**PRAKTIKUM MODUL 6**

**PENANGANAN PENGECUALIAN**

**PEMROGRAMAN BERORIENTASI OBJEK**

Code tersedia di: <https://github.com/afrzl/Pemrograman-Berorientasi-Objek/tree/main/Week%206/Praktikum>

1. Orang.java

import java.text.SimpleDateFormat;

import java.util.Calendar;

*/\*\**

*\* Orang*

*\*/*

public abstract class Orang {

  private String nama;

  private Calendar tanggalLahir;

  public **Orang**(String *nama*) {

    this.nama = *nama*;

  }

  public **Orang**(String *nama*, Calendar *tanggalLahir*) {

    this.nama = *nama*;

    this.tanggalLahir = *tanggalLahir*;

  }

  public String **getNama**() {

    return nama;

  }

  public void **setNama**(String *nama*) {

    this.nama = *nama*;

  }

  void **validatetanggallahir**() throws Exception {

    if (tanggalLahir.**get**(Calendar.YEAR) < 2000) {

      throw new **Exception**("Dibawah umur");

    } else {

      System.out.**println**("Cukup umur");

    }

  }

  public String **getTanggalLahir**() {

    try {

      SimpleDateFormat sdf = new **SimpleDateFormat**("yyyy MMM dd");

**validatetanggallahir**();

      return sdf.**format**(tanggalLahir.**getTime**()).**toString**();

    } catch (Exception *e*) {

      return "Belum tersedia";

    } finally {

      System.out.**println**("error tertangani");

    }

  }

  public void **setTanggalLahir**(Calendar *tanggalLahir*) {

    this.tanggalLahir = *tanggalLahir*;

  }

  public String **getNamaPanggilan**() {

    return nama.**substring**(0, Math.**min**(nama.**length**(), 3));

  }

}

1. Dosen.java

*/\*\**

*\* Dosen*

*\*/*

public interface Dosen {

  String **getNIDN**();

  void **setNIDN**(String *NIDN*);

  String **getKeahlian**();

  String **getPekerjaan**();

  void **setKeahlian**(String *keahlian*);

}

1. Pegawai.java

import java.util.Calendar;

*/\*\**

*\* Pegawai*

*\*/*

public class Pegawai extends Orang implements Dosen {

  private String NIP;

  private String kantor;

  private String unitKerja;

  private String NIDN;

  private String kelompokKeahlian;

  public **Pegawai**(

    String *nama*,

    String *NIP*,

    String *kantor*,

    String *unitKerja*,

    String *NIDN*,

    String *kelompokKeahlian*

  ) {

    super(*nama*);

    this.NIP = *NIP*;

    this.kantor = *kantor*;

    this.unitKerja = *unitKerja*;

    this.NIDN = *NIDN*;

    this.kelompokKeahlian = *kelompokKeahlian*;

  }

  public **Pegawai**(

    String *nama*,

    Calendar *tanggalLahir*,

    String *NIP*,

    String *kantor*,

    String *unitKerja*,

    String *NIDN*,

    String *kelompokKeahlian*

  ) {

    super(*nama*, *tanggalLahir*);

    this.NIP = *NIP*;

    this.kantor = *kantor*;

    this.unitKerja = *unitKerja*;

    this.NIDN = *NIDN*;

    this.kelompokKeahlian = *kelompokKeahlian*;

  }

  public String **getNIP**() {

    return NIP;

  }

  public String **getKantor**() {

    return kantor;

  }

  public void **setNIP**(String *NIP*) {

    this.NIP = *NIP*;

  }

  public void **setKantor**(String *kantor*) {

    this.kantor = *kantor*;

  }

  @Override

  public String **getNIDN**() {

    return NIDN;

  }

  @Override

  public void **setNIDN**(String *NIDN*) {

    this.NIDN = *NIDN*;

  }

  @Override

  public String **getKeahlian**() {

    return kelompokKeahlian;

  }

  @Override

  public String **getPekerjaan**() {

    return "mengajar";

  }

  @Override

  public void **setKeahlian**(String *keahlian*) {

    kelompokKeahlian = *keahlian*;

  }

}

1. Programmer.java

import java.util.Calendar;

*/\*\**

*\* Programmer*

*\*/*

public class Programmer extends Pegawai {

  String bahasaPemrograman;

  String platform;

  public **Programmer**(

    String *nama*,

    Calendar *tanggalLahir*,

    String *NIP*,

    String *kantor*,

    String *unitKerja*,

    String *NIDN*,

    String *kelompokKeahlian*,

    String *bahasaPemrograman*,

    String *platform*

  ) {

    super(*nama*, *tanggalLahir*, *NIP*, *kantor*, *unitKerja*, *NIDN*, *kelompokKeahlian*);

    this.bahasaPemrograman = *bahasaPemrograman*;

    this.platform = *platform*;

  }

  public String **getBahasaPemrograman**() {

    return bahasaPemrograman;

  }

  public void **setBahasaPemrograman**(String *bahasaPemrograman*) {

    this.bahasaPemrograman = *bahasaPemrograman*;

  }

  public String **getPlatform**() {

    return platform;

  }

  @Override

  public String **getPekerjaan**() {

    return "Coding all along day";

  }

  public void **setPlatform**(String *platform*) {

    this.platform = *platform*;

  }

}

1. MainTest.java

import java.util.GregorianCalendar;

*/\*\**

*\* MainTest*

*\*/*

public class MainTest {

  public static void **main**(String[] *args*) {

    Pegawai aku = new **Pegawai**(

      "aku",

      new **GregorianCalendar**(2015, 8, 31),

      "123456",

      "Polstat STIS",

      "BAU",

      "555555",

      "Ngoding"

    );

    System.out.**println**(

      "Object Pegawai:\nNama: " +

      aku.**getNama**() +

      "\nNama Panggilan: " +

      aku.**getNamaPanggilan**() +

      "\nNIP:" +

      aku.**getNIP**() +

      "\nKantor: " +

      aku.**getKantor**() +

      "\nPekerjaan: " +

      aku.**getPekerjaan**() +

      "\nTanggal Lahir: " +

      aku.**getTanggalLahir**()

    );

    Pegawai dia = new **Pegawai**(

      "aku",

      "123456",

      "Polstat STIS",

      "BAU",

      "555555",

      "Ngoding"

    );

    System.out.**println**(

      "\nObject Pegawai:\nNama: " +

      dia.**getNama**() +

      "\nNama Panggilan: " +

      dia.**getNamaPanggilan**() +

      "\nNIP:" +

      dia.**getNIP**() +

      "\nKantor: " +

      dia.**getKantor**() +

      "\nPekerjaan: " +

      dia.**getPekerjaan**() +

      "\nTanggal Lahir: " +

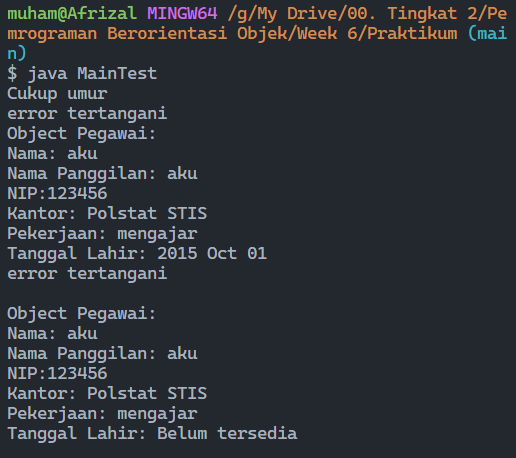
      dia.**getTanggalLahir**()

    );

  }

}

1. Output



1. Penjelasan

Pada class Orang di method getTanggalLahir(), dilakukan try catch untuk mengecek apakah terdapat error. Di dalam block try, memanggil function validatetanggallahir() yang berisi pengecekan apakah tahun lahir sebelum atau sesudah 2000. Jika tahun lahir sebelum 2000, maka akan di throw exception dengan pesan “Dibawah umur”. Sehingga jika class Orang memiliki tahun lahir dibawah 2000, akan gagal pada saat try catch di method getTanggalLahir().