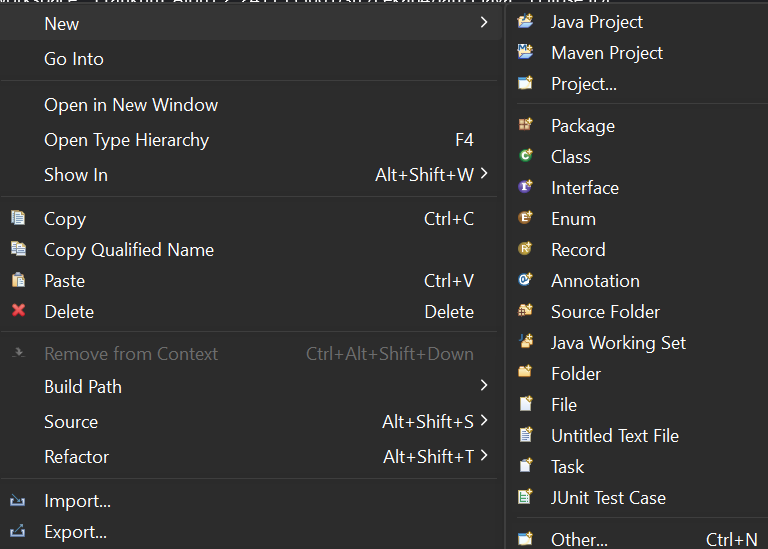
Jika pernyataan pertama tidak sesuai, program akan mencetak pernyataan “else”, seperti berikut,



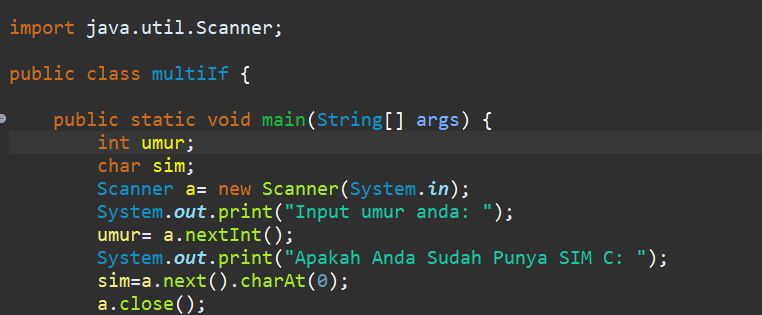
1. MultiIf

Program ini menggunakana pernyataan If independent. Artinya, semua kondisi If akan di uji satu per satu. Setiap blok if tidak berkaitan, sehingga ketika lebih dari satu kondisi terpenuhi, program akan menjalankan lebih dari satu pernyataan.

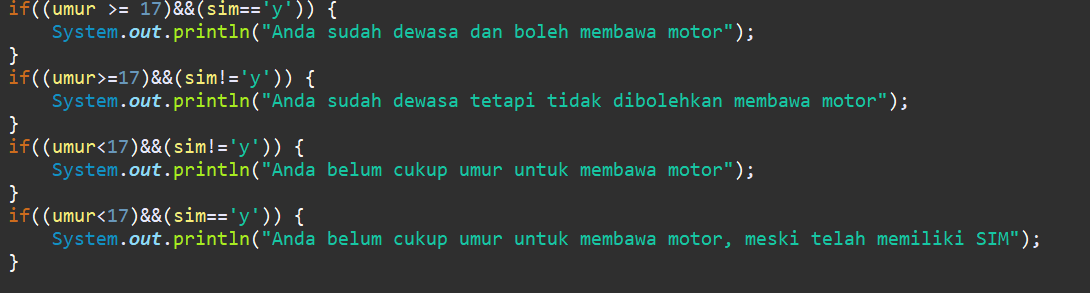
1. Langkah awal adalah membuat class dan memberi nama class sesuai kebutuhan.



1. Tambahkan Scanner untuk menginput data dari konsol atau pengguna.



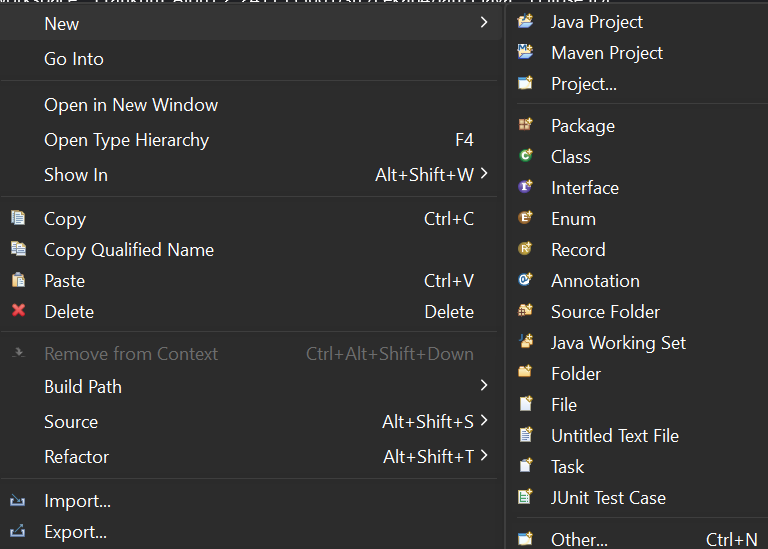
1. Tambahkan pernyataan kondisionalnya.



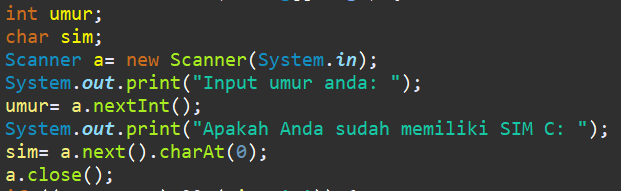
1. If Else (2)

Class “If else” ke dua akan membuat program untuk menentukan apakah seseorang telah diizinkan untuk membawa motor berdasarkan umur dan kepemilikan SIM C.

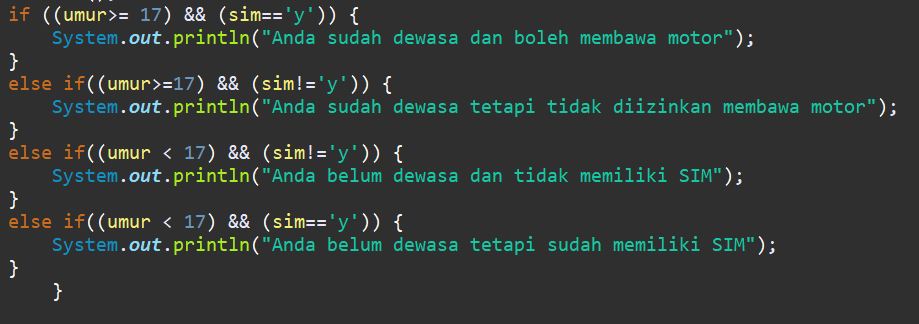
1. Langkah pertama untuk membuat program adalah bentuk class baru dengan Langkah yang sama seperti class-class sebelumnya. Beri nama sesuai kebutuhan program.



1. Sama halnya dengan program sebelumnya, gunakan scanner untuk menginput data dari pengguna.



1. Setelah menambahkan Scanner, buat pernyataan kondisional berupa if else seperti gambar di bawah



1. Masing-masing pernyataan akan menghasilkan input yang berbeda-beda sesuai dengan perintah yang buat.
2. Kondisi 1: Jika umur ≥ 17 dan memiliki SIM (sim=='y'):

* Output: "Anda sudah dewasa dan boleh membawa motor"
* Artinya: Pengguna memenuhi syarat untuk membawa motor.

1. Kondisi 2: Jika umur ≥ 17 dan tidak memiliki SIM (sim!='y'):

* Output: "Anda sudah dewasa tetapi tidak diizinkan membawa motor"
* Artinya: Meskipun sudah cukup umur, pengguna belum memenuhi syarat legal karena tidak memiliki SIM.

1. Kondisi 3: Jika umur < 17 dan tidak memiliki SIM (sim!='y'):

* Output: *"Anda belum dewasa dan tidak memiliki SIM"*
* Artinya: Pengguna belum cukup umur dan tidak memenuhi syarat untuk membawa motor.

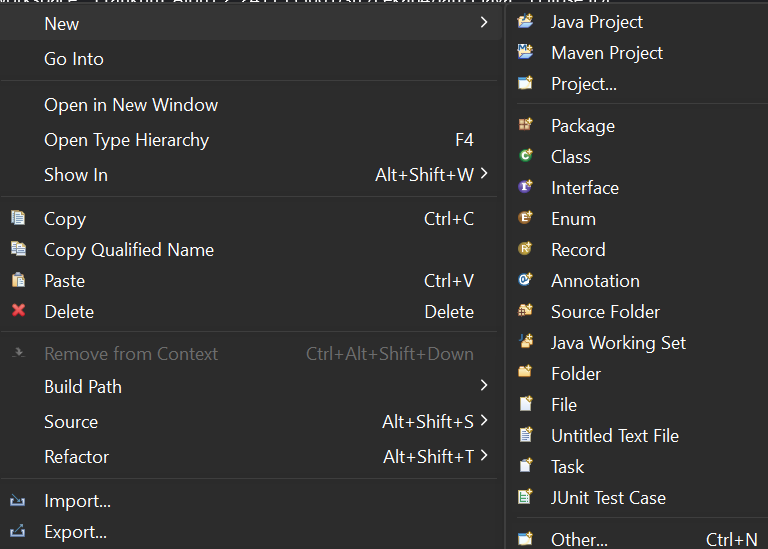
1. Kondisi 4: Jika umur < 17 dan memiliki SIM (sim=='y'):

* Output: "Anda belum dewasa tetapi sudah memiliki SIM"
* Artinya: Hal ini mungkin menunjukkan kesalahan data karena SIM seharusnya hanya diberikan pada usia ≥ 17 tahun.

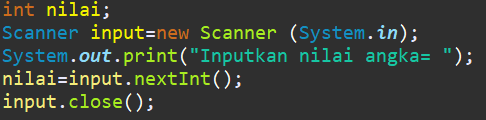
1. Nilai

Class ini menggunakan pernyataan kondisional berupa “If else” juga. Berbeda dengan program codingan “If else 2”, pada class nilai kode yang digunakan lebih sederhana.

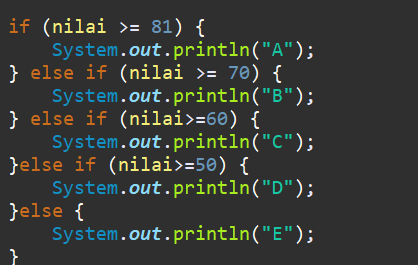
* 1. Buat class seperti biasa dan beri nama sesuai kebutuhan, pada kali ini nama classnya adalah “Nilai”



* 1. Tambahkan scanner dan int variabel nilai.



* 1. Kemudian buat pernyataan kondisional seperti di bawah ini,



* 1. Hasil dari masing-masing pernyataan adalah,

1. ika nilai ≥ 81, program mencetak huruf A.
2. Jika nilai berada di rentang 70 hingga 80 (inklusif 70), program mencetak huruf B.
3. Jika nilai berada di rentang 60 hingga 69, program mencetak huruf C.
4. Jika nilai berada di rentang 50 hingga 59, program mencetak huruf D.
5. Jika nilai < 50, program mencetak huruf E.
6. Kesimpulan

Pernyataan kondisional di java adalah struktur kontrol untuk mengatur program berdasarkan kondisi tertentu. Fungsi pernyataan kondisional pada java adalah mengontorl program, membuat program lebih efisien, meningkatkan feksebilitas, dan mempermudah memperbaiki kode.