Nama : Kelvianto Pratama Harum

NIM : 200210500016

MK : Pemrograman Lanjut

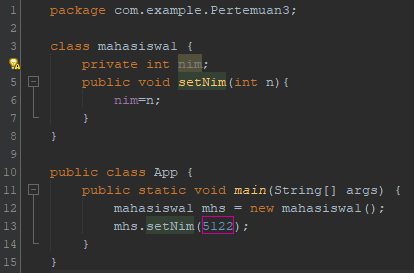
Dosen : Muhammad Fajar B, S.Pd., M.Cs.

**PERTEMUAN III**

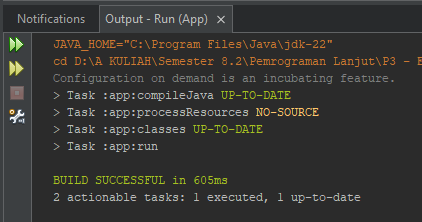
**ENKAPSULASI**

1. **Contoh 3.1 – Method**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **App.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **mahasiswa1**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **8**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **mahasiswa1**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **private** bernama **nim** dengan tipe data **integer**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setNim**, dalam method ini parameter yang digunakan adalah variabel **n** dengan tipe data **integer**. Scope method ini mulai dari baris **5** hingga baris **7**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **6**, diberikan nilai pada field **nim** dengan nilai dari argumen variabel **n** pada saat pemanggilan method **setNim**.

Pada baris **10**, dideklarasikan Class bernama **App**, class ini mencakup semua kode yang ada pada baris **10** hingga baris **15**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **App**.

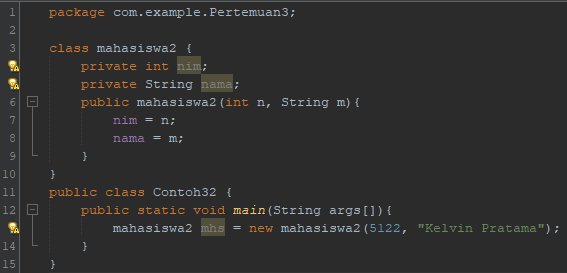
Pada baris **11**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **11** hingga baris **14**.

Pada baris **12**, dibuat sebuah object dari class **mahasiswa** dengan nama **mhs** tanpa argumen konstruktor.

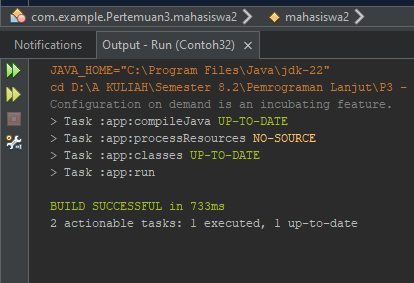
Pada baris **13**, dipanggil method **setNim** pada object **mhs** dengan argumen nilai **5122**. Pemanggilan method ini berarti akan mengubah nilai **field nim** pada object **mhs** dengan nilai **5122**.

1. **Contoh 3.2 – Konstruktor**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh32.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **mahasiswa2**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **10**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **mahasiswa2**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **private** bernama **nim** dengan tipe data **integer**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **private** bernama **nama** dengan tipe data **String**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **6**, dideklarasikan **konstruktor** untuk class ini (**mahasiswa2**). Konstruktor adalah bagian kode yang paling pertama dipanggil pada saat Object Initialization atau pembentukan object. Konstruktor ini dapat menggunakan parameter atau tidak. Pada dasarnya, konstruktor adalah method yang pertama kali dijalankan pada saat object creation/loading. Konstruktor ini memiliki parameter variabel **n** dengan tipe data **integer**, dan variabel **m** dengan tipe data **String**.

Pada baris **7**, diberikan nilai pada field **nim** dengan nilai dari argumen variabel **n** pada saat konstruktor dijalankan.

Pada baris **8**, diberikan nilai pada field **nama** dengan nilai dari argumen variabel **m** pada saat konstruktor dijalankan.

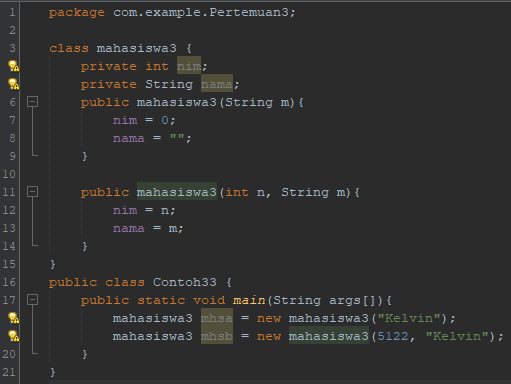
Pada baris **11**, dideklarasikan Class bernama **Contoh32**, class ini mencakup semua kode yang ada pada baris **11** hingga baris **15**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh32**.

Pada baris **12**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **11** hingga baris **14**.

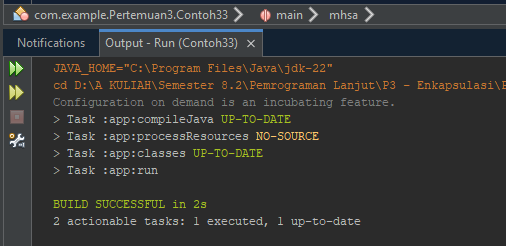
Pada baris **13**, dibuat sebuah object dari class **mahasiswa2** dengan nama **mhs** dan argumen konstruktor nilai **5122** pada parameter **n**, dan nilai **Kelvin Pratama** pada parameter **m**.

1. **Contoh 3.3 – Konstruktor**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh33.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **mahasiswa3**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **15**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **mahasiswa3**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **private** bernama **nim** dengan tipe data **integer**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **private** bernama **nama** dengan tipe data **String**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **6**, dideklarasikan **konstruktor** untuk class ini (**mahasiswa2**). Konstruktor adalah bagian kode yang paling pertama dipanggil pada saat Object Initialization atau pembentukan object. Konstruktor ini dapat menggunakan parameter atau tidak. Pada dasarnya, konstruktor adalah method yang pertama kali dijalankan pada saat object creation/loading. Konstruktor ini memiliki parameter variabel **m** dengan tipe data **String**.

Pada baris **7**, diberikan nilai **0** pada field **nim** pada saat konstruktor dijalankan.

Pada baris **8**, diberikan nilai **“” (string kosong)** pada field **nama** pada saat konstruktor dijalankan.

Pada baris **11**, dibuat sebuah konstruktor untuk class ini (**mahasiswa3**). Konstruktor dapat dideklarasikan lebih dari satu dengan catatan memiliki parameter yang berbeda dengan konstruktor lainnya, konsep ini dinamakan **constructor overloading**. Pada saat **runtime**, program akan menyesuaikan konstruktor mana yang akan digunakan pada saat pembuatan object tergantung pada argumen yang diberikan pada saat pembuatan object. Konstruktor ini memiliki parameter variabel **n** dengan tipe data **integer**, dan variabel **m** dengan tipe data **String**.

Pada baris **12**, diberikan nilai pada field **nim** dengan nilai dari argumen variabel **n** pada saat konstruktor dijalankan.

Pada baris **13**, diberikan nilai pada field **nama** dengan nilai dari argumen variabel **m** pada saat konstruktor dijalankan.

Pada baris **16**, dideklarasikan Class bernama **Contoh33**, class ini mencakup semua kode yang ada pada baris **16** hingga baris **21**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh33**.

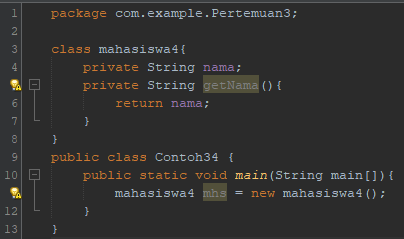
Pada baris **17**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **17** hingga baris **20**.

Pada baris **21**, dibuat sebuah object dari class **mahasiswa3** dengan nama **mhsa** dan argumen konstruktor nilai **Kelvin** pada parameter **m**. Pembuatan object ini akan menggunakan konstruktor ke-1 (berdasarkan urutan deklarasi)

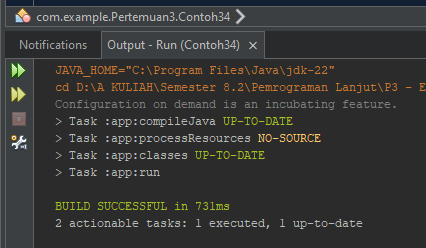
Pada baris **22**, dibuat sebuah object dari class **mahasiswa3** dengan nama **mhsb** dan argumen konstruktor nilai **5122** pada parameter **n**, dan nilai **Kelvin** pada parameter **m**. Pembuatan object ini akan menggunakan konstruktor ke-2 (berdasarkan urutan deklarasi)

1. **Contoh 3.4**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh34.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **mahasiswa4**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **8**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **mahasiswa4**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **private** bernama **nama** dengan tipe data **String**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan method dengan modifier **private**, tipe data balik **String**, nama **getNim**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **5** hingga baris **7**. Modifier **private** dalam deklarasi ini berarti method ini hanya dapat dipanggil **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **6**, nilai **field nama** dimuat dan dikembalikan kepada pemanggil method.

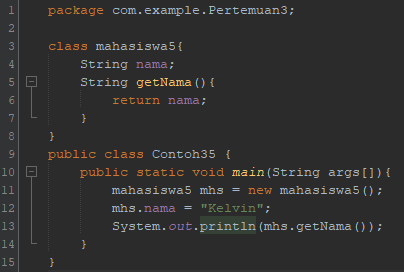
Pada baris **9**, dideklarasikan Class bernama **Contoh34**, class ini mencakup semua kode yang ada pada baris **9** hingga baris **13**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh34**.

Pada baris **10**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **10** hingga baris **12**.

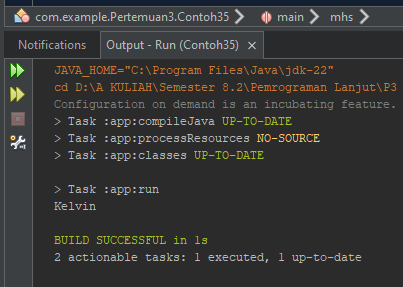
Pada baris **11**, dibuat sebuah object dari class **mahasiswa4** dengan nama **mhs** tanpa argumen konstruktor.

1. **Contoh 3.5**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh35.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **mahasiswa5**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **8**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **mahasiswa5**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **default** (tanpa keyword modifier seperti **public, private, protected**) bernama **nama** dengan tipe data **String**. Modifier **default** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** package yang sama dengan class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan method dengan modifier **default** (tanpa keyword modifier seperti **public, private, protected**), tipe data balik **String**, nama **getNama**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **5** hingga baris **7**. Modifier **default** dalam deklarasi ini berarti method ini hanya dapat dipanggil **didalam** package yang sama dengan class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **6**, nilai **field nama** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **9**, dideklarasikan Class bernama **Contoh35**, class ini mencakup semua kode yang ada pada baris **9** hingga baris **15**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh35**.

Pada baris **10**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **10** hingga baris **12**.

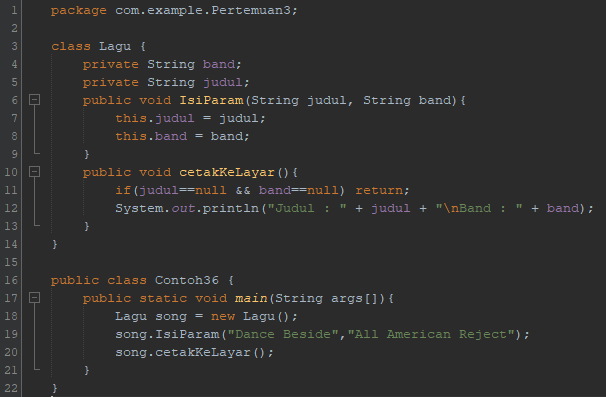
Pada baris **11**, dibuat sebuah object dari class **mahasiswa5** dengan nama **mhs** tanpa argumen konstruktor.

Pada baris **12**, diubah field **nama** dalam object **mhs**, dengan nilai **“Kelvin”**.

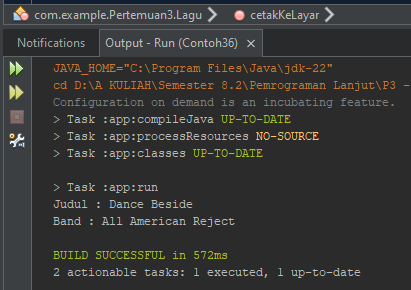
Pada baris **13**, digunakan untuk menampilkan nilai balik dari pemanggilan method **getNama()** dari object **mhs**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

1. **Contoh 3.6**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh36.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **Lagu**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **14**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Lagu**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **private** bernama **band** dengan tipe data **String**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **private** bernama **judul** dengan tipe data **String**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **6**, dideklarasikan sebuah method tanpa nilai balik (**void**), dengan nama **IsiParam** dan parameter variabel **judul** dengan tipe data **String**, dan parameter variabel **band** dengan tipe data **String**. Scope method ini dari baris **6** hingga baris **9**.

Pada baris **7**, diberikan nilai pada field **judul** dengan nilai dari argumen variabel **judul** pada saat method dipanggil.

Pada baris **8**, diberikan nilai pada field **band** dengan nilai dari argumen variabel **band** pada saat method dipanggil.

Pada baris **10**, dideklarasikan sebuah method tanpa nilai balik (**void**), dengan nama **cetakKeLayar**. Scope method ini dari baris **10** hingga baris **13**.

Pada baris **11**, dideklarasikan seleksi kondisi dengan kondisi jika **member variabel judul** tidak memiliki nilai **dan** **member variabel band** tidak memiliki nilai maka akhiri eksekusi method ini.

Pada baris **12**, digunakan untuk string **“Judul: “** diikuti dengan nilai **member variable** **judul** lalu baris baru, kemudian string “**Band : “** dan nilai **member variabel band**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **16**, dideklarasikan Class bernama **Contoh36**, class ini mencakup semua kode yang ada pada baris **16** hingga baris **22**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh36**.

Pada baris **17**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **17** hingga baris **21**.

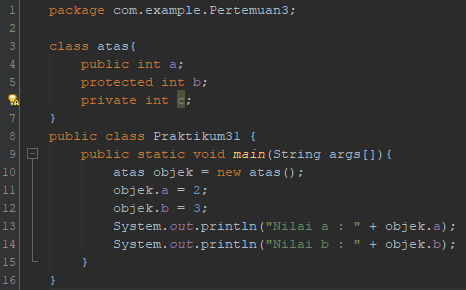
Pada baris **18**, dibuat sebuah object dari class **Lagu** dengan nama **song** tanpa argumen konstruktor.

Pada baris **19**, dipanggil method **IsiParam** dari object **song** dengan argumen **“Dance Beside”** dan **“All American Reject”**. Pemanggilan ini akan mengubah nilai field **judul** pada object **song** dengan nilai **“Dance Beside”** dan nilai field **band** pada object **song** dengan nilai **“All American Reject”**.

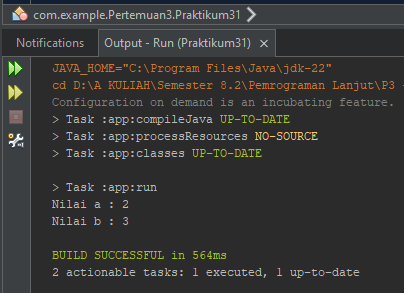
Pada baris **20**, dipanggil method **cetakKeLayar** dari object **song**.

1. **Praktikum 3.1**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Praktikum31.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **atas**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **7**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **atas**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **public** bernama **a** dengan tipe data **int**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class maupun package lain.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **protected** bernama **b** dengan tipe data **int**. Modifier **protected** dalam deklarasi ini berarti field ini hanya dapat diakses **oleh** class yang sama maupun class/object turunannya dalam package yang sama.

Pada baris **6**, dideklarasikan sebuah field dengan modifier **private** bernama **c** dengan tipe data **int**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **8**, dideklarasikan Class bernama **Praktikum31**, class ini mencakup semua kode yang ada pada baris **8** hingga baris **16**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Praktikum31**.

Pada baris **9**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **9** hingga baris **15**.

Pada baris **10**, dibuat object dari class **atas** dengan nama **objek** tanpa argumen konstruktor.

Pada baris **11**, dimodifikasi nilai field **a** pada object **objek** dengan nilai **2**.

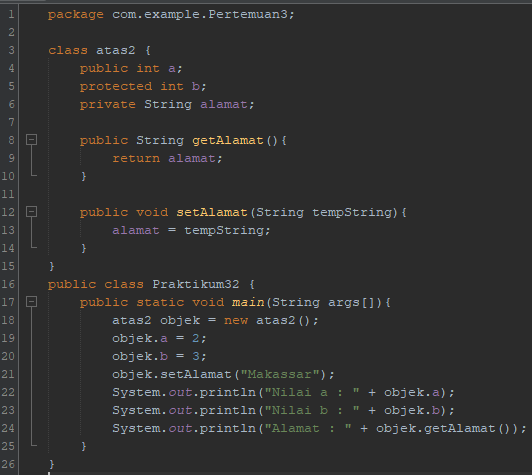
Pada baris **12**, dimodifikasi nilai field **b** pada object **objek** dengan nilai **3**.

Pada baris **13**, digunakan untuk string **“Nilai a: “** diikuti dengan nilai field **a** pada object **objek**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

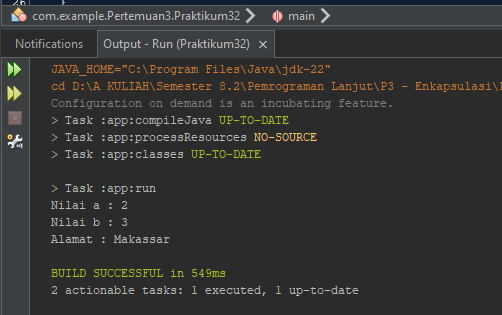
Pada baris **14**, digunakan untuk string **“Nilai b: “** diikuti dengan nilai field **b** pada object **objek**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

1. **Praktikum 3.2**

Source code:



Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Praktikum32.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **atas2**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **15**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **atas2**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **public** bernama **a** dengan tipe data **int**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class maupun package lain.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **protected** bernama **b** dengan tipe data **int**. Modifier **protected** dalam deklarasi ini berarti field ini hanya dapat diakses **oleh** class yang sama maupun class/object turunannya dalam package yang sama.

Pada baris **6**, dideklarasikan sebuah field dengan modifier **private** bernama **alamat** dengan tipe data **String**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **8**, dideklarasikan method dengan nama **getAlamat** dan tipe data balik **String** tanpa parameter. Scope method ini mulai dari baris **8** hingga baris **10**.

Pada baris **9**, dimuat nilai field **alamat** pada class ini lalu digunakan sebagai nilai balik method ini.

Pada baris **12**, dideklarasikan method dengan nama **setAlamat** tanpa tipe data balik dan parameter variabel **tempString** dengan tipe data **String**. Scope method ini mulai dari baris **12** hingga baris **14**.

Pada baris **13**, dilakukan modifikasi nilai field **alamat** pada class ini dengan nilai variabel argumen **tempString**.

Pada baris **16**, dideklarasikan Class bernama **Praktikum32**, class ini mencakup semua kode yang ada pada baris **16** hingga baris **26**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Praktikum32**.

Pada baris **17**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **17** hingga baris **25**.

Pada baris **18**, dibuat object dari class **atas2** dengan nama **objek** tanpa argumen konstruktor.

Pada baris **19**, dimodifikasi nilai field **a** pada object **objek** dengan nilai **2**.

Pada baris **20**, dimodifikasi nilai field **b** pada object **objek** dengan nilai **3**.

Pada baris **21**, dipanggil method **setAlamat** dari object **objek** dengan argumen nilai **“Makassar”**. Pemanggilan method ini akan mengubah nilai field **alamat** pada object **objek** dengan nilai **“Makassar”**.

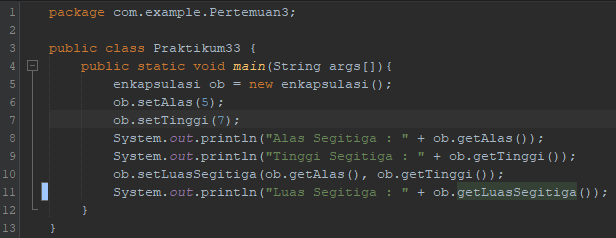
Pada baris **22**, digunakan untuk string **“Nilai a: “** diikuti dengan nilai field **a** pada object **objek**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

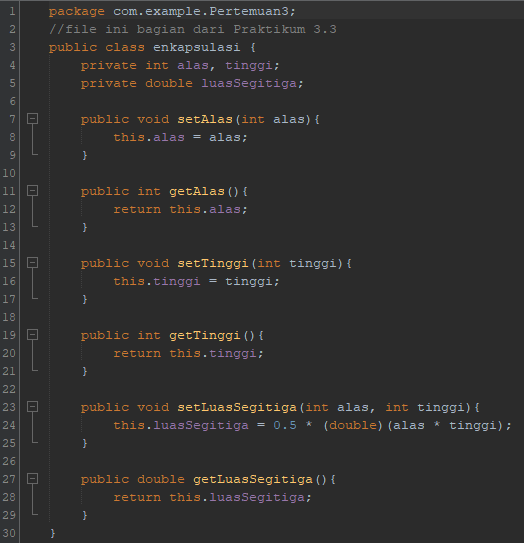
Pada baris **23**, digunakan untuk string **“Nilai b: “** diikuti dengan nilai field **b** pada object **objek**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **24**, digunakan untuk string **“Alamat: “** diikuti dengan nilai balik pemanggilan method **getAlamat** pada object **objek**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

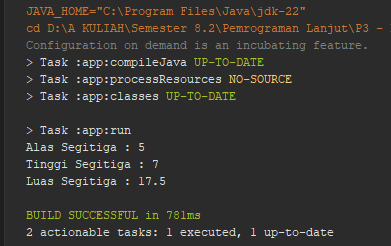
1. **Praktikum 3.3**

Source code:





Output:



Penjelasan:

Praktikum33.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Praktikum33.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **Praktikum33**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **13**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Praktikum33**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **12**.

Pada baris **5**, dibuat object dari class **enkapsulasi** dengan nama **ob** tanpa argumen konstruktor.

Pada baris **6**, dipanggil method **setAlas** dari object **ob** dengan argumen nilai **5**. Pemanggilan method ini akan mengubah nilai field **alas** pada object **ob** dengan nilai **5**.

Pada baris **7**, dipanggil method **setTinggi** dari object **ob** dengan argumen nilai **7**. Pemanggilan method ini akan mengubah nilai field **tinggi** pada object **ob** dengan nilai **7**.

Pada baris **8**, digunakan untuk string **“Alas Segitiga: “** diikuti dengan nilai balik pemanggilan method **getAlas** pada object **ob**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **9**, digunakan untuk string **“Tinggi Segitiga: “** diikuti dengan nilai balik pemanggilan method **getTinggi** pada object **ob** Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **10**, dipanggil method **setLuasSegitiga** dari object **ob** dengan argumen nilai balik pemanggilan method **getAlas** pada object **ob** dan nilai balik pemanggilan method **getTinggi** pada object **ob.**

Pada baris **11**, digunakan untuk string **“Luas Segitiga: “** diikuti dengan nilai balik pemanggilan method **getLuasSegitiga** pada object **ob**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

enkapsulasi.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **enkapsulasi.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **enkapsulasi**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **30**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **enkapsulasi**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **private** bernama **alas** dan **tinggi** dengan tipe data **integer**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **private** bernama **luasSegitiga** dengan tipe data **double**. Modifier **private** dalam deklarasi ini berarti field ini hanya dapat diakses **didalam** class ini saja, tidak pada object turunannya maupun class lain atau inheritancenya.

Pada baris **7**, dideklarasikan method dengan nama **setAlas** tanpa tipe data balik dan parameter variabel **alas** dengan tipe data **int**. Scope method ini mulai dari baris **7** hingga baris **9**.

Pada baris **8**, dilakukan modifikasi nilai field **alas** pada class ini dengan nilai variabel argumen **alas**.

Pada baris **11**, dideklarasikan method dengan nama **getAlamat** dan tipe data balik **int** tanpa parameter. Scope method ini mulai dari baris **11** hingga baris **13**.

Pada baris **12**, dimuat nilai field **alas** pada class ini lalu digunakan sebagai nilai balik method ini.

Pada baris **15**, dideklarasikan method dengan nama **setTinggi** tanpa tipe data balik dan parameter variabel **tinggi** dengan tipe data **int**. Scope method ini mulai dari baris **15** hingga baris **17**.

Pada baris **16**, dilakukan modifikasi nilai field **tinggi** pada class ini dengan nilai variabel argumen **tinggi**.

Pada baris **19**, dideklarasikan method dengan nama **getTinggi** dan tipe data balik **int** tanpa parameter. Scope method ini mulai dari baris **19** hingga baris **21**.

Pada baris **20**, dimuat nilai field **tinggi** pada class ini lalu digunakan sebagai nilai balik method ini.

Pada baris **23**, dideklarasikan method dengan nama **setLuasSegitiga** tanpa tipe data balik dan parameter variabel **alas** dan **tinggi** dengan tipe data **int**. Scope method ini mulai dari baris **23** hingga baris **25**.

Pada baris **24**, dilakukan modifikasi nilai field **luasSegitiga** pada class ini dengan hasil perhitungan **0.5 \* (alas \* tinggi)**.

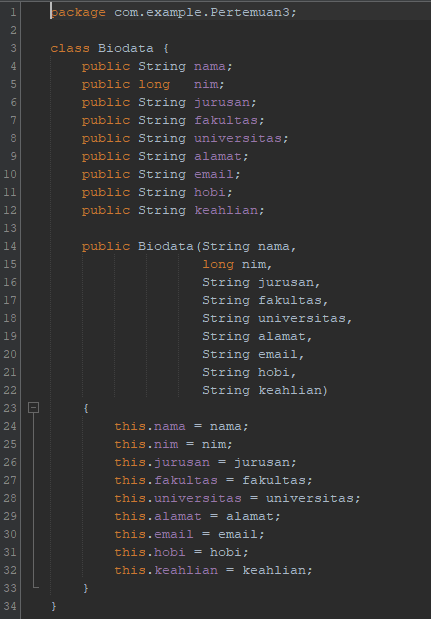
Pada baris **27**, dideklarasikan method dengan nama **getTinggi** dan tipe data balik **int** tanpa parameter. Scope method ini mulai dari baris **27** hingga baris **29**.

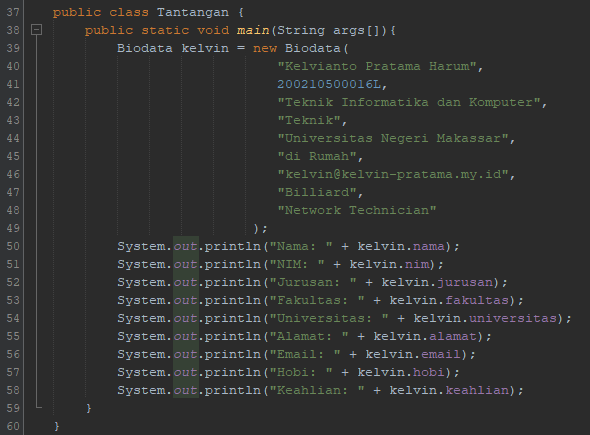
Pada baris **28**, dimuat nilai field **luasSegitiga** pada class ini lalu digunakan sebagai nilai balik method ini.

1. **Soal Tantangan**

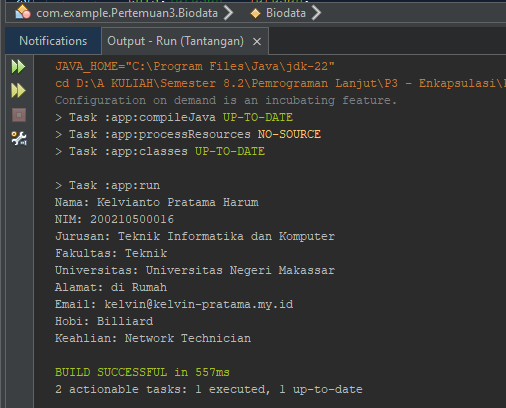
Buatlah program enkapsulasi sederhana yang dapat menampilkan biodata Anda, berupa : nama, nim, jurusan, fakultas, universitas, alamat, email, hobi, dan keahlian.

Source code:





Output:



Penjelasan:

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Tantangan.java** tergabung pada package **com.example.Pertemuan3**

Pada baris **3**, dideklarasikan Class bernama **Biodata**, class ini mencakup semua kode yang ada pada baris **3** hingga baris **34**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Biodata**.

Pada baris **4**, dideklarasikan sebuah field dengan modifier **public** bernama **nama** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **5**, dideklarasikan sebuah field dengan modifier **public** bernama **nim** dengan tipe data **long**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **6**, dideklarasikan sebuah field dengan modifier **public** bernama **jurusan** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **7**, dideklarasikan sebuah field dengan modifier **public** bernama **fakultas** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **8**, dideklarasikan sebuah field dengan modifier **public** bernama **universitas** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **9**, dideklarasikan sebuah field dengan modifier **public** bernama **alamat** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **10**, dideklarasikan sebuah field dengan modifier **public** bernama **email** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **11**, dideklarasikan sebuah field dengan modifier **public** bernama **hobi** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **12**, dideklarasikan sebuah field dengan modifier **public** bernama **keahlian** dengan tipe data **String**. Modifier **public** dalam deklarasi ini berarti field ini dapat diakses **semua** class baik dalam package yang sama maupun berbeda.

Pada baris **14**, dideklarasikan konstruktor untuk class ini dengan modifier **public** dan parameter variabel **nama** tipe data **String**, variabel **nim** tipe data **long**, variabel **jurusan** tipe data **String**, variabel **fakultas** tipe data **String**, variabel **universitas** tipe data **String**, variabel **alamat** tipe data **String**, variabel **email** tipe data **String**, variabel **hobi** tipe data **String**, dan variabel **keahlian** tipe data **String**.

Pada baris **24**, dilakukan modifikasi pada field **nama** dengan nilai parameter variabel **nama**.

Pada baris **25**, dilakukan modifikasi pada field **nim** dengan nilai parameter variabel **nim**.

Pada baris **26**, dilakukan modifikasi pada field **jurusan** dengan nilai parameter variabel **jurusan**.

Pada baris **27**, dilakukan modifikasi pada field **fakultas** dengan nilai parameter variabel **fakultas**.

Pada baris **28**, dilakukan modifikasi pada field **universitas** dengan nilai parameter variabel **universitas**.

Pada baris **29**, dilakukan modifikasi pada field **alamat** dengan nilai parameter variabel **alamat**.

Pada baris **30**, dilakukan modifikasi pada field **email** dengan nilai parameter variabel **email**.

Pada baris **31**, dilakukan modifikasi pada field **hobi** dengan nilai parameter variabel **hobi**.

Pada baris **32**, dilakukan modifikasi pada field **keahlian** dengan nilai parameter variabel **keahlian**.

Pada baris **37**, dideklarasikan Class bernama **Tantangan**, class ini mencakup semua kode yang ada pada baris **37** hingga baris **60**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Tantangan**.

Pada baris **38**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **38** hingga baris **59**.

Pada baris **39**, dibuat object dari class **Biodata** dengan nama **kelvin** dengan argumen konstruktor nilai **“Kelvianto Pratama Harum”**, **200210500016, “Teknik Informatika dan Komputer”, “Teknik”, “Universitas Negeri Makassar”, “di Rumah”,** [**kelvin@kelvin-pratama.my.id**](mailto:kelvin@kelvin-pratama.my.id)**, “Billiard”,** dan **“Network Technician”**.

Pada baris **50**, digunakan untuk menampilkan **“Nama: “** diikuti dengan nilai field **nama** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **51**, digunakan untuk menampilkan **“NIM: “** diikuti dengan nilai field **nim** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **52**, digunakan untuk menampilkan **“Jurusan: “** diikuti dengan nilai field **jurusan** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **53**, digunakan untuk menampilkan **“Fakultas: “** diikuti dengan nilai field **fakultas** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **54**, digunakan untuk menampilkan **“Universitas: “** diikuti dengan nilai field **universitas** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **55**, digunakan untuk menampilkan **“Alamat: “** diikuti dengan nilai field **alamat** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **56**, digunakan untuk menampilkan **“Email: “** diikuti dengan nilai field **email** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **57**, digunakan untuk menampilkan **“Hobi: “** diikuti dengan nilai field **hobi** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **58**, digunakan untuk menampilkan **“Keahlian: “** diikuti dengan nilai field **keahlian** pada object **kelvin**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.