LAPORAN 8
Pemrograman Berorientasi Objek

“Pewarisan & Koleksi Heterogen”

Dosen Pengampu : **Paulina Heruningsih Prima Rosa.**

DIBUAT OLEH :
Nama : Yohanis Celvin D.P.U.Pati
NIM : 245314033

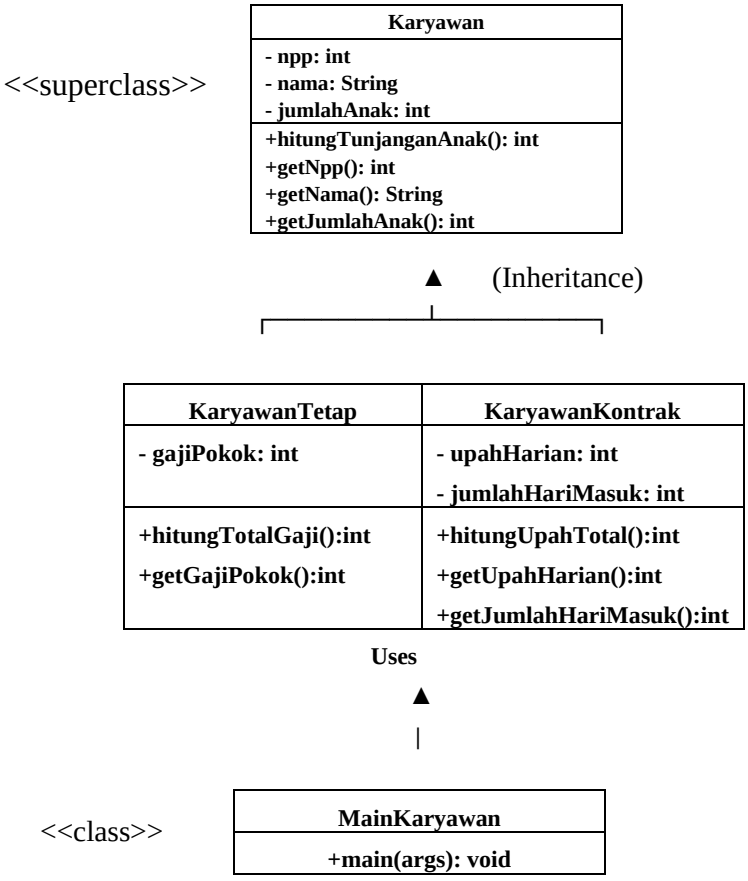
KELAS : BP

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2024

Kasus 1

A. DIAGRAM CLASS

Penggambaran diagram kelas



B. LISTING PROGRAM

```java
class Karyawan {

    // Deklarasi atribut private
    private int npp;            // Nomor Pegawai
    private String nama;        // Nama karyawan
    private int jumlahAnak;     // Jumlah anak karyawan

    // Variabel static untuk subsidi per anak (nilai sama untuk semua karyawan)
    public static int subsidiPerAnak = 100000;

    // Constructor untuk inisialisasi objek Karyawan
    public Karyawan(int npp, String nama, int jumlahAnak) {
        this.npp = npp;         // Set nilai npp
        this.nama = nama;       // Set nilai nama
        this.jumlahAnak = jumlahAnak; // Set nilai jumlahAnak
    }

    // Method menghitung tunjangan anak
    public int hitungTunjanganAnak() {
        return jumlahAnak * subsidiPerAnak;  // Hitung total subsidi anak
    }

    // Getter methods untuk mengakses atribut private
    public int getNpp() {
        return npp;
    }

    public String getNama() {
        return nama;
    }

    public int getJumlahAnak() {
        return jumlahAnak;
    }
}
```

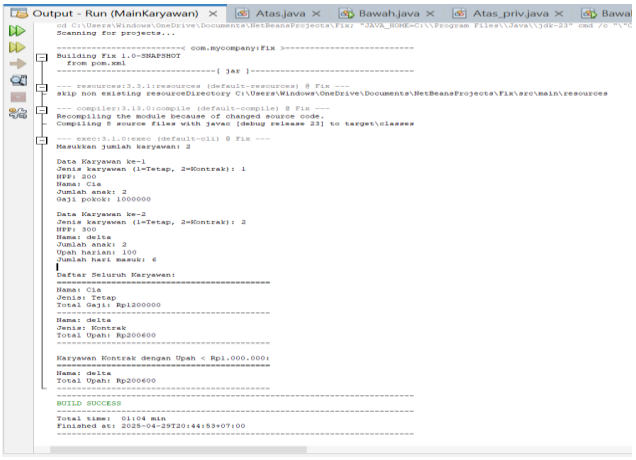
```java
/**
 *
 * @author Celvin Pati
 */
class KaryawanKontrak extends Karyawan {

    private int upahHarian;       // Upah per hari
    private int jumlahHariMasuk;  // Jumlah hari kerja

    // Constructor karyawan kontrak
    public KaryawanKontrak(int npp, String nama, int jumlahAnak, int upahHarian, int jumlahHariMasuk) {
        super(npp, nama, jumlahAnak);  // Panggil constructor parent class
        this.upahHarian = upahHarian;  // Set nilai upahHarian
        this.jumlahHariMasuk = jumlahHariMasuk;  // Set nilai jumlahHariMasuk
    }

    // Method hitung total upah (upah harian * hari kerja + tunjangan anak)
    public int hitungUpahTotal() {
        return (upahHarian * jumlahHariMasuk) + super.hitungTunjanganAnak();
    }

    // Getter methods
    public int getUpahHarian() {
        return upahHarian;
    }

    public int getJumlahHariMasuk() {
        return jumlahHariMasuk;
    }
}
```

```java
/**
 *
 * @author Celvin Pati
 */
class KaryawanTetap extends Karyawan {

    private int gajiPokok;    // Gaji pokok karyawan tetap

    // Constructor karyawan tetap
    public KaryawanTetap(int npp, String nama, int jumlahAnak, int gajiPokok) {
        super(npp, nama, jumlahAnak);  // Panggil constructor parent class
        this.gajiPokok = gajiPokok;    // Set nilai gajiPokok
    }

    // Method hitung total gaji (gaji pokok + tunjangan anak)
    public int hitungTotalGaji() {
        return gajiPokok + super.hitungTunjanganAnak();
    }

    public int getGajiPokok() {
        return gajiPokok;
    }
}
```

```java
/**
 *
 * @author Celvin Pati
 */
public class MainKaryawan {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);  // Buat objek Scanner
        ArrayList<Karyawan> daftarKaryawan = new ArrayList<>();  // Buat koleksi heterogen

        // Input jumlah karyawan
        System.out.print("Masukkan jumlah karyawan: ");
        int jumlah = scanner.nextInt();
        scanner.nextLine();  // Bersihkan newline di buffer

        // Loop untuk input data setiap karyawan
        for(int i=0; i<jumlah; i++) {
            System.out.println("\nData Karyawan ke-" + (i+1));

            // Input jenis karyawan
            System.out.print("Jenis karyawan (1=Tetap, 2=Kontrak): ");
            int jenis = scanner.nextInt();
            scanner.nextLine();  // Bersihkan newline di buffer

            // Input data umum semua karyawan
            System.out.print("NPP: ");
            int npp = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Nama: ");
            String nama = scanner.nextLine();
            System.out.print("Jumlah anak: ");
            int anak = scanner.nextInt();

            // Input data spesifik berdasarkan jenis karyawan
            if(jenis == 1) {
                // Input data karyawan tetap
                System.out.print("Gaji pokok: ");
                int gaji = scanner.nextInt();
                daftarKaryawan.add(new KaryawanTetap(npp, nama, anak, gaji));
            } else {
                // Input data karyawan kontrak
                System.out.print("Upah harian: ");
                int upah = scanner.nextInt();
                System.out.print("Jumlah hari masuk: ");
                int hari = scanner.nextInt();
                daftarKaryawan.add(new KaryawanKontrak(npp, nama, anak, upah, hari));
            }
        }

        // Tampilkan semua karyawan
        System.out.println("\nDaftar Seluruh Karyawan:");
        System.out.println("-----------------------------------------------");
        for(Karyawan k : daftarKaryawan) {
            System.out.println("Nama: " + k.getNama());

            // Cek jenis karyawan menggunakan instanceof
            if(k instanceof KaryawanTetap) {
                KaryawanTetap tetap = (KaryawanTetap)k;  // Downcasting
                System.out.println("Jenis: Tetap");
                System.out.println("Total Gaji: Rp" + tetap.hitungTotalGaji());
            } else {
                KaryawanKontrak kontrak = (KaryawanKontrak)k;  // Downcasting
                System.out.println("Jenis: Kontrak");
                System.out.println("Total Upah: Rp" + kontrak.hitungUpahTotal());
            }
            System.out.println("-----------------------------------------------");
        }

        // Tampilkan karyawan kontrak dengan upah < 1 juta
        System.out.println("\nKaryawan Kontrak dengan Upah < Rp1.000.000:");
        System.out.println("-----------------------------------------------");
        for(Karyawan k : daftarKaryawan) {
            if(k instanceof KaryawanKontrak) {
                KaryawanKontrak kontrak = (KaryawanKontrak)k;
                int total = kontrak.hitungUpahTotal();
                if(total < 1000000) {
                    System.out.println("Nama: " + k.getNama());
                    System.out.println("Total Upah: Rp" + total);
                    System.out.println("-----------------------------------------------");
                }
            }
        }

        scanner.close();  // Tutup Scanner
    }
}
```

C. OUTPUT Unit Test nya

- Setelah program nya saya perbaiki dan saya rapikan strukturnya maka program dapat tampil lebih bagus berikut utputnya:



D. ANALISA

- **Class Karyawan**

No	Syntax	penjelasan
1	public class Karyawan{ private int npp; private String nama; private int jumlahAnak;	// Atribut/variabel instan dari class Karyawan yang diset private agar hanya bisa diakses melalui method (encapsulation).
2	public static int subsidiPernk = 100;	// Atribut static berarti dimiliki bersama oleh semua objek dari class Karyawan.

3	public Karyawan(int npp, String nama, int jumlahAnak) {
	this.npp = npp;
	this.nama = nama;
	this.jumlahAnak = jumlahAnak;
	}
		// Constructor dengan parameter yang digunakan untuk menginisialisasi data saat objek dibuat.

4	public int hitungTunjanganAnak() { return jumlahAnak * subsidiPerAnak; }	//Method yang menghitung tunjangan anak berdasarkan jumlah anak dan subsidi per anak.
5	public int getNpp() { return npp; } public String getNama() { return nama; } public int getJumlahAnak() { return jumlahAnak; }	//Method getter untuk mengambil nilai atribut npp, nama, dan jumlahAnak.

- **Class KaryawanTetap**

No	Syntax	penjelasan
1	class KaryawanTetap extends Karyawan{	// KaryawanTetap adalah subclass (anak) dari Karyawan, artinya mewarisi atribut dan method dari superclass Karyawan (hubungan pewarisan/inheritance).
2	Private int gajiPokok;	// Atribut khusus untuk karyawan tetap, bersifat private, menyimpan nilai gaji pokok.
3	public KaryawanTetap(int npp, String nama, int jumlahAnak, int gajiPokok) { super(npp, nama, jumlahAnak); this.gajiPokok = gajiPokok; }	//Constructor dengan parameter. Menggunakan super(...) untuk memanggil constructor superclass, lalu menyetel nilai gajiPokok.
4	public int hitungTotalGaji() { return gajiPokok + super.hitungTunjanganAnak(); }	//Method untuk menghitung total gaji, yaitu gaji pokok ditambah tunjangan anak yang dihitung dari superclass.
5	public int getGajiPokok() { return gajiPokok; }	//Getter untuk mengakses nilai atribut gajiPokok. Mengikuti prinsip encapsulation.

- **Class KaryawanKontrak**

No	Syntax	penjelasan
1	class KaryawanKontrak extends Karyawan {	// Mendeklarasikan kelas KaryawanKontrak yang merupakan subclass dari kelas Karyawan, sehingga mewarisi atribut dan metode dari kelas Karyawan.
2	private int upahHarian;	// Mendeklarasikan atribut upahHarian, yang menyimpan informasi tentang upah per hari untuk karyawan kontrak, bersifat private.
3	private int jumlahHariMasuk;	//Mendeklarasikan atribut jumlahHariMasuk, yang menyimpan jumlah hari kerja bagi karyawan kontrak, bersifat private.

4	public KaryawanKontrak(int npp, String nama, int jumlahAnak, int upahHarian, int jumlahHariMasuk) {	//Constructor untuk kelas KaryawanKontrak dengan parameter yang menerima data seperti nomor pokok pegawai (NPP), nama, jumlah anak, upah harian, dan jumlah hari masuk. Menggunakan super(...) untuk memanggil konstruktor kelas induk Karyawan dan menginisialisasi atribut kelas.
5	super(npp, nama, jumlahAnak);	//Panggilan konstruktor superclass Karyawan untuk menginisialisasi atribut npp, nama, dan jumlahAnak pada kelas induk.
6	this.upahHarian = upahHarian;	//Menginisialisasi atribut upahHarian dengan nilai yang diterima sebagai parameter pada konstruktor.
7	this.jumlahHariMasuk = jumlahHariMasuk;	//Menginisialisasi atribut jumlahHariMasuk dengan nilai yang diterima sebagai parameter pada konstruktor.
8	public int hitungUpahTotal() {	//Mendeklarasikan metode hitungUpahTotal() yang digunakan untuk menghitung total upah karyawan kontrak.
9	return (upahHarian * jumlahHariMasuk) + super.hitungTunjanganAnak();	//Menghitung total upah dengan mengalikan upah harian dengan jumlah hari masuk, lalu menambahkan tunjangan anak yang dihitung dari superclass Karyawan.
10	public int getUpahHarian() {	//Mendeklarasikan metode getter untuk mengakses nilai upahHarian.
11	return upahHarian;	//Mengembalikan nilai dari atribut upahHarian.
12	public int getJumlahHariMasuk() {	//Mendeklarasikan metode getter untuk mengakses nilai jumlahHariMasuk.
13	return jumlahHariMasuk;	//Mengembalikan nilai dari atribut jumlahHariMasuk.

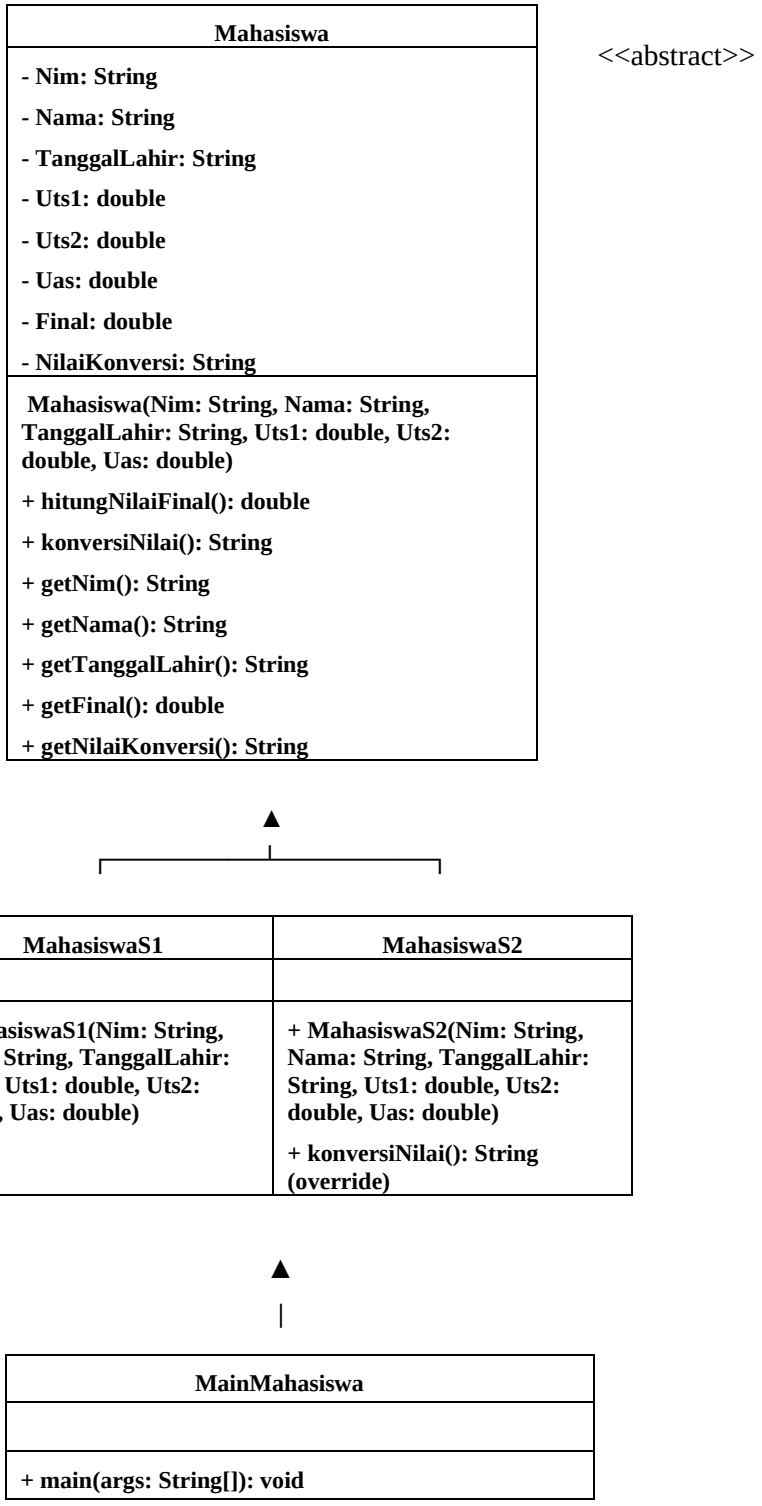
- **Class MainKaryawan**

No	Syntax	penjelasan
1	public class MainKaryawan {	// Mendeklarasikan kelas utama MainKaryawan tempat program dieksekusi.
2	Scanner scanner = new Scanner(System.i n);	// Membuat objek Scanner untuk membaca input dari pengguna.
3	ArrayList<Karyawan> daftarKaryawan = new ArrayList<>();	//Membuat list daftarKaryawan yang dapat menyimpan objek dari subclass Karyawan, seperti KaryawanTetap dan KaryawanKontrak.
4	System.out.print("Masukkan jumlah karyawan: ");	//Minta input jumlah karyawan.
5	int jumlah = scanner.nextInt();	//Membaca jumlah karyawan dari input.
6	scanner.nextLine();	//Membersihkan karakter newline di buffer agar tidak mengganggu input berikutnya.
7	for(int i=0; i<jumlah; i++) { ... }	//Loop untuk menginput data setiap karyawan.
8	System.out.print("Jenis karyawan (1=Tetap, 2=Kontrak): ");	//Meminta input jenis karyawan.
9	int jenis = scanner.nextInt();	//Membaca input jenis karyawan.
10	System.out.print("NPP: "); ... System.out.print("Jumlah anak: ");	//Input data umum karyawan (NPP, Nama, dan jumlah anak).
11	if(jenis == 1) { ... } else { ... }	//Percabangan untuk membedakan input karyawan tetap dan kontrak.

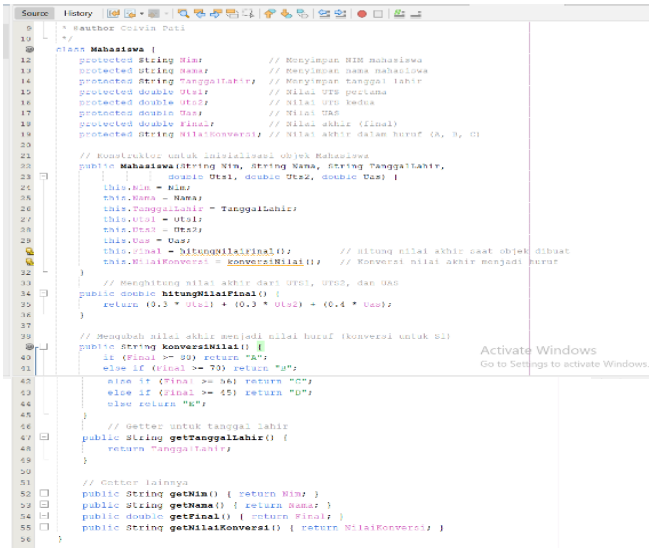
12	daftarKaryawan.add(new KaryawanTetap(...));	//Menambahkan objek KaryawanTetap ke list setelah input data lengkap.
13	daftarKaryawan.add(new KaryawanKontrak(...));	//Menambah objek KaryawanKontrak ke list.
14	for(Karyawan k : daftarKaryawan) { ... }	//Loop untuk menampilkan seluruh data karyawan.
15	if(k instanceof KaryawanTetap) { ... } else { ... }	//Memakai instanceof untuk memeriksa jenis objek karyawan, lalu melakukan downcasting ke jenis aslinya.
16	System.out.println("Total Gaji: Rp" + tetap.hitungTotalGaji());	//Menampilkan total gaji untuk karyawan tetap.
17	System.out.println("Total Upah: Rp" + kontrak.hitungUpahTotal());	//Menampilkan total upah untuk karyawan kontrak.
18	for(Karyawan k : daftarKaryawan) { ... if(total < 1000000) { ... } }	//Loop untuk menampilkan hanya karyawan kontrak dengan total upah di bawah Rp1.000.000.
19	scanner.close();	//Menutup objek Scanner setelah selesai digunakan.

Kasus 2

A. DIAGRAM CLASS

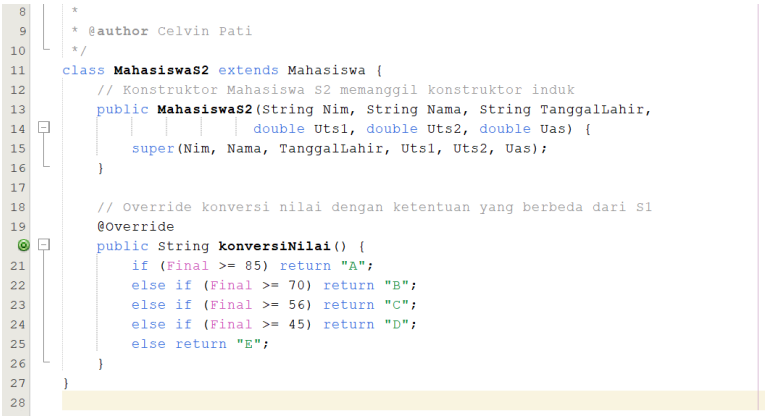


B. LISTING PROGRAM

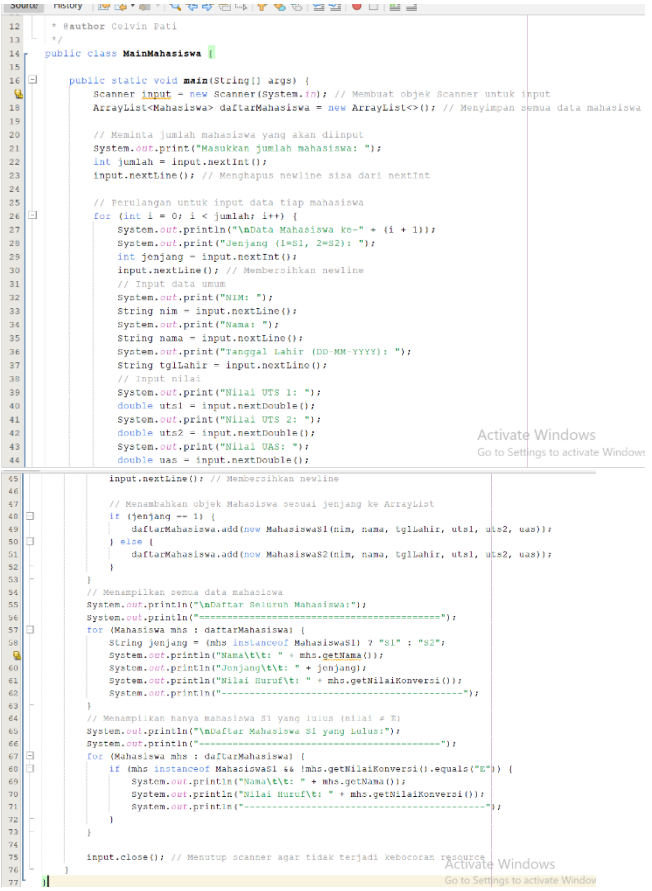


```
/**
 * @author Celvin Pati
 */
class Mahasiswa {
    protected String Nim;           // Menyimpan NIM mahasiswa
    protected String Nama;          // Menyimpan nama mahasiswa
    protected String TanggalLahir;  // Menyimpan tanggal lahir
    protected double Uts1;          // Nilai UTS pertama
    protected double Uts2;          // Nilai UTS kedua
    protected double Uas;           // Nilai UAS
    protected double Final;         // Nilai akhir (final)
    protected String NilaiKonversi; // Nilai akhir dalam huruf (A, B, C)

    // Konstruktor untuk inisialisasi objek Mahasiswa
    public Mahasiswa(String Nim, String Nama, String TanggalLahir,
                     double Uts1, double Uts2, double Uas) {
        this.Nim = Nim;
        this.Nama = Nama;
        this.TanggalLahir = TanggalLahir;
        this.Uts1 = Uts1;
        this.Uts2 = Uts2;
        this.Uas = Uas;
        this.Final = hitungNilaiFinal();        // hitung nilai akhir saat objek dibuat
        this.NilaiKonversi = konversiNilai();   // Konversi nilai akhir menjadi huruf
    }

    // Menghitung nilai akhir dari UTS1, UTS2, dan UAS
    public double hitungNilaiFinal() {
        return (0.3 * Uts1) + (0.3 * Uts2) + (0.4 * Uas);
    }

    // Mengubah nilai akhir menjadi nilai huruf (konversi untuk S1)
    public String konversiNilai() {
        if (Final >= 80) return "A";
        else if (Final >= 70) return "B";
        else if (Final >= 56) return "C";
        else if (Final >= 45) return "D";
        else return "E";
    }

    // Getter untuk tanggal lahir
    public String getTanggalLahir() {
        return TanggalLahir;
    }

    // Getter lainnya
    public String getNim() { return Nim; }
    public String getNama() { return Nama; }
    public double getFinal() { return Final; }
    public String getNilaiKonversi() { return NilaiKonversi; }
}
```

```
/**
 * @author Celvin Pati
 */
class MahasiswaS1 extends Mahasiswa {
    // Konstruktor Mahasiswa S1 memanggil konstruktor induk
    public MahasiswaS1(String Nim, String Nama, String TanggalLahir,
                       double Uts1, double Uts2, double Uas) {
        super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas);
    }
}
```



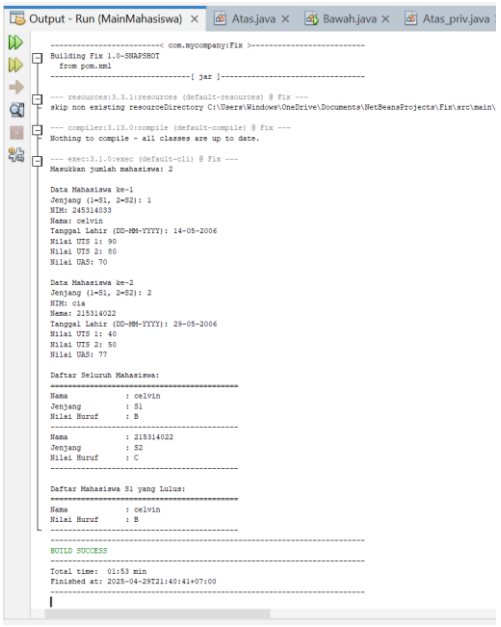
```
/**
 * @author Celvin Pati
 */
class MahasiswaS2 extends Mahasiswa {
    // Konstruktor Mahasiswa S2 memanggil konstruktor induk
    public MahasiswaS2(String Nim, String Nama, String TanggalLahir,
                       double Uts1, double Uts2, double Uas) {
        super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas);
    }

    // Override konversi nilai dengan ketentuan yang berbeda dari S1
    @Override
    public String konversiNilai() {
        if (Final >= 85) return "A";
        else if (Final >= 70) return "B";
        else if (Final >= 56) return "C";
        else if (Final >= 45) return "D";
        else return "E";
    }
}
```



```
/**
 * @author Celvin Pati
 */
public class MainMahasiswa {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in); // Membuat objek Scanner untuk input
        ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>(); // Menyimpan semua data mahasiswa

        // Meminta jumlah mahasiswa yang akan diinput
        System.out.print("Masukkan jumlah mahasiswa: ");
        int jumlah = input.nextInt();
        input.nextLine(); // Menghapus newline sisa dari nextInt

        // Perulangan untuk input data tiap mahasiswa
        for (int i = 0; i < jumlah; i++) {
            System.out.println("\nData Mahasiswa ke-" + (i + 1));
            System.out.print("Jenjang (1=S1, 2=S2): ");
            int jenjang = input.nextInt();
            input.nextLine(); // Membersihkan newline
            // Input data umum
            System.out.print("NIM: ");
            String nim = input.nextLine();
            System.out.print("Nama: ");
            String nama = input.nextLine();
            System.out.print("Tanggal Lahir (DD-MM-YYYY): ");
            String tglLahir = input.nextLine();
            // Input nilai
            System.out.print("Nilai UTS 1: ");
            double uts1 = input.nextDouble();
            System.out.print("Nilai UTS 2: ");
            double uts2 = input.nextDouble();
            System.out.print("Nilai UAS: ");
            double uas = input.nextDouble();
            input.nextLine(); // Membersihkan newline

            // Menambahkan objek Mahasiswa sesuai jenjang ke ArrayList
            if (jenjang == 1) {
                daftarMahasiswa.add(new MahasiswaS1(nim, nama, tglLahir, uts1, uts2, uas));
            } else {
                daftarMahasiswa.add(new MahasiswaS2(nim, nama, tglLahir, uts1, uts2, uas));
            }
        }
        // Menampilkan semua data mahasiswa
        System.out.println("\nDaftar Seluruh Mahasiswa:");
        System.out.println("========================================");
        for (Mahasiswa mhs : daftarMahasiswa) {
            String jenjang = (mhs instanceof MahasiswaS1) ? "S1" : "S2";
            System.out.println("Nama\t\t: " + mhs.getNama());
            System.out.println("Jenjang\t\t: " + jenjang);
            System.out.println("Nilai Huruf\t: " + mhs.getNilaiKonversi());
            System.out.println("----------------------------------------");
        }
        // Menampilkan hanya mahasiswa S1 yang lulus (nilai ≠ E)
        System.out.println("\nDaftar Mahasiswa S1 yang Lulus:");
        System.out.println("----------------------------------------");
        for (Mahasiswa mhs : daftarMahasiswa) {
            if (mhs instanceof MahasiswaS1 && !mhs.getNilaiKonversi().equals("E")) {
                System.out.println("Nama\t\t: " + mhs.getNama());
                System.out.println("Nilai Huruf\t: " + mhs.getNilaiKonversi());
                System.out.println("----------------------------------------");
            }
        }

        input.close(); // Menutup scanner agar tidak terjadi kebocoran resource
    }
}
```

C. OUTPUT

- Setelah program nya saya perbaiki dan saya rapikan strukturnya maka program dapat menampilkan hasil yang lebih baik dari pada hasil kerja saa di lab:



D. ANALISA

- **Class Mahasiswa**

No	Syntax	penjelasan
1	java public Mahasiswa(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) { this.Nim = Nim; this.Nama = Nama; this.TanggalLahir = TanggalLahir; this.Uts1 = Uts1; this.Uts2 = Uts2; this.Uas = Uas; this.Final = hitungNilaiFinal(); this.NilaiKonversi = konversiNilai(); }	//Konstruktor untuk menginisialisasi atribut dan langsung menghitung nilai akhir (Final) serta nilai huruf (NilaiKonversi).
2	java public double hitungNilaiFinal()	// Method untuk menghitung nilai akhir berdasarkan bobot: UTS1 30%, UTS2 30%, dan UAS 40%.

	{ return (0.3 * Uts1) + (0.3 * Uts2) + (0.4 * Uas); }	
3	java public String konversiNilai() { if (Final >= 80) return "A"; else if (Final >= 70) return "B"; else if (Final >= 56) return "C"; else if (Final >= 45) return "D"; else return "E"; }	// Method untuk mengubah nilai Final menjadi huruf berdasarkan skala penilaian S1.
4	java public String getTanggalLahir() { return TanggalLahir; }	// Getter untuk mengambil nilai dari atribut TanggalLahir.
5	java public String getNim() { return Nim; }	// Getter untuk mengambil nilai dari atribut Nim.
6	java public String getNama() { return Nama; }	// Getter untuk mengambil nilai dari atribut Nama.
7	java public double getFinal() { return Final; }	// Getter untuk mengambil nilai akhir (Final).
8	java public String getNilaiKonversi() { return NilaiKonversi; }	// Getter untuk mengambil nilai huruf (NilaiKonversi).

- **Class MahasiswaS1**

No	Syntax	penjelasan
1	java public MahasiswaS1(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) { super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas); }	// Konstruktor kelas MahasiswaS1 yang memanggil konstruktor kelas induk (Mahasiswa) menggunakan super untuk mengisi data mahasiswa.

- **Class MahasiswaS2**

No	Syntax	penjelasan
1	java public MahasiswaS2(String Nim, String Nama, String TanggalLahir, double Uts1, double Uts2, double Uas) { super(Nim, Nama, TanggalLahir, Uts1, Uts2, Uas); }	// Konstruktor MahasiswaS2 yang memanggil konstruktor induk (Mahasiswa) menggunakan super untuk menginisialisasi data.
2	java @Override public String konversiNilai() { if (Final >= 85) return "A"; else if (Final >= 70) return "B"; else if (Final >= 56) return "C"; else if (Final >= 45) return "D"; else return "E"; }	//Method override dari konversiNilai() yang mengganti aturan konversi nilai khusus untuk jenjang S2.

- **MainMahasiswa**

No	Syntax	penjelasan
1	java Scanner input = new Scanner(System.in);	// Membuat objek Scanner untuk membaca input dari pengguna.
2	java ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>();	//Membuat ArrayList untuk menyimpan objek dari class Mahasiswa (baik S1 maupun S2).

3	javaSystem.out.print("Masu kkan jumlah mahasiswa: "); int jumlah = input.nextInt(); input.nextLine();	//Menerima jumlah mahasiswa yang akan diinput, lalu membersihkan newline.
4	java for (int i = 0; i < jumlah; i++) { ... }	//Melakukan pengulangan untuk input data tiap mahasiswa sebanyak jumlah yang dimasukkan.
5	java System.out.print("Jenjang (1=S1, 2=S2): "); int jenjang = input.nextInt(); input.nextLine();	//Menerima input jenjang pendidikan (S1 atau S2), lalu membersihkan newline.
6	java System.out.print("NIM: "); String nim = input.nextLine(); ...	//Mengambil input identitas mahasiswa: NIM, Nama, Tanggal Lahir, dan nilai ujian.
7	java if (jenjang == 1) { daftarMahasiswa.add(new MahasiswaS1(...)); } else { daftarMahasiswa.add(new MahasiswaS2(...)); }	//Menambahkan objek ke ArrayList sesuai jenjang: MahasiswaS1 atau MahasiswaS2.
8	java for (Mahasiswa mhs : daftarMahasiswa) { ... }	//Menampilkan semua data mahasiswa, termasuk jenjang dan nilai huruf.
9	java if (mhs instanceof MahasiswaS1 && !mhs.getNilaiKonversi().equals("E")) { ... }	//Menampilkan hanya mahasiswa S1 yang nilai hurufnya bukan "E" (berarti lulus).
10	java input.close();	Menutup objek Scanner untuk menghindari kebocoran resource.