LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

“METHOD DAN OBJEK”



Nama : Nia Ramadhani

NIM : 2311531006

Dosen Pengampu : DR. WAHYUDI,S.T,MT.

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERITAS ANDALAS

PADANG

**BAB I**

**PENDAHULUAN**

1. **Tujuan**
   * + 1. Mahasiswa dapat mengetahui penggunaan method dan Objek.
2. **Kajian Teori**

OOP (Object Oriented Programming) adalah salah satu paradigma atau teknik pemrograman yang populer dalam pengembangan aplikasi. Dengan paradigma tersebut, kita dapat dengan mudah memvisualisasikan kode karena OOP sendiri mirip dengan skenario kehidupan nyata. Dalam penerapannya, OOP menggabungkan kumpulan fungsi atau atribut yang memiliki kesamaan dalam sebuah unit yang kita sebut sebagai objek. Class, merupakan sebuah blueprint yang dapat dikembangkan untuk membuat sebuah objek. Di dalam class bisa terdapat beberapa attribut ataupun behaviour. Property, adalah data yang ada dalam sebuah class yang merepresentasikan karakteristik dari sebuah class. Setiap property memiliki struktur dasar, diantaranya adalah modifier, tipe data, nama konstanta dan nilai. Function, sebuah prosedur yang memiliki keterkaitan dengan pesan dan objek.

Ada 4 pilar dalam OOP, yaitu Inheritance, Encapsulation, Abstraction, dan Polymorphism.

Inheritance, mekanisme pewarisan sifat yang dilakukan dari sebuah kelas (induk) ke class baru (anak), di mana beberapa fungsi maupun property yang ada pada induk kelas akan bisa diakses oleh kelas baru tersebut.

Ada beberapa istilah dalam inheritance, yaitu superclass dan subclass. Superclass adalah class yang fitur-fiturnya akan diwariskan. Sedangkan subclass adalah class baru yang akan mewarisi member milik superclass.

Inheritance di kenal ada beberapa jenis, yaitu :

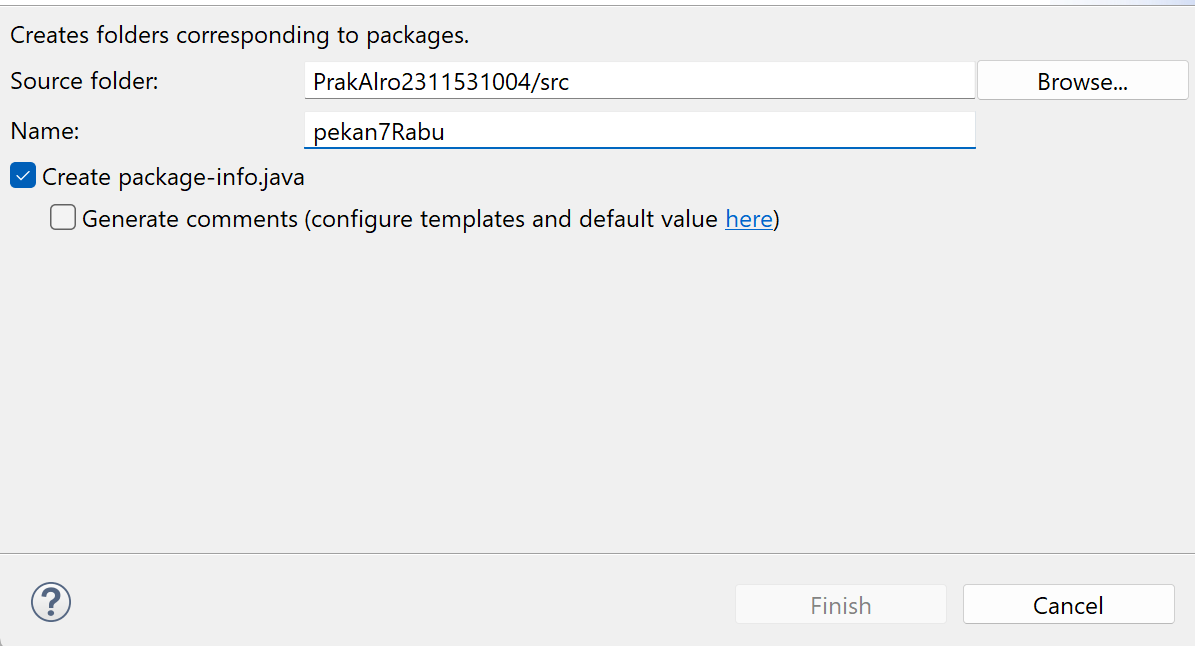
1. Single Inheritance, class yang dibuat hanya mewarisi satu class saja.
2. Multiple Inheritance, class yang mewarisi lebih dari satu superclass
3. Hierarchical Inheritance, sebuah superclass yang diwarisi oleh beberapa subclass.
4. Hybrid Inheritance, kombinasi dari beberapa inheritance.

* Encapsulation, adalah proses di mana penanganan data diisolasi dan ditempatkan di dalam sebuah class. Data yang terisolasi tidak dapat diakses langsung dari luar.
* Abstraction, adalah mekanisme saat proses dalam sebuah objek disembunyikan dan hanya akan menyediakan apa yang benar-benar perlu digunakan.
* Polymorphism, merupakan kemampuan objek, variabel, atau fungsi yang dapat memiliki berbagai bentuk. Polymorphism terbagi menjadi 2 yaitu:
* Compile Time Polymorphism, adalah sebuah proses di mana sebuah method atau fungsi dipanggil saat kompilasi. Contoh dari ini adalah konsep overloading.
* Runtime Polymorphism, adalah proses di mana sebuah fungsi dipanggil pada saat runtime. Contohnya adalah konsep overriding.

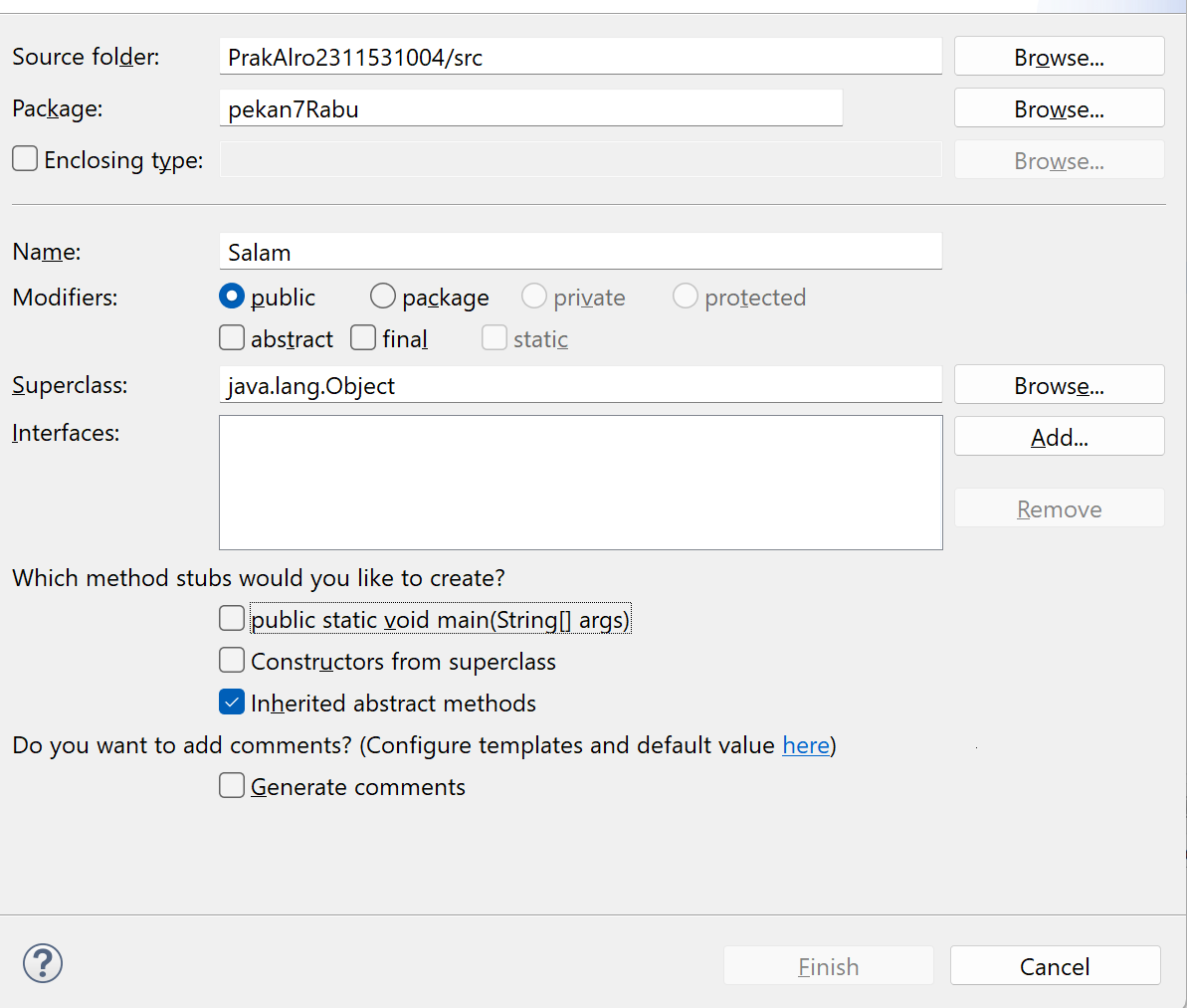
**BAB II**

**CARA KERJA**

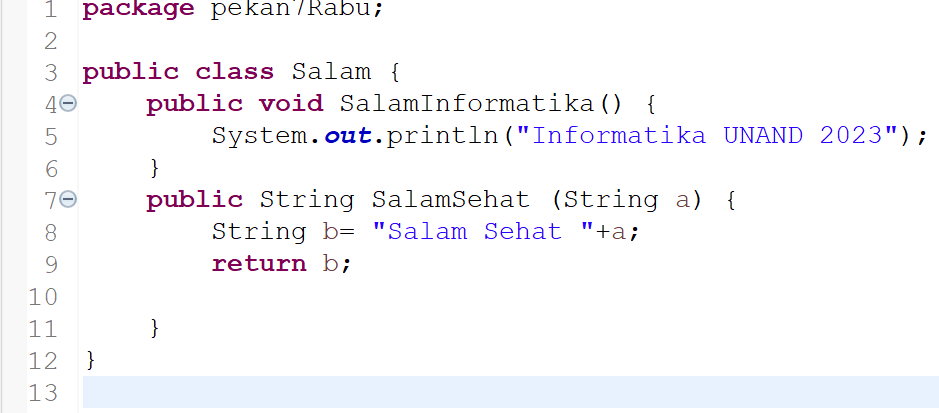
1. **Salam**
2. Buat package baru dengan nama Pekan7Rabu.



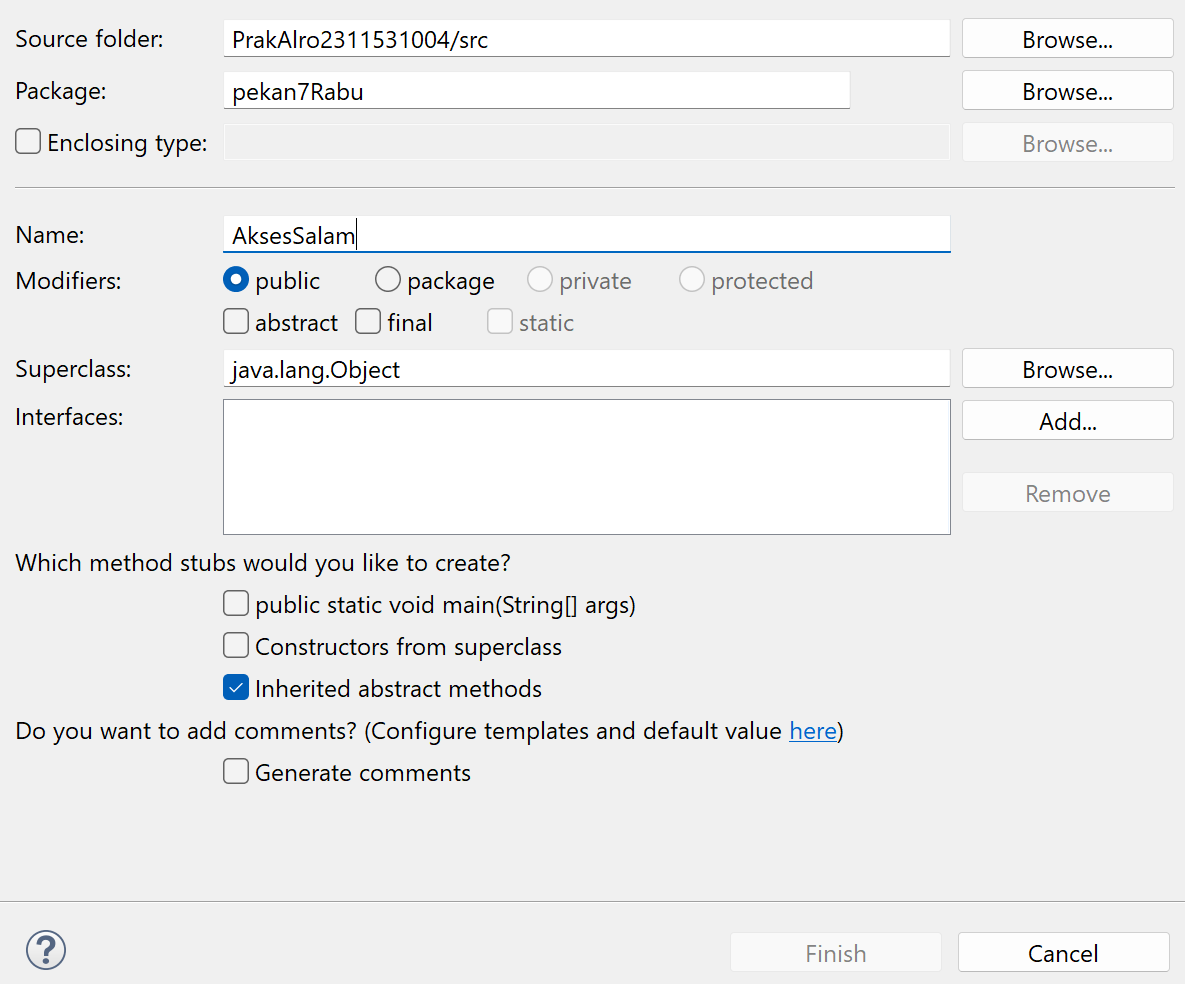
1. Setelah package baru terbentuk, kemudian buat class baru dengan nama Salam.



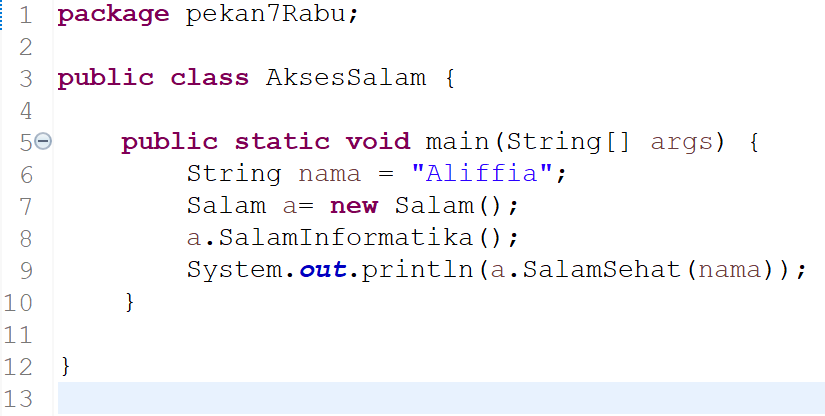
1. Kemudian masukkan codingan seperti gambar dibawah.



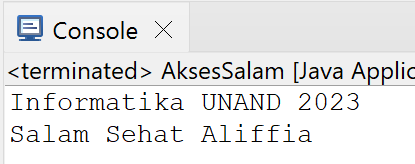
1. Setelah semua codingan selesai, bisa lakukan Run dengan cara klik panah hijau diatas. Outputnya tidak akan keluar karna method nya tidak menggunakan main.
2. **AksesSalam**
3. Buat class baru dengan nama AksesSalam.



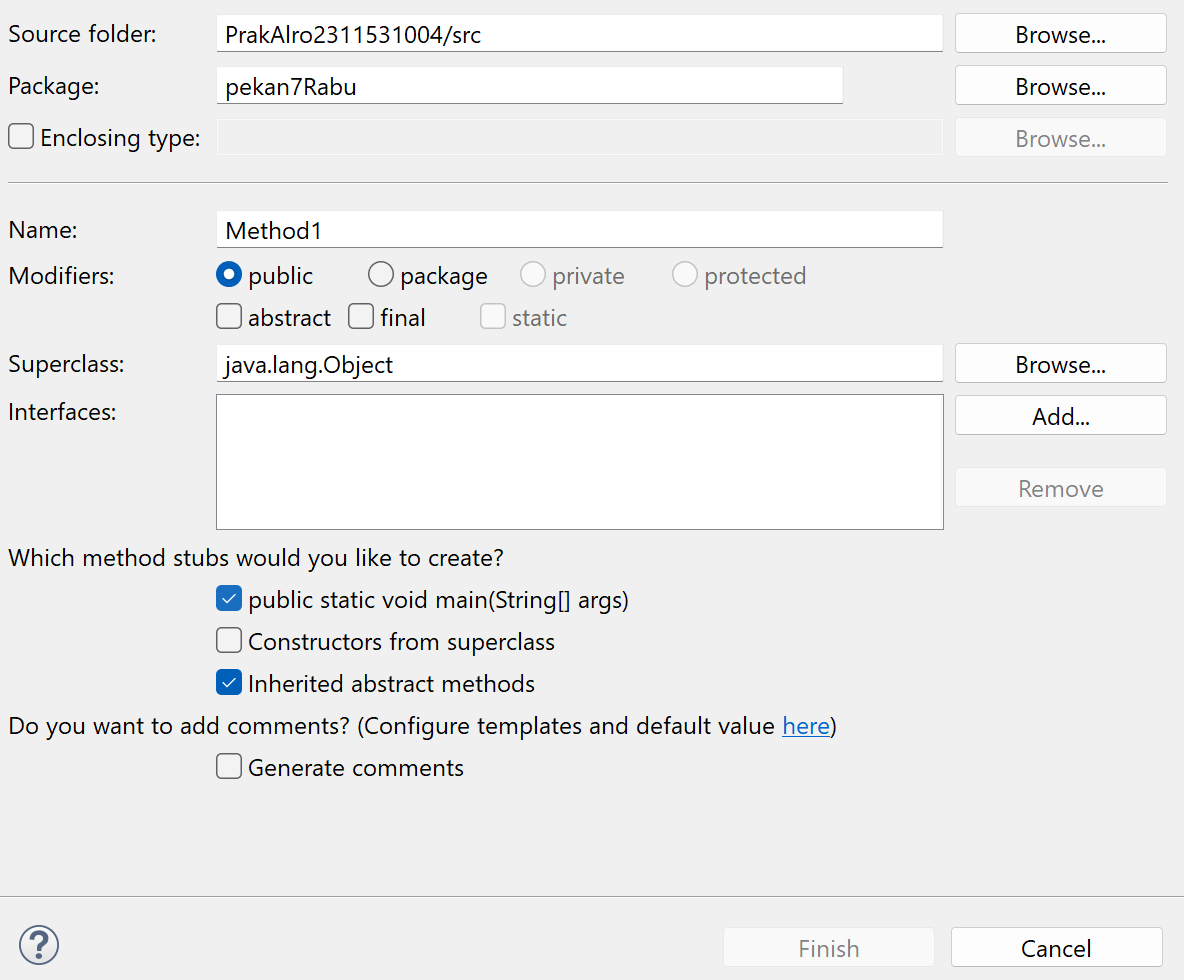
1. Kemudian masukkan codingan seperti gambar dibawah.



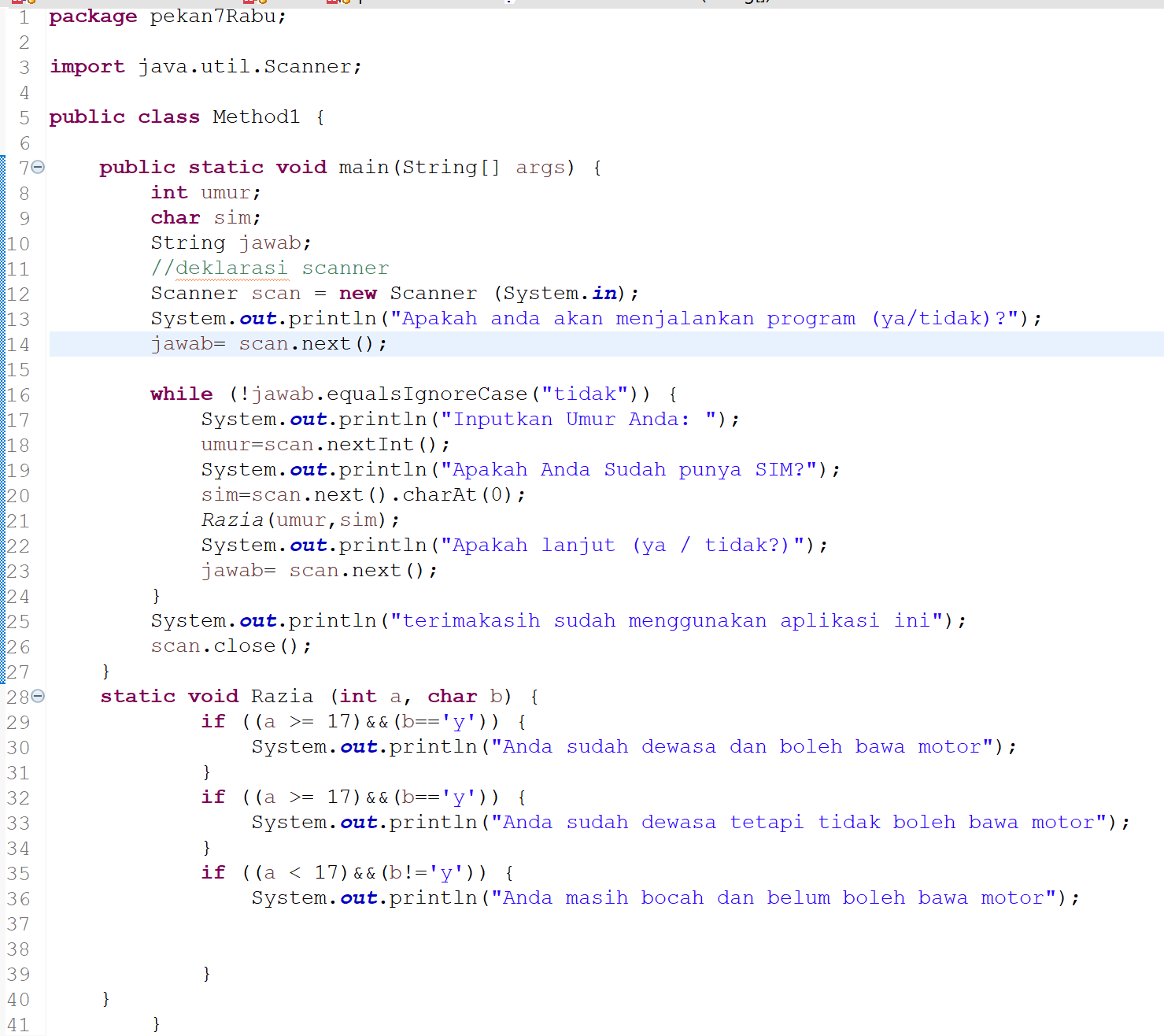
1. Setelah semua codingan selesai, bisa lakukan Run dengan cara klik panah hijau diatas. Jika codingan anda berhasil maka akan muncul seperti gambar dibawah.



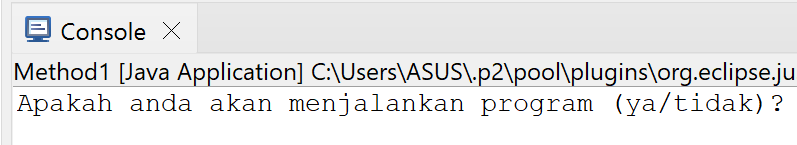
1. Setelah muncul seperti gambar diatas, berarti codingan sudah benar.
2. **Method1**
3. Buat class baru dengan nama Method1.



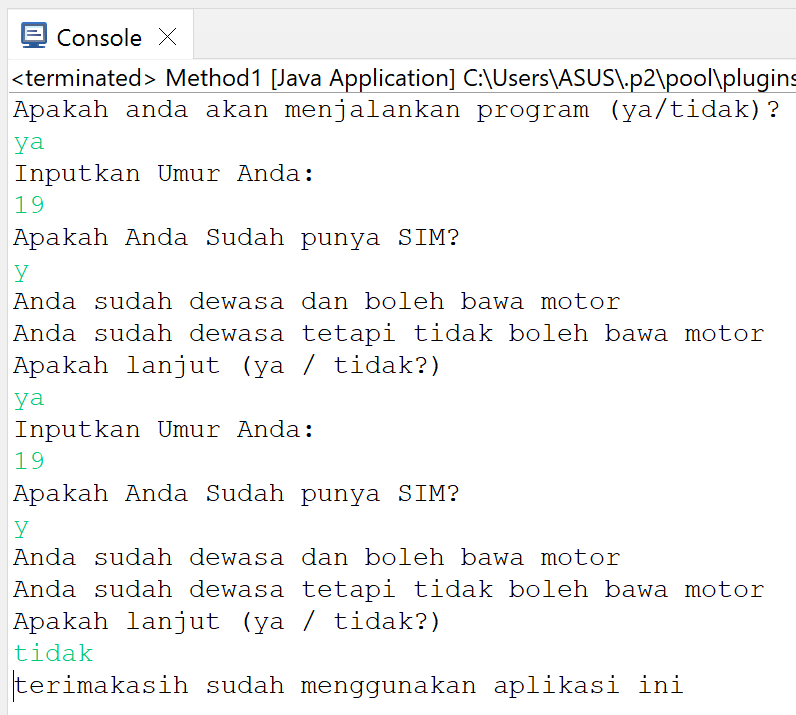
1. Kemudian masukkan codingan seperti gambar dibawah.



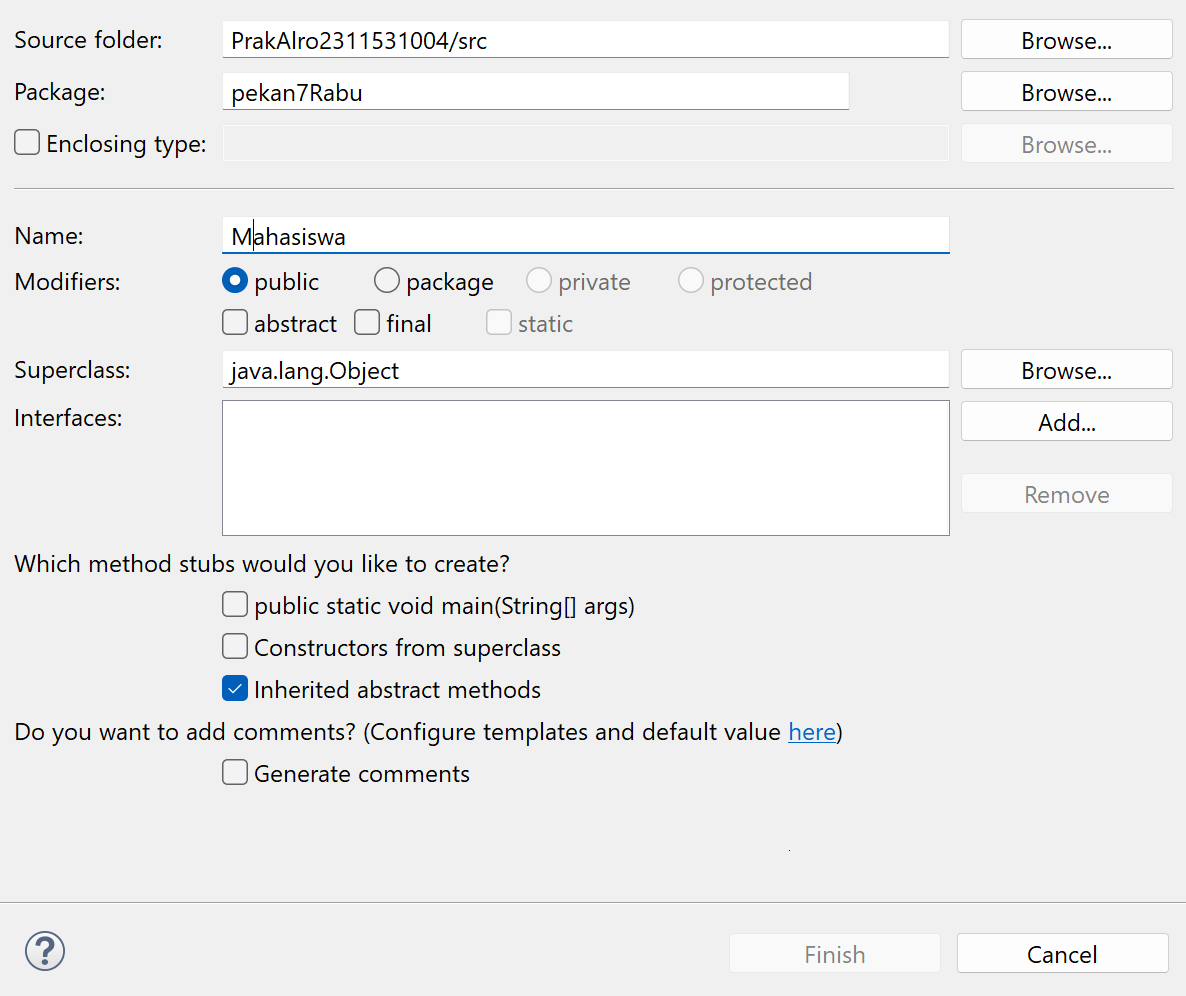
1. Setelah semua codingan selesai, bisa lakukan Run dengan cara klik panah hijau diatas. Jika codingan anda berhasil maka akan muncul seperti gambar dibawah.



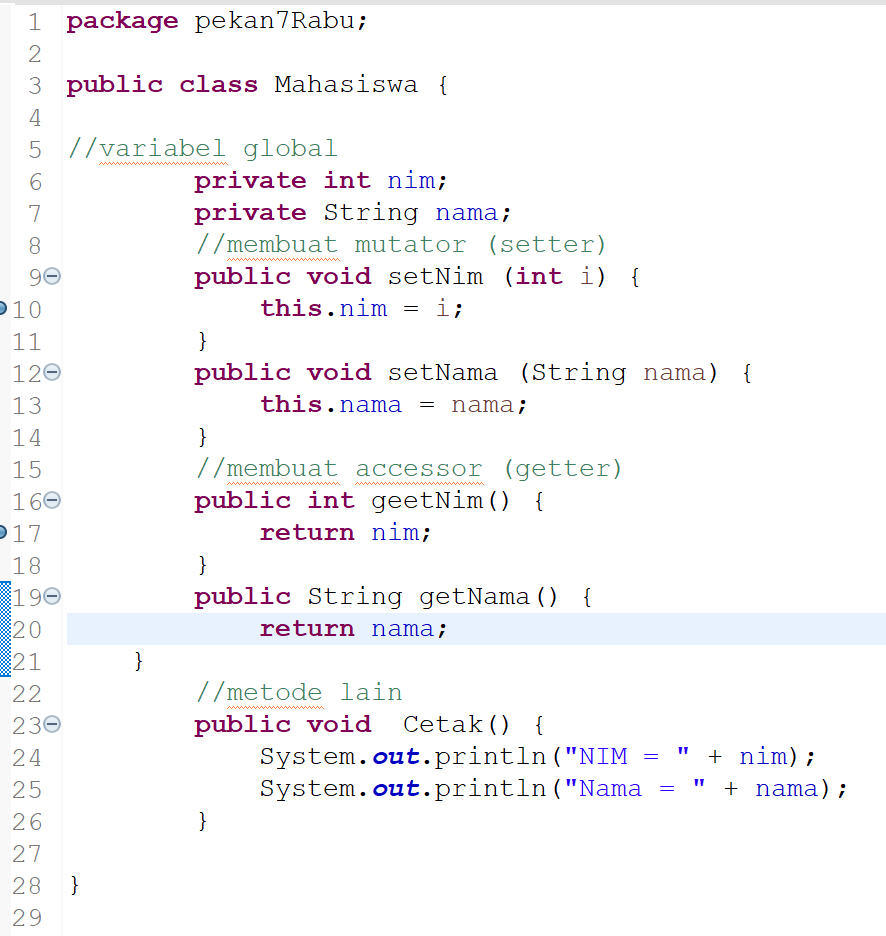
1. Setelah muncul seperti gambar diatas, maka kita bisa coba inputkan untuk tes codingan tersebut. Seperti gambar dibawah. Berarti codingan sudah benar.



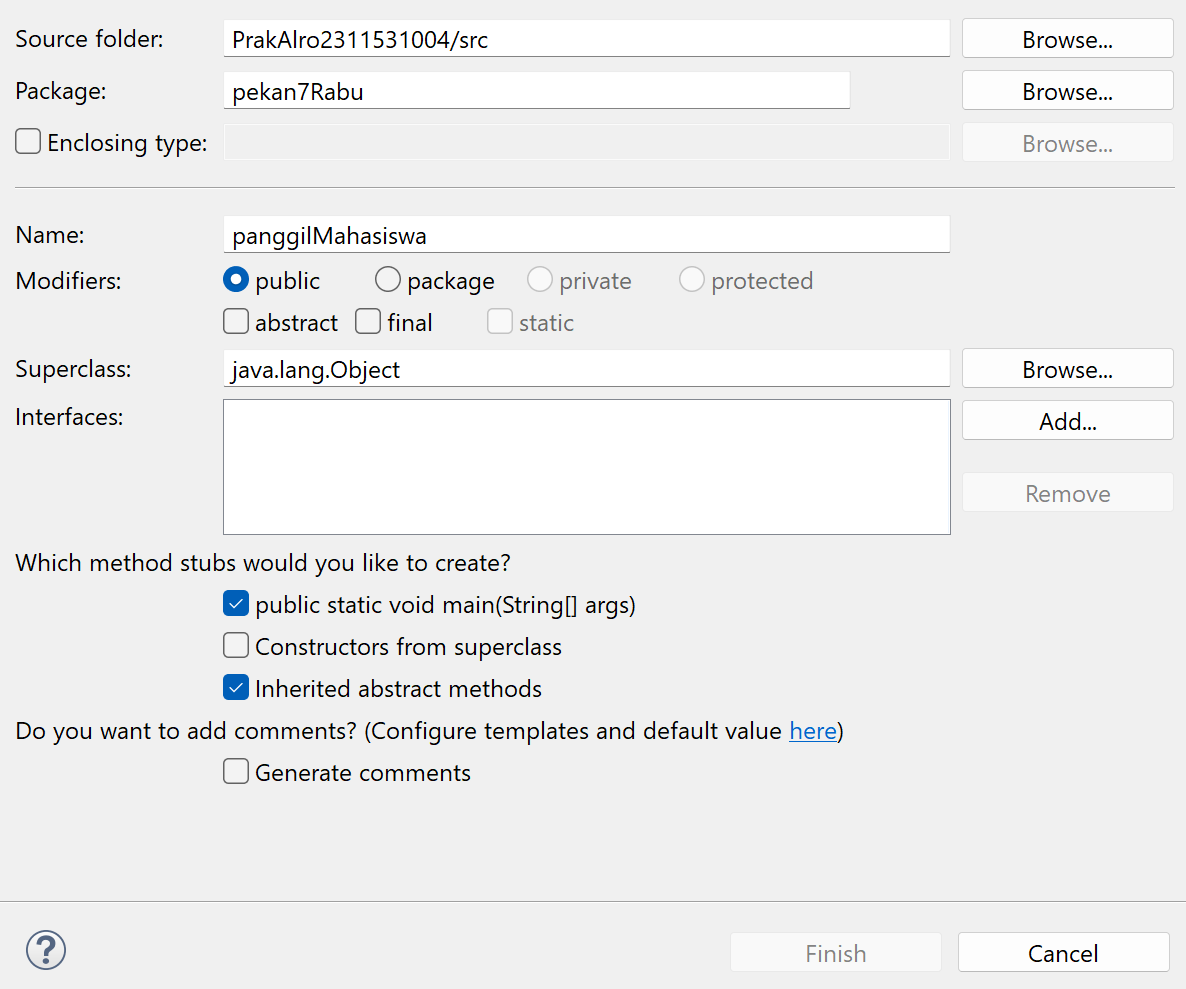
1. Setelah muncul seperti gambar diatas, berarti codingan sudah benar.
2. **Mahasiswa**
3. Buat class baru dengan nama Mahasiswa.



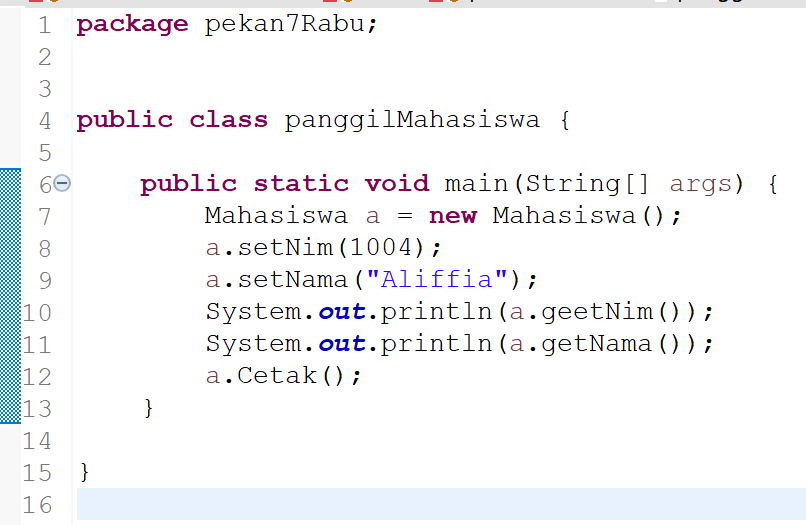
1. Kemudian masukkan codingan seperti gambar dibawah.



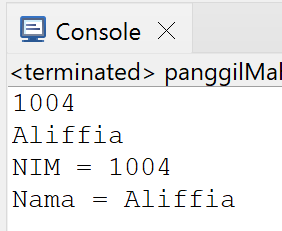
1. Setelah semua codingan selesai, bisa lakukan Run dengan cara klik panah hijau diatas. Outputnya tidak akan keluar karna method nya tidak menggunakan main.
2. **PanggilMahasiswa**
3. Buat class baru dengan nama panggilMahasiswa.



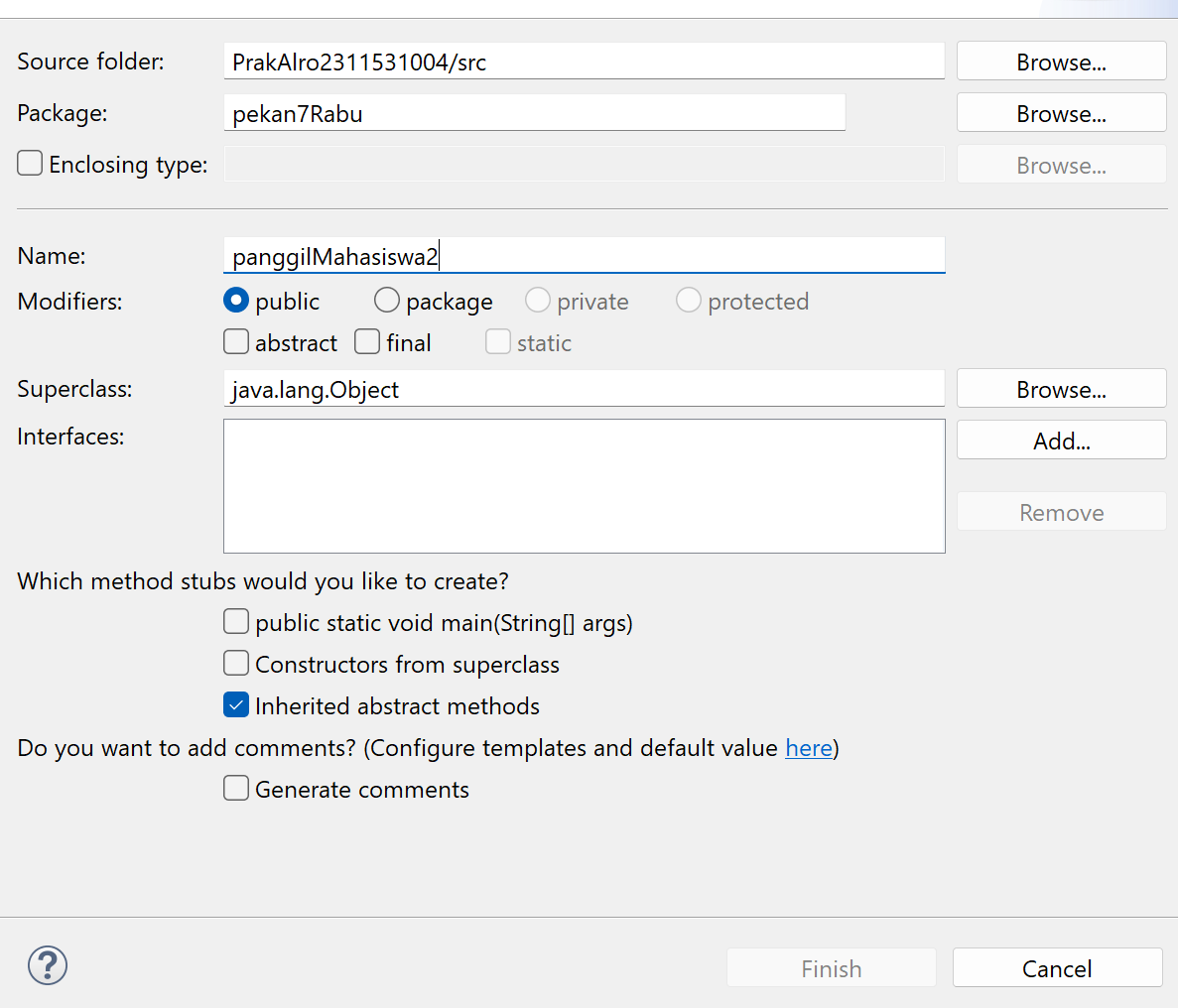
1. Kemudian masukkan codingan seperti gambar dibawah.



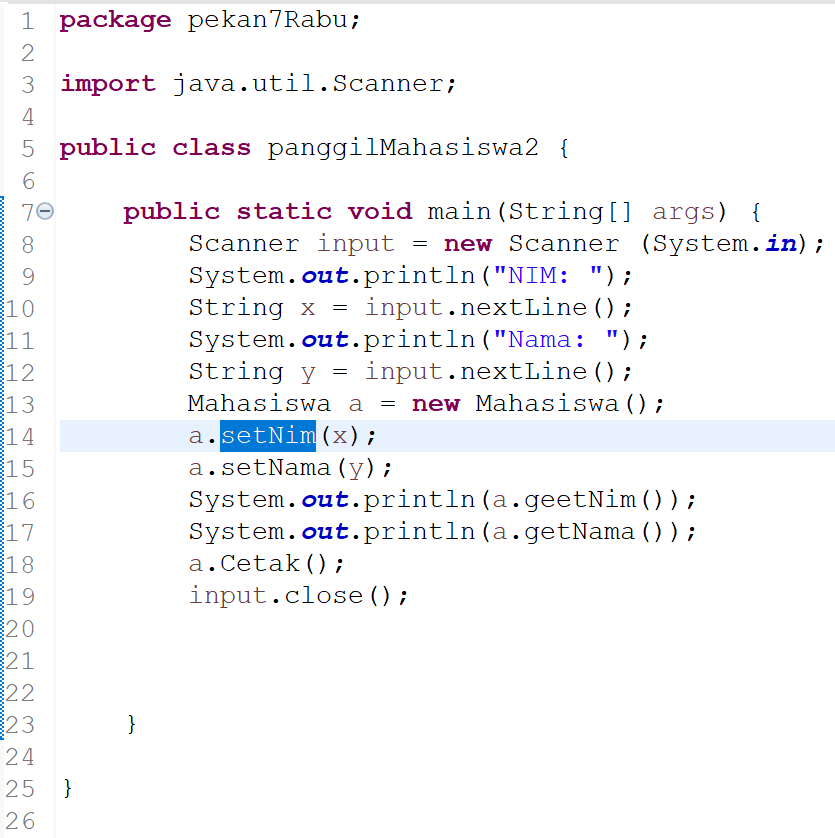
1. Setelah semua codingan selesai, bisa lakukan Run dengan cara klik panah hijau diatas. Seperti gambar dibawah. Berarti codingan sudah benar.



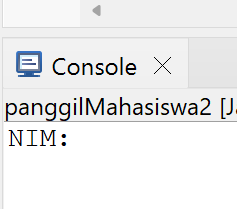
1. **PanggilMahasiswa2**
2. Buat class baru dengan nama panggilMahasiswa.



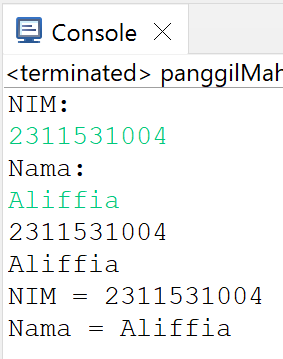
1. Kemudian masukkan codingan seperti gambar dibawah.



1. Setelah semua codingan selesai, bisa lakukan Run dengan cara klik panah hijau diatas. Jika codingan anda berhasil maka akan muncul seperti gambar dibawah.



1. Setelah muncul seperti gambar diatas, maka kita bisa coba inputkan untuk tes codingan tersebut. Seperti gambar dibawah. Berarti codingan sudah benar.



**BAB III**

**PENUTUP**

1. **Kesimpulan**

**Method adalah kumpulan program yang mempunyai nama. Method merupakan sarana bagi programmer untuk memecah program menjadi bagian-bagian yang kecil agar jadi lebih kompleks sehingga dapat di gunakan berulang-ulang. Objek pada java adalah sekumpulan software yang terdiri dari variable dan method-method yang terkait. Objek juga merupakan benda nyata yang di buat berdasarkan rancangan yang di definisikan di dalam class. Pada praktikum kali ini kita dapat mengetahui penggunaan dan fungsi method dan objek.**