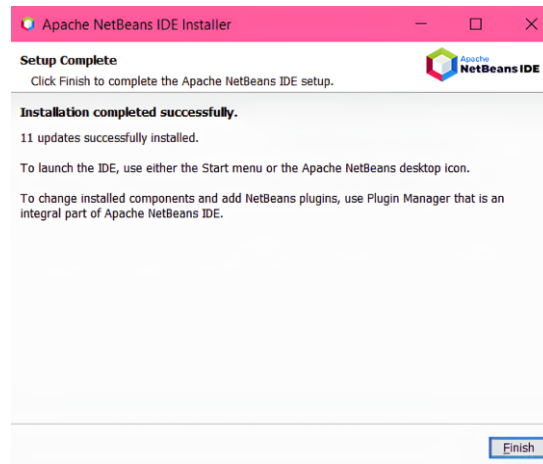


Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

MODUL 2 PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

Laporkan hasil praktikum berikut dengan hasil penugasan dan penjelasannya ke Dosen dalam bentuk file pdf dengan format nama <<nim>>_modul2.

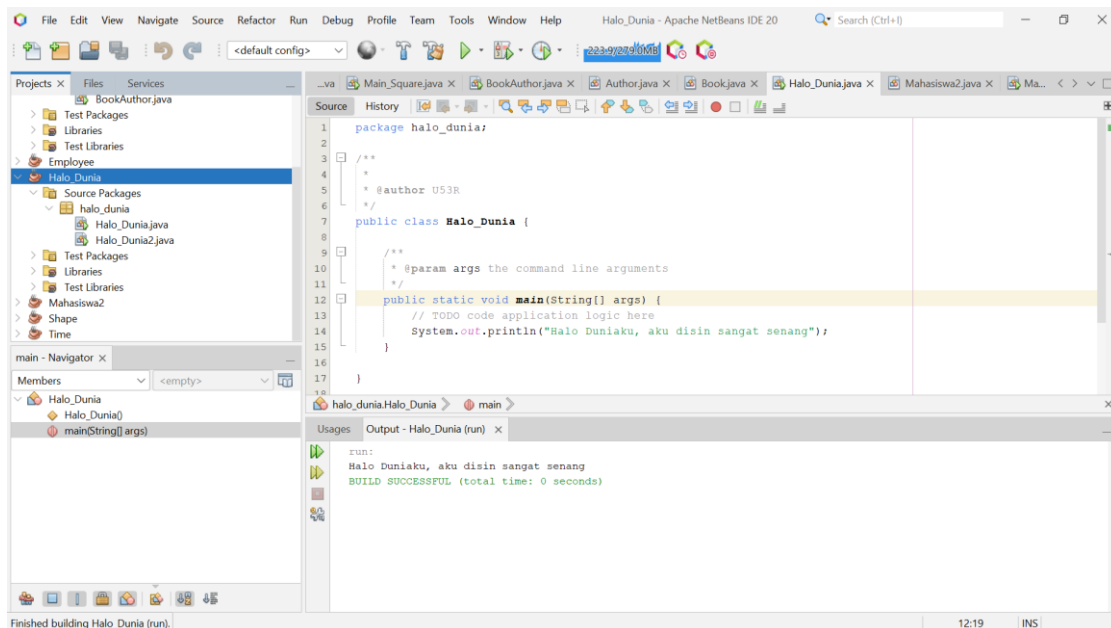
1. Tangkapan layar hasil akhir instalasi



2. Tangkapan layar hasil running

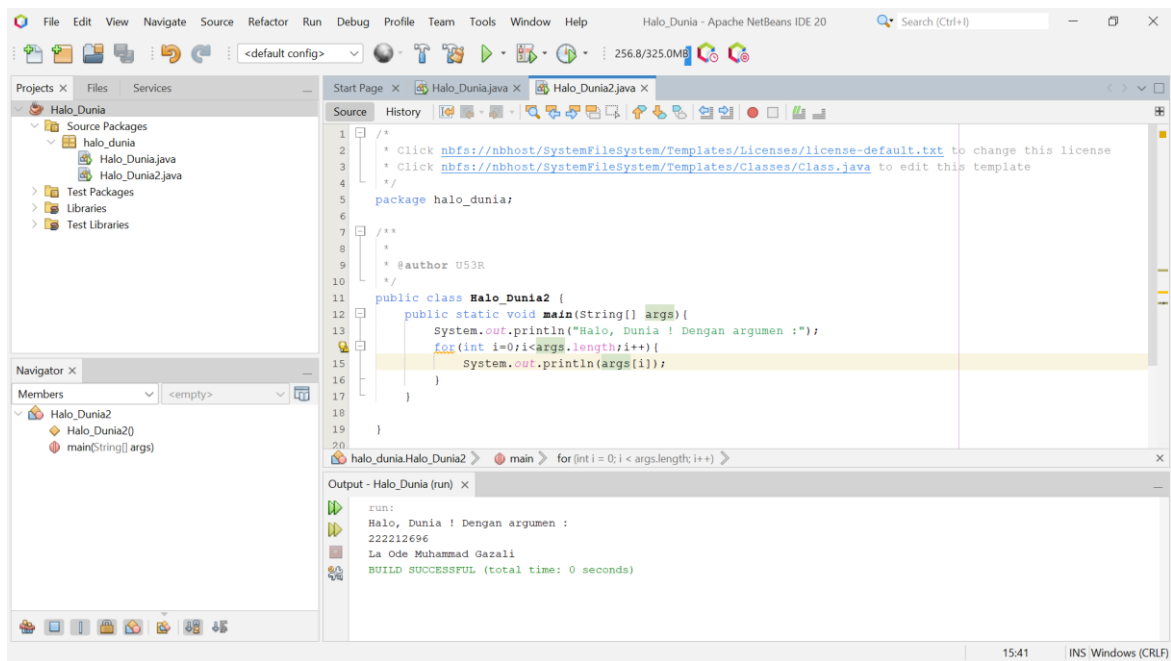
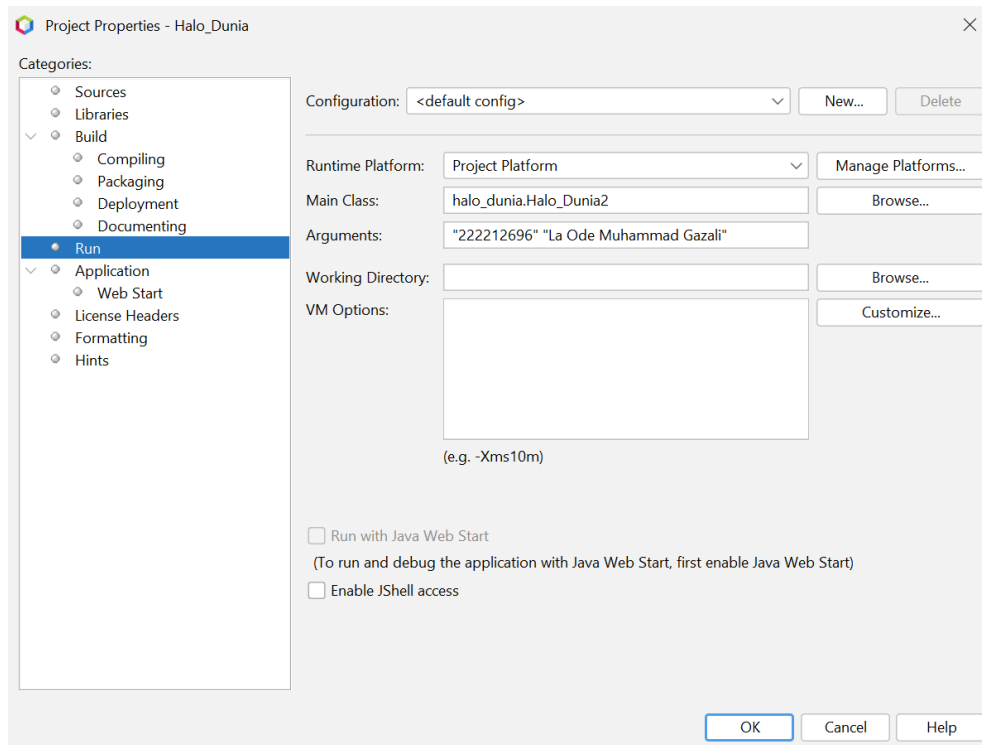
A. Mencoba pemrograman Java di Netbeans

- HaloDunia.java



- **HaloDunia2.java**

Terlebih dahulu mengatur Main Class dan memberikan argument Nim dan nama sendiri

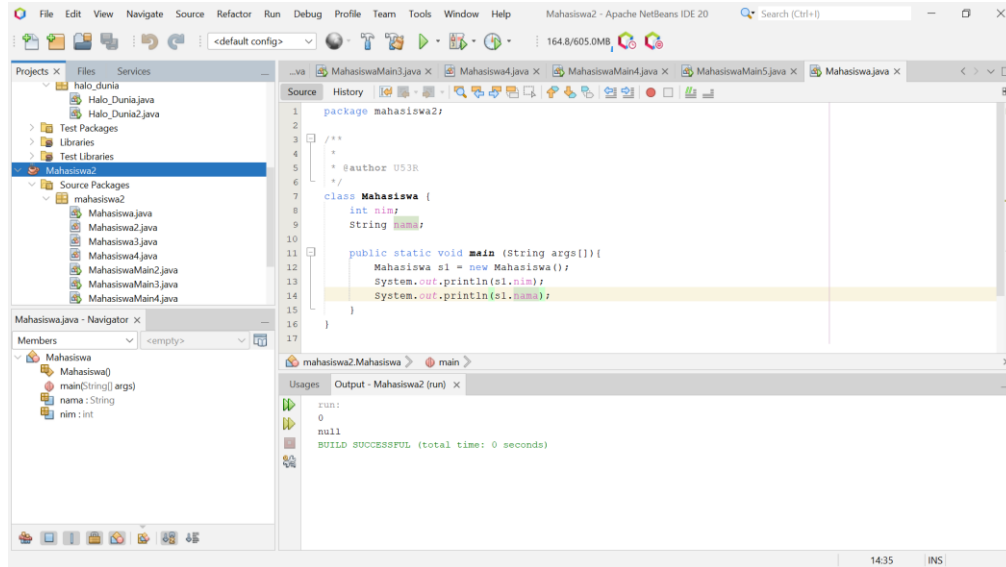


B. Objek vs Class

- Class

Secara sederhana, pembuatan class dalam java dapat dilakukan sebagai berikut:

Mahasiswa.java



Karena variabel nim dan nama belum diinisiasi nilainya, maka saat di run menghasilkan nim=0 dan nama=null.

Namun, seringkali dalam *real-time development* seringkali dibuat dalam file yang berbeda namun masih dalam satu file sumber yang sama.

- Objek

Terdapat 3 cara dalam menginisialisasi objek di Java, yaitu:

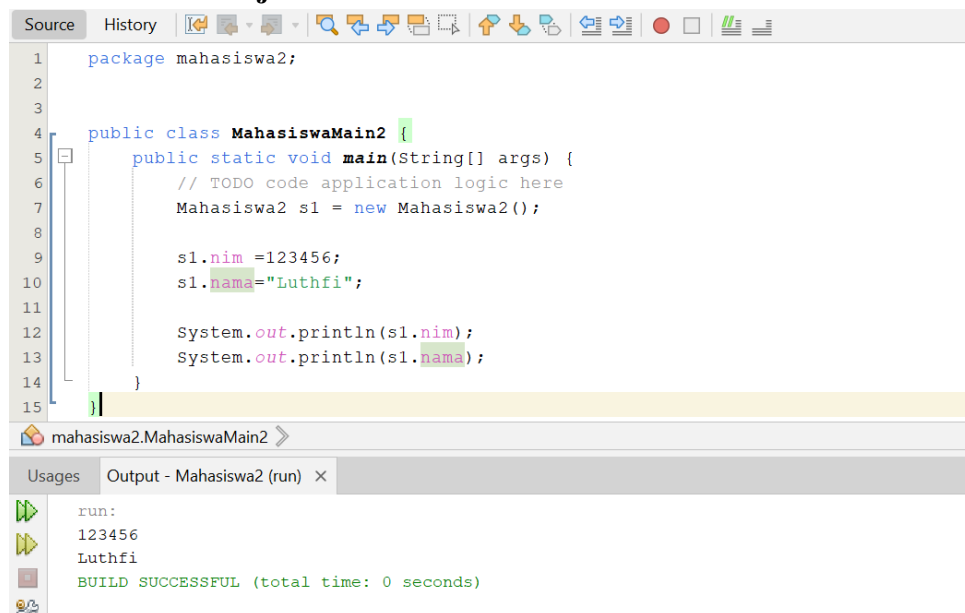
- Inisialisasi melalui referensi

Mahasiswa2.java

```
1 package mahasiswa2;
2
3 public class Mahasiswa2 {
4     int nim;
5     String nama;
6 }
```

Program tersebut hanya mendeklarasikan variabel nim bertipe integer dan nama bertipe String. Untuk membuat objek dapat dilakukan seperti pada program MahasiswaMain2.java dan MahasiswaMain3.java berikut :

MahasiswaMain2.java



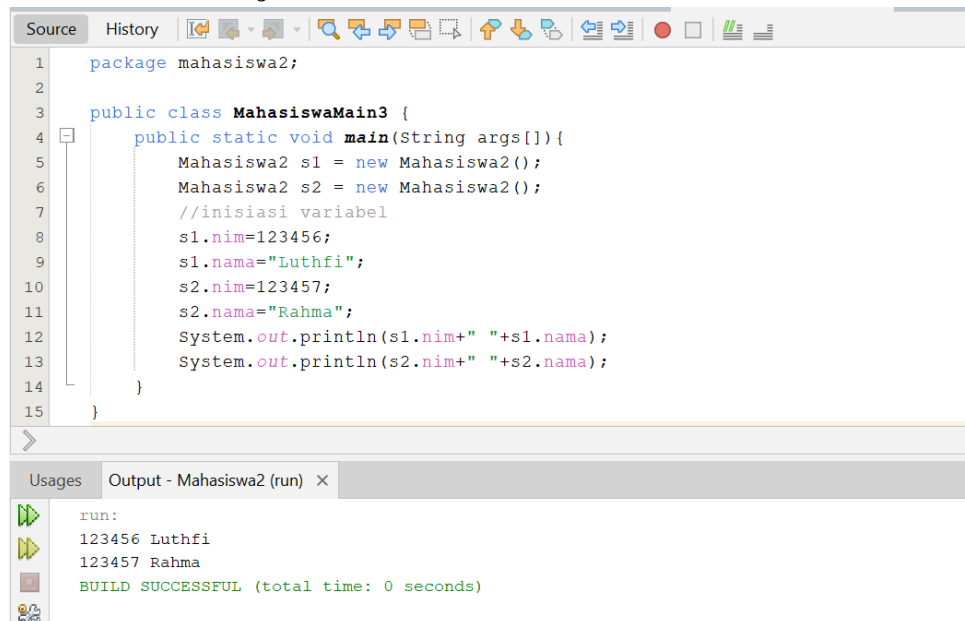
```
1 package mahasiswa2;
2
3
4 public class MahasiswaMain2 {
5     public static void main(String[] args) {
6         // TODO code application logic here
7         Mahasiswa2 s1 = new Mahasiswa2();
8
9         s1.nim = 123456;
10        s1.nama = "Luthfi";
11
12        System.out.println(s1.nim);
13        System.out.println(s1.nama);
14    }
15 }
```

run:

```
123456
Luthfi
BUILD SUCCESSFUL (total time: 0 seconds)
```

Dalam program tersebut, dibuat sebuah objek bernama **s1** dari kelas Mahasiswa2. Kemudian objek tersebut di inisialisasi nim dan namanya, kemudia hasil run –nya seperti gambar diatas.

MahasiswaMain3.java



```
1 package mahasiswa2;
2
3 public class MahasiswaMain3 {
4     public static void main(String args[]){
5         Mahasiswa2 s1 = new Mahasiswa2();
6         Mahasiswa2 s2 = new Mahasiswa2();
7         //inisiasi variabel
8         s1.nim=123456;
9         s1.nama="Luthfi";
10        s2.nim=123457;
11        s2.nama="Rahma";
12        System.out.println(s1.nim+" "+s1.nama);
13        System.out.println(s2.nim+" "+s2.nama);
14    }
15 }
```

run:

```
123456 Luthfi
123457 Rahma
BUILD SUCCESSFUL (total time: 0 seconds)
```

Program MahasiswaMain3.java diatas membuat dua buah objek yaitu **s1** dan **s2** dari kelas Mahasiswa2 kemudian menginisialisasi kedua objek dengan value yang berbeda. Hasil run-nya dapat dilihat pada gambar diatas.

- Inisialisasi melalui metode

Mahasiswa3.java

```
1 package mahasiswa2;
2
3 public class Mahasiswa3 {
4     int nim;
5     String nama;
6
7     void tambahData(int vnim, String vnama){
8         nim = vnim;
9         nama = vnama;
10    }
11
12    void tampilkanInfo(){
13        System.out.println(nim + " " +nama);
14    }
15 }
```

Dalam program tersebut, selain mendaklarasikan variabel dan tipe datanya, terdapat pula dua buah metode bertipe void, yaitu metode tambahData dengan parameter vnim dan vnama untuk proses memasukan nilai dari sebuah objek kedalam variabel, dan juga terdapat metode tampilkanInfo() untuk menampilkan isi dari sebuah objek.

MahasiswaMain4.java

```
1 package mahasiswa2;
2
3 public class MahasiswaMain4 {
4     public static void main (String args[]){
5         Mahasiswa3 s1 = new Mahasiswa3();
6         Mahasiswa3 s2 = new Mahasiswa3();
7
8         s1.tambahData(123456, "Luthfi");
9         s2.tambahData(123457, "Rahma");
10        s1.tampilkanInfo();
11        s2.tampilkanInfo();
12    }
13
14 }
15 }
```

mahasiswa2.MahasiswaMain4

Usages Output - Mahasiswa2 (run) ×

run:
123456 Luthfi
123457 Rahma
BUILD SUCCESSFUL (total time: 0 seconds)

Pada kelas main pada program tersebut, proses insialiasasi objek-objek dapat dilakukan dengan langsung memanggil metode yang dimaksud dan memberikan value pada parameternya. Sehingga ketika di-run hasilnya seperti gambar diatas.

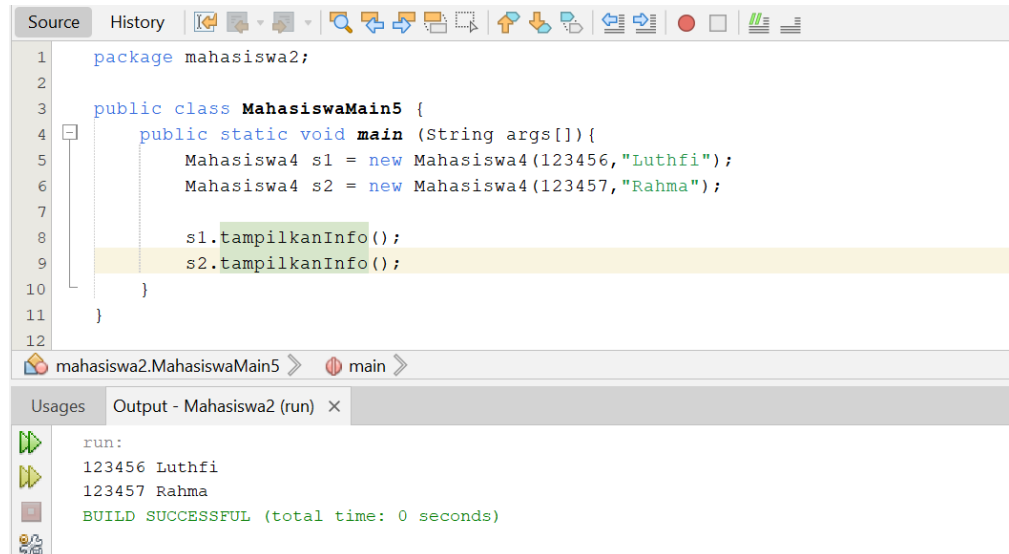
- Inisialisasi melalui konstruktor

Mahasiswa4.java

```
1 package mahasiswa2;
2
3 public class Mahasiswa4 {
4     int nim;
5     String nama;
6
7     Mahasiswa4(int vnim, String vnama){
8         nim = vnim;
9         nama = vnama;
10    }
11
12    void tampilkanInfo(){
13        System.out.println(nim + " " + nama);
14    }
15 }
```

Dalam program tersebut, selain mendaklarasikan variabel dan tipe datanya, terdapat pula sebuah konstrukot yang memiliki nama yang sama dengan nama kelasnya, yang juga disertai dengan parameter/argument. Kemudian untuk menampilkan isi dari sebuah objek, terdapat sebuah metode tampilkanInfo().

MahasiswaMain5.java



```
Source History
1 package mahasiswa2;
2
3 public class MahasiswaMain5 {
4     public static void main (String args[]){
5         Mahasiswa4 s1 = new Mahasiswa4(123456,"Luthfi");
6         Mahasiswa4 s2 = new Mahasiswa4(123457,"Rahma");
7
8         s1.tampilkanInfo();
9         s2.tampilkanInfo();
10    }
11 }
mahasiswa2.MahasiswaMain5 main
Usages Output - Mahasiswa2 (run) x
run:
123456 Luthfi
123457 Rahma
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pada fungsi main() diatas, proses inisialisasi nilai dari sebuah objek dapat langsung dengan mengisi parameter pada konstruktor yang ada, sehingga tidak perlu lagi dilakukan dengan mereferensi variabel ataupun memanggil sebuah metode tertentu. Kemudian untuk menjalankan program dapat langsung memanggil metode tampilkanInfo() untuk setiap objek.