

Membangun Aplikasi Java Menyenangkan



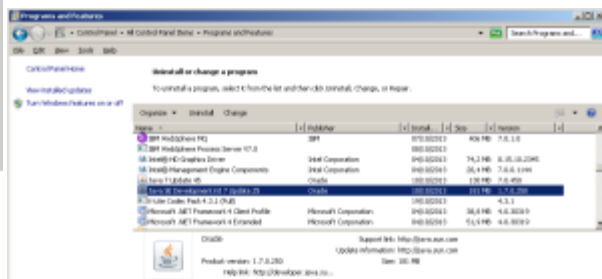
Membangun Aplikasi Java Menggunakan Eclipse

Pada tulisan ini dibahas beberapa hal mendasar yang saya rasa penting diketahui dalam penggunaan Eclipse untuk membangun aplikasi Java. Beberapa hal yang dibahas adalah: pembuatan workspace, cara membuat project, cara membuat Java class, dan cara membuat package.

Konsep abstraction, encapsulation, inheritance dan polymorphism yang telah dibahas pada Tutorial Object Oriented Programming dengan Java Bagian 2, 3 dan 4 dapat diterapkan menggunakan Eclipse. Yang berbeda di sini hanyalah Integrated Development Environment (IDE) saja. Pada artikel-artikel tersebut IDE yang digunakan adalah Netbeans.

Java Development Kit

Pastikan Java Development Kit (JDK) sudah terinstall pada komputer. Bila belum, install JDK terbaru pada komputer Anda. Pada contoh ini di komputer saya telah terinstal JDK 1.7 seperti terlihat pada gambar di bawah:



Bila JDK belum terinstall, download saja JDK ini dari situs Oracle. Kemudian lakukan proses install (cari petunjuk cara install JDK di google, gampang kok). Tanpa JDK, kita tidak akan bisa mengcompile program yang kita buat. Dengan kata lain, kita tidak bisa membuat program Java tanpa JDK.

Persiapan

Buat sebuah direktori pada C:\ dengan nama "TUTORIAL". Direktori ini akan kita gunakan sebagai workspace. Workspace merupakan direktori tempat kita bekerja. Semua file-file project yang dibuat akan disimpan dalam workspace ini.

Download Eclipse. Kemudian ekstrak file hasil download tersebut. Proses ekstrak ini menghasilkan sebuah direktori bernama eclipse. Pindahkan (copy-kan) direktori eclipse ini ke C:\TUTORIAL.

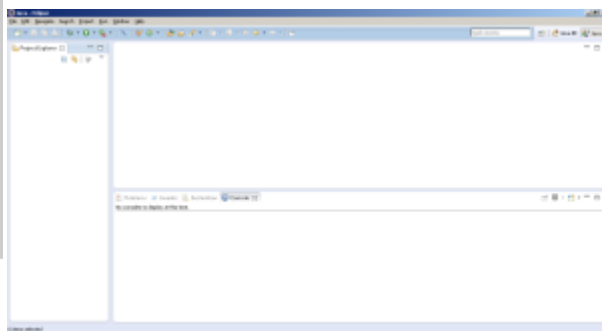
Eclipse dapat dijalankan tanpa harus diinstall. Cukup didownload, diekstrak, lalu tinggal dijalankan. Karena itu eclipse dapat dijalankan walau disimpan dalam flashdisk. Eclipse yang digunakan pada artikel ini adalah Eclipse Juno.

Di dalam direktori eclipse tersebut terdapat sebuah file bernama eclipse.exe. Untuk menjalankan eclipse, jalankan file eclipse.exe tersebut. Bila perlu, buat shortcut pada desktop untuk file eclipse.exe ini.

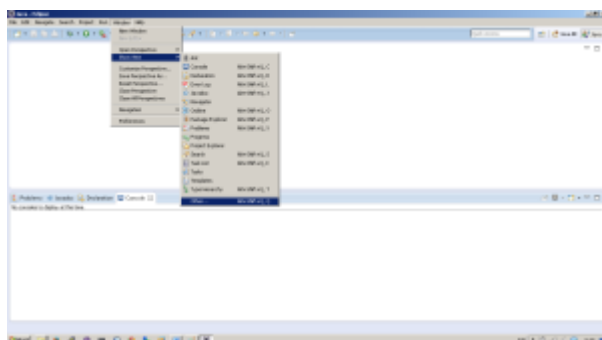
Ketika menjalankan eclipse pertama kali, eclipse akan meminta kita untuk memilih workspace. Pilihlah C:\TUTORIAL sebagai workspacenya. Bila Anda kehendaki, centang pada bagian "Use this as the default and do not ask again" untuk membuat C:\TUTORIAL sebagai default workspace. Dengan pencentangan ini eclipse tidak akan bertanya tentang workspace ketika dinyalakan kembali.



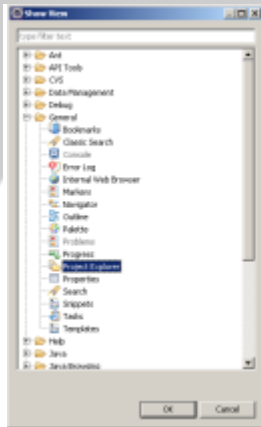
Setelah eclipse menyala, pada bagian kiri terdapat panel "project explorer" (lihat pada gambar).



Namun bila panel "project explorer" tidak ada, panel ini dapat ditampilkan dengan cara mengklik Window -> Show View -> Other. Setelah itu akan muncul window "Show View".



Pada window "Show View" ini pilih General -> Project Explorer. Kemudian klik "OK".

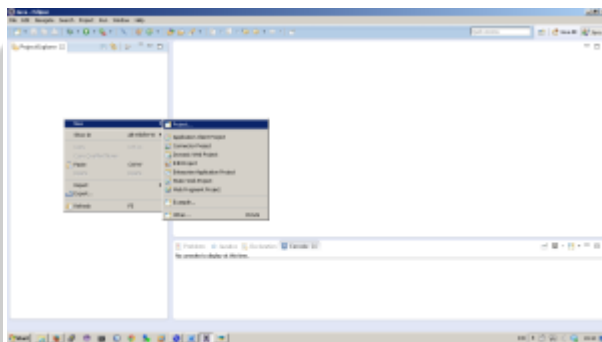


Hasilnya adalah panel "Project Explorer" akan muncul dalam eclipse di sebelah kiri.

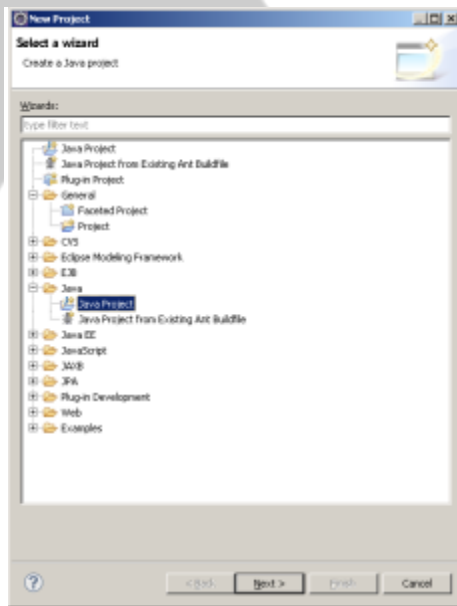
Membuat Project Baru

Dalam eclipse, setiap pembuatan program Java harus dilakukan dalam sebuah project. Untuk itu, hal pertama yang harus dilakukan adalah membuat project dalam eclipse.

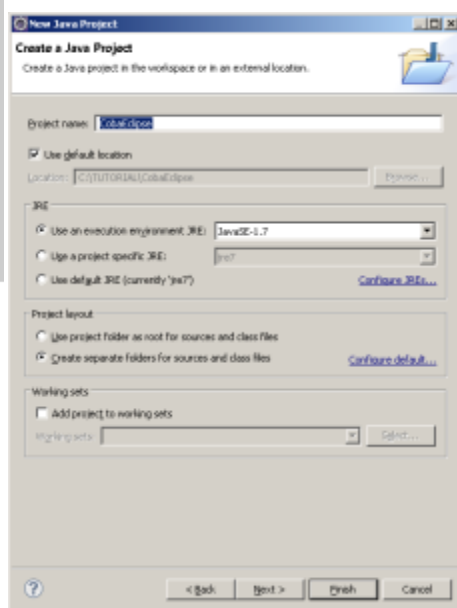
Untuk membuat project baru, lakukan klik kanan pada panel "Project Explorer". Kemudian pilih New -> Project.



Setelah muncul window "New Project", pilih Java -> Java Project. Lalu tekan tombol "Next".



Pada window "New Java Project", isikan nama project pada field "Project name:". Misalkan di sini kita memberi nama projectnya "CobaEclipse". Perhatikan juga pada bagian checkbox "Use default location". Pada bagian bawah checkbox tersebut ditunjukkan lokasi tempat menyimpan file-file project kita. Pada gambar terlihat project ini disimpan pada lokasi default yaitu: C:\TUTORIAL\CobaEclipse. Kemudian klik "Next".



Pada window "New Java Project" berikutnya klik saja "Finish".



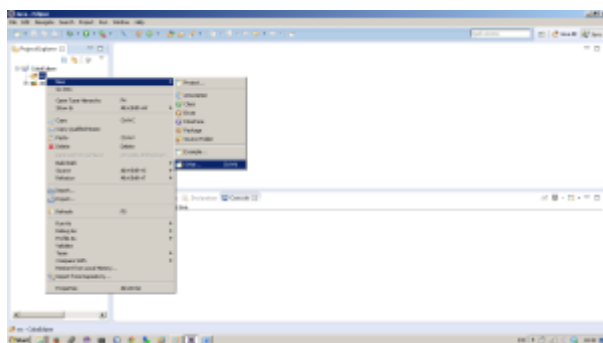
Kini pada panel "Project Explorer" di sebelah kiri muncul sebuah node bertuliskan "CobaEclipse". Node itulah project yang baru kita buat barusan. Di dalam node tersebut terdapat folder "src". Di dalam folder itulah nantinya kita akan menyimpan semua source code program java yang kita buat.

Bila Anda lihat menggunakan windows explorer, pada direktori C:\TUTORIAL akan muncul sebuah direktori baru dengan nama CobaEclipse. Setiap kali membuat project baru, Eclipse akan membuatkan sebuah folder pada file system di sistem operasi untuk menyimpan file-file project tersebut. Nama folder ini sama dengan nama project yang dibuat.

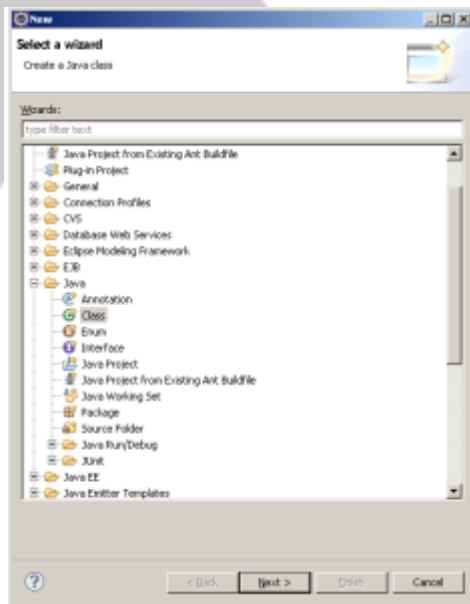
Untuk mencegah kehilangan data, direktori-direktori inilah yang perlu Anda backup bila Anda bekerja menggunakan Eclipse. Di dalam direktori project inilah tersimpan semua source code program yang kita buat.

Membuat File Java Class

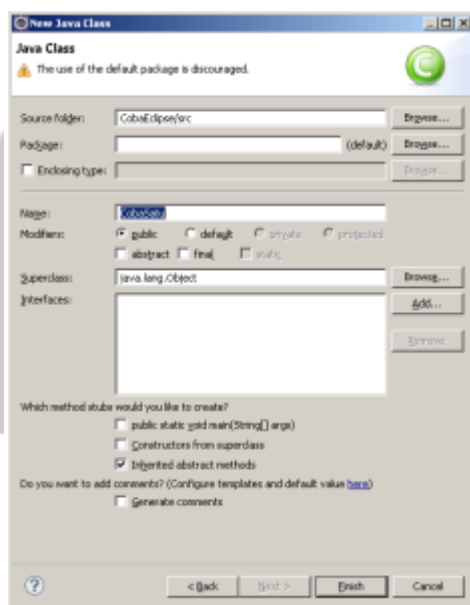
Untuk membuat file source code kelas Java, klik kanan pada folder "src". Kemudian pilih New -> Other.



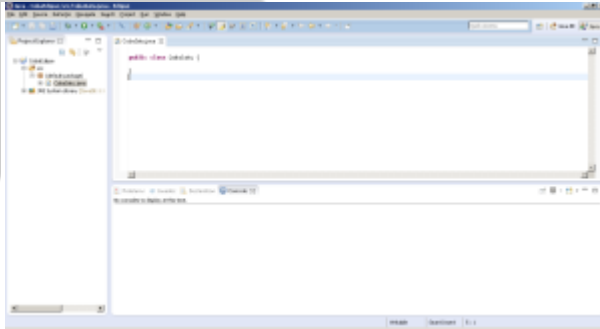
Pada window "New" pilih Java -> Class. Kemudian klik "Next".



Pada window "New Java Class" isikan field "Name: " dengan "CobaSatu". Ini artinya kita akan membuat sebuah class java dengan nama "CobaSatu". Kemudian klik tombol "Finish".



Setelah menekan tombol "Finish" kini pada folder "src" muncul file class yang barusan kita buat. Di sini ditunjukkan dengan nama CobaSatu.java. Pada panel editor, di sebelah kanan, ditampilkan source code class CobaSatu.java ini.



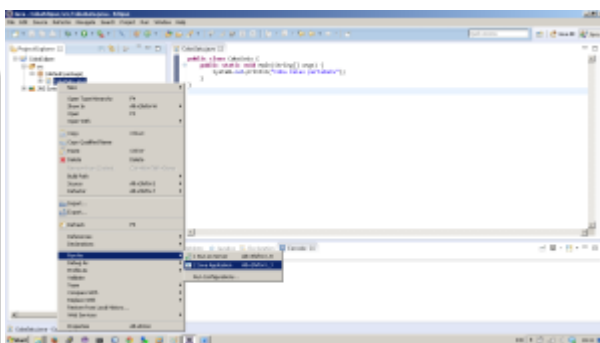
Ubah koding kelas CobaSatu ini menjadi seperti ini:

```

1 public class CobaSatu {
2     public static void main(String[] args) {
3         System.out.println("Coba kelas pertamaku");
4     }
5 }

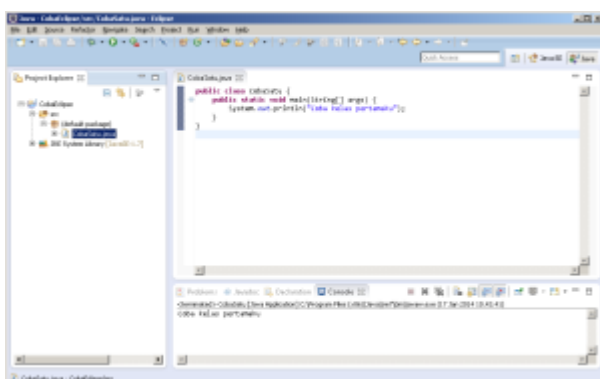
```

Kemudian simpan dengan cara File -> Save All. Setelah itu klik kanan pada node CobaSatu.java di panel Project Explorer. Kemudian pilih Run As -> Java Application. Proses ini akan menjalankan/mengeksekusi source code program yang baru kita buat barusan.



Hasil eksekusi program ini ditampilkan pada panel Console yang ada di bagian bawah Editor eclipse. Output yang dihasilkan adalah tulisan pada console:

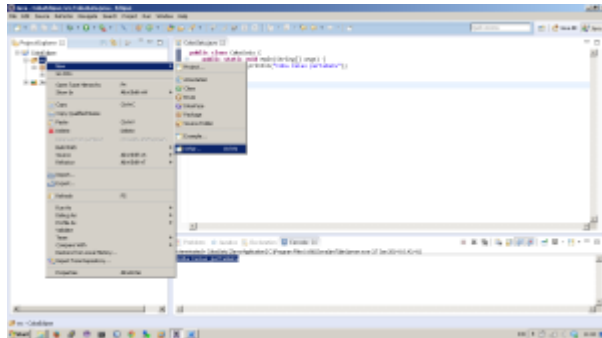
Coba kelas pertamaku



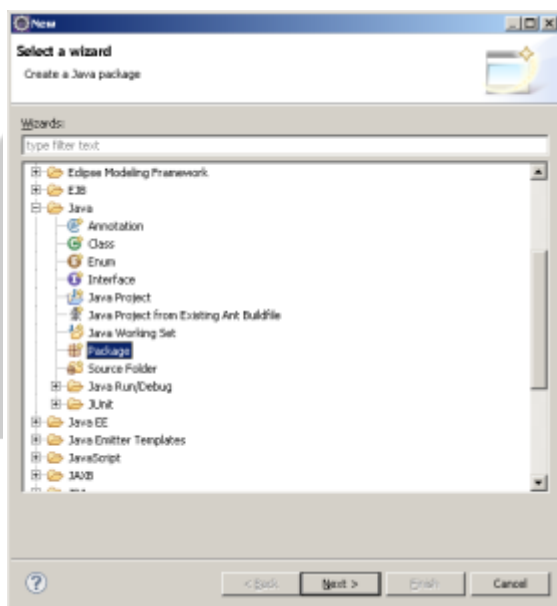
Sampai di sini kita sudah selesai membuat sebuah project pada eclipse sekaligus membuat sebuah kelas dan menjalankan kelas tersebut pada eclipse. Kira-kira seperti inilah gambaran umum cara membuat program menggunakan eclipse.

Membuat Package

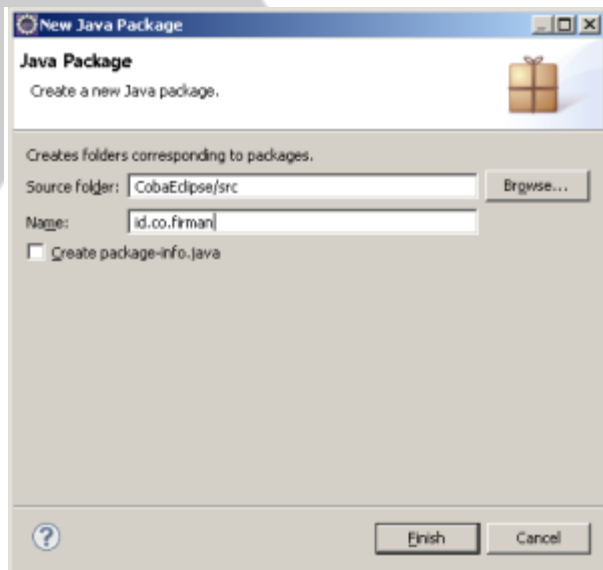
Pada panel Project Explorer, klik kanan pada node "src" kemudian pilih New -> Other.



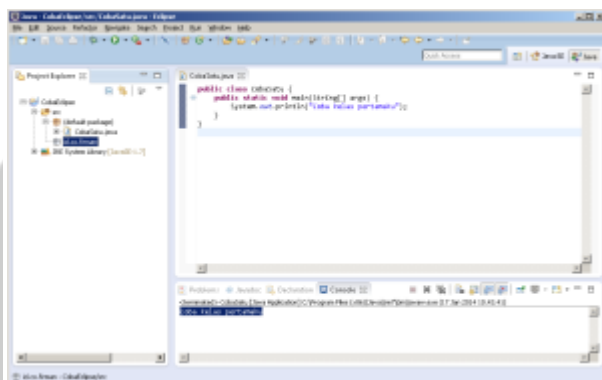
Pada window "New" pilih Java -> Package. Kemudian klik "Next".



Pada window "New Java Package" isikan nama package dengan nama "id.co.firman" pada field "Name:". Kemudian klik tombol Finish.



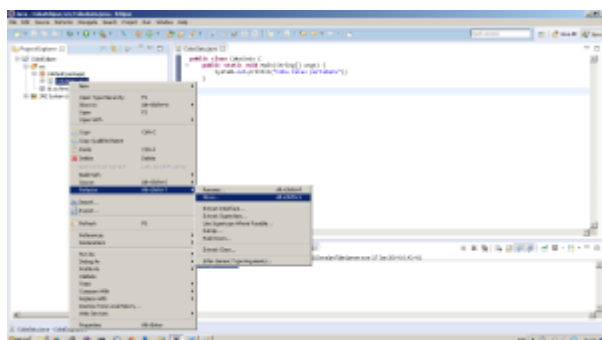
Kini coba perhatikan pada panel Project Explorer. Di dalam node "src" kini bertambah package yang barusan kita buat yaitu node "id.co.firman".



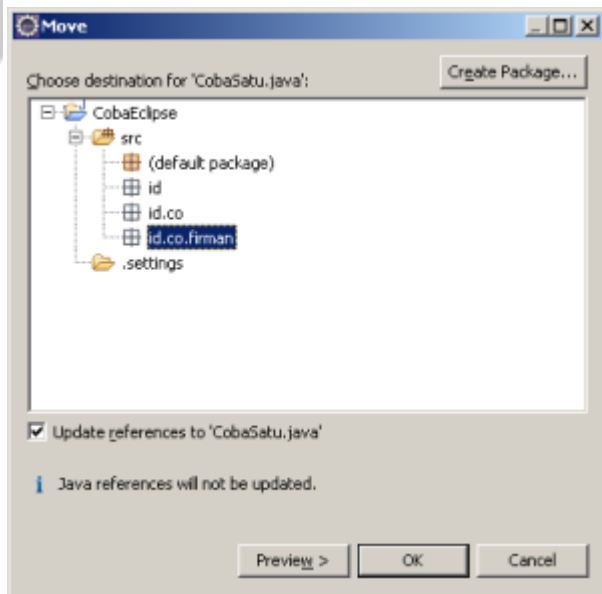
Memindahkan File Class Ke Package Lain

File CobaSatu.java masih berada di luar package id.co.firman. Dalam hal ini dalam eclipse terlihat file CobaSatu.java berada pada default package. Kita akan memindahkan file CobaSatu.java ini ke dalam id.co.firman.

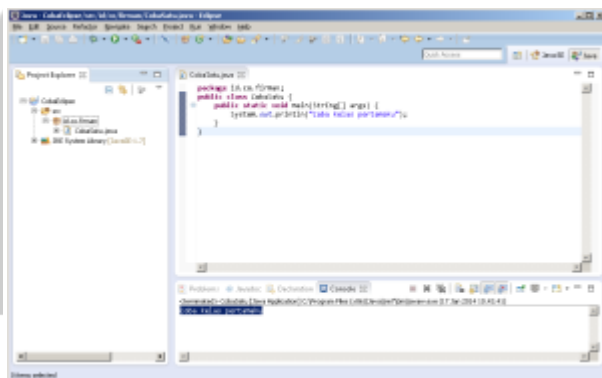
Pada panel Project Explorer, klik kanan pada class CobaSatu.java. Kemudian pilih Refactor -> Move.

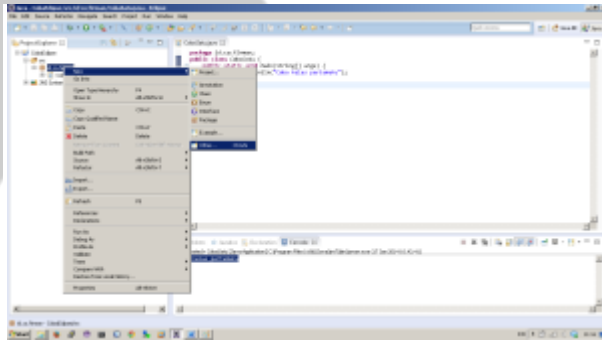


Pada window "Move" pilih package tujuan pemindahan file. Dalam hal ini kita pilih "id.co.firman". Kemudian klik "OK".

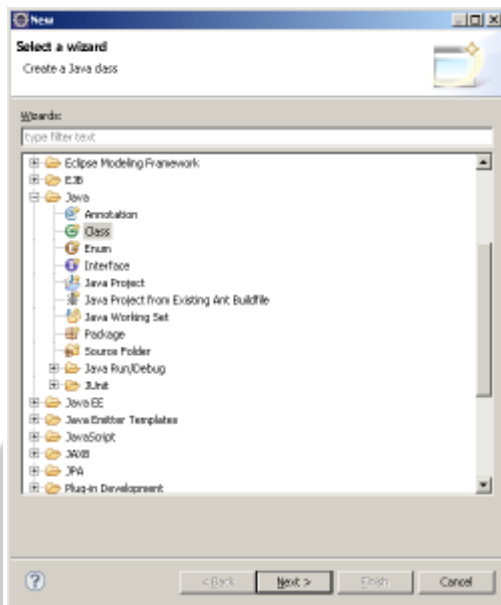


Coba perhatikan di panel Project Explorer. Kini file kelas CobaSatu.java sudah berada di dalam package id.co.firman.

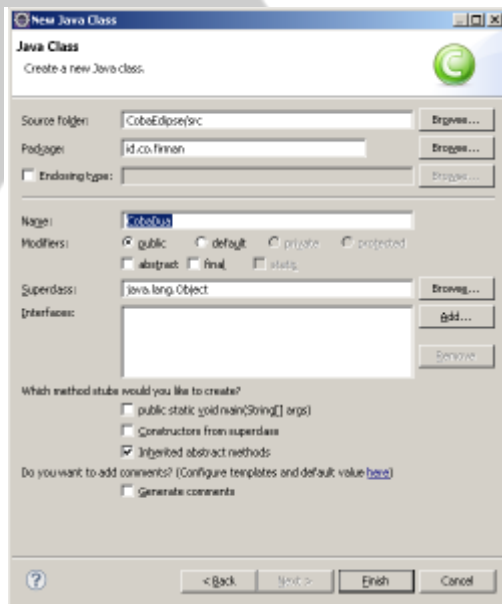




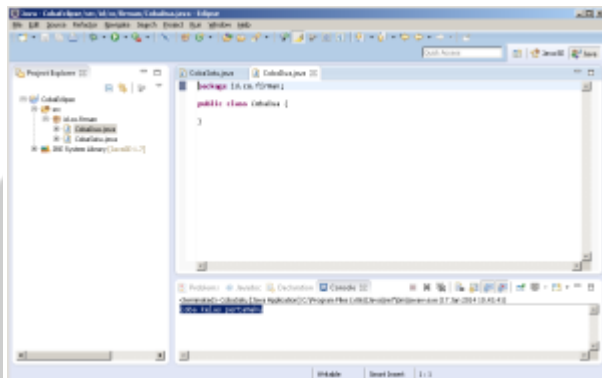
Pada window "New" pilih Java -> Class. Kemudian klik "Next".



Pada window "New Java Class", beri nama kelas yang baru ini dengan nama "CobaDua" dengan mengisi field "Name:" dengan nilai "CobaDua". Coba perhatikan field "Package:" tertulis nilai "id.co.firman". Ini artinya kelas yang kita buat ini akan berada di dalam package id.co.firman. Kemudian klik "Finish".



Pada panel Project Explorer, kini file kelas CobaDua.java berada dalam package id.co.firman.



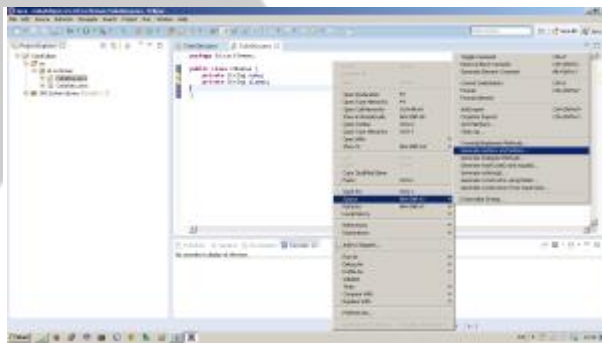
Membuat Getter dan Setter pada Eclipse

Seperti pernah diutarakan pada artikel sebelumnya ([Getter dan Setter pada Kelas Java](#)), sebaiknya atribut sebuah kelas menggunakan akses level selain public. Untuk itu, di sini kita akan membuat method getter dan setter dengan bantuan eclipse.

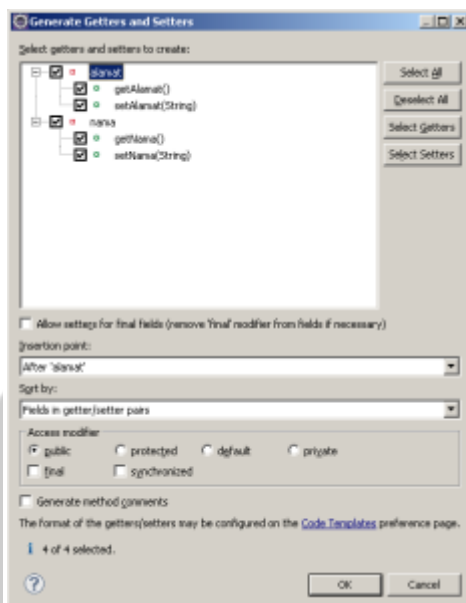
Pada kelas CobaDua, buat dua buah atribut String dengan level akses private. Kedua atribut ini adalah nama dan alamat.

```
1 private String nama;
2 private String alamat;
```

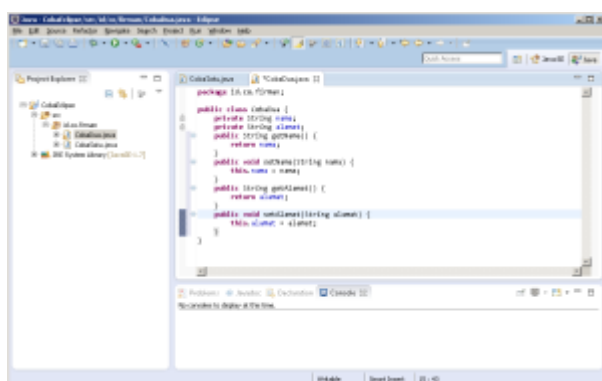
Kemudian, tempatkan kursor di bawah kedua atribut tersebut. Lalu klik kanan pada daerah tersebut. Pilih Source -> Generate Getters and Setters...



Pada window "Generate Getters and Setters", beri centang pada checkbox atribut alamat dan nama seperti pada gambar. Lalu tekan "OK".



Coba perhatikan pada source code kini sudah ditambahkan method getter dan setter untuk atribut nama dan alamat. Simpan semua perubahan ini dengan memilih File -> Save All.



Secara lengkap kode pada kelas CobaDua menjadi seperti ini:

```
1 package id.co.firman;
2
3 public class CobaDua {
4     private String nama;
5     private String alamat;
```

```

5      public String getNama() {
6          return nama;
7      }
8      public void setNama(String nama) {
9          this.nama = nama;
10     }
11     public String getAlamat() {
12         return alamat;
13     }
14     public void setAlamat(String alamat) {
15         this.alamat = alamat;
16     }
17 }
18

```

Dengan cara ini kita tidak perlu menuliskan method getter dan setter secara manual. Eclipse sudah cukup pintar untuk membuatkan getter dan setter ini. Kita cukup memodifikasi method-method ini bila perlu.

Memanggil Method Getter dan Setter

Sekarang Anda coba sendiri membuat koding untuk memanggil method getter dan setter pada kelas CobaDua dari kelas CobaSatu. Bila Anda belum tahu caranya, coba baca artikel [Getter dan Setter pada Kelas Java](#) dan [Tutorial Object Oriented Programming dengan Java Bagian 2](#). Anda perlu menguasai teknik komunikasi antarobjek yaitu teknik "instansiasi -> panggil resource" seperti pada tutorial bagian 2 tersebut.



Nama:Dicka Ananda Abdi Negara

Sekarang saya bersekolah di SMKN 13 kota bandung,saya mengambil jurusan RPL (rekayasa perangkat lunak) atau software engineering.Saya cukup tertarik dan berminat dibagian Game Developer,Saya sekarang sedang mempelajari Pembuatan Game 3D,Saya juga lumayan mengerti dalam hal yang berkaitan dengan software(ngoding)