# LAPORAN PRAKTIKUM

# PEMROGRAMAN BERORIENTASI OBJECT

# JOBSHEET 4



# Sahrul Ramadhani

# 244107020058

# 2H

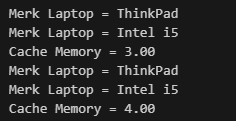
# D-IV Teknik Informatika

# Teknologi Informasi

# Politeknik Negeri Malang

# 2025

# Hasil Percobaan 1



# Jawaban pertanyaan Percobaan 1

## - Setter digunakan untuk mengubah nilai atribut/memberikan nilai isi sesuai tipedatanya.

* Getter ini diunakan untuk mengambil isi atribut yang telah di tetapkan oleh si setter tadi

## Konstruktor default itu parameternya di kosongin dulu, bisa di pakai atau di isi belakangan dengan method setter, sedangkan konstruktor berparameter langsung di berikan nilai saat object dibuat.

## Proc yang menyimpan nilai dari object Processor

## Pada kode :

public Laptop(String merk, Processor proc) {

this.merk = merk;

this.proc = proc;

}

Menandakan laptop punya Processor dengan merelasikan keduanya.

## Menampilkan processor yang di pilih laptop tersebut, jadi tiap melakukan laptop.info juga akan memanggil dan menampilkan processornya

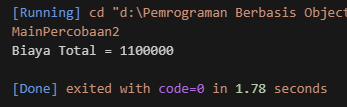
## Bisa kok , berarti langsung buat object processor baru di dalam konstruktor laptop tanpa membuat baru secara terpisah contoh kode :

Laptop L = new Laptop("ThinkPad", new Processor("Intel i5", 3) );

L.info();

# Percobaan 2

Hasil RUN:



# Jawaban Pertanyaan Percobaan 2

## Relasinya bisa di lihat pada class pelanggan ada variable yang mengggunakan tipe data Mobil dan Sopir, lalu pada construktor pelanggan juga ada parameter ber tipe Mobil dan Sopir itu adalah relasinya.

## Ya karena biayanya itu dihitung perhari bedasaarkan jumlah hari, dan tarif harian akan di kalikan dengan jumlah hari

## - mobil.hitungBiayaMobil(hari): menghitung **total biaya sewa mobil** selama berapa hari (biayaMobilPerHari \* hari).

* sopir.hitungBiayaSopir(hari): menghitung **total biaya sopir** selama berapa hari (biayaSopirPerHari \* hari).

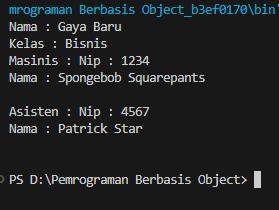
## Untuk menghubungkan objek Mobil m dan Sopir s ke object Pelanggan p, agar Pelanggan p memiliki object Mobil m dan Sopir s tertentu yang sudah dibuat

## Untuk menghitung total semuanya, sewa mobil dan sopir, dengan data biaya Sopir dan Mobil x jumlah hari.

## Mengambil Object mobil Pelanggan p dan menampilkan merknya, namun pada Class Mobil harus diberikan getmobil untuk mengambil nama/merk mobilnya.

# Percobaan 3

Hasil RUN :



# Jawaban Pertanyaan Percobaan 3

## Keduanya untuk memanggil method info yang telah dibuat pada kedua class, yaitu class pegawai dan keretaApi, tujuanya adalah untuk menampilkan info yang ada di dalam masing masing method tsbt.

## Sudah aku modifikasi

## Outputnya eror karena argumen yang ada di dalam construktor tidak sesuai dengan jumlah wadah konstrutornya yang berjumlah 4, hanya di masukan 3.

## Ya tinggal tmabahkan sebuah konstruktor lagi yang menerima 3 argumen tanpa asisten dan menambahkan ternary pada keretaApi.infonya

public KeretaApi (String nama, String kelas, Pegawai masinis){

this.nama = nama;

this.kelas = kelas;

this.masinis = masinis;

}

public String info (){

String info = "";

info += "Nama : " + this.nama + "\n";

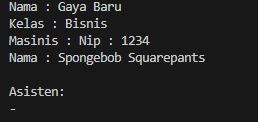
info += "Kelas : " + this.kelas + "\n";

info += "Masinis : " + this.masinis.info() + "\n";

info += "Asisten:\n" + (asisten != null ? asisten.info() : "-\n");

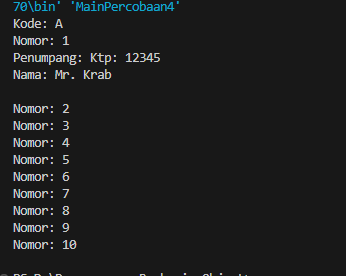
return info;

}



# Percobaan 4

Hasil RUN:



# Jawaban Pertanyaan Perccobaan 4

## Jumlah kursi di masukan saat inisiasi new Gerbong tadi sebanyak 10 jumlah Kursi

## Menggunakan ternary agar hanya menampilkan info jika kursi = terisi agar mencegah nullpointer

## Ya karena nomor kan pada awal di inisiasikan = 1, jadi ketika akan di isikan ke index harus nomor -1 agar menempati index ke 0 begitu juga nomor dan index selanjutnya.

## Ya yang terjadi budi akan menempari kursi no1, isi sebelumnya di tulisulang (overwrite), karena program tidak ada pengecekan kursi yang sudah terisi tidak boleh di timpa

## Disini saya akan modify pada method setpenumpang yang ada diclass gerbong dan memasukan data ke main pada kursi yang sama

public void setPenumpang(Penumpang penumpang, int nomor) {

int idx = nomor - 1

if (arrayKursi[idx].getPenumpang() != null) {

System.out.println("Kursi " + nomor + " sudah terisi. Silakan pilih kursi lain.");

Return;

}

arrayKursi[idx].setPenumpang(penumpang);

}

Pada Main :

public class MainPercobaan4 {

    public static void main(String[] args) {

        Penumpang p = new Penumpang("12345", "Mr. Krab");

        Gerbong gerbong = new Gerbong("A", 10);

        gerbong.setPenumpang(p, 1);

        System.out.println(gerbong.info());

        Penumpang p1 = new Penumpang("4628", "Rama");

        gerbong.setPenumpang(p1, 1);

        System.out.println(gerbong.info());

    }

}

Output :

