LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

PROJEK UTS 2

Bank Sampah

Kelompok 2 Kelas CP :

Emmanuela Narisa Widasari / 225314065 25%

Gregorius Langgeng Dharmaputra / 225314066 25%

Edward Vito Brata Yuwana / 225314067 25%

Bonifasius Wisnu Raditya Atmaji / 225314085 25%

PRODI STUDI INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS SANATA DHARMA

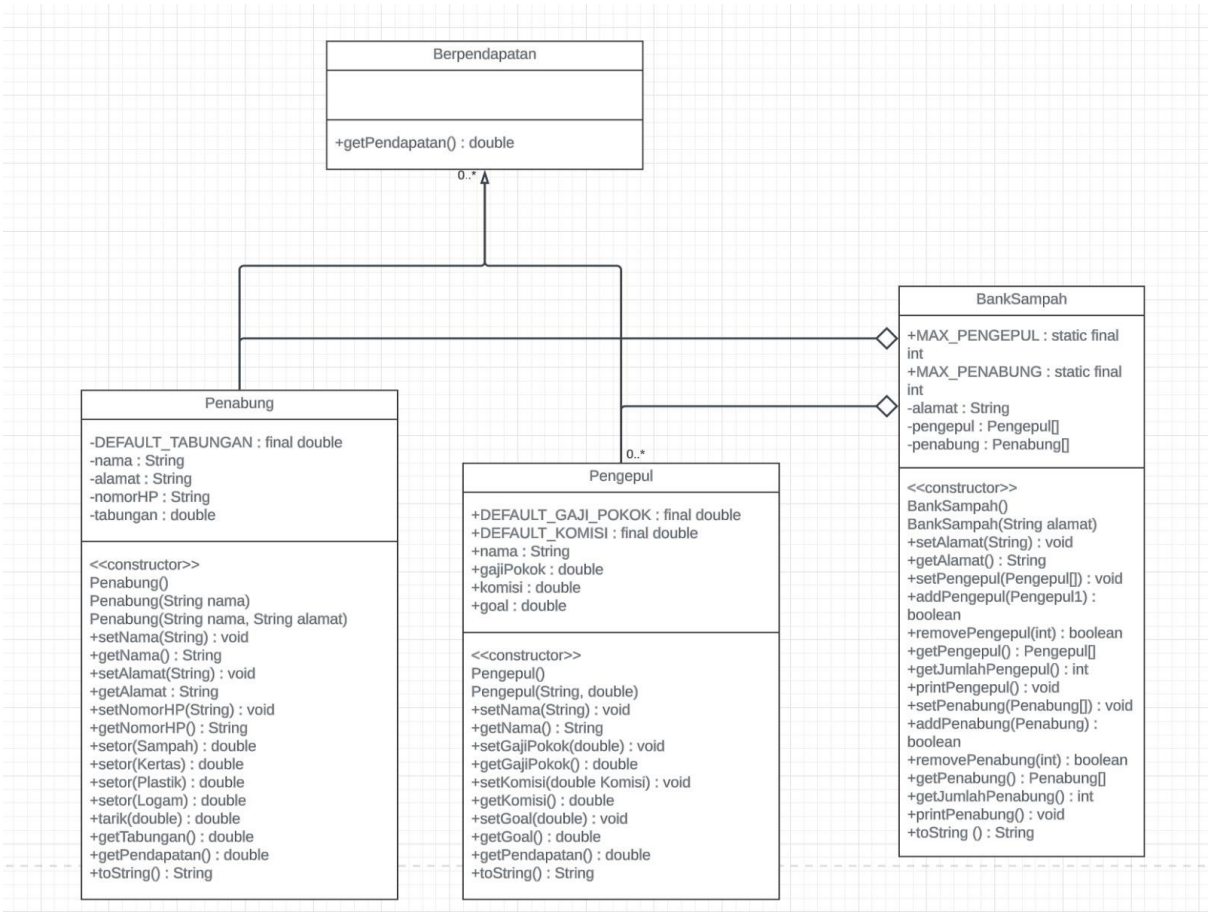
SEMESTER GENAP

A. Latar Belakang Pemilihan

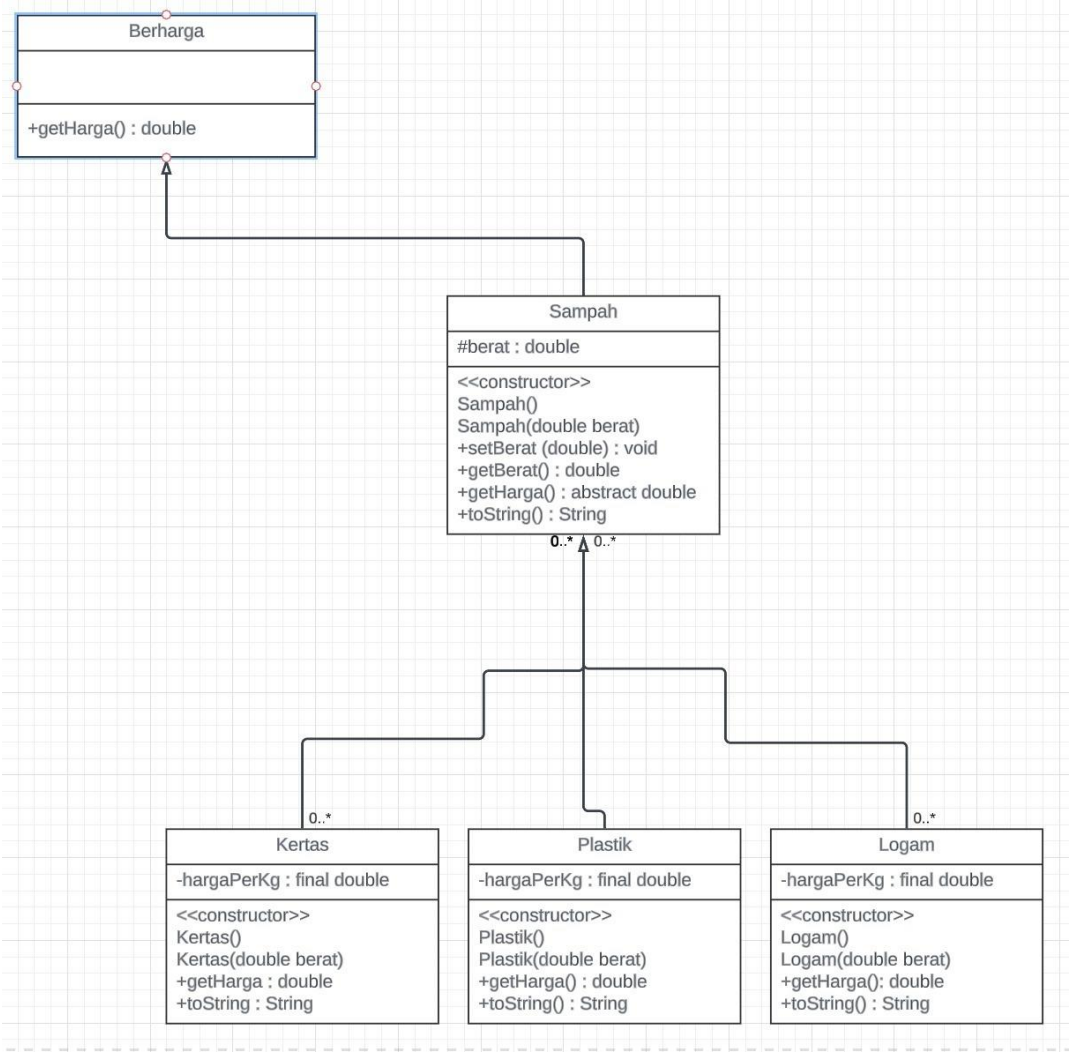
Kelompok membahas mengenai Bank Sampah. Dalam kasus ini, penabung dapat menyetorkan sampah ke Bank Sampah. Sampah tersebut bisa berupa plastik, kertas, dan logam. Dari hasil setor ke Bank Sampah, penabung bisa memperoleh pendapatan. Harga-harga sampah tersebut bisa didapatkan melalui berat sampah yang dikalikan dengan harga sampah per kg nya. Sampah-sampah yang sudah dikumpulkan di Bank Sampah tersebut kemudian diteruskan ke pengepul. Pengepul pun juga mendapatkan pendapatan dari hal tersebut.

B. Tujuan

Tujuan dalam pembuatan program ini adalah untuk menampilkan tabungan yang dimiliki penabung, menampilkan data sampah yang disetorkan penabung ke Bank Sampah dan juga pendapatan yang dihasilkan dari hasil setor tersebut, dan juga menampilkan data pengepul beserta pendapatannya dalam membeli sampah dari Bank Sampah tersebut.



C. UML



Penjelasan

Pada diagram UML, terdapat dua kelas interface yaitu Berpendapatan dan Berharga.

Pada interface Berharga, memiliki method getHarga bertipe double. Kemudian, kelas interface tersebut diturunkan ke kelas abstrak yaitu kelas Sampah. Pada kelas Sampah, terdapat set dan get method berat. Kemudian, terdapat juga getHarga yang bersifat abstrak. Method tersebut kemudian digunakan untuk sub kelasnya yaitu Kertas, Plastik, dan Logam. Method getHarga dibuat abstrak supaya memaksa dalam sub kelasnya

terdapat method itu juga. Kemudian, di sub kelasnya dihitunglah getHarga dengan kriteria penghitungan masing-masing dengan data berat yang diambil dari kelas Sampah.

Kelas Berpendapatan juga menjadi kelas interface dengan terdapat method getPendapatan. Kemudian, kelas Berpendapatan di implements kan ke kelas Penabung dan Pengepul. Terdapat set dan get method untuk nama, alamat, dan nomor HP. Kemudian, terdapat method untuk setor dengan parameter dari kelas Sampah, Kertas, Plastik, dan Logam. Method setor ini digunakan penabung untuk menyetorkan sampah-sampah ke Bank Sampah. Ada juga method tarik untuk mengambil uang dari hasil tabungan dari penabung. Lalu, juga ada getPendapatan yang mengembalikan nilai dari getTabungan. Kemudian untuk pengepul, terdapat set dan get method untuk nama, gaji pokok, komisi, dan goal. Ada juga getPendapatan untuk mendapatkan nilai pendapatan dari pengepul. Lalu, terdapat juga komposisi antara kelas BankSampah dengan kelas Penabung dan Pengepul. BankSampah dapat mengambil data dari kelas Penabung dan Pengepul dengan teknik komposisi. Pada kelas BankSampah, terdapat method set dan get alamat untuk BankSampah, serta set (memasukkan data), add (menambahkan data), dan remove (menghapus data) untuk pengepul dan penabung. Ada juga get method untuk jumlah pengepul dan penabung. Lalu, terdapat print pengepul dan penabung untuk menampilkan data-data dari pengepul dan penabung.

Pada program ini sudah menerapkan konstanta. Konstanta ini terdapat dalam kelas BankSampah yaitu untuk menentukan nilai maksimal dari jumlah penabung dan pengepul. Selain itu konstanta digunakan dalam kelas Penabung dimana nilai default dari tabungan setiap penabung adalah 0 dan dalam kelas Pengepul sebagai nilai default dari gaji pokok yang diterima oleh setiap pengepul disertai dengan nilai default dari komisi untuk dikalikan dengan sampah yang berhasil dibeli. Konstanta juga digunakan dalam subclass dari kelas super Sampah untuk menentukan harga per kilogram dari masing-masing jenis sampah tersebut. Dengan demikian, nilai konstanta digunakan sebagai nilai tetap dan tidak dapat diubah.

```
//  Constant
final double hargaPerKg=2500;
```

Pada program ini sudah menggunakan overloading. Nama dari constructor tidak bisa sama dalam satu kelas. Overloading digunakan untuk membedakan beberapa

constructor tersebut dengan parameter yang berbeda-beda. Contohnya seperti pada kelas Penabung, terdapat tiga constructor : Penabung(), Penabung(String nama), dan Penabung(String nama, String alamat). Walaupun nama constructornya sama, tetapi dengan menggunakan overloading constructor, perlu menggunakan parameter yang berbeda di setiap constructornya.

Pada program ini sudah menggunakan sifat enkapsulasi. Pada program, beberapa kelas menggunakan set dan get method. Method set digunakan untuk memasukkan data dengan parameter tertentu. Misal, contohnya yaitu setNama pada kelas Pengepul yang

```
//  Tabungan
public double setor(Sampah sampah) {
    if (sampah instanceof Plastik) {
        Plastik temp =((Plastik) sampah);
        this.tabungan += temp.getHarga();
        return temp.getHarga();
    } else if (sampah instanceof Kertas) {
        Kertas temp =((Kertas) sampah);
        this.tabungan += temp.getHarga();
        return temp.getHarga();
    } else if (sampah instanceof Logam) {
        Logam temp =((Logam) sampah);
        this.tabungan += temp.getHarga();
        return temp.getHarga();
    }
    return -1;
}
//  Overloading setor
public double setor(Kertas kertas){
    this.tabungan += kertas.getHarga();
    return kertas.getHarga();
}

public double setor(Plastik plastik){
    this.tabungan += plastik.getHarga();
    return plastik.getHarga();
}

public double setor(Logam logam){
    this.tabungan += logam.getHarga();
    return logam.getHarga();
}
```

digunakan untuk memasukkan nama dari pengepul. Kemudian, setelah mendapatkan data dari setNama, nama pengepul di return melalui get method.

```
protected double berat;
```

```
//  Nama
public void setNama(String nama) {
    this.nama = nama;
}

public String getNama() {
    return nama;
}
```

Pada program ini sudah menggun i memiliki atribut yang bertipe data ke ,

memanggil sebuah method dengan menggunakan tipe objek tersebut untuk mengakses kelas lain.

```
private Pengepul[] pengepul;
private Penabung[] penabung;
```

Pada program ini sudah menggunakan sifat pewarisan. Pewarisan ditandai dengan extends dari kelas turunan ke kelas super. Pewarisan diterapkan pada kelas jenis-jenis sampah yang di extends kee kelas Sampah. Contohnya yaitu : public class Plastik extends Sampah, public class Kertas extends Sampah, dan public class Logam extends Sampah. Karena sudah di extends ke kelas Sampah, maka dari itu kelas Plastik, Kertas, dan Logam dapat mengakses atribut-atribut yang ada pada kelas Sampah contohnya yaitu getBerat. Selain itu, supaya constructor dapat diwariskan, maka perlunya fungsi super pada constructor kelas turunannya.

```
public class Plastik extends Sampah{
```

Pada program ini sudah menggunakan sifat polimorfisme. Program ini menggunakan casting dan instanceOf. Sifat polimorfisme terletak di kelas Penabung pada method setor dengan parameter dari kelas Sampah. Penggunaan instanceOf digunakan pada perulangan if else. InstanceOf bisa digunakan dengan cara casting ke kelas Sampah atau bisa juga dengan membuat objek Sampah menggunakan kelas jenis-jenis sampah.

Pada

```
if (sampah instanceof Plastik) {
    Plastik temp =((Plastik) sampah);
    this.tabungan += temp.getHarga();
    return temp.getHarga();
} else if (sampah instanceof Kertas) {
    Kertas temp =((Kertas) sampah);
    this.tabungan += temp.getHarga();
    return temp.getHarga();
} else if (sampah instanceof Logam) {
    Logam temp =((Logam) sampah);
    this.tabungan += temp.getHarga();
    return temp.getHarga();
}
```

program ini sudah menggunakan kelas Abstrak dan Interface. Kelas Abstrak ditandai dengan adanya keterangan abstract pada kelasnya, contohnya seperti public abstract class Sampah implements Berharga. Selain

itu, pada kelas Abstrak pasti terdapat setidaknya satu method yang bersifat abstrak, seperti contohnya yaitu : public abstract double getHarga(). Method abstrak tersebut hanya dituliskan seperti itu tanpa adanya isian. Pada program ini, ada kelas yang digunakan sebagai kelas Abstrak yaitu kelas Sampah. Kemudian, kelas yang dijadikan turunan dari kelas abstrak perlu diberi extends, contohnya seperti : public class Plastik extends Sampah, public class Kertas extends Sampah, dan public class Logam extends Sampah.

```
public abstract class Sampah implements Berharga{
```

Pada program ini juga menggunakan kelas Interface. Kelas Interface dapat diketahui ketika dalam penulisan kelasnya ditulis seperti ini public interface Berharga. Pada program ini, terdapat dua kelas Interface yaitu kelas Berharga dan Berpendapatan. Method dari kelas Berharga yaitu untuk getHarga dan method dari kelas Berpendapatan yaitu untuk getPendapatan. Kelas yang diturunkan dari kelas interface menggunakan implements, contohnya seperti : public class Penabung implements Berpendapatan.

```
public interface Berpendapatan {
    public double getPendapatan();
}
```

Pada program ini juga menggunakan penerapan overriding. Overriding dilakukan ketika pada kelas turunan memiliki nama method yang sama dengan nama method pada kelas super. Contohnya seperti pada kelas Sampah, terdapat method toString. Pada kelas turunannya yaitu kelas Kertas, Plastik, dan Logam juga terdapat method toString. Maka dari itu, perlunya dituliskan overriding pada kelas turunannya.

```
@Override
public String toString() {
    return String.format(format:"Sampah Plastik Seberat : %1$s Kg",super.berat);
}
```

D. Output Program

```
run:
=======================KERTAS=======================
Nama : Dodi, Alamat : Jln Sumantri, Tabungan : Rp 1050.00
=======================PLASTIK======================
Nama : Dodi, Alamat : Jln Sumantri, Tabungan : Rp 3300.00
=======================LOGAM========================
Nama : Dodi, Alamat : Jln Sumantri, Tabungan : Rp 303300.00

=======================PENGEPUL=====================
Nama : Dodot Pendapatan : 2000.0 :

=======================PENARIKAN====================
Nama : Dodi, Alamat : Jln Sumantri, Tabungan : Rp 303300.00
Nama : Dodi, Alamat : Jln Sumantri, Tabungan : Rp 203300.00

=======================DATA SAMPAH==================
Sampah Kertas Seberat : 0.7 Kg
Sampah Plastik Seberat : 0.9 Kg
Sampah Logam Seberat : 100.0 Kg

=======================DATA PENABUNG================
Nama : Dodi, Alamat : Jln Sumantri, Tabungan : Rp 203300.00
Nama : Budi, Alamat : Jln Sumantri, Tabungan : Rp 0.00
Nama : Budimin, Alamat : Jln Sumantri, Tabungan : Rp 0.00

Nama : Budi, Alamat : Jln Sumantri, Tabungan : Rp 0.00
Nama : Budimin, Alamat : Jln Sumantri, Tabungan : Rp 0.00

=======================BANK SAMPAH==================
Bank Sampah
Alamat : null
Jumlah Pengepul : 0
Jumlah Penabung : 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

E. Analisis Program

Program ini memiliki konsep objek berbasis kelas, di mana setiap entitas di dalam program (seperti sampah, penabung, dan pengepul) direpresentasikan oleh kelas-kelas yang sesuai. Kelas Kertas, Plastik, dan Logam merupakan subkelas dari kelas Sampah. Masing-masing kelas sampah memiliki nilai konstanta harga per kilogramnya dan memiliki method dengan parameter berat untuk menyimpan berat dari sampah yang telah dikumpulkan oleh penabung ketika mereka menyetor sampah. Dalam method main beberapa objek sampah dan pengguna seperti penabung dan pengepul dibuat dan digunakan untuk melakukan setor sampah dan mencetak informasi penabung dan juga

pengepul. Program ini menggambarkan implementasi dasar dari sistem Bank Sampah, dimana penabung dapat menyetor sampah dan mendapatkan pendapatan, dan pengepul bertugas mengumpulkan sampah dari penabung. Program ini masih dapat dikembangkan namun teknik-teknik di atas dapat digunakan untuk menyimpan data dari bank sampah sehingga para penabung dapat menyetorkan sampah dan ditukar menjadi pendapatan dari hasil mengumpulkan sampah dengan berat tertentu. Selain itu, para pengepul juga dapat mengetahui gaji yang ia dapatkan dari hasil mengumpulkan sampah yang berhasil dibeli. Bank sampah itu sendiri pun juga dapat mendata jumlah berat sampah secara keseluruhan yang diperoleh dari sampah para penabung. Seperti bank pada umumnya, maka dalam bank sampah ini pun, setiap penabung berhak mengambil hasil dari hasil sampah yang telah dikumpulkan namun pengambilan atau penarikan tersebut dapat dilakukan bila telah memenuhi minimal sampah yang telah dikumpulkan.