**LAPORAN**

**PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Nifal

Stambuk : 13020220039

Kelas : A1

Dosen : Mardiyyah Hasnawi, S.Kom., M.T.,MTA

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

**MAKASSAR**

**2024**

# EVALUASI PRAKTIKUM

1. Apakah perbedaan antara struktur kontrol percabangan if-else dan switch-case?

* Kondisi:
  + - * + if-else digunakan ketika terdapat satu kondisi yang harus dievaluasi.
        + switch-case digunakan ketika terdapat beberapa kemungkinan kondisi yang akan dievaluasi.

* Tipe Kondisi:

o if-else dapat mengevaluasi ekspresi boolean (true/false) atau kondisi lain yang menghasilkan nilai boolean. o switch-case biasanya digunakan dengan nilai-nilai konstan atau enum.

* Eksekusi:

o Dalam if-else, setiap kondisi dievaluasi secara berurutan, dan hanya satu blok pernyataan yang dieksekusi. o Dalam switch-case, nilai ekspresi switch dievaluasi sekali, dan kemudian pernyataan yang sesuai dengan nilai tersebut dieksekusi. Setelah itu, eksekusi dilanjutkan hingga akhir blok switch atau hingga ditemukan pernyataan break.

* Kondisi Default:

o Dalam if-else, tidak ada yang mirip dengan kondisi default. Jika tidak ada kondisi yang benar, tidak ada blok pernyataan khusus yang dieksekusi. o Dalam switch-case, terdapat blok default yang dieksekusi jika tidak ada nilai case yang cocok dengan nilai switch.

* Comparasi Nilai:

o if-else menggunakan operasi perbandingan untuk mengevaluasi kondisi. o switch-case membandingkan nilai ekspresi switch dengan nilai-nilai case secara langsung.

1. Kapan digunakan struktur kontrol if-else dan switch-case

* if-else:

o Gunakan if-else ketika Anda perlu mengevaluasi kondisi yang kompleks, seperti kombinasi dari beberapa kondisi atau kondisi bersarang. o Jika kondisi berhubungan dengan ekspresi boolean yang kompleks, if-else lebih fleksibel dalam menangani logika tersebut. o Ketika setiap kondisi memerlukan tindakan atau pernyataan yang berbeda, if-else memungkinkan Anda untuk menentukan tindakan yang sesuai untuk setiap kondisi.

* switch-case:

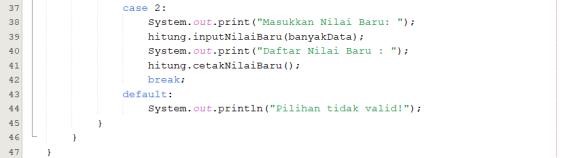
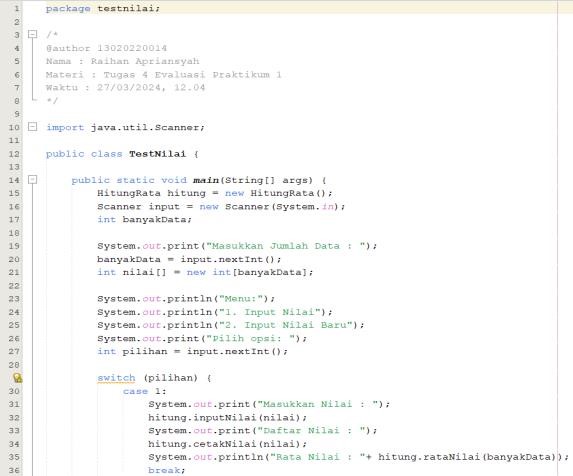
o Gunakan switch-case ketika Anda memiliki beberapa pilihan kondisi yang akan dievaluasi berdasarkan satu ekspresi atau variabel. o Jika Anda membandingkan nilai yang sama dengan beberapa nilai konstan atau enum, switch-case lebih cocok dan membuat kode lebih mudah dibaca. o Jika setiap kasus memerlukan eksekusi yang sama, atau hampir sama, switch-case dapat mengurangi pengulangan kode.

1. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case.

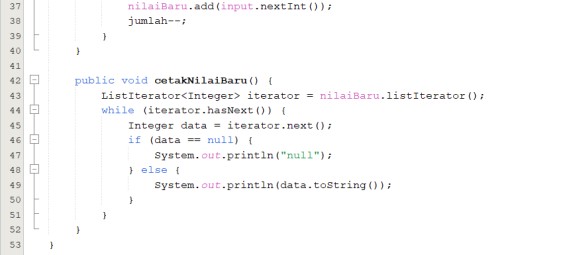
opsi pilihan 1=inputNilai() Pilihan 2=inputNilaiBaru()

* Coding :

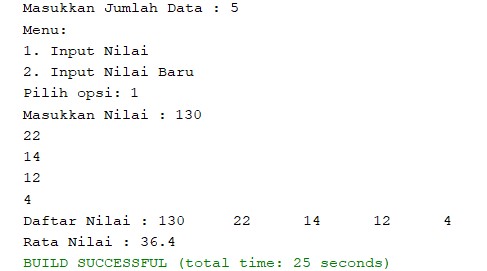
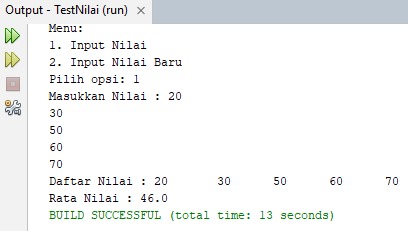




* + - * + Class HitungRata



* + - * + Output



1. Apakah perbedaan antara struktur kontrol perulangan while dan do-while?

* while:

o Evaluasi kondisi dilakukan sebelum blok pernyataan dieksekusi. Jika kondisi awalnya salah, blok pernyataan tidak akan dieksekusi sama sekali. o Cocok digunakan ketika jumlah iterasi tidak pasti dan perlu dicek kondisi sebelum masuk ke dalam perulangan.

Contoh penggunaan while:

int i = 0; while (i < 5) {

System.out.println("Nilai i: " + i); i++;

}

* do-while:

o Evaluasi kondisi dilakukan setelah blok pernyataan dieksekusi, sehingga setidaknya blok pernyataan akan dieksekusi sekali bahkan jika kondisi awalnya salah. o Cocok digunakan ketika setidaknya satu iterasi harus dilakukan, meskipun kondisi awalnya salah. Contoh penggunaan do-while:

int i = 0; do {

System.out.println("Nilai i: " + i); i++;

} while (i < 5);

1. Kapan digunakan struktur kontrol for?

* Struktur kontrol for digunakan ketika kita tahu jumlah iterasi yang akan dilakukan sebelumnya atau ketika kita ingin menggunakan loop untuk melakukan iterasi dengan baik melalui array, koleksi, atau rentang bilangan tertentu. Sintaks for juga cocok untuk melakukan inisialisasi, pengujian kondisi, dan peningkatan variabel loop dalam satu baris.

1. Apakah perbedaan antara Array dan ArrayList?berilah contoh masing-masing!

* Array:
  + - * + Array adalah struktur data dalam bahasa pemrograman yang dapat menyimpan sejumlah elemen dengan tipe data yang sama.
        + Ukuran array biasanya tetap (statis), artinya setelah array dibuat, ukurannya tidak dapat diubah. o Untuk mengakses elemen array, Anda menggunakan indeks berbasis nol.

Contoh penggunaan array dalam Java:

int[] numbers = {1, 2, 3, 4, 5};

System.out.println(numbers[0]);

* + ArrayList:

o ArrayList adalah kelas dalam Java yang merupakan bagian dari kerangka kerja koleksi (collections framework) dan digunakan untuk menyimpan kumpulan elemen. o Ukuran ArrayList dapat berubah (dinamis), artinya Anda dapat menambah atau menghapus elemen tanpa harus membuat array baru. o ArrayList menyediakan berbagai metode untuk menambah, menghapus, atau mengakses elemen, serta untuk melakukan operasi lainnya.

Contoh penggunaan ArrayList dalam Java:

import java.util.ArrayList;

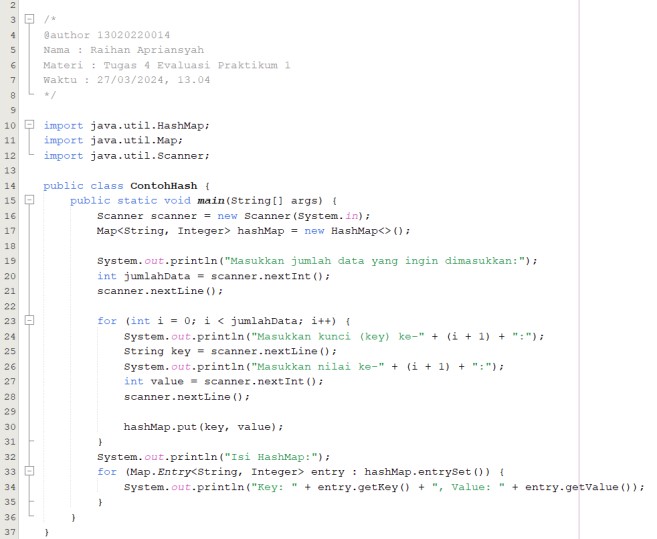
ArrayList<Integer> numbers = new ArrayList<>(); numbers.add(1); numbers.add(2); numbers.add(3);

System.out.println(numbers.get(0));

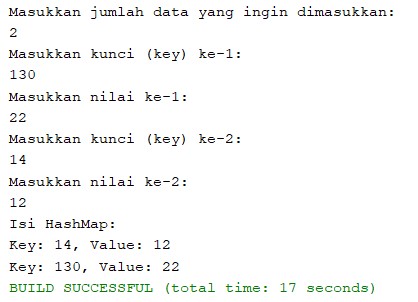
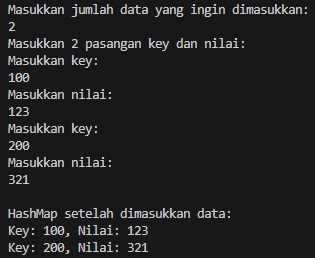
1. Buatlah contoh program yang mengimplementasikan HashMap dengan memasukkan nilai dan key

* + Codingan :





* + Output



# EVALUASI PRAKTIKUM

1. Berdasarkan ke tiga program di atas Class utama, Class Orang dan Class Mahasiswa, manakah yang menunjukkan konsep pewarisan dan polimorfisme! Jelaskan konsep tersebut sesuai program tersebut!

* + - Dalam ketiga program di atas, konsep pewarisan dan polimorfisme terlihat pada kelas Mahasiswa.

* + - Pewarisan (Inheritance):

Kelas Mahasiswa merupakan subkelas dari kelas Orang, yang berarti Mahasiswa mewarisi semua atribut dan metode dari kelas Orang. Pada konstruktor tanpa parameter di kelas Mahasiswa, terdapat pemanggilan super() yang mengacu pada konstruktor kelas induk (Orang), sehingga konstruktor kelas induk akan dieksekusi terlebih dahulu sebelum konstruktor kelas Mahasiswa.Dengan kata lain, kelas Mahasiswa mengambil semua karakteristik yang dimiliki oleh kelas Orang dan menambahkan atribut tambahan stb.

* + - Polimorfisme:

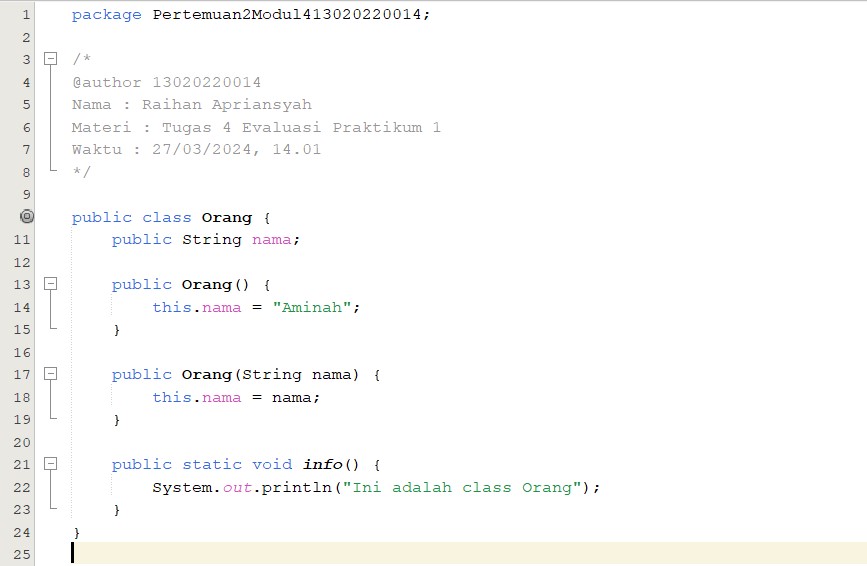
Terdapat dua versi dari konstruktor kelas Mahasiswa: satu tanpa parameter dan satu dengan dua parameter (nama dan stb).Konsep polimorfisme memungkinkan penggunaan metode dengan nama yang sama di kelas yang berbeda Dalam hal ini, metode konstruktor Mahasiswa() tanpa parameter adalah contoh polimorfisme, di mana kelas Mahasiswa memiliki versi konstruktor yang berbeda dari kelas induknya (Orang). Meskipun kelas Mahasiswa mewarisi konstruktor tanpa parameter dari kelas Orang, kelas Mahasiswa juga memiliki versi konstruktor tambahan yang memungkinkan inisialisasi atribut nama dan stb sekaligus.

1. Tambahkan static pada method info() Class Orang dan Class Mahasiswa kemudian lakukan pemanggilan method info() pada program utama (Class utama)!

* + - * Utama.java



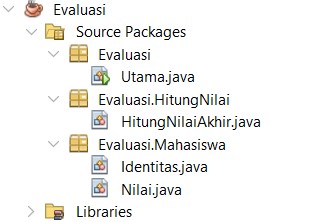
* + - * Orang.java



* + - * Mahasiswa.java



1. Buatlah sebuah project dengan nama project stambuk anda dan buatlah pengorganisasian package dan class seperti berikut!



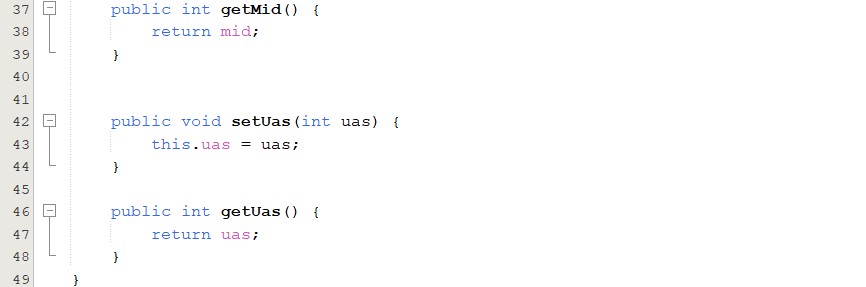
Setelah Itu Lengkapi Program pada soal yang telah diberikan

* + - Identitas.java



* + Nilai.java





* + HitungNilaiAkhir.java



* + Utama.java Output

