

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL IV
INHERITANCE DAN POLYMORPHISM**



**Disusun Oleh:
Nisrina Nurhaliza
21102218
S1 IF-09-0**

**Dosen Pengampu :
Dedy Agung Prabowo, S.Kom., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2023**

BAB I

TUJUAN PRAKTIKUM

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat :

1. Mahasiswa diharapkan mampu mengerti dan memahami konsep Inheritance (Pewarisan)
2. Mahasiswa diharapkan mampu menerapkan Inheritance dalam Pemrograman Java

BAB II

TOOLS

1. IntelliJ Community Edition
2. Java SE Development Kit 19

BAB III

DASAR TEORI

a. Abstraction

Abstraction adalah menggunakan hal hal sederhana untuk mewakili hal hal rumit atau kompleks. **Abstraction** adalah menemukan hal hal yang penting pada suatu objek dan mengabaikan hal hal yang sifatnya incidental. Dapat dikatakan juga Proses untuk menyembunyikan detail implementasi dan hanya sisi fungsionalitas (gambaran umum) saja yang ditampilkan. Abstraction di java terbagi menjadi dua yaitu class abstract dan interface.

b. Enkapsulasi

Enkapsulasi (Encapsulation) merupakan cara untuk melindungi **property** (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program. Cara untuk melindungi data yaitu dengan menggunakan access modifiers (hak akses). Ada 4 hak akses yang tersedia yaitu default, public, protected dan private.

No	Modifier	Pada Class dan Interface	Pada Method dan Variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method method yang sepaket
4	Privat	Tidak bias diterapkan	Tidak dapat diakses dimanapun kecuali oleh method method yang ada dalam kelas itu sendiri

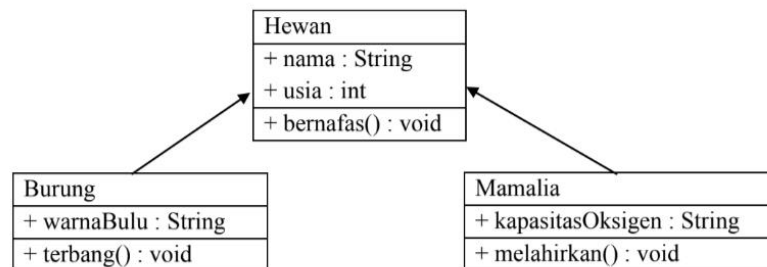
Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

c. Inheritance

Konsep yang mendasari **Inheritance** adalah Generalization. Generalization digambarkan sebagai hubungan dari subclass ke superclass, sedangkan Inheritance adalah proses pewarisan data dan behaviour (method) dari superclass ke subclass. Contohnya, hewan

adalah generalization dari singa, tumbuhan adalah generalization dari rumput. Sebaliknya, singa adalah inherits dari hewan dan rumput inherits dari tumbuhan.

Pewarisan (Inheritance) merupakan proses pembentukan kelas baru dari kelas yang sudah ada (reusability). Kelas yang mewariskan disebut Kelas Super (Kelas Induk), sedangkan kelas yang diwariskan disebut Sub Kelas (Kelas Anak). Pewarisan ini bersifat menyeluruh, sehingga semua data dan method yang dimiliki oleh kelas asalnya akan diturunkan kepada kelas baru. Konsep dari inheritance sendiri yaitu untuk membuat struktur class pada pemrograman yang dimana struktur tersebut terdapat sebuah Parentclass atau Superclass sebagai induk kelas dan Subclass sebagai anak kelas.



d. Polymorphism

Polimorfisme dapat berarti “mempunyai banyak bentuk” sehingga dapat disimpulkan, **Polimorfisme** adalah kemampuan untuk meminta objek yang berbeda untuk melaksanakan tugas yang sama dan membuat objek tahu bagaimana untuk mencapainya dengan caranya sendiri. **Polimorfisme** menunjukkan kemampuan untuk

menangani dua atau lebih bentuk obyek yang berlainan saat eksekusi berlangsung.

Polimorfisme dapat diilustrasikan sebagai berikut, perhatikanlah penggunaan kata “mentah” dalam beberapa kalimat. “Sayuran itu masih mentah, belum dimasak”, “Pukulan petinju itu berhasil dimentahkan oleh lawannya”, “Gagasan ini masih mentah sehingga perlu di bahas kembali”. Kata “mentah” pada contoh di atas dapat diaplikasikan pada berbagai objek dan dapat di-interpretasikan ke dalam beberapa makna. **Polymorphism** adalah konsep pemrograman yang berorientasi pada objek yang mengacu pada kemampuan variabel, fungsi atau objek untuk mengambil beberapa bentuk. Polymorphism adalah penggunaan salah satu item seperti fungsi, atribut, atau interface pada berbagai jenis objek yang berbeda dalam bahasa pemrograman. Dalam bahasa pemrograman yang menunjukkan polimorfisme, objek kelas memiliki hierarki yang sama yang diwariskan dari kelas induk yang sama, mungkin memiliki fungsi dengan nama yang sama, tetapi dengan perilaku berbeda.

Sumber :

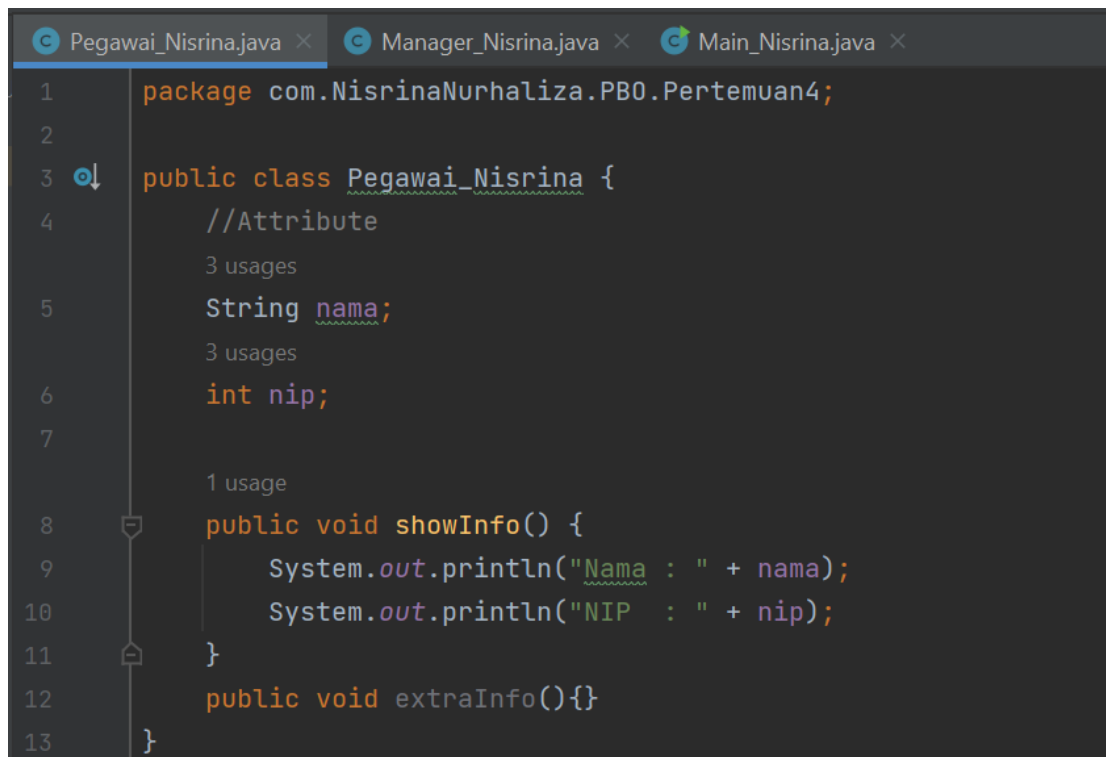
1. <http://imamfarisi.com/abstraction-di-java/>
Diakses pada tanggal 13 April 2023.
2. www.gamelab.id/news/1171-java--menggunakan-konsep-inheritance-agar-kode-pemrograman-menjadi-lebih-terstruktur#:~:text=Konsep%20dari%20inheritance%20sendiri%20yaitu,dan%20Subclass%20sebagai%20anak%20kelas .
Diakses pada tanggal 13 April 2023.
3. <https://www.ekrut.com/media/polymorphism-adalah>
Diakses pada tanggal 13 April 2023.

BAB IV

GUIDED

1. Uji Coba Inheritance

- Sebuah class baru dengan nama **Pegawai**
 - Source Code

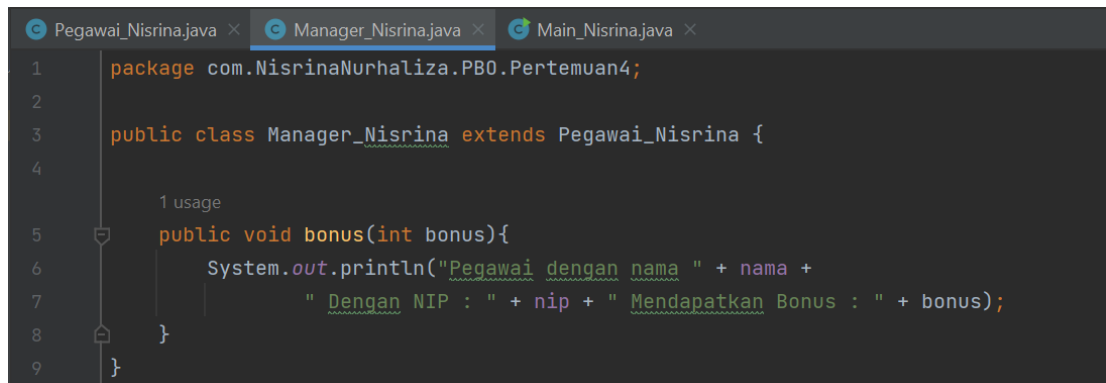


```
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class Pegawai_Nisrina {
4     //Attribute
5     String nama;
6     int nip;
7
8     public void showInfo() {
9         System.out.println("Nama : " + nama);
10        System.out.println("NIP : " + nip);
11    }
12    public void extraInfo(){}
13 }
```

The screenshot shows an IDE with three tabs: Pegawai_Nisrina.java, Manager_Nisrina.java, and Main_Nisrina.java. The Pegawai_Nisrina.java tab is active, displaying the source code for the Pegawai_Nisrina class. The code is as follows:

- Sebuah class baru dengan nama **Manager**

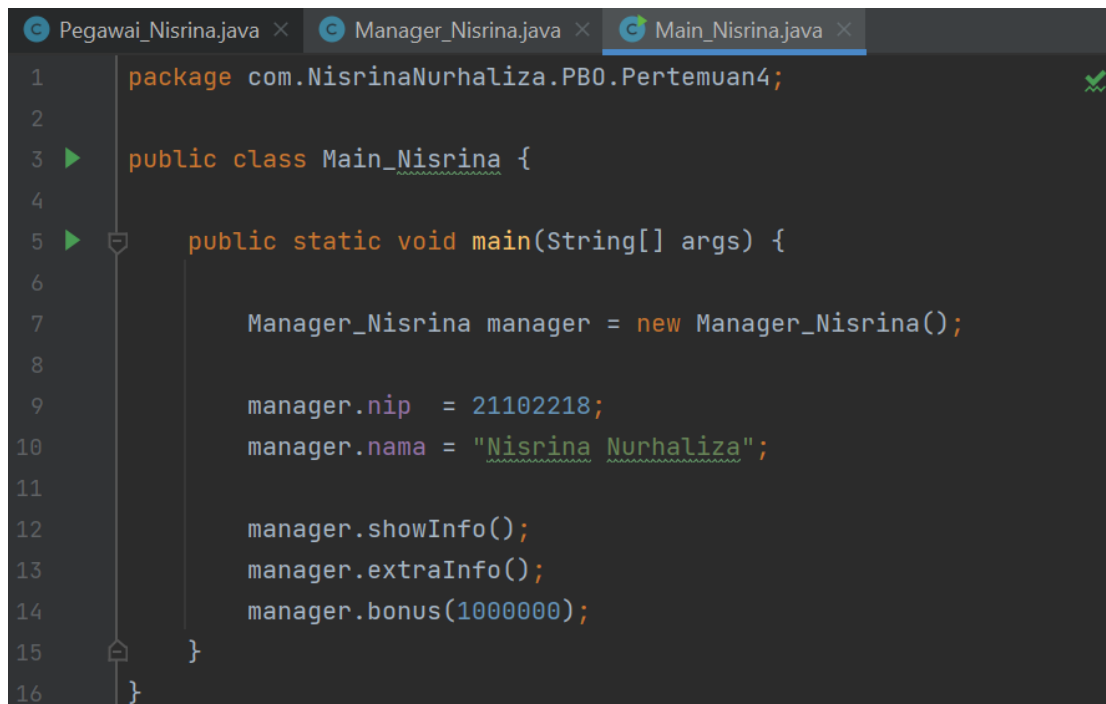
- Source Code



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class Manager_Nisrina extends Pegawai_Nisrina {
4
5     1 usage
6     public void bonus(int bonus){
7         System.out.println("Pegawai dengan nama " + nama +
8             " Dengan NIP : " + nip + " Mendapatkan Bonus : " + bonus);
9     }
10 }
```

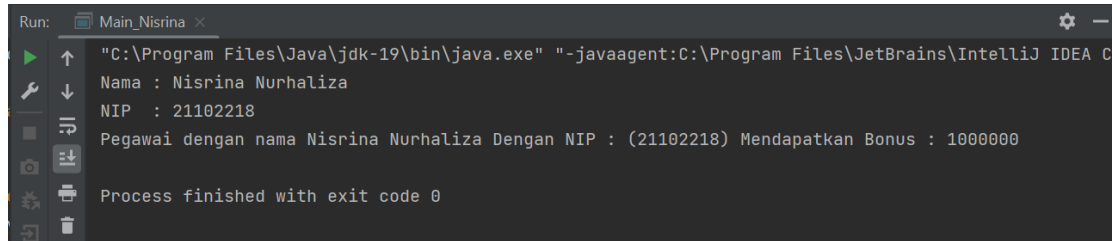
- Agar program dapat dijalankan, sekarang buatlah java class baru dengan nama **Main**

- Source Code



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class Main_Nisrina {
4
5     public static void main(String[] args) {
6
7         Manager_Nisrina manager = new Manager_Nisrina();
8
9         manager.nip = 21102218;
10        manager.nama = "Nisrina Nurhaliza";
11
12        manager.showInfo();
13        manager.extraInfo();
14        manager.bonus(1000000);
15    }
16 }
```

- Output Program

A screenshot of a Java IDE's console window. The window title is "Run: Main_Nisrina x". The console output shows the execution of a Java program. The first line is the command: "C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA C". The subsequent lines are the program's output: "Nama : Nisrina Nurhaliza", "NIP : 21102218", and "Pegawai dengan nama Nisrina Nurhaliza Dengan NIP : (21102218) Mendapatkan Bonus : 1000000". The final line indicates "Process finished with exit code 0".

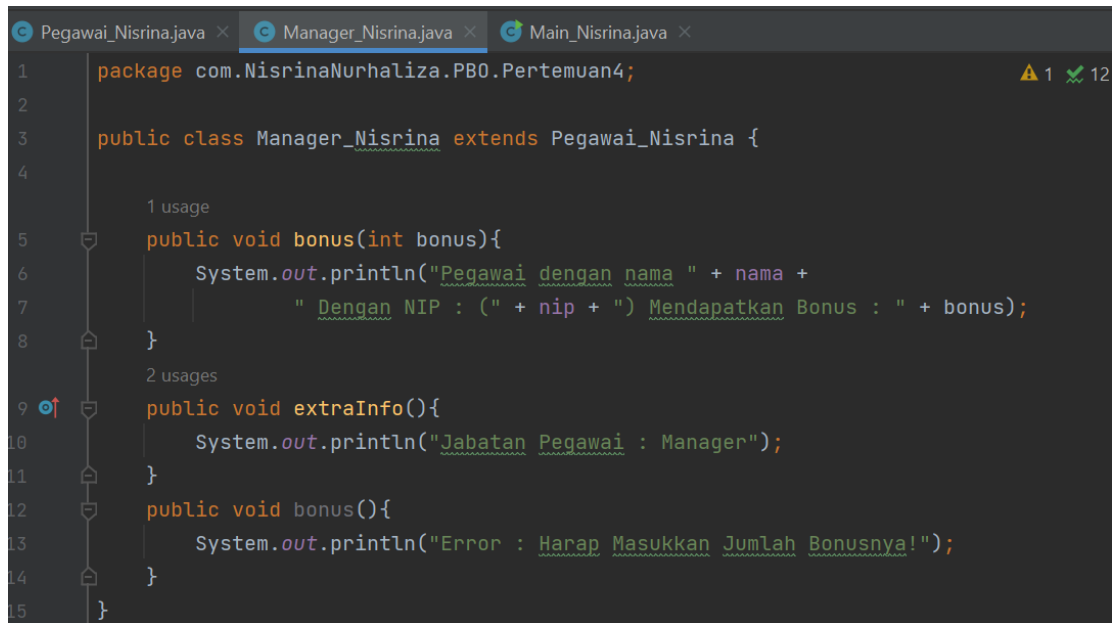
```
Run: Main_Nisrina x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA C
Nama : Nisrina Nurhaliza
NIP : 21102218
Pegawai dengan nama Nisrina Nurhaliza Dengan NIP : (21102218) Mendapatkan Bonus : 1000000
Process finished with exit code 0
```

- Deskripsi Program

Program di atas adalah sebuah program Java yang menggunakan kelas **Main** untuk menjalankan program. Program ini membuat satu objek, yaitu objek **Manajer**. Objek **Manajer** adalah instance dari kelas **Manajer** dan **Pegawai** masing-masing. Setelah membuat objek, program ini mengisi nilai variabel instance "nama" dan "nip" untuk masing-masing objek. Objek **Manajer** diberi nilai NIP 21102218 dan nama "Nisrina Nurhaliza. Program ini kemudian memanggil beberapa metode dari objek **Manajer**. Metode "showInfo()" dipanggil untuk menampilkan informasi pegawai ke layar console. Metode "extrainfo()" dipanggil untuk menampilkan informasi tambahan mengenai jabatan pegawai (hanya untuk objek "Manajer"). Metode "bonus(int bonus)" dipanggil untuk memberikan bonus pada objek "Manajer". Metode "bonus()" juga dipanggil untuk memberikan instruksi kepada pengguna agar memasukkan jumlah bonus saat akan memanggil metode "bonus(int bonus)".

2. Uji Coba Polymorphism

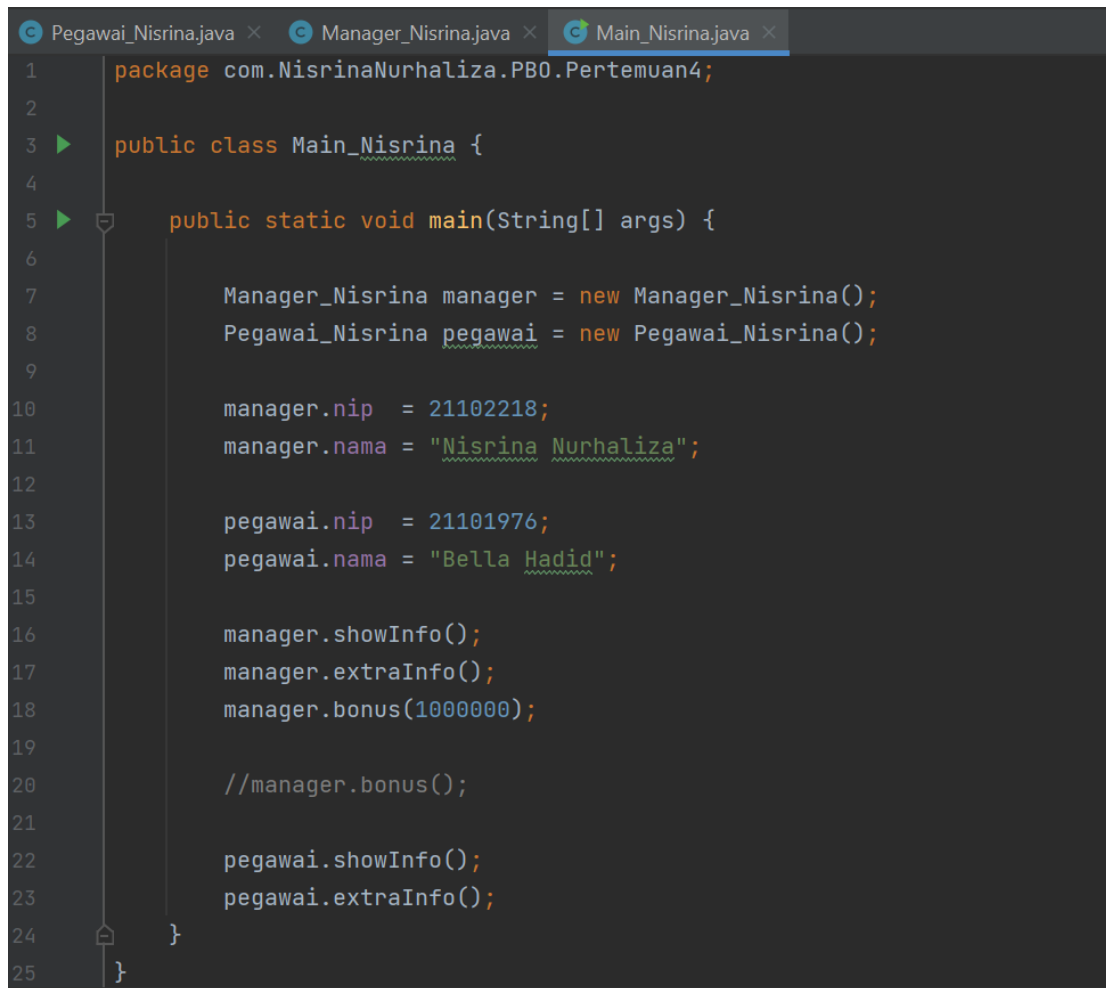
- Modifikasi class **Manager** dengan tambahan script
 - Source Code



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class Manager_Nisrina extends Pegawai_Nisrina {
4
5     1 usage
6     public void bonus(int bonus){
7         System.out.println("Pegawai dengan nama " + nama +
8             " Dengan NIP : (" + nip + ") Mendapatkan Bonus : " + bonus);
9     }
10    2 usages
11    public void extraInfo(){
12        System.out.println("Jabatan Pegawai : Manager");
13    }
14    public void bonus(){
15        System.out.println("Error : Harap Masukkan Jumlah Bonusnya!");
16    }
17 }
```

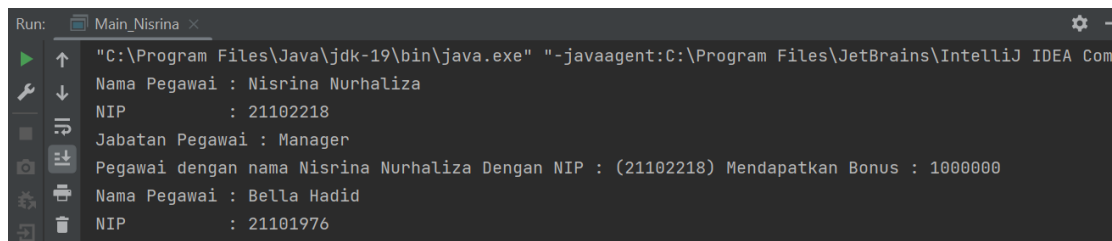
- Modifikasi class **Main** dengan tambahan script

- Source Code



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class Main_Nisrina {
4
5     public static void main(String[] args) {
6
7         Manager_Nisrina manager = new Manager_Nisrina();
8         Pegawai_Nisrina pegawai = new Pegawai_Nisrina();
9
10        manager.nip = 21102218;
11        manager.nama = "Nisrina Nurhaliza";
12
13        pegawai.nip = 21101976;
14        pegawai.nama = "Bella Hadid";
15
16        manager.showInfo();
17        manager.extraInfo();
18        manager.bonus(1000000);
19
20        //manager.bonus();
21
22        pegawai.showInfo();
23        pegawai.extraInfo();
24    }
25 }
```

- Output Program



```
Run: Main_Nisrina
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Com
Nama Pegawai : Nisrina Nurhaliza
NIP          : 21102218
Jabatan Pegawai : Manager
Pegawai dengan nama Nisrina Nurhaliza Dengan NIP : (21102218) Mendapatkan Bonus : 1000000
Nama Pegawai : Bella Hadid
NIP          : 21101976
```

- Deskripsi Program

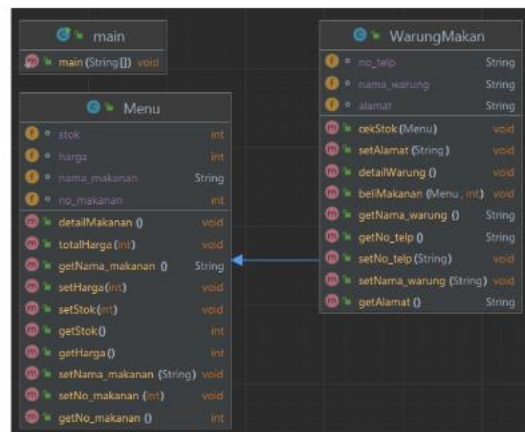
Program di atas adalah sebuah program Java yang menggunakan kelas **Main** untuk menjalankan program. Program ini membuat dua objek, yaitu objek **Manajer** dan objek **Pegawai**. Objek **Manajer** dan **Pegawai** adalah instance dari kelas **Manajer** dan **Pegawai** masing-masing. Setelah membuat objek, program ini mengisi nilai variabel instance "nama" dan "nip" untuk masing-masing objek. Objek **Manajer** diberi nilai NIP 21102218 dan nama "Nisrina Nurhaliza", sedangkan objek **Pegawai** diberi nilai NIP 21101976 dan nama "Bella Hadid". Program ini kemudian memanggil beberapa metode dari objek **Manajer** dan **Pegawai**. Metode "showInfo()" dipanggil untuk menampilkan informasi pegawai ke layar console. Metode "extrainfo()" dipanggil untuk menampilkan informasi tambahan mengenai jabatan pegawai (hanya untuk objek "Manajer"). Metode "bonus(int bonus)" dipanggil untuk memberikan bonus pada objek "Manajer". Metode "bonus()" juga dipanggil untuk memberikan instruksi kepada pengguna agar memasukkan jumlah bonus saat akan memanggil metode "bonus(int bonus)". Terakhir, metode "extraInfo()" dipanggil pada objek **Pegawai**.

UNGUIDED

1. Buatlah sebuah program warung makan sederhana dengan ketentuan sebagai berikut dengan item yang jual :

No Makanan	Nama Makanan	Harga	Stock
1	Nasi Goreng	10.000	10
2	Nasi Goreng Special	15.000	4
3	Nasi Goreng Special + Telur	20.000	20

Desain diagram :



Dengan ketentuan sebagai berikut pada class WarungMakan.java :

- Terdapat method getter dan setter
- **Method detailWarung()** menampilkan detail seperti alamat, nama warung dan no telp
- **Method beliMakanan()** menampilkan detail pembelian makanan dan melakukan pengecekan stock pada menu jika habis maka akan menampilkan stock tidak cukup dan jika stock tersedia, stock dikurangi jumlah makanan yang dibeli

- **Method cekStock()** menampilkan jumlah stock terkini setelah dilakukan pembelian dari menu yang dipilih
- Class **WarungMakan** extends class menu

Dengan ketentuan sebagai berikut Class Makanan.java:

- Terdapat method getter dan setter
- **Method detailMakanan()** menampilkan detail seperti no makanan, nama makanan, harga dan stock
- **Method totalHarga()** menampilkan total harga dari hasil jumlah pembelian makanan.

Buatlah Class Main.java di dalamnya dapat mengisi semua isian di Class WarungMakan dan dapat mengisi Menu makanan seperti detail tabel item diatas, kemudian tampilkan detail warung dan makanan yang telah diisi. Lakukan operasi pembelian dengan method yang telah dibuat sebelumnya

Source Code :

- **WarungMakan.java**

```
WarungMakan_Nisrina.java x Menu_Nisrina.java x MainMakanan_Nisrina.java x
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class WarungMakan_Nisrina {
4     String no_telp;
5     String nama_warung;
6     String alamat;
7
8     public void detailWarung(){
9         System.out.println("===== WARUNG MAKAN SEDERHANA ALA NISRINA =====");
10        System.out.println("Nama Warung Makan : " + nama_warung);
11        System.out.println("Alamat : " + alamat);
12        System.out.println("No Telp : " + no_telp);
13        System.out.println();
14    }
15
16    //Membuat Fungsi Cetak
17    public void cekStock(Menu_Nisrina m) {
18        if (m.no_makanan == 1) {
19            System.out.println("Stock Nasi Goreng : " + m.stock);
20        } else if (m.no_makanan == 2) {
21            System.out.println("Stock Nasi Goreng Spesial : " + m.stock);
22        } else if (m.no_makanan == 3) {
23            System.out.println("Stock Nasi Goreng Spesial + Telur : " + m.stock);
24        }
25        System.out.println();
26    }
27
28 }
```

```

3 usages
29 @ public void beliMakanan(Menu_Nisrina m, int jumlah) {
30     if (jumlah >= m.stock){
31         System.out.println("Mohon Maaf Stock Tidak Cukup");
32         m.harga = 0;
33     }
34     } else {
35         System.out.println();
36         System.out.println("==== Detail Pembelian Makanan =====");
37         System.out.println();
38         System.out.println("Nama Makanan : " + m.nama_makanan);
39         System.out.println("Harga Makanan : " + m.harga);
40         System.out.println("Jumlah : " + jumlah);
41         m.harga = m.harga * jumlah;
42         m.stock -= jumlah;
43     }
44 }

```

```

46 //Getter and Setter
47 public String getNo_telp() {
48
49     return no_telp;
50 }
51
52 public void setNo_telp(String no_telp) {
53
54     this.no_telp = no_telp;
55 }
56
57 public String getNama_warung() {
58
59     return nama_warung;
60 }

```

```
61
62     public void setName_warung(String nama_warung) {
63
64         this.nama_warung = nama_warung;
65     }
66
67     public String getAlamat() {
68
69         return alamat;
70     }
71
72     public void setAlamat(String alamat) {
73
74         this.alamat = alamat;
75     }
76 }
```

➤ Public void beliMakanan(menu m, int jumlah)

- **Menu.java**

```
WarungMakan_Nisrina.java × Menu_Nisrina.java × MainMakanan_Nisrina.java ×
1 package com.NisrinaNurhaliza.PB0.Pertemuan4;
2
3 public class Menu_Nisrina extends WarungMakan_Nisrina {
4     int stock;
5     int harga;
6     String nama_makanan;
7     int no_makanan;
8
9     //Fungsi untuk Menampilkan Menu
10    public void detailMakanan() {
11        System.out.println("No Makanan : " + no_makanan);
12        System.out.println("Nama Makanan : " + nama_makanan);
13        System.out.println("Harga : " + harga);
14        System.out.println();
15    }
```

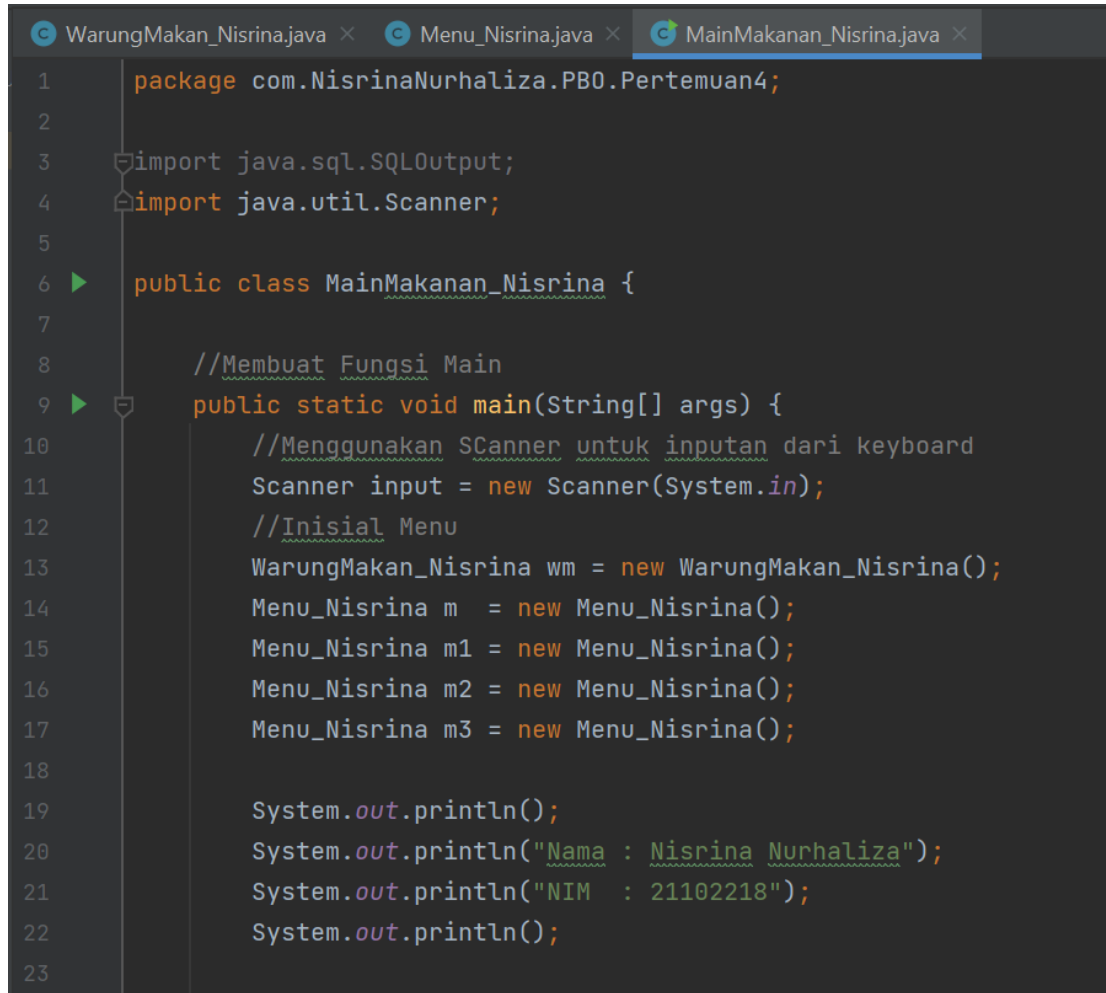
```
WarungMakan_Nisrina.java × Menu_Nisrina.java × MainMakanan_Nisrina.java ×
17 //Membuat Fungsi TotalHarga
18    public void totalHarga() {
19        System.out.println("Total Harga : " + harga);
20        System.out.println();
21    }
```

```
WarungMakan_Nisrina.java × Menu_Nisrina.java × MainMakanan_Nisrina.java ×
25     public int getStock() {
26
27         return stock;
28     }
29
30     public void setStock(int stock) {
31
32         this.stock = stock;
33     }
34
35     public int getHarga() {
36
37         return harga;
38     }
39
40     public void setHarga(int harga) {
41
42         this.harga = harga;
43     }
44
45     public String getNama_makanan() {
46
47         return nama_makanan;
48     }
```

```
48
49     public void setNama_makanan(String nama_makanan) {
50
51         this.nama_makanan = nama_makanan;
52     }
53
54     public int getNo_makanan() {
55
56         return no_makanan;
57     }
58
59     public void setNo_makanan(int no_makanan) {
60
61         this.no_makanan = no_makanan;
62     }
63 }
```

- Public void totalHarga berfungsi membuat total harga
- Public void detailMakanan berfungsi untuk melihat menu

- **Main.java**



```
1 package com.NisrinaNurhaliza.PBO.Pertemuan4;
2
3 import java.sql.SQLOutput;
4 import java.util.Scanner;
5
6 public class MainMakanan_Nisrina {
7
8     //Membuat Fungsi Main
9     public static void main(String[] args) {
10         //Menggunakan Scanner untuk inputan dari keyboard
11         Scanner input = new Scanner(System.in);
12         //Inisial Menu
13         WarungMakanan_Nisrina wm = new WarungMakanan_Nisrina();
14         Menu_Nisrina m = new Menu_Nisrina();
15         Menu_Nisrina m1 = new Menu_Nisrina();
16         Menu_Nisrina m2 = new Menu_Nisrina();
17         Menu_Nisrina m3 = new Menu_Nisrina();
18
19         System.out.println();
20         System.out.println("Nama : Nisrina Nurhaliza");
21         System.out.println("NIM : 21102218");
22         System.out.println();
23     }
```

```
23
24     wm.nama_warung = "Warung Bottega Nisrina";
25     wm.alamat      = "Ashta District 8 Teluk";
26     wm.no_telp     = "087724411250";
27
28     m1.no_makanan  = 1;
29     m1.nama_makanan = "Nasi Goreng";
30     m1.harga       = 10000;
31     m1.stock       = 10;
32
33     m2.no_makanan  = 2;
34     m2.nama_makanan = "Nasi Goreng Spesial";
35     m2.harga       = 15000;
36     m2.stock       = 4;
37
38     m3.no_makanan  = 3;
39     m3.nama_makanan = "Nasi Goreng Spesial + Telur";
40     m3.harga       = 20000;
41     m3.stock       = 20;
42
```

```
WarungMakan_Nisrina.java × Menu_Nisrina.java × MainMakanan_Nisrina.java ×
43     wm.detailWarung();
44     m1.detailMakanan();
45     m2.detailMakanan();
46     m3.detailMakanan();
47
```



```

WarungMakan_Nisrina.java × Menu_Nisrina.java × MainMakanan_Nisrina.java ×
48      System.out.print("Masukkan Nomor Makanan      : ");
49      m.no_makanan = input.nextInt();
50      //Menggunakan Percabangan
51      //Untuk Memilih No 1
52      if(m.no_makanan == 1){
53
54          System.out.print("Masukkan Jumlah Pesanan      : ");
55          wm.beliMakanan(m1, input.nextInt());
56          m1.totalHarga();
57          wm.cekStock(m1);
58
59          //Untuk Memilih No 2
60      } else if (m.no_makanan == 2){
61          System.out.print("Masukkan Jumlah Pesanan      : ");
62          wm.beliMakanan(m2, input.nextInt());
63          m2.totalHarga();
64          wm.cekStock(m2);
65
66          //Untuk Memilih No 3
67      } else if (m.no_makanan == 3) {
68          System.out.print("Masukkan Jumlah Pesanan      : ");
69          wm.beliMakanan(m3, input.nextInt());
70          m3.totalHarga();
71          wm.cekStock(m3);
72      }
73
74          //Berhenti Memilih Menu
75          input.close();
76      }

```

- Public static void main (String[] args) berfungsi sebagai fungsi main
- Script Main menggunakan percabangan
- WarungMakan_Nisrina wm = new WarungMakan_Nisrina(); berfungsi membuat inisial menu

Output :

➤ Jika Sesuai Stock Dan Stock Masih Tersedia

```
Run: MainMakanan_Nisrina x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Fi

Nama : Nisrina Nurhaliza
NIM : 21102218

===== WARUNG MAKAN SEDERHANA ALA NISRINA =====
Nama Warung Makan : Warung Bottega Nisrina
Alamat : Ashta District 8 Teluk
No Telpon : 087724411250

No Makanan : 1
Nama Makanan : Nasi Goreng
Harga : 10000

No Makanan : 2
Nama Makanan : Nasi Goreng Spesial
Harga : 15000

No Makanan : 3
Nama Makanan : Nasi Goreng Spesial + Telur
Harga : 20000

Masukkan Nomor Makanan : 1
Masukkan Jumlah Pesanan : 2
```

```
===== Detail Pembelian Makanan =====
```

```
Nama Makanan   : Nasi Goreng
```

```
Harga Makanan  : 10000
```

```
Jumlah         : 2
```

```
Total Harga    : 20000
```

```
Stock Nasi Goreng      : 8
```

User memasukkan jumlah makanan dan terlihat detail pembelian dan sisa stock setelah pembelian

➤ **Jika Stock Melebihi Batas**

```
Run: MainMakanan_Nisrina x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrain

Nama : Nisrina Nurhaliza
NIM : 21102218

===== WARUNG MAKAN SEDERHANA ALA NISRINA =====
Nama Warung Makan : Warung Bottega Nisrina
Alamat : Ashta District 8 Teluk
No Telpon : 087724411250

No Makanan : 1
Nama Makanan : Nasi Goreng
Harga : 10000

No Makanan : 2
Nama Makanan : Nasi Goreng Spesial
Harga : 15000

No Makanan : 3
Nama Makanan : Nasi Goreng Spesial + Telur
Harga : 20000

Masukkan Nomor Makanan : 2
Masukkan Jumlah Pesanan : 30
```

```
No Makanan      : 3
Nama Makanan    : Nasi Goreng Spesial + Telur
Harga           : 20000

Masukkan Nomor Makanan      : 2
Masukkan Jumlah Pesanan    : 30
Mohon Maaf Stock Tidak Cukup
Total Harga      : 0

Stock Nasi Goreng Spesial :4
```

Jika user membeli makanan dengan jumlah melebihi stock maka akan menampilkan stock tidak cukup.

Deskripsi Program

Program di atas merupakan program yang menyediakan informasi mengenai warung makan dan daftar menu makanan yang disediakan oleh warung tersebut. Program ini memiliki tiga kelas yaitu kelas WarungMakan, Menu, dan Main. Kelas WarungMakan memiliki atribut no_telp, nama_warung, dan alamat. Kelas ini juga memiliki tiga metode yaitu detailWarung(), cekStok(), dan beliMakanan(). Kelas Menu merupakan turunan dari kelas WarungMakan dan memiliki empat atribut yaitu stok, harga, nama_makanan, dan no_makanan. Kelas ini juga memiliki empat metode yaitu detailMakanan(), totalHarga(), getStok(), dan getHarga(). Kelas MainWM merupakan kelas yang berisi method main() dan menjadi penghubung dari

kedua kelas di atas. Pada kelas ini, program akan menampilkan informasi mengenai warung dan menu makanan yang tersedia. Setelah itu, program akan meminta input nomor makanan dari pengguna dan menampilkan detail pembelian serta stok makanan yang tersisa. Secara keseluruhan, program ini digunakan untuk menampilkan informasi mengenai warung makan dan menu makanan yang disediakan oleh warung tersebut serta untuk melakukan pembelian makanan.

BAB V

KESIMPULAN

Pada praktikum pertemuan kedua, diharapkan mahasiswa mampu memahami mengenai Inheritance dan Polymorphism yang memiliki fungsi dan kegunaan masing masing dalam penyelesaian sebuah masalah dalam suatu pemrograman. Inheritance dan Polymorphism merupakan dasar dalam pemrograman berorientasi objek dengan bentuk pewarisan dengan kemampuan untuk meminta objek yang berbeda untuk melaksanakan tugas yang sama dan membuat objek tahu bagaimana untuk mencapainya dengan caranya sendiri..

Dalam praktikum ini, belajar tentang konsep dasar dan implementasi dari konsep-konsep tersebut. Abstraksi adalah proses menyembunyikan rincian implementasi dari penggunaan objek sehingga hanya aspek yang relevan ditampilkan. Hal ini dilakukan dengan membuat kelas abstrak dan mengimplementasikan metode yang dapat diwarisi oleh kelas turunan. Enkapsulasi adalah proses menyembunyikan data dan penggunaan objek dan hanya memberikan akses melalui metode yang ditentukan. Hal dilakukan dengan membatasi akses ke variabel dan metode kelas, sehingga penggunaanya dapat mengakses apa yang diperbolehkan. Inheritance adalah proses membuat kelas baru yang mewarisi sifat-sifat dari kelas yang ada. Hal ini dilakukan dengan membuat kelas turunan yang mewarisi metode dan variabel kelas induknya. Polimorfisme adalah kemampuan objek untuk menampilkan perilaku yang berbeda tergantung pada konteks penggunaannya. Hal ini dilakukan dengan menggunakan metode yang sama namun menghasilkan keluaran yang berbeda tergantung pada objek yang digunakan.