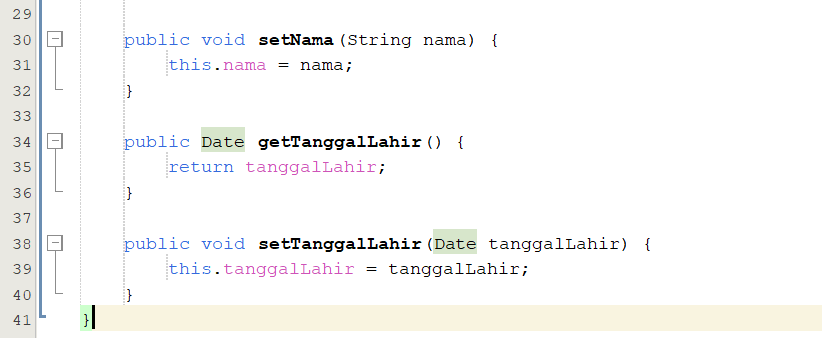
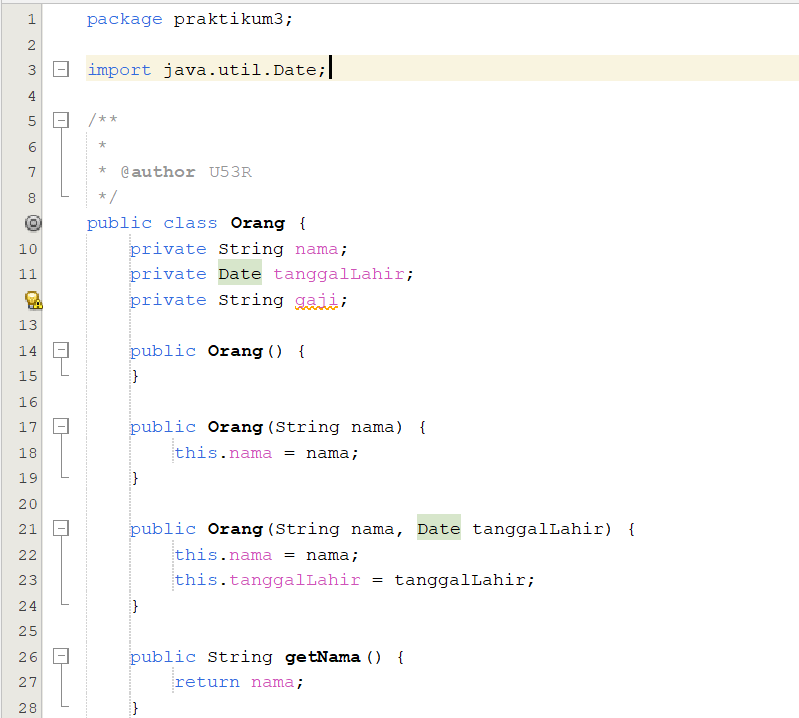
|  |  |
| --- | --- |
| Nama | : La Ode Muhammad Gazali |
| NIM | : 222212696 |
| Kelas | : 2KS2 |

**MODUL 3 PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK**

1. **Inheritance**

* Orang.java

****

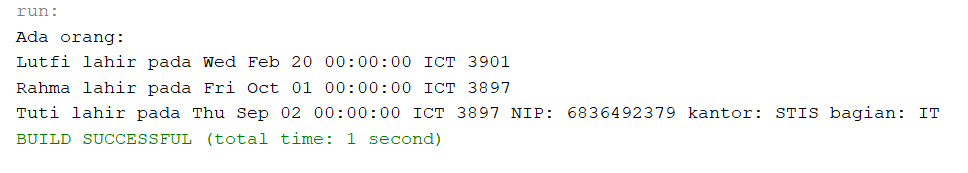
* Pegawai.java

****

* Kantor.java (Kelas Main)

****

* Hasil Running

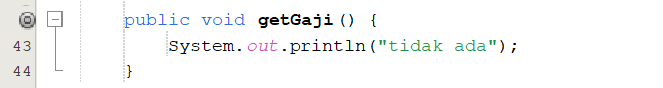
****

Program tersebut merupakan program heritance, dimana kelas orang memiliki turunan kelas pegawai. Pada main program, dibuat 3 buah objek, anatara lain 2 buah objek orang dan 1 objek pegawai. Hasil running program dapat dilihat pada compiler diatas.

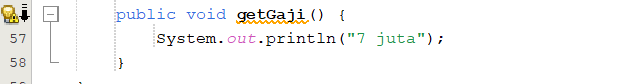
1. **Polymorphism**

Polymorphism memungkinkan suatu objek memiliki banyak bentuk atau perilaku. Pada program sebelumnya, akan ditambahkan variabel gaji pada kelas Orang dan kelas Pegawai, sebagai berikut:

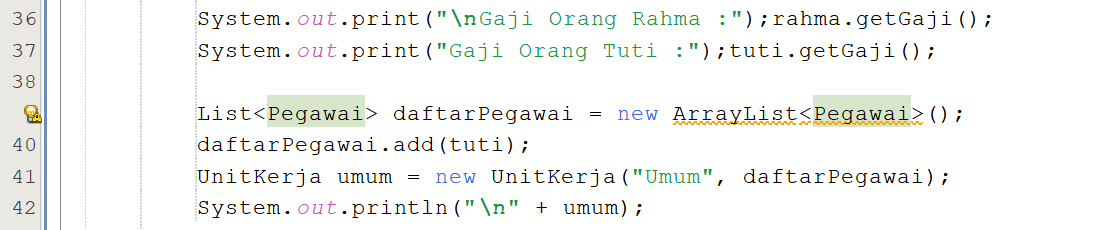
* Menambahkan method getGaji() pada kelas Orang, namun outputnya “tidak ada”



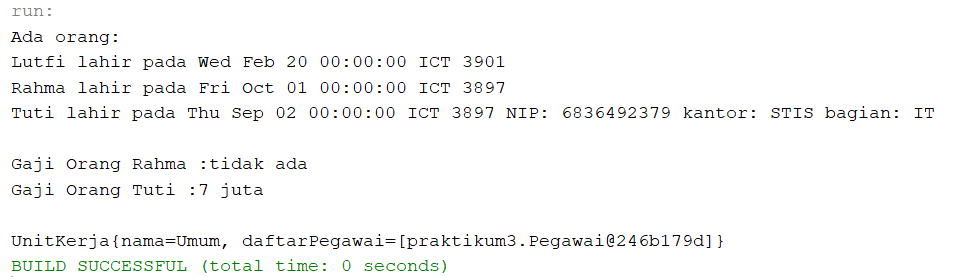
* Menambahkan method getGaji() pada kelas Pegawai dengan gaji sebesar 7 juta



* Memanggil method gaji dari objek Orang dan Pegawai pada kelas Kantor



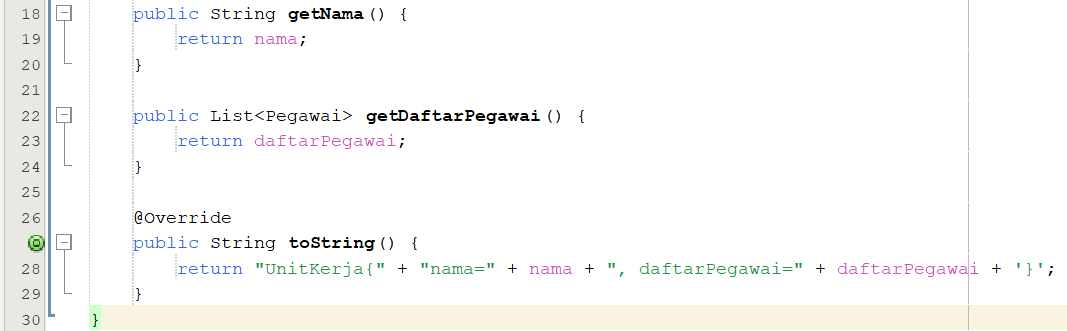
* Hasil running

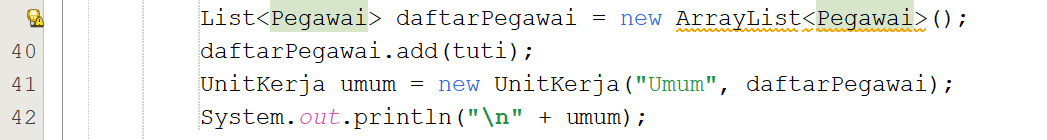


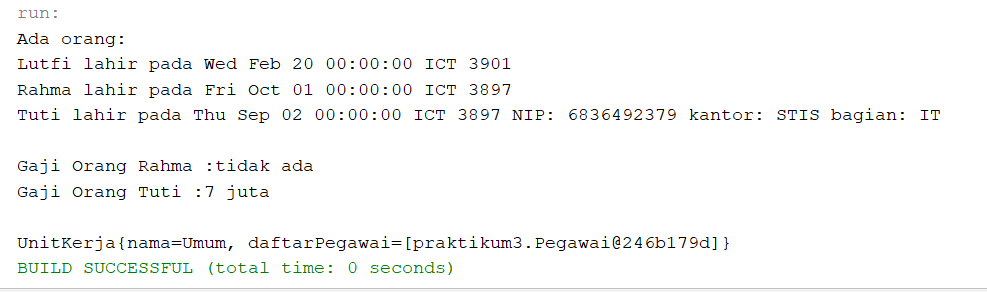
1. **Aggregation**

Dalam kasus program diatas, misalkan di suatu Unit Kerja yang mempunyai nama di kantor, ada beberapa pegawai yang bekerja. Kita bisa membuat kelas Unit Kerja dan menambahkan Pegawai sebagai bagian dari Unit Kerja tersebut.

* UnitKerja.java



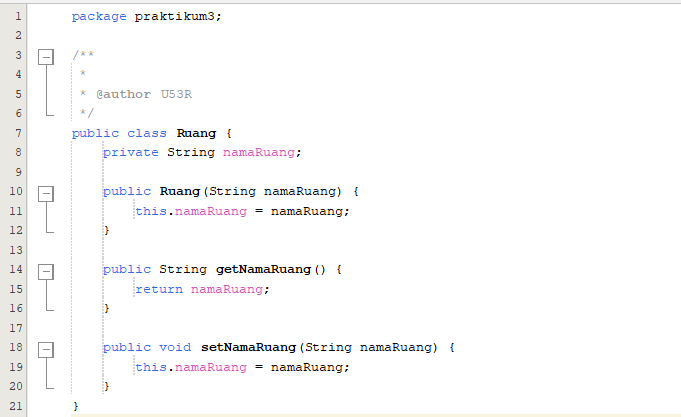
* Memanggil method gaji dari objek Orang dan Pegawai pada kelas Kantor
* UnitKerja.java



1. **Composition**

Komposisi adalah bentuk agregasi terbatas di mana dua entitas sangat bergantung satu sama lain. Dalam kasus program diatas, misalnya Gedung yang terdiri dari ruangan. Setiap Gedung pasti punya ruangan. Jika tidak ada ruangan, maka gedung tidak akan ada.

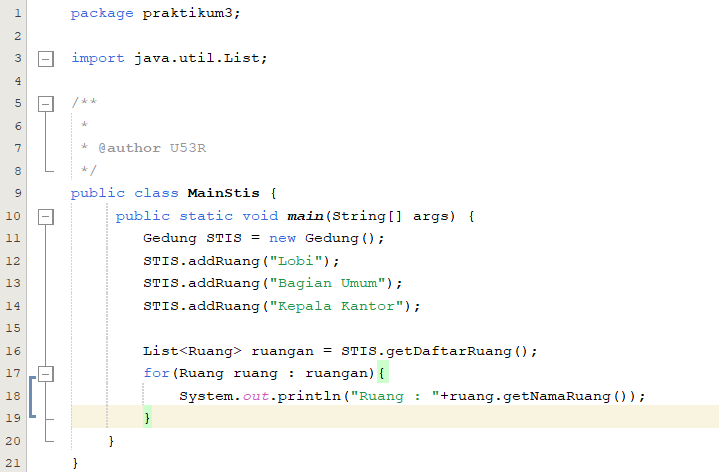
* Ruang.java

****

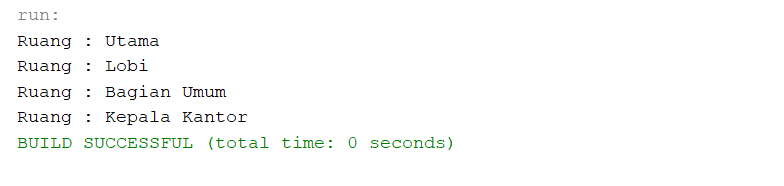
* Gedung.java

****

* MainStis.java



* Hasil running

****