

Nama : Viona Asya

ArindaNIM : 2109106108

Kelas : INFORMATIKA B 2021

PERHATIAN !!!

1. Bersifat individu dan wajib dikerjakan bagi setiap peserta praktikum
2. Kerjakan dengan sungguh-sungguh dan semaksimal mungkin
3. Percaya pada diri sendiri , Tidak perlu mencontek Teman kalian
4. Jika terindikasi meniru teman (hanya ganti tema, hanya ganti variabel, hanya ganti tipe data, hanya ganti tampilan), Keduanya mendapat nilai E.
5. Jangan lupa turn in jika sudah pull request di github.
6. Semangat Mengerjakan
7. Jika ingin ubah judul / tema silahkan konfirmasi ke aslab yang mengajar di kelasnya.

INSTRUKSI PADA PROGRAM JAVA

Lanjutkan project posttest sebelumnya, buat folder baru Posttest4 dan copy file posttest sebelumnya ke folder baru

Instruksi pada posttest sebelumnya tetap wajib diterapkan pada posttest ini.

Terapkan konsep Polymorphism keduanya [**Minimal 1 fungsi overloading dan override**]

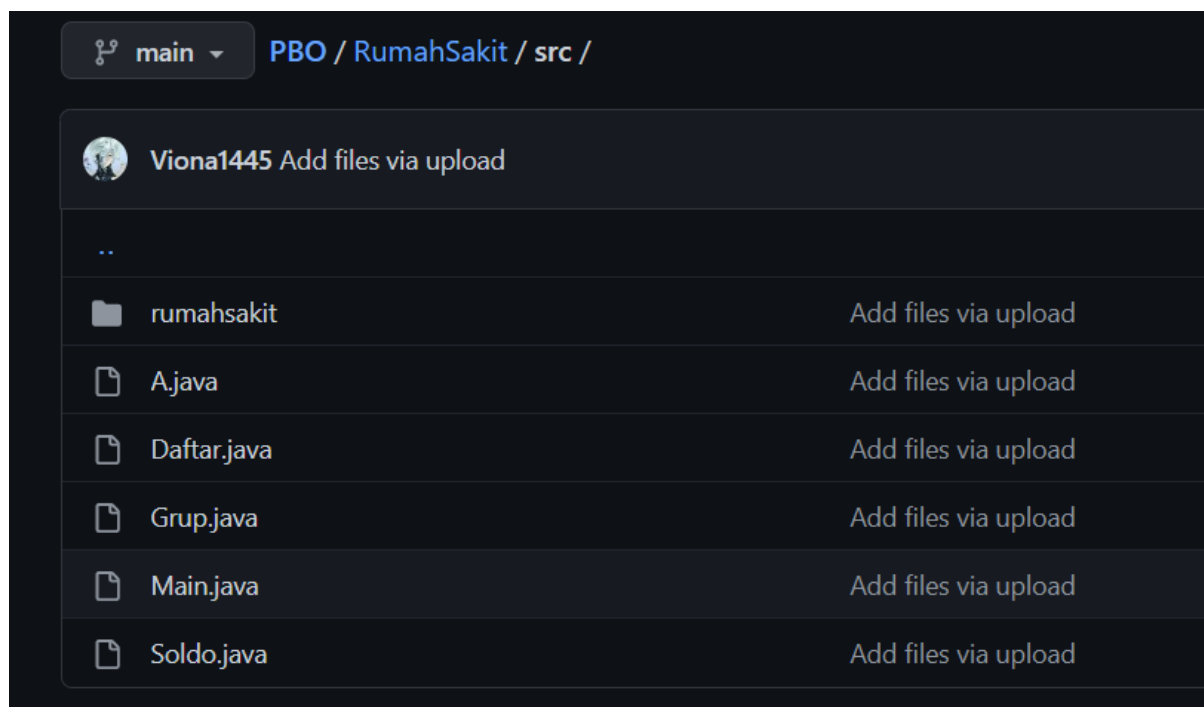
Penerapan

Polymorphism harus logis atau tidak "mengada - ada"

FORMAT PENGUMPULAN

SS program dalam bentuk pdf, letakan didalam folder ssProgram

Source Code :



Main.java

```
package RumahSakit;
```

```

import java.io.*;
import java.lang.reflect.Array;
import java.time.Year;
import java.util.Arrays;
import java.util.Scanner;
import java.util.StringTokenizer;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader input = new BufferedReader(new
InputStreamReader(System.in));
        //arraylist grup
        ArrayList<Daftar> listGrup = new ArrayList<>();

        //arraylist soloduo
        ArrayList<Daftar> listSoldu = new ArrayList<>();

        Scanner terminalInput = new Scanner(System.in);
        String pilihanUser;
        boolean isLanjutkan = true;

        while (isLanjutkan) {
            clearScreen();
            System.out.print("\n\n");

System.out.println("
\n");
                System.out.println("                || SISTEM || MANAJEMEN || RUMAH SAKIT
||                ");

System.out.println("
\n" +
                "
\n" +
                "
\n" +
                "
\n" +
                "

\n");

                System.out.println("                [1]\tLihat Seluruh Data Pasien");
                System.out.println("                [2]\tCari Data Pasien");
                System.out.println("                [3]\tTambah Data Pasien");
                System.out.println("                [4]\tUbah Data Pasien");
                System.out.println("                [5]\tHapus Data Pasien\n");

System.out.println("
\n");
                System.out.print("Pilih : ");
                pilihanUser = terminalInput.next();
                System.out.print("\n\n");

                switch (pilihanUser) {
                    case "1":

System.out.println("
\n");
                        System.out.println("                LIST SELURUH DATA
PASIEN                ");

System.out.println("
\n");
                        tampilkanData();

```

```
        break;  
case "2":
```

```

System.out.println("
");
        System.out.println("                                CARI DATA PASIEN
");
System.out.println("
\n");
        cariData();
        break;
        case "3":
System.out.println("
");
        System.out.println("                                TAMBAH DATA
PASIEN                                ");
System.out.println("
\n");
        tambahData();
        tampilkanData();
        break;
        case "4":
System.out.println("
");
        System.out.println("                                UBAH DATA
PASIEN                                ");
System.out.println("
\n");
        updateData();
        break;
        case "5":
System.out.println("
");
        System.out.println("                                HAPUS DATA PASIEN
");
System.out.println("
\n");
        deleteData();
        break;
        default:
            System.err.println("\nInput anda tidak ditemukan\nSilahkan
pilih [1-5]");
    }

    isLanjutkan = getYesorNo("        Apakah Anda Ingin Melanjutkan");
}
}

private static void updateData() throws IOException{
    // database original
    File database = new File("database.txt");
    FileReader fileInput = new FileReader(database);
    BufferedReader bufferedInput = new BufferedReader(fileInput);

    // database sementara
    File tempDB = new File("tempDB.txt");
    FileWriter fileOutput = new FileWriter(tempDB);
    BufferedWriter bufferedOutput = new BufferedWriter(fileOutput);

    // tampilkan data

```

```
System.out.println("List pasien");
```

```
tampilkanData();

// ambil user input / pilihan data
Scanner terminalInput = new Scanner(System.in);
System.out.print("\nMasukan Nomor Pasien Yang Akan DiUpdate : ");

System.out.println(" ");

int updateNum = terminalInput.nextInt();

// tampilkan data yang ingin diupdate

String data = bufferedInput.readLine();
int entryCounts = 0;

while (data != null){
    entryCounts++;

    StringTokenizer st = new StringTokenizer(data, ",");

    // tampilkan entrycounts == updateNum
    if (updateNum == entryCounts){
        System.out.println("\nData yang ingin anda update adalah:");

        System.out.println(" ");

        System.out.println("dokter          : " + st.nextToken());
        System.out.println("kamar          : " + st.nextToken());
        System.out.println("nama           : " + st.nextToken());
        System.out.println("penyakit       : " + st.nextToken());
        System.out.println("umur           : " + st.nextToken());

        // update data

        // mengambil input dari user

        String[] fieldData = {"kamar", "nama", "penyakit", "umur"};
        String[] tempData = new String[4];

        st = new StringTokenizer(data, ",");
        String originalData = st.nextToken();

        for(int i=0; i < fieldData.length ; i++) {
            boolean isUpdate = getYesorNo("apakah anda ingin merubah " +
fieldData[i]);
            originalData = st.nextToken();
            if (isUpdate){
                //user input

                if (fieldData[i].equalsIgnoreCase("kamar")){
                    System.out.print("masukan kamar, format=(YYYY): ");
                    tempData[i] = ambilkamar();
                } else {
                    terminalInput = new Scanner(System.in);
                    System.out.print("\nMasukan " + fieldData[i] + " baru:");

                    tempData[i] = terminalInput.nextLine();
                }
            } else {
                tempData[i] = originalData;
            }
        }

        // tampilkan data baru ke layar
        st = new StringTokenizer(data, ",");
```

```

        st.nextToken();
        System.out.println("\nData baru anda adalah ");

System.out.println("
");

        System.out.println("kamar          : " + st.nextToken() + " -
-> " + tempData[0]);
        System.out.println("nama          : " + st.nextToken() + " -->
" + tempData[1]);
        System.out.println("penyakit       : " + st.nextToken() + " -
-> " + tempData[2]);
        System.out.println("umur          : " + st.nextToken() + " --
> " + tempData[3]);

        boolean isUpdate = getYesOrNo("apakah anda yakin ingin mengupdate
data tersebut");

        if (isUpdate){

            // cek data baru di database
            boolean isExist = cekPasienDiDatabase(tempData,false);

            if(isExist){
                System.err.println("data pasien sudah ada di database,
proses update dibatalkan, \nsilahkan delete data yang bersangkutan");
                // copy data
                bufferedOutput.write(data);

            } else {

                // format data baru kedalam database
                String kamar = tempData[0];
                String nama = tempData[1];
                String penyakit = tempData[2];
                String umur = tempData[3];

                // kita bikin primary key
                long nomorEntry = ambilEntryPerkamar(nama, kamar) + 1;

                String punulisTanpaSpasi = nama.replaceAll("\\s+", "");
                String primaryKey =
punulisTanpaSpasi+"_"+kamar+"_"+nomorEntry;

                // tulis data ke database
                bufferedOutput.write(primaryKey + "," + kamar + "," + nama
+"," + penyakit + "," + umur);
            }
        } else {
            // copy data
            bufferedOutput.write(data);
        }
    } else {
        // copy data
        bufferedOutput.write(data);
    }
    bufferedOutput.newLine();

    data = bufferedInput.readLine();
}

// menulis data ke file
bufferedOutput.flush();

// delete original database
database.delete();

```

```

        // rename file tempDB menjadi database
        tempDB.renameTo(database);

    }

    private static void deleteData() throws IOException{
        // kita ambil database original
        File database = new File("database.txt");
        FileReader fileInput = new FileReader(database);
        BufferedReader bufferedInput = new BufferedReader(fileInput);

        // kita buat database sementara
        File tempDB = new File("tempDB.txt");
        FileWriter fileOutput = new FileWriter(tempDB);
        BufferedWriter bufferedOutput = new BufferedWriter(fileOutput);

        // tampilkan data
        System.out.println("List pasien");
        tampilkanData();

        // kita ambil user input untuk mendelete data
        Scanner terminalInput = new Scanner(System.in);
        System.out.print("\nMasukan nomor pasien yang akan dihapus: ");
        int deleteNum = terminalInput.nextInt();

        // looping untuk membaca tiap data baris dan skip data yang akan didelete

        boolean isFound = false;
        int entryCounts = 0;

        String data = bufferedInput.readLine();

        while (data != null){
            entryCounts++;
            boolean isDelete = false;

            StringTokenizer st = new StringTokenizer(data, ",");

            // tampilkan data yang ingin di hapus
            if (deleteNum == entryCounts){
                System.out.println("\nData Yang Ingin Anda Hapus Adalah:");

                System.out.println("
                System.out.println("dokter          : " + st.nextToken());
                System.out.println("kamar          : " + st.nextToken());
                System.out.println("nama           : " + st.nextToken());
                System.out.println("penyakit       : " + st.nextToken());
                System.out.println("umur          : " + st.nextToken());

                isDelete = getYesorNo("Apakah anda yakin akan menghapus?");
                isFound = true;
            }

            if(isDelete){
                //skip pindahkan data dari original ke sementara
                System.out.println("Data berhasil dihapus");
            } else {
                // kita pindahkan data dari original ke sementara
                bufferedOutput.write(data);
                bufferedOutput.newLine();
            }
            data = bufferedInput.readLine();
        }

        if(!isFound){

```



```

        System.err.println("pasien tidak ditemukan");
    }

    // menulis data ke file
    bufferedOutput.flush();
    // delete original file
    database.delete();
    // rename file sementara ke database
    tempDB.renameTo(database);
}

private static void tampilkanData() throws IOException{

    FileReader fileInput;
    BufferedReader bufferInput;

    try {
        fileInput = new FileReader("database.txt");
        bufferInput = new BufferedReader(fileInput);
    } catch (Exception e){
        System.err.println("Database Tidak ditemukan");
        System.err.println("Silahkan tambah data terlebih dahulu");
        tambahData();
        return;
    }

    System.out.println("\n| No | \tkamar | \tnama | \tpenyakit | \tumur pasien");
    System.out.println(" ");

    String data = bufferInput.readLine();
    int nomorData = 0;
    while(data != null) {
        nomorData++;

        StringTokenizer stringToken = new StringTokenizer(data, ",");

        stringToken.nextToken();
        System.out.printf("| %2d |", nomorData);
        System.out.printf("| \t%4s |", stringToken.nextToken());
        System.out.printf("| \t%-20s |", stringToken.nextToken());
        System.out.printf("| \t%-20s |", stringToken.nextToken());
        System.out.printf("| \t%s |", stringToken.nextToken());
        System.out.print("\n");

        data = bufferInput.readLine();
    }

    System.out.println(" ");

}

private static void cariData() throws IOException{

    // membaca database ada atau tidak

    try {
        File file = new File("database.txt");
    } catch (Exception e){
        System.err.println("Database Tidak ditemukan");
        System.err.println("Silahkan tambah data terlebih dahoeleoe");
    }
}

```

[illegible]

```

penyakit + "," + umur);
        bufferOutput.newLine();
        bufferOutput.flush();
    }

    } else {
        System.out.println("pasien yang anda akan masukan sudah tersedia di
database dengan data berikut:");
        cekpasienDiDatabase(keywords, true);
    }

    bufferOutput.close();
}

private static long ambilEntryPerkamar(String nama, String kamar) throws
IOException {
    FileReader fileInput = new FileReader("database.txt");
    BufferedReader bufferInput = new BufferedReader(fileInput);

    long entry = 0;
    String data = bufferInput.readLine();
    Scanner dataScanner;
    String primaryKey;

    while(data != null){
        dataScanner = new Scanner(data);
        dataScanner.useDelimiter(",");
        primaryKey = dataScanner.next();
        dataScanner = new Scanner(primaryKey);
        dataScanner.useDelimiter(" ");

        nama = nama.replaceAll("\\s+", "");

        if (nama.equalsIgnoreCase(dataScanner.next()) &&
kamar.equalsIgnoreCase(dataScanner.next()) ) {
            entry = dataScanner.nextInt();
        }

        data = bufferInput.readLine();
    }

    return entry;
}

private static boolean cekpasienDiDatabase(String[] keywords, boolean
isDisplay) throws IOException{
    FileReader fileInput = new FileReader("database.txt");
    BufferedReader bufferInput = new BufferedReader(fileInput);

    String data = bufferInput.readLine();
    boolean isExist = false;
    int nomorData = 0;

    if (isDisplay) {
        System.out.println("\n| No | \tkamar | \tnama | \tpenyakit
| \tumur pasien");
        System.out.println("-----");
    }

    while(data != null){
        // cek keywords didalam baris
        isExist = true;

```

```

        for(String keyword:keywords){
            isExist = isExist &&
data.toLowerCase().contains(keyword.toLowerCase());
        }

        // jika keywordsnya cocok maka tampilkan

        if(isExist){
            if(isDisplay) {
                nomorData++;
                StringTokenizer stringToken = new StringTokenizer(data, ",");

                stringToken.nextToken();
                System.out.printf("| %2d ", nomorData);
                System.out.printf("| \t%4s ", stringToken.nextToken());
                System.out.printf("| \t%-20s ", stringToken.nextToken());
                System.out.printf("| \t%-20s ", stringToken.nextToken());
                System.out.printf("| \t%s ", stringToken.nextToken());
                System.out.print("\n");
            } else {
                break;
            }
        }

        data = bufferInput.readLine();
    }

    if (isDisplay){
System.out.println("
");

    }

    return isExist;
}

private static String ambilkamar() throws IOException{
    boolean kamarValid = false;
    Scanner terminalInput = new Scanner(System.in);
    String kamarInput = terminalInput.nextLine();
    while(!kamarValid) {
        try {
            Year.parse(kamarInput);
            kamarValid = true;
        } catch (Exception e) {
            System.out.println("Format kamar yang anda masukan salah,
format=(YYYY)");
            System.out.print("silahkan masukan kamar lagi: ");
            kamarValid = false;
            kamarInput = terminalInput.nextLine();
        }
    }

    return kamarInput;
}

private static boolean getYesorNo(String message){
    Scanner terminalInput = new Scanner(System.in);
    System.out.print("\n"+message+" (y/n)? ");
    String pilihanUser = terminalInput.next();

    while(!pilihanUser.equalsIgnoreCase("y") &&
!pilihanUser.equalsIgnoreCase("n")) {
        System.err.println("Pilihan anda bukan y atau n");
        System.out.print("\n"+message+" (y/n)? ");
    }
}

```



```

    }

    return pilihanUser.equalsIgnoreCase("y");

}

private static void clearScreen() {
    try {
        if (System.getProperty("os.name").contains("Windows")) {
            new
ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();
        } else {
            System.out.print("\033\143");
        }
    } catch (Exception ex) {
        System.err.println("tidak bisa clear screen");
    }
}
}

```

Grup.Java

```

package RumahSakit;

public class Grup extends Daftar{
    int jumlahAnggota;

    public Grup(String nama, String Penyakit, String judul, int urut, int
jumlahAnggota) {
        super(nama, penyakit, judul, urut);
        this.jumlahAnggota = jumlahAnggota;
    }

    public void setJumlahAnggota(int jumlahAnggota) {
        this.jumlahAnggota = jumlahAnggota;
    }

    public int getJumlahAnggota() {
        return jumlahAnggota;
    }
}

```

Soldo.Java

```

package RumahSakit;

public class SoloDuo extends Daftar{
    int jumlahPasien;

    public SoloDuo(String nama, String Penyakit, String judul, int urut, int
jumlahArtis){
        super(nama, agensi, judul, urut);
        this.jumlahArtis = jumlahArtis;
    }

    public void setJumlahArtis(int jumlahPenhyakit) {
        this.jumlahArtis = jumlahArtis;
    }

    public int getJumlahArtis() {
        return jumlahPasien;
    }
}

```

Daftar.java

```
package posttest3;

public class Daftar {
    String nama;
    String Penyakit;
    String judul;
    int urut;
    int jumlahAnggota;
    int jumlahArtis;

    public Daftar(String nama, String penyakit, String judul, int urut){
        this.nama = nama;
        this.agensi = agensi;
        this.judul = judul;
        this.urut = urut;

        this.jumlahAnggota = jumlahAnggota;

        this.jumlahArtis = jumlahArtis;
    }

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

    public String getNama() {
        return nama;
    }

    public void setAgensi(String agensi) {
        this.agensi = agensi;
    }

    public String getAgensi() {
        return agensi;
    }

    public void setJudul(String judul) {
        this.judul = judul;
    }

    public String getJudul() {
        return judul;
    }

    public void setUrut(int urut) {
        this.urut = urut;
    }

    public int getUrut() {
        return urut;
    }

    public void setJumlahAnggota(int jumlahAnggota) {
        this.jumlahAnggota = jumlahAnggota;
    }

    public int getJumlahAnggota() {
        return jumlahAnggota;
    }

    public void setJumlahArtis(int jumlahPasien) {
        this.jumlahPasien = jumlahArtis;
    }
}
```

```
public int getJumlahArtis() {  
    return jumlahArtis;  
}  
}
```

Output :

SS OUTPUT PROGRAM :

1. Menu Utama



2. Lihat Seluruh Data Pasien

LIST SELURUH DATA PASIEN					
No	kamar	nama	penyakit	umur pasien	
1	4001	Viona	Magh	19 Tahun	
2	4002	Sahriah	Pusing	20 Tahun	
3	4003	Cici	DBD	25 Tahun	
4	4004	Dhea	Muntaber	30 Tahunssss_9000_1	

3. Cari Data Pasien

CARI DATA PASIEN					
Masukan kata kunci untuk mencari pasien: 4001					
No	kamar	nama	penyakit	umur pasien	
1	4001	Viona	Magh	19 Tahun	

4. Tambah Data Pasien

```
TAMBAH DATA PASIEN

masukan dokter nama: dodo
masukan umur pasien: 18 Tahun
masukan dokter penyakit: Batuk
masukan kamar, format=(YYYY): 2001
[2001,dodo,Batuk,18 Tahun]
0

Data yang akan anda masukan adalah

primary key : dodo_2001_1
kamar : 2001
nama      : dodo
umur      : 18 Tahun
penyakit   : Batuk

Apakah akan ingin menambah data tersebut? (y/n)?
```

5. Ubah Data Pasien

```
Data baru anda adalah

kamar      : 4001 --> 7009
nama       : Viona --> dimas
penyakit    : Magh --> magh
umur       : 19 Tahun --> 90 tahun

apakah anda yakin ingin mengupdate data tersebut (y/n)?
```

6. Hapus Data Pasien

```

=====
                                HAPUS DATA PASIEN
=====

List pasien

| No | kamar | nama          | penyakit          | umur pasien
=====
| 1 | 4001 | Viona         | Magh              | 19 Tahun
| 2 | 4002 | Sahriah       | Pusing            | 20 Tahun
| 3 | 4003 | Cici          | DBD               | 25 Tahun
| 4 | 4004 | Dhea          | Muntaber          | 30 Tahunssss_9000_1
| 5 | 2004 | dodo          | demam             | nana
| 6 | 2001 | dodo          | Batuk             | 18 Tahun
=====

Masukan nomor pasien yang akan dihapus: 5

Data Yang Ingin Anda Hapus Adalah:
=====

dokter      : dodo_2004_1
kamar       : 2004
nama        : dodo

```

7. Tampil Data Pasien

```

=====
                                LIST SELURUH DATA PASIEN
=====

| No | kamar | nama          | penyakit          | umur pasien
=====
| 1 | 4001 | Viona         | Magh              | 19 Tahun
| 2 | 4002 | Sahriah       | Pusing            | 20 Tahun
| 3 | 4003 | Cici          | DBD               | 25 Tahun
| 4 | 4004 | Dhea          | Muntaber          | 30 Tahunssss_9000_1
| 5 | 2004 | dodo          | demam             | nana
| 6 | 2001 | dodo          | Batuk             | 18 Tahun
=====

```

Link Github :

<https://github.com/Viona1445/PBO>