**LAPORAN PRAKTIKUM**

**OOP**

**KUIS 1**

**Disusun oleh:**

**Pascalis Dewangga S. L. 2241720140**

**TI - 2D**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023**

1. Program Bangun Datar
2. Kode Program

Class Bangun Datar

public class BangunDatar {

    private String jenis;

    public String getJenis() {

        return jenis;

    }

    public void setJenis(String jenis) {

        this.jenis = jenis;

    }

    public float hitungKeliling() {

        float p = 0;

        float l = 0;

        return 2 \* (p + l);

    }

    public float hitungLuas() {

        float p = 0;

        float l = 0;

        return p \* l;

    }

}

Class PersegiPanjang

public class PersegiPanjang extends BangunDatar {

    private float p;

    private float l;

    public float getP() {

        return p;

    }

    public float getL() {

        return l;

    }

    public void setP(float p) {

        this.p = p;

    }

    public void setL(float l) {

        this.l = l;

    }

    @Override

    public float hitungKeliling() {

        return 2 \* (p + l);

    }

    @Override

    public float hitungLuas() {

        return p \* l;

    }

}

Class Segitiga

public class Segitiga extends BangunDatar {

    private float alas;

    private float tinggi;

    private float sisi;

    public float getA() {

        return alas;

    }

    public float getS() {

        return sisi;

    }

    public float getT() {

        return tinggi;

    }

    public void setA(float alas) {

        this.alas = alas;

    }

    public void setS(float sisi) {

        this.sisi = sisi;

    }

    public void setT(float tinggi) {

        this.tinggi = tinggi;

    }

    @Override

    public float hitungKeliling() {

        return 3 \* sisi;

    }

    @Override

    public float hitungLuas() {

        return (alas \* tinggi) / 2;

    }

}

Class Lingkaran

public class Lingkaran extends BangunDatar {

    private float phi;

    private float jari;

    public float getJ() {

        return jari;

    }

    public float getPhi() {

        return phi;

    }

    public void setJ(float jari) {

        this.jari = jari;

    }

    public void setPhi(float phi) {

        this.phi = phi;

    }

    @Override

    public float hitungKeliling() {

        return phi \* (2 \* jari);

    }

    @Override

    public float hitungLuas() {

        return phi \* jari \* jari;

    }

}

Class MainBangunDatar

public class MainBangunDatar {

    public static void main(String[] args) {

        PersegiPanjang perPan = new PersegiPanjang();

        Segitiga setiga = new Segitiga();

        Lingkaran lingkar = new Lingkaran();

        perPan.setJenis("Persegi Panjang");

        setiga.setJenis("Segitiga");

        lingkar.setJenis("Lingkaran");

        perPan.setP(4);

        perPan.setL(5);

        setiga.setS(6);

        setiga.setA(6);

        setiga.setT(8);

        lingkar.setPhi(3.14f);

        lingkar.setJ(15);

        System.out.println("Jenis Bangun Datar: " + perPan.getJenis());

        System.out.println("Keliling: " + perPan.hitungKeliling());

        System.out.println("Luas: " + perPan.hitungLuas());

        System.out.println();

        System.out.println("Jenis Bangun Datar: " + setiga.getJenis());

        System.out.println("Keliling: " + setiga.hitungKeliling());

        System.out.println("Luas: " + setiga.hitungLuas());

        System.out.println();

        System.out.println("Jenis Bangun Datar: " + lingkar.getJenis());

        System.out.println("Keliling: " + lingkar.hitungKeliling());

        System.out.println("Luas: " + lingkar.hitungLuas());

    }

}

1. Hasil

A screen shot of a computer

Description automatically generated

1. Pertanyaan
2. Percobaan 2
3. Kode Program

Class Pegawai

public class Pegawai {

    public String nip, nama, alamat;

    Pegawai(String nip, String nama, String alamat) {

        this.nip = nip;

        this.nama = nama;

        this.alamat = alamat;

    }

    public String getNama() {

        return this.nama;

    }

    public String getNip() {

        return nip;

    }

    public int getGaji(Dosen dosen) {

        return dosen.getGaji();

    }

}

Class Dosen

public class Dosen extends Pegawai{

    public int jumlahSKS;

    public int tarif\_SKS;

    public Dosen(String nip, String nama, String alamat) {

        super(nip, nama, alamat);

    }

    public void setSKS(int jumlahSKS) {

        this.jumlahSKS = jumlahSKS;

    }

    public int getGaji() {

        return jumlahSKS \* tarif\_SKS;

    }

}

Class DaftarGaji

// import java.util.ArrayList;

public class DaftarGaji {

    public Pegawai[] listPegawai;

    public int jumlahPegawai = 0;

    public DaftarGaji(int kapasitas) {

        listPegawai = new Pegawai[kapasitas];

        kapasitas = 0;

    }

    public void addPegawai(Pegawai pegawai) {

        if (jumlahPegawai < listPegawai.length) {

            listPegawai[jumlahPegawai] = pegawai;

            jumlahPegawai++;

        } else {

            System.out.println("Daftar pegawai sudah penuh.");

        }

    }

    public void printSemuaGaji() {

        for (int i = 0; i < jumlahPegawai; i++) {

            if (listPegawai[i] instanceof Dosen) {

                Dosen dosen = (Dosen) listPegawai[i];

                System.out.println(

                        "NIP: " + dosen.getNip() + "  Nama: " + dosen.getNama() + ", Gaji: " + dosen.getGaji());

            }

        }

    }

}

Class MainPegawai

public class MainPegawai {

    public static void main(String[] args) {

        Dosen d1 = new Dosen("222884", "Pascalis", "Malang");

        Dosen d2 = new Dosen("222024", "Dewa", "Purwokerto");

        Dosen d3 = new Dosen("222234", "Sandi", "Bekasi");

        d1.tarif\_SKS = 50000;

        d1.jumlahSKS = 10;

        d2.tarif\_SKS = 50000;

        d2.jumlahSKS = 8;

        d3.jumlahSKS = 9;

        d3.tarif\_SKS = 50000;

        DaftarGaji dg = new DaftarGaji(2);

        dg.addPegawai(d1);

        dg.addPegawai(d2);

        dg.addPegawai(d3);

        dg.printSemuaGaji();

    }

}

1. Hasil

A computer screen shot of a computer program

Description automatically generated

Hasil pertama adalah kondisi ketika sesuai dengan kapasitas, sedangkan hasil kedua adalah ketika pendaftaran pegawai melebihi kapasitas.