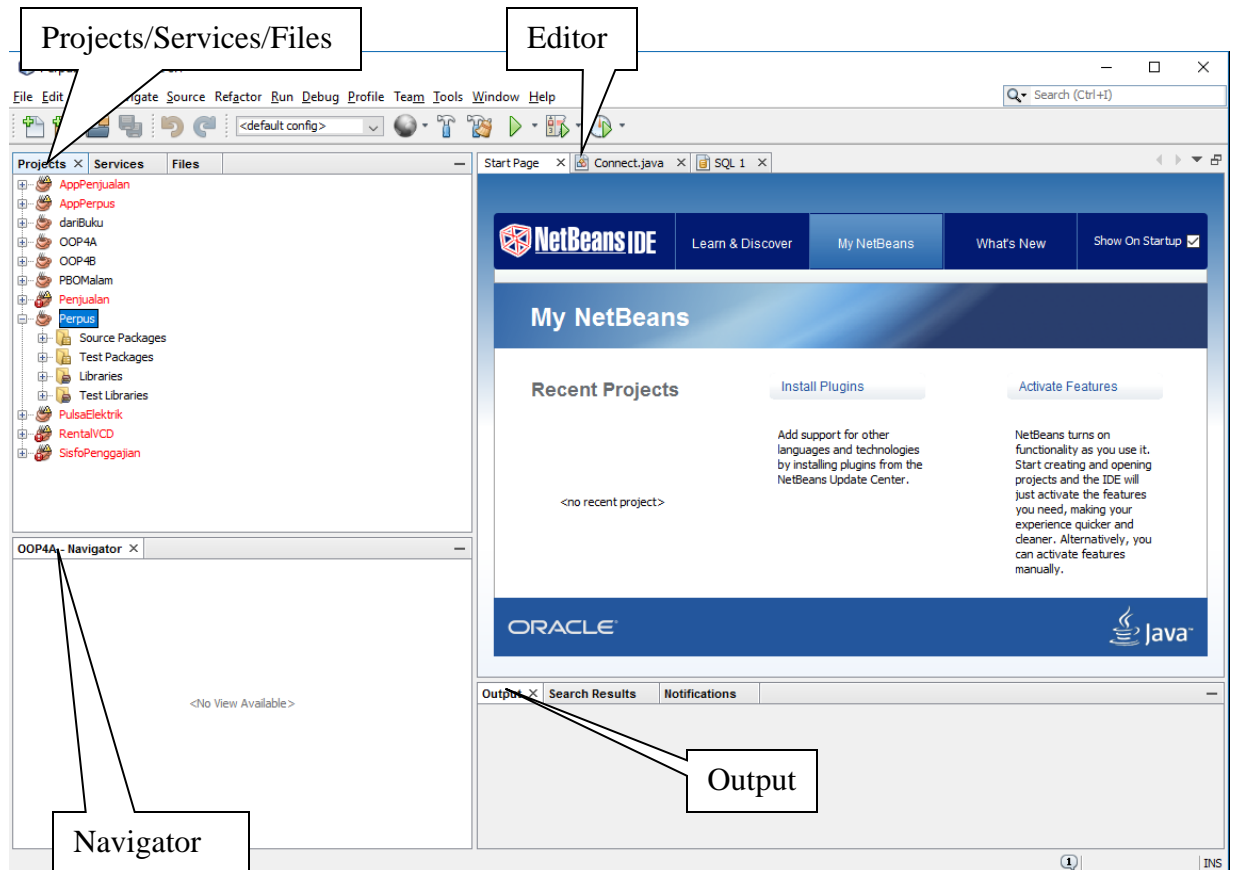


# PRAKTEK 1

## CLASS, ATTRIBUTE, METHOD, DAN OBJEK

### 1.1 NetBeans 8.1

Sebelum membuat project ada baiknya kenali dulu windows yang ada pada NetBeans 8.1



**Projects** berisikan daftar project yang dikelola oleh NetBeans, **Services** berisikan layanan-layanan termasuk apabila nantinya kita menggunakan MySQL maka akan terhubung melalui Services ini, **Files** menampilkan file-file yang dikelola oleh NetBeans.

**Editor** berisikan Source, History, dan Design (hanya ada jika merupakan visual Form). Source berisikan text editor untuk menuliskan source code, History menampilkan riwayat perubahan, dan Design menampilkan desain form.

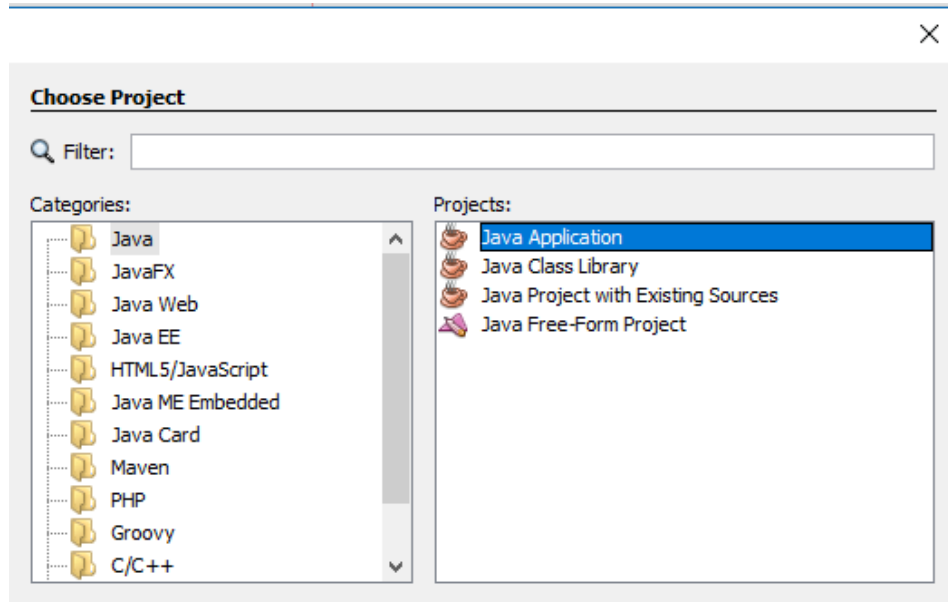
**Navigator** menampilkan daftar atribut, method, dan komponen visual yang terdapat dalam satu class, sedangkan **Output** menampilkan hasil output program yang bisa berisikan Text, Pesan Error, dan pesan berjalannya program.

Beberapa hal yang sering tidak menyebabkan kesalahan dalam praktek dan harus diingat dalam menuliskan program di NetBeans adalah:

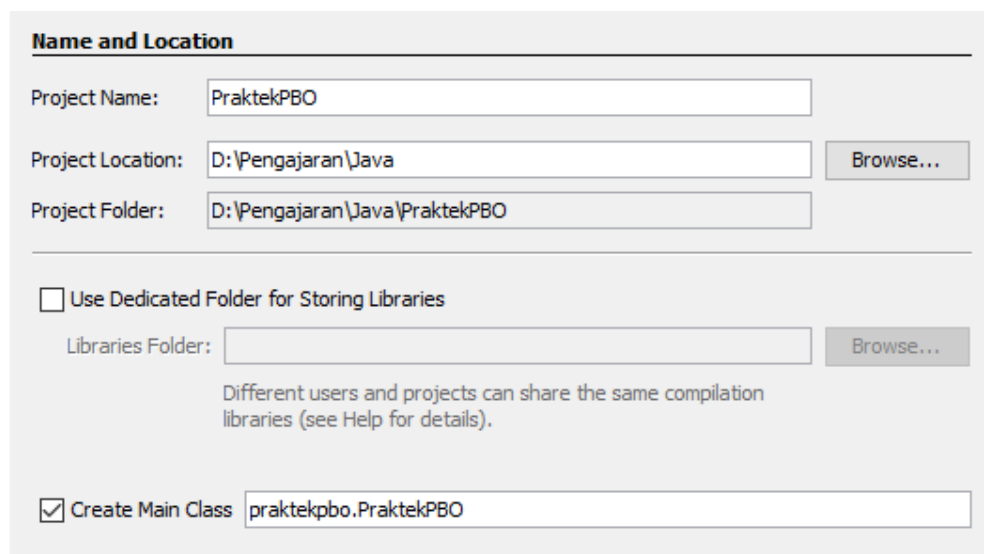
1. **Menuliskan Nama** : dalam menuliskan nama baik attribute, variable, method, class, nama komponen visual dan lainnya
  - a. Hanya boleh huruf, angka dan garis bawah ( \_ ), **tidak boleh menggunakan spasi**,
    - i. bingka barandam
    - ii. bingka-barandam
    - iii. bingka+barandam
  - b. tidak boleh juga diawali dengan angka,
    - i. 7bingkabarandam
    - ii. 41macamwadai
  - c. tidak boleh menggunakan **reserved words** atau kata yang dipakai oleh java
    - i. System
    - ii. int
2. jadi jika kita akan memberikan nama maka bisa ditulis dengan
  - i. **bingkabarandam1**
  - ii. **bingkaBarandam1**
  - iii. **BINGKABARANDAM1**
  - iv. **bingka\_barandam1**
3. **Case Sensitive** : Java membedakan antara huruf besar dan kecil sehingga apabila kita membuat suatu atribut **jukung**, kemudian **Jukung**, lalu **JUKUNG**, ketiganya dianggap berbeda. Sama halnya ketika kita membuat atribut **Manday**, tapi kita panggil dengan **manday**, maka akan terjadi kesalahan.

## 1.2 Membuat Project

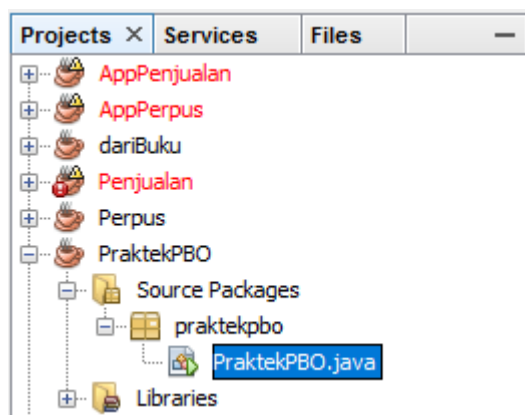
1. Awali dengan Do'a Bismillahir Rohmaanir Rohiim, Robbiy zidniy 'ilma warzuqniy fahma ☺
2. Pilih File -> New Project atau langsung dengan shortcut Ctrl+Shift+N
3. Pada Categories Pilih **Java** kemudian pada Projects pilih **Java Application** kemudian **Next**



4. Tuliskan nama project **PraktekPBO** pilih lokasi project dengan **Browse**, pastikan **Create Main Class** tercentang

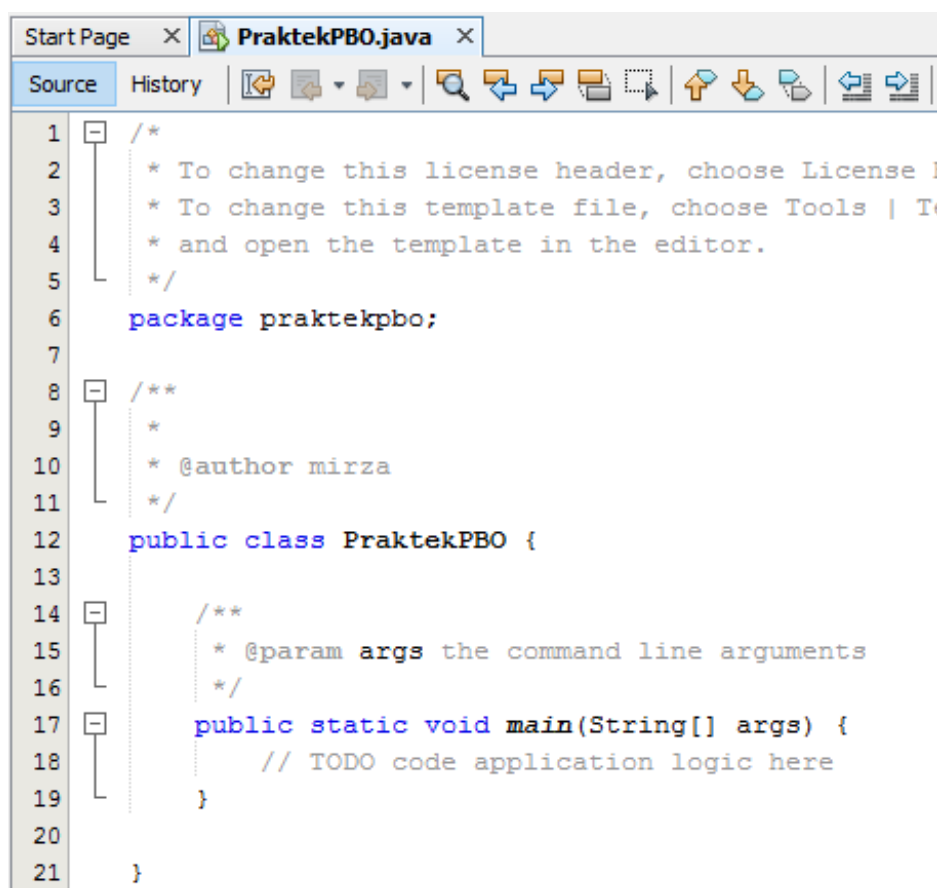


5. Perhatikan pada windows **Project**



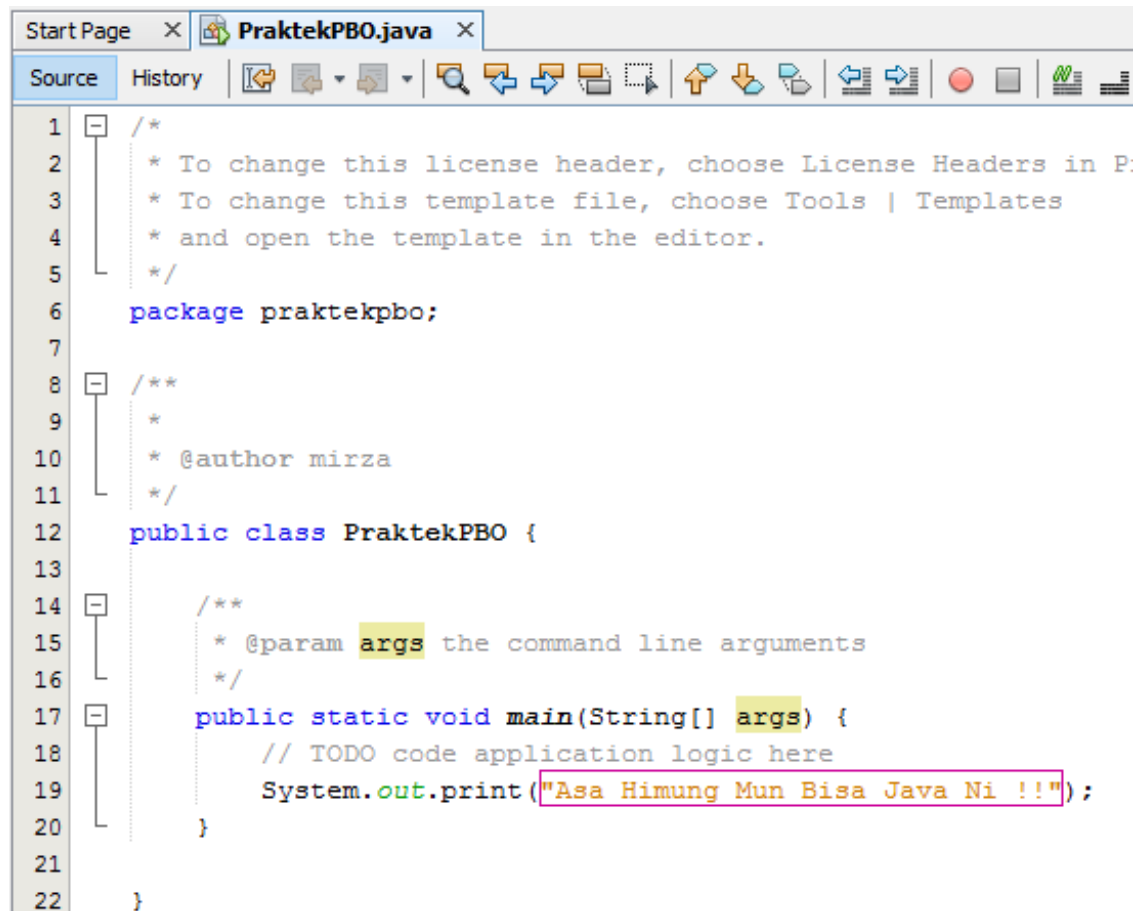
Nama project **PraktekPBO** nama package pada Source Packages adalah **praktekpbo** dan main class otomatis dibuat dengan nama **PraktekPBO.java**

6. Perhatikan pada windows **Source** isi dari **PraktekPBO.java**



sudah terdapat method main **public static void main (String[]args)** yang mana didalamnya ini yang akan kita isi dengan source code.

7. Isi dengan perintah sebagai berikut

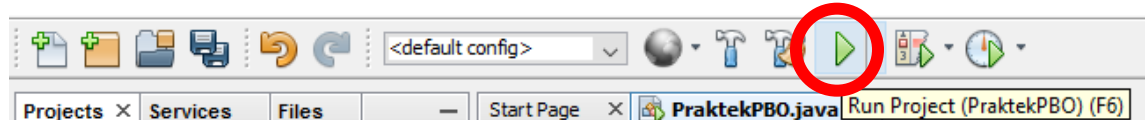


```

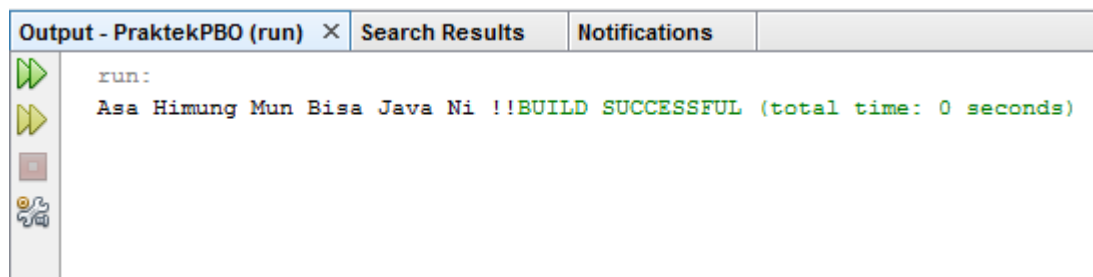
1  /*
2   * To change this license header, choose License Headers in P
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package praktekpbo;
7
8  /**
9   *
10   * @author mirza
11   */
12  public class PraktekPBO {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.print("Asa Himung Mun Bisa Java Ni !!");
20      }
21
22  }

```

8. Jalankan dengan tombol F6 atau klik tombol berikut



9. Perhatikan hasilnya pada windows **Output**



10. Bagaimana ? sudah merasa himung ? ☺

11. Berikutnya coba tambahkan perintah berikut:

```

12  public class PraktekPBO {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          String s;
20          int i;
21          double d;
22
23          s="Ini isi variable String";
24          i=1;
25          d=0.875;
26
27          System.out.println("Asa Himung Mun Bisa Java Ni !!");
28          System.out.println(s);
29          System.out.println(i);
30          System.out.println(d);
31
32      }
33
34  }

```

Baris 19 adalah cara mendeklarasikan variable s dengan tipe data String






Baris 20 adalah cara mendeklarasikan variable i dengan tipe data integer

Baris 21 adalah cara mendeklarasikan variable d dengan tipe data double

Baris 23-25 adalah cara mengisi variable

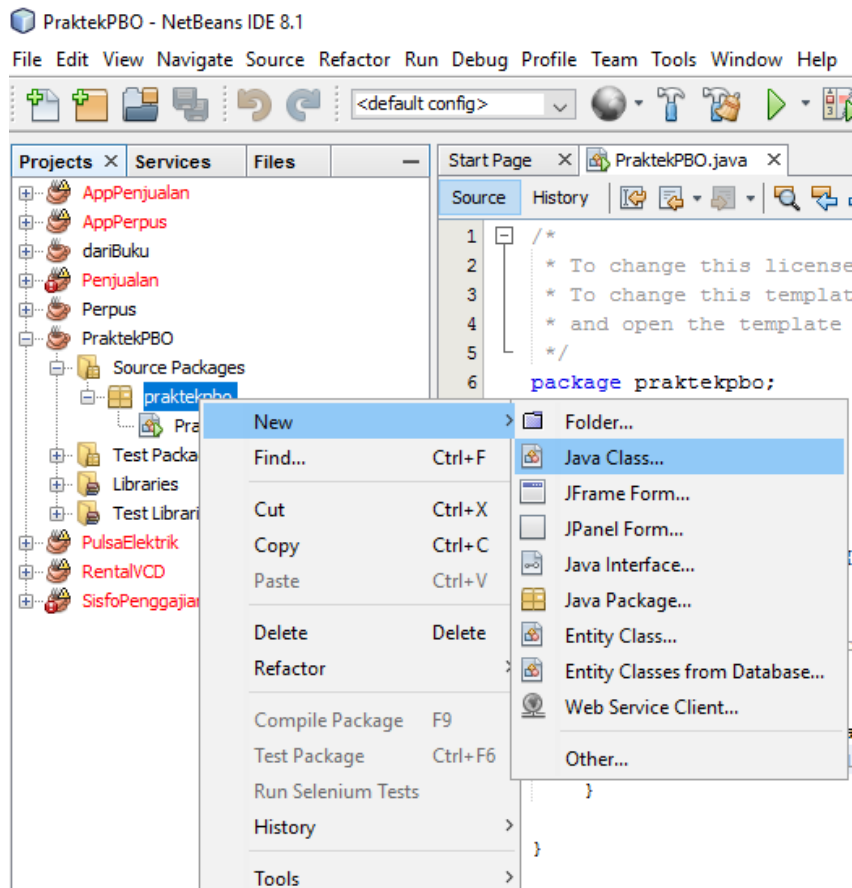
Baris 28-30 adalah cara menampilkan isi variable

12. Perhatikan hasilnya pada windows **Output**

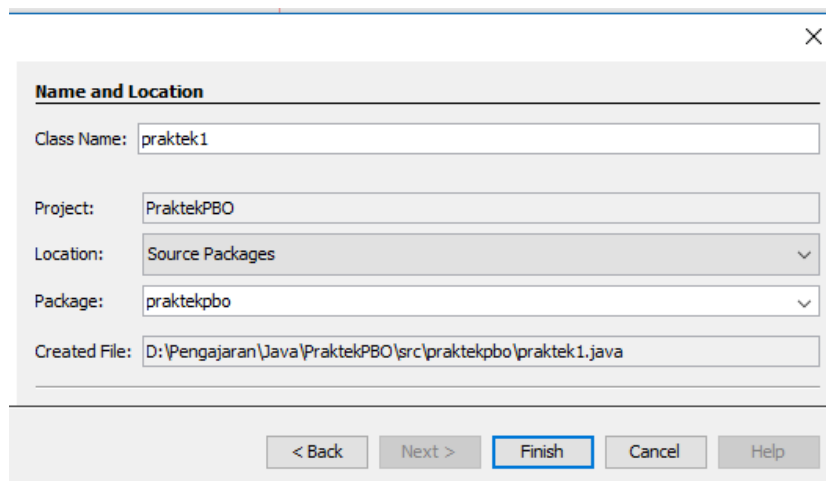
Output - PraktekPBO (run) ×	Search Results	Notifications
 run:  Asa Himung Mun Bisa Java Ni !!  Ini isi variable String  1  0.875 BUILD SUCCESSFUL (total time: 0 seconds)		

### 1.3 Membuat Class

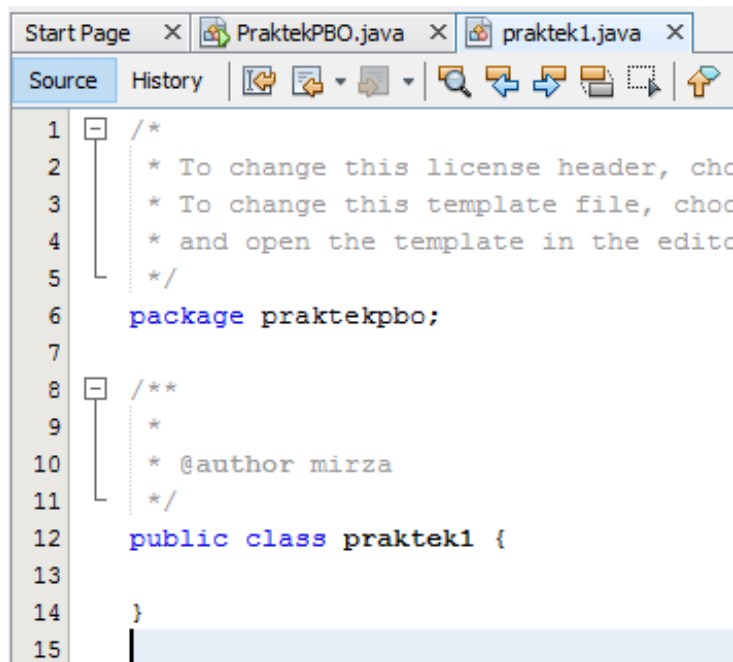
1. Klik kanan pada package yang diinginkan, dalam praktek ini pada package **praktekpbo** kemudian pilih New -> Java Class..



2. Isi nama class **praktek1** kemudian klik Finish



3. Perhatikan pada windows **Source**



```

1  /*
2   * To change this license header, choose the
3   * To change this template file, choose the
4   * and open the template in the editor.
5   */
6  package praktekpbo;
7
8  /**
9   *
10   * @author mirza
11   */
12  public class praktek1 {
13
14  }
15

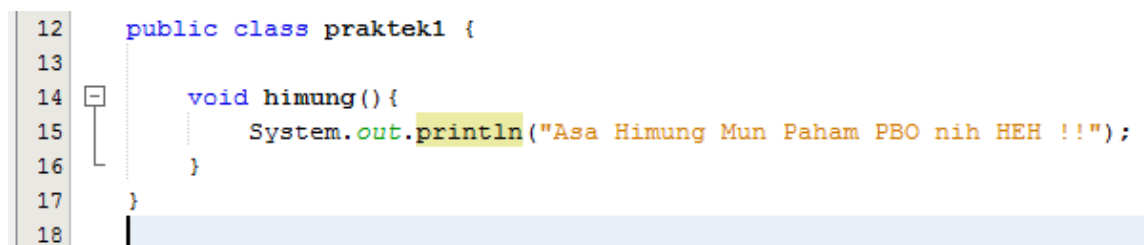
```

Ya gini deh hasilnya

4. Kita sekarang punya 2 file terbuka di windows Source yaitu class yang mengandung method main pada **PraktekPBO.java** dan class yang kita buat di **praktek1.java** attribute dan method akan kita isi di **praktek1.java** sedangkan untuk menjalankannya ataupun mengisi attributnya perlu dibuat object (instance) terlebih dahulu pada **PraktekPBO.java**

#### 1.4 Membuat Method

1. Tuliskan perintah berikut (baris 14-16) didalam class praktek1



```

12  public class praktek1 {
13
14      void himung() {
15          System.out.println("Asa Himung Mun Paham PBO nih HEH !!");
16      }
17  }
18

```

Untuk menjalankannya perlu dilakukan instance (membuat object) pada kelas yang mengandung method main.



2. Tuliskan perintah berikut (baris 20-21) pada file **PraktekPBO.java** (perintah dari sub bab 1.2 bisa dihapus)

```

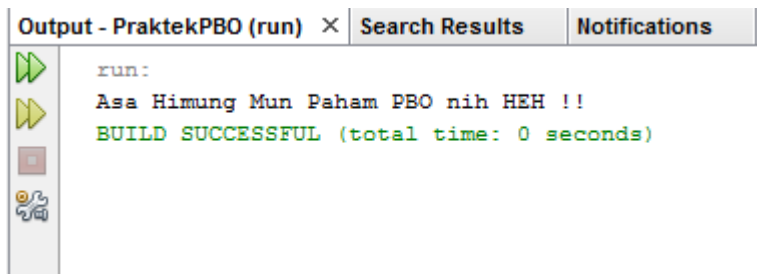
12  public class PraktekPBO {
13
14      /**
15       * @param args the command line arguments
16       */
17
18      public static void main(String[] args) {
19          // TODO code application logic here
20          praktek1 objekP = new praktek1();
21          objekP.himung();
22      }
23  }

```

Baris 20 adalah cara instance objek

Baris 21 adalah cara menjalankan method dari class yang sudah kita buat sebelumnya

3. Perhatikan hasilnya pada windows **Output**



The screenshot shows the 'Output - PraktekPBO (run)' window. It contains the following text:

```

run:
Asa Himung Mun Paham PBO nih HEH !!
BUILD SUCCESSFUL (total time: 0 seconds)

```

## 1.5 Membuat Attribute

1. Kembali ke file **praktek1.java** kemudian tambahkan perintah berikut (18-20)

```

12  public class praktek1 {
13
14      void himung() {
15          System.out.println("Asa Himung Mun Paham PBO nih HEH !!");
16      }
17
18      String s;
19      int i;
20      double d;
21  }

```

Sebagaimana method, attribute ini pun hanya akan diisi sesudah dilakukan instance terlebih dahulu

2. Pindah ke file **PraktekPBO.java** dan isi perintah berikut (23-29)

```







12  public class PraktekPBO {
13
14      /**
15       * @param args the command line arguments
16       */
17
18      public static void main(String[] args) {
19          // TODO code application logic here
20          praktek1 objekP = new praktek1();
21          objekP.himung();
22
23          objekP.s="Ulun ni harat meolah program, han..";
24          objekP.i=90;
25          objekP.d=0.875;
26
27          System.out.println(objekP.s);
28          System.out.println(objekP.i);
29          System.out.println(objekP.d);
30      }
31  }

```

Baris 23-25 adalah cara mengisi attribute dari suatu object

Baris 27-29 adalah cara menampilkan isinya

3. Perhatikan hasilnya pada windows **Output**

Output - PraktekPBO (run) ×	Search Results	Notifications
 run:  Asa Himung Mun Paham PBO nih HEH !!  Ulun ni harat meolah program, han..  90  0.875  BUILD SUCCESSFUL (total time: 0 seconds)		

## 1.6 Latihan 1

Ikuti langkah-langkah pada gambar berikut, tanyakan bila ada yang tidak dimengerti atau terdapat error.

1.

```

12 public class praktek1 {
13
14     void himung() {
15         System.out.println("Asa Himung Mun Paham PBO nih HEH !!");
16     }
17
18     String s;
19     int i;
20     double d;
21
22     void cetakBerataan() {
23         System.out.println(s);
24         System.out.println(i);
25         System.out.println(d);
26     }
27 }

```

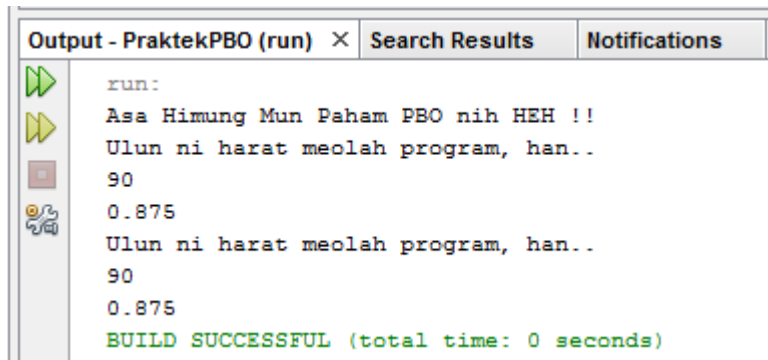
2.

```

12 public class PraktekPBO {
13
14     /**
15      * @param args the command line arguments
16      */
17
18     public static void main(String[] args) {
19         // TODO code application logic here
20         praktek1 objekP = new praktek1();
21         objekP.himung();
22
23         objekP.s="Ulun ni harat meolah program, han..";
24         objekP.i=90;
25         objekP.d=0.875;
26
27         System.out.println(objekP.s);
28         System.out.println(objekP.i);
29         System.out.println(objekP.d);
30
31         objekP.cetakBerataan();
32     }
33 }

```

3.



The screenshot shows an IDE's Output window with three tabs: "Output - PraktekPBO (run)", "Search Results", and "Notifications". The "Output" tab is active, displaying the following text:

```
run:  
Asa Himung Mun Paham PBO nih HEH !!  
Ulun ni harat meolah program, han..  
90  
0.875  
Ulun ni harat meolah program, han..  
90  
0.875  
BUILD SUCCESSFUL (total time: 0 seconds)
```

On the left side of the Output window, there is a vertical toolbar with icons for running (green play button), stepping through code (yellow play button), stopping (red square), and debugging (bug icon).

## PRAKTEK 2

### CLASS SCANNER

#### 2.1 Class Scanner

Scanner adalah class yang disediakan oleh Java untuk menginput nilai melalui keyboard, class ini terdapat pada class java.util sehingga perlu diimport terlebih dahulu sebelum digunakan, pun karena merupakan class maka menggunakan Scanner harus melalui instance terlebih dahulu.

1. Buat class dengan nama **mahasiswa2** (langkah membuat class bisa dilihat di 1.3) dan isikan perintah berikut

```

14     public class mahasiswa2 {
15         String NPM,nama,predikat;
16         int tahunLulus;
17         double IPK;
18         boolean penghargaan;
19
20     void cetakSemua() {
21         System.out.println("NPM           : "+NPM);
22         System.out.println("Nama           : "+nama);
23         System.out.println("Tahun Lulus   : "+tahunLulus);
24         System.out.println("IPK           : "+IPK);
25         System.out.println("Predikat      : "+predikat);
26         System.out.println("Penghargaan   : "+penghargaan);
27     }
28 }
29 
```

2. Buat class dengan nama **praktek2** dan isikan perintah berikut

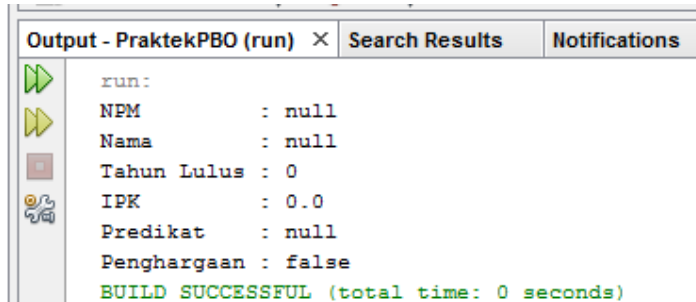
```

14     public class praktek2 {
15     public static void main(String[] args) {
16         mahasiswa2 mukidi = new mahasiswa2();
17         mukidi.cetakSemua();
18     }
19 }

```

3. Fokus.. jangan senyum-senyum liat Mukidi ☺

4. Jalankan dengan **Shift+F6** (karena class ini bukan main class)
5. Perhatikan hasilnya pada windows **Output**



Java memberikan nilai awal default untuk tipe data String berupa null, tipe data int berupa 0 tipe data double berupa 0.0 tipe data Boolean berupa false. Salah satu cara untuk mengisinya pada source code, dapat dibaca pada 1.5 sedangkan untuk kali ini akan kita isi dengan dituliskan pada saat program berjalan.

6. Ubah isi **praktek2.java** seperti pada gambar berikut

```

13  import java.util.Scanner;
14  public class praktek2 {
15      public static void main(String[] args) {
16          Scanner masUkin = new Scanner(System.in);
17
18          mahasiswa2 mukidi = new mahasiswa2();
19
20          System.out.print("Masukin NPM      :");
21          mukidi.NPM=masUkin.nextLine();
22
23          System.out.print("Masukin Nama      :");
24          mukidi.nama=masUkin.nextLine();
25
26          System.out.print("Masukin Tahun Lulus :");
27          mukidi.tahunLulus=masUkin.nextInt();
28
29          System.out.print("Masukin IPK      :");
30          mukidi.IPK=masUkin.nextDouble();
31          masUkin.nextLine();
32
33          System.out.print("Masukin Predikat   :");
34          mukidi.predikat=masUkin.nextLine();
35

```

```

36         System.out.print("Masukin Penghargaan :");
37         mukidi.penghargaan=masUkin.nextBoolean();
38
39         mukidi.cetakSemua();
40     }
41 }

```

7. Jalankan dengan **Shift+F6**
8. Perhatikan hasilnya pada windows **Output**, jangan lupa untuk meng-klik windows Output karena kita akan mengisi data Mukidi sesuai dengan source code yang sudah kita tuliskan. Kemudian hasilnya akan ditampilkan oleh method **cetakSemua()**

## 2.2 Validasi Input (IF)

Pada saat melakukan input dengan keyboard ada kalanya programmer menguji jalannya program dengan menginput secara sembrono, sehingga muncul pesan error bahwa nilai yang kita input tidak cocok untuk variable yang tersedia. Hayo.. ngaku aja deeeh. Pada kasus ini misalkan menginput **tahunLulus** dengan huruf.

```

Output - PraktekPBO (run) × Search Results Notifications
run:
Masukin NPM :a
Masukin Nama :b
Masukin Tahun Lulus :ini sengaja salah
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at praktekpbo.praktek2.main(praktek2.java:28)
C:\Users\mirza\AppData\Local\NetBeans\Cache\8.1\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 15 seconds)

```

Maka untuk menyembuhkan penyakit ini bisa dilakukan pengecekan terlebih dahulu sebelum terjadi input dengan perintah **hasNextInt()**. Ubah sedikit pada file **praktek2.java** pada bagian menyimpan **tahunLulus** menjadi seperti pada gambar

```

27         System.out.print("Masukin Tahun Lulus :");
28
29         if (masUkin.hasNextInt()) {
30             mukidi.tahunLulus=masUkin.nextInt();
31         }
32         else{
33             masUkin.next();
34             System.out.println("SALAH INPUT");
35         }

```

### 2.3 Validasi Input (Exception)

Exception adalah salah satu cara untuk menangkap error sebelum terjadi dan mencegah aplikasi berhenti berjalan kemudian memberikan pesan error yang mudah dipahami oleh user. Untuk kasus mukidi lakukan langkah berikut:

1. Tambahkan **import java.util.InputMismatchException;** sebelum **class praktek2**
2. Tambahkan perintah berikut

```
try{
    mukidi.tahunLulus=masUkin.nextInt();
}
catch(InputMismatchException e){
    System.out.println("Input Tidak Sesuai");
    masUkin.next();
}
```

### 2.4 Latihan 2

1. Pada 2.1 nomor 6, terdapat gambar, coba periksa baris 31, apa ya fungsinya?
2. Lanjutkan Validasi Input menggunakan IF dan Exception untuk tipe data double (IPK) dan boolean (penghargaan)



## PRAKTEK 3

### KONDISI

#### 3.1 IF

Pernyataan `if-then` dan `if-then-else` merupakan pernyataan paling mendasar dalam pengendalian program untuk melaksanakan suatu instruksi tertentu hanya ketika suatu kondisi terpenuhi. Misalkan kita buat aturan **jika SPP lunas maka terbitkan kartu ujian**. Dalam hal ini untuk bisa menerbitkan kartu ujian maka kondisi yang harus dipenuhi adalah SPP lunas.

Sering kali yang menjadi sulit adalah menuangkannya kedalam bahasa pemrograman. Kondisi SPP lunas bisa kita buat menjadi suatu variable misalkan `SPPLunas`, untuk tipe datanya kita buat berdasarkan kemungkinan isi dari variable `SPPLunas`. Karena `SPPLunas` kemungkinan isinya adalah 'lunas' dan 'belum lunas' maka bisa saja kita berikan tipe data `String`:

```
String SPPLunas;
```

Akan tetapi karena kemungkinan isinya hanya dua maka akan lebih efektif jika kita gunakan tipe data **boolean** yang berisikan **true** dan **false**

```
boolean SPPLunas;
```

selanjutnya tinggal menuliskan aturan diatas disesuaikan dengan bahasa java:

**jika SPP lunas maka terbitkan kartu ujian**

```
if(SPPLunas==true){
    System.out.println("Cetak Kartu Ujian");
}
```

Kemudian lengkapi beserta class main dan inialisasi nilai awal dari variable `SPPLunas`

```
12 public class praktek3 {  
13     public static void main(String[] args) {  
14         boolean SPPLunas;  
15  
16         SPPLunas=true;  
17  
18         if(SPPLunas==true) {  
19             System.out.println("Cetak Kartu Ujian");  
20         }  
21     }  
22 }
```

