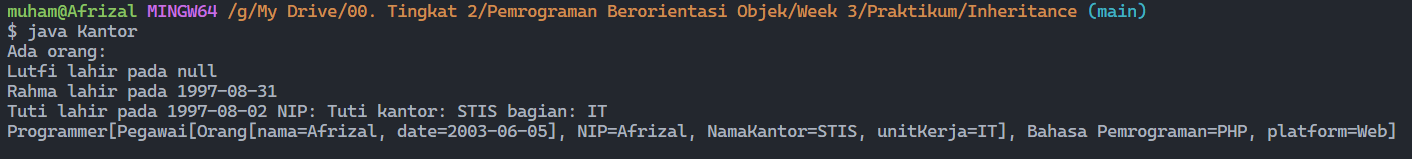
**MODUL 3**

**PEMROGRAMAN BERORIENTASI OBJEK**

1. **INHERITANCE**

Source Code: [Orang.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Inheritance/Orang.java), [Pegawai.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Inheritance/Pegawai.java), [Programmer.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Inheritance/Programmer.java), [Kantor.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Inheritance/Kantor.java)

Output:



Penjelasan:

Line 1: hasil print dari method getNama() dan getTanggalLahir() dari object Orang yang bernama Lutfi.

Line 2: hasil print dari method getNama() dan getTanggalLahir() dari object Orang yang bernama Rahma.

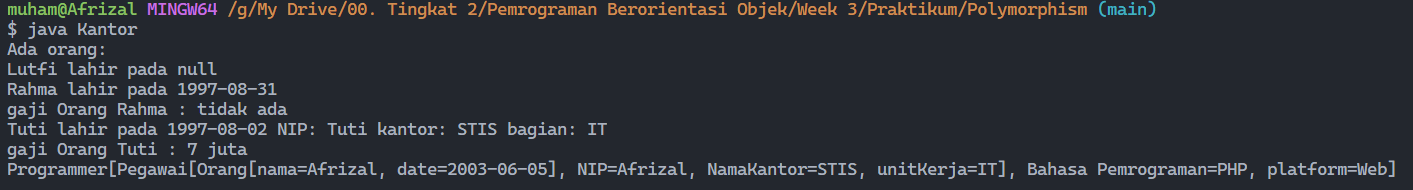
Line 3: hasil print dari method getNama(), getTanggalLahir(), getNIP(), getNamaKantor(), dan getUnitKerja() dari object Pegawai yang bernama Tuti.

Line 4: hasil print dari method toString() dari object Programmer yang bernama Afrizal.

1. **POLYMORPHISM**

Source Code: [Orang.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Polymorphism/Orang.java), [Pegawai.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Polymorphism/Pegawai.java), [Programmer.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Polymorphism/Programmer.java), [Kantor.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Polymorphism/Kantor.java)

Output:



Penjelasan:

Terdapat tambahan method getGaji(). Pada object bernama Rahma, akan menjalankan method getGaji() di object Orang. Sedangkan pada object bernama Tuti, akan menjalankan method getGaji() di object Pegawai dan mengoverload method getGaji() yang ada di object Orang.

1. **AGGREGATION**

Source Code: [UnitKerja.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Aggregation/UnitKerja.java), [Kantor.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Aggregation/Kantor.java)

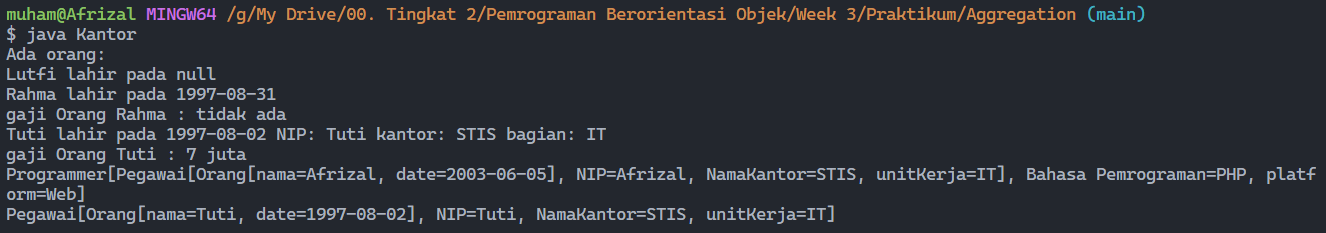
List<Pegawai> daftarPegawai = new **ArrayList**<Pegawai>();

daftarPegawai.**add**(tuti);

UnitKerja Umum = new **UnitKerja**("Umum", daftarPegawai);

System.out.**println**(Umum);

Output:



Penjelasan:

Dengan menggunakan class UnitKerja, maka pegawai dapat diagregatkan didalam List yang bernama daftarPegawai. Sehingga, bisa diprint pegawai yang masuk ke dalam List.

1. **COMPOSITION**

Source Code: [Ruang.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Composition/Ruang.java), [Gedung.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Composition/Gedung.java), [GedungTest.java](https://github.com/afrzl/Pemrograman-Berorientasi-Objek/blob/main/Week%203/Praktikum/Composition/GedungTest.java)

Gedung STIS = new **Gedung**();

STIS.**addRuang**("Lobi");

STIS.**addRuang**("Bagian Umum");

STIS.**addRuang**("Kepala Kantor");

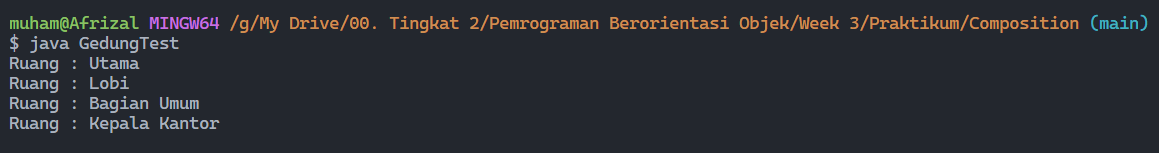
List<Ruang> ruangan = STIS.**getDaftarRuang**();

for (Ruang ruang : ruangan) {

  System.out.**println**("Ruang : " + ruang.**getNamaRuang**());

}

Output:



Penjelasan:

Object Gedung yang berisi List Ruang dibuat dengan nama STIS lalu dengan method constructor tanpa parameter sehingga terbuat object Ruang dengan nama Utama. Setelah itu ditambahkan lagi beberapa ruang. Sehingga, bisa diprint semua Ruang yang ada di STIS dengan for.