

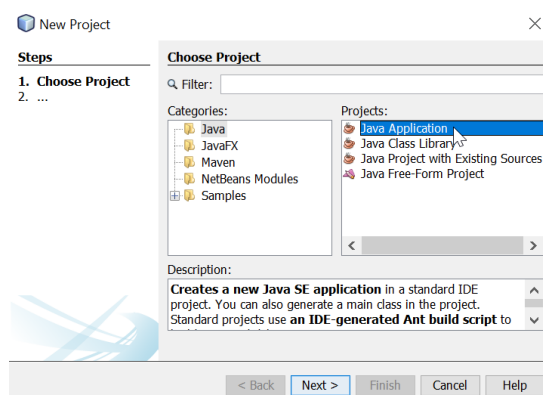
Lembar Praktikum 01

NIM : K3520025
Nama Mahasiswa : Fahira Zahra

- Mata Pelajaran : Pemrograman Berorientasi Obyek
 - Pokok Bahasan : Pendahuluan PBO
 - Tujuan Pembelajaran :
 - Melalui serangkaian langkah praktikum dengan Netbeans, peserta didik mampu membuat package dengan benar
 - Melalui serangkaian langkah praktikum dengan Netbeans, peserta didik mampu membuat class dan main-class Java dengan benar
 - Melalui serangkaian langkah praktikum dengan Netbeans, peserta didik mampu mendefinisikan atribut dari sebuah class dengan benar
 - Melalui serangkaian langkah praktikum dengan Netbeans, peserta didik mampu membuat method sederhana di dalam sebuah class dengan benar
 - Melalui serangkaian langkah praktikum dengan Netbeans, peserta didik mampu membuat obyek dengan benar dari class yang telah didefinisikan
 - Melalui serangkaian langkah praktikum dengan Netbeans, peserta didik mampu menjalankan project Java dengan benar
-

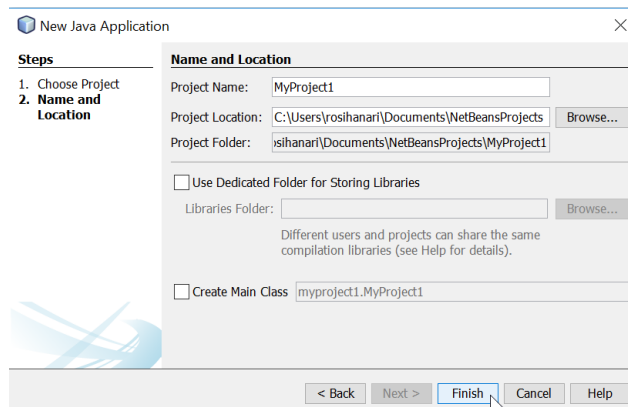
Langkah Membuat Project Java Baru di Netbeans

1. Buka Netbeans
2. Klik File - New Project
3. Pilih Categories -> Java, Projects -> Java Application

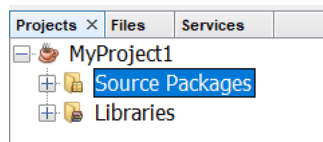


4. Klik Next
5. Isikan 'MyProject1' pada Project Name

6. Tentukan lokasi penyimpanan project
7. Hilangkan tanda centang pada 'Create Main Class'



8. Klik Finish
9. Selanjutnya akan muncul struktur project di bagian sidebar sebelah kiri



Kegiatan:

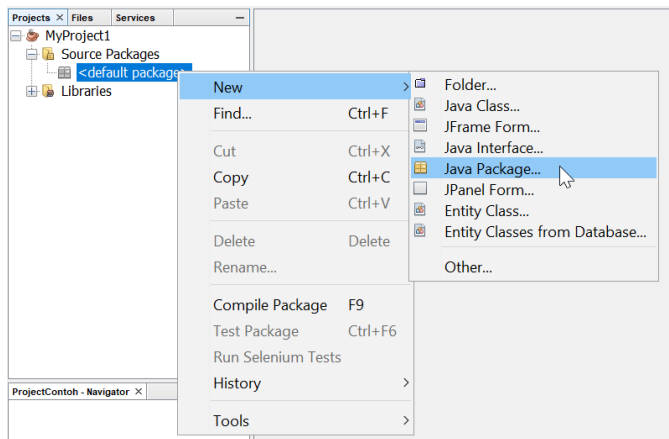
1. Lakukan langkah di atas!
2. Kemudian amati struktur direktori dari project yang sudah dibuat di atas, melalui Windows Explorer! Apa kesimpulan Anda terhadap struktur direktorinya?

Struktur direktori dari project Java di Netbeans ini hampir sama cara kerjanya dengan bahasa pemrograman PHP yang harus dijalankan dengan XAMPP. File yang dibuat akan diarahkan secara otomatis untuk disimpan pada folder dari aplikasi yang digunakan, walaupun pada akhirnya akan tetap bisa diatur sendiri ingin disimpan dimana sesuai keinginan. Kesimpulan mengenai struktur direktori Netbeans ini menurut saya cukup praktis, Netbeans langsung mengarahkan kemana file seharusnya disimpan sehingga jika para pengguna semisalnya lupa telah menyimpan file dimana, maka pengguna itu sendiri akan segera dengan mudahnya menemukan file tersebut karena cukup dengan membuka folder aplikasi dimana file project tersebut sebelumnya dibuat.

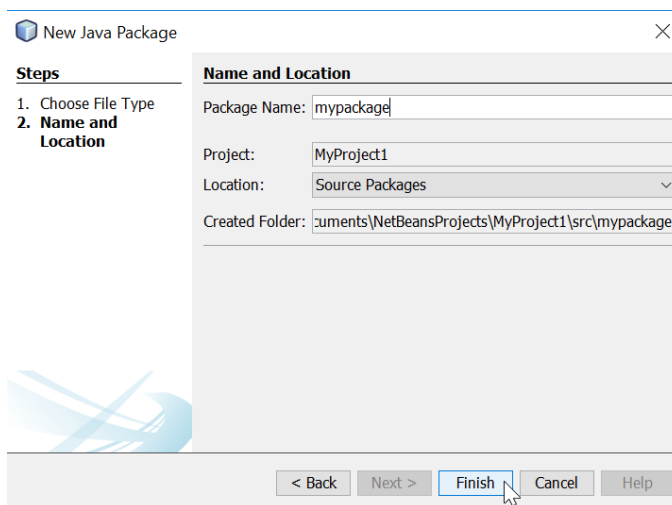
Langkah Membuat Package

Sebuah package di dalam project, sama seperti folder. Di dalam sebuah package dapat diisi oleh beberapa class. Serta di dalam sebuah project bisa terdiri dari beberapa package.

1. Klik Source Packages
2. Klik kanan pada Default Package – New – Java Package



3. Beri nama package 'mypackage' pada 'Package Name'



4. Klik Finish

Kegiatan:

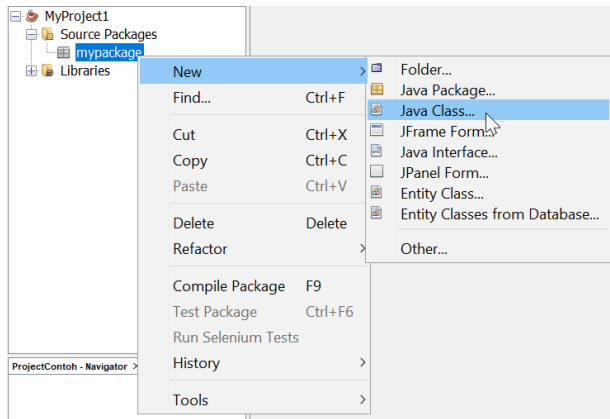
1. Lakukan langkah untuk membuat package di atas
2. Amati kembali efek pembuatan package tersebut di dalam direktori projectnya melalui Windows Explorer! Apa yang terjadi terhadap struktur direktori project setelah pembuatan package?

Dari percobaan pembuatan package diatas, tidak banyak menyebabkan perubahan pada folder project yang sebelumnya telah dibuat. Setelah pembuatan folder mypackage hanya terjadi penambahan folder mypackage tersebut yang telah dibuat pada percobaan diatas pada **MyProject1 > src > mypackage**.

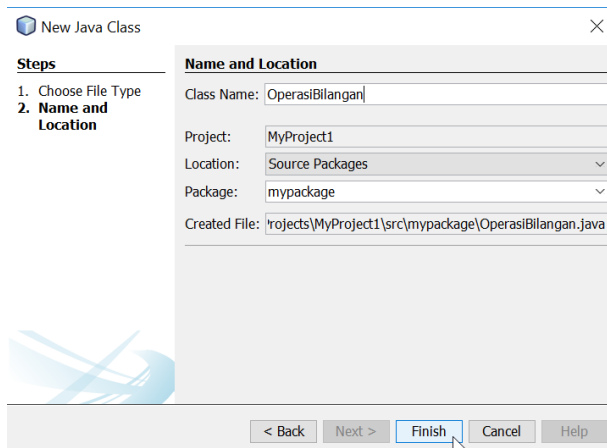
Langkah Membuat Class

Setelah package dibuat, barulah bisa dibuat class-class di dalamnya

1. Klik kanan pada package yang sudah dibuat
2. Klik New – Java Class



3. Beri nama 'OperasiBilangan' pada Class Name



4. Klik Finish

Kegiatan:

1. Lakukan langkah membuat class di atas
2. Amati perubahan tampilan di Netbeans setelah sebuah class dibuat! Tampilan apa yang berubah?

Perubahan yang tampil pada Netbeans setelah file OperasiBilangan adalah berhasil munculnya file java OperasiBilangan tersebut yang telah dibuat pada percobaan diatas.

3. Amati perubahan struktur direktori di project melalui Windows Explorer! Perubahan apakah yang terjadi setelah sebuah class dibuat?

Struktur dari direktori MyProject1 tidak terdapatnya perubahan setelah file java OperasiBilangan dibuat. Hanya terjadinya penambahan file OperasiBilangan tersebut pada **MyProject1 > src > mypackage > OperasiBilangan.java**.

4. Menurut Anda, kira-kira apa kegunaan dari baris perintah
`package mypackage;`
 yang terletak di atas class 'OperasiBilangan'?

Kegunaan dari baris perintah tersebut adalah untuk menandakan bahwa file OperasiBilangan ini adalah file java yang tersimpan dan terletak di package yang bernama mypackage.

Menambahkan Atribut dan Method di dalam Class

Setelah sebuah class dibuat, atribut dan method dapat ditambahkan ke dalamnya.

Kegiatan:

1. Tambahkan kode berikut ini di dalam class 'OperasiBilangan'

```
public class OperasiBilangan {  
  
    // atribut  
    public int bilPertama;  
    public int bilKedua;  
  
    // methods  
    public void hitungPenjumlahan(){  
        int hasil = bilPertama + bilKedua;  
        System.out.println("Hasil penjumlahannya: " + hasil);  
    }  
  
    public void hitungPengurangan(){  
        int hasil = bilPertama - bilKedua;  
        System.out.println("Hasil pengurangannya: " + hasil);  
    }  
  
}
```

2. Amati cara penulisan atribut dan method di atas. Apa perbedaan mencolok cara pendefinisian keduanya dalam hal penulisan nama dan strukturnya ?

Yang membedakan dari penulisan atribut dan methods diatas adalah mungkin pada atribut yang digunakan untuk penulisan variabel yang akan digunakan, sedangkan untuk methods digunakan sebagai cara pengoperasian dari penjumlahan dan pengurangan bilangan yang akan dioperasikan.

Membuat Obyek (Instansiasi)

Setelah sebuah class selesai didefinisikan, selanjutnya dapat dilakukan proses instansiasi yaitu proses pembuatan obyek dari class tersebut.

Kegiatan:

1. Buatlah class baru 'MyClass' di dalam package 'mypackage' pada project 'MyProject1'
2. Tuliskan kode program berikut ini

```

public class MyClass {
    public static void main(String[] args) {
        OperasiBilangan op1 = new OperasiBilangan();
        op1.bilPertama = 10;
        op1.bilKedua = 20;
        op1.hitungPenjumlahan();
        op1.hitungPengurangan();
    }
}

```

Keterangan:

Sebuah class yang di dalamnya terdapat method **main()** dinamakan **main class**.

3. Jalankan project 'MyProject1' dengan mengklik Run – Run Project
4. Amati tampilan outputnya. Apa maksud dari program Java yang baru saja dibuat?

Maksud dari program Java yang telah dibuat adalah, file MyClass yang berperan sebagai main class menampilkan variabel yang menunjukkan besarnya suatu bilangan yang akan dioperasikan. Sedangkan pada file OperasiBilangan adalah berisi pemrograman yang menampilkan cara pengoperasian dari bilangan yang akan dijumlahkan dan dikurangkan. Lalu setelah keduanya dibuat, maka project yang berisikan dua file class tersebut dijalankan (run) lalu keluarlah hasil pengoperasian dari dua bilangan tersebut.

5. Dalam class 'MyClass', buatlah obyek baru dengan nama 'op2'. Lalu gunakan obyek 'op2' ini untuk mencari hasil penjumlahan serta pengurangan dari 25 dan 55.
6. Buatlah class baru dengan nama 'MyNewClass' di dalam package 'mypackage' dengan isi class sebagai berikut.

```

public class MyNewClass {
    public static void main(String[] args) {
        OperasiBilangan op1 = new OperasiBilangan();
        op1.bilPertama = 50;
        op1.bilKedua = 60;
        op1.hitungPenjumlahan();
        op1.hitungPengurangan();
    }
}

```

7. Jalankan project 'MyProject1' dengan mengklik Run – Run Project! Amati apa yang terjadi?

Tidak ada yang berubah dari MyProject1 saat dijalankan setelah dibuatnya file MyNewClass. Yang ditampilkan saat dijalankan adalah masih hasil dari pengoperasian bilangan yang dicantumkan pada file java MyClass dan OperasiBilangan.

8. Bagaimana supaya perintah yang ada di dalam main() method MyNewClass ini dapat dijalankan?

Karena yang disebutkan adalah `main()` method, berarti menandakan bahwa file `MyNewClass` adalah main class. Oleh karena itu agar perintah yang terdapat di file `MyNewClass` bisa dijalankan, maka perintah tersebut harus dipindahkan ke file class `MyClass` yang sudah lebih dulu menjadi main class. Hal tersebut dikarenakan bahwa di dalam bahasa pemrograman Java OOP sebuah project bisa memiliki banyak class, namun hanya 1 file class saja yang bisa menjadi main class.

9. Apa kesimpulan Anda tentang main class dalam sebuah project?

Main class adalah sebuah file yang mempunyai fungsi utama dalam menjalankan sebuah objek, pada main class biasanya berisikan objek yang akan dijalankan tersebut. Seperti pada contoh diatas, pada main class berisikan nilai-nilai dari bilangan-bilangan yang akan dieksekusi atau dioperasikan berdasarkan perintah yang terdapat di file `OperasiBilangan`. Perlu diingat bahwa pada Java OOP, di dalam suatu package boleh terdapat dari beberapa class, namun yang boleh menjadi main class hanya satu file class saja.

10. Tukarkan posisi urutan penulisan atribut dan method yang ada di class '`OperasiBilangan`'. Kemudian jalankan project. Apakah ada efeknya terhadap output?

Tidak terjadi perubahan pada output saat atribut dan method dibalik posisinya pada file class `OperasiBilangan`.

Project 1

Sampai tahap ini, Anda sudah bisa membuat project baru, membuat package, class, kemudian mendefinisikan atribut, method, serta membuat obyek, dan juga memahami kegunaan main-class. Selanjutnya kerjakan project berikut ini

1. Dalam class '`OperasiBilangan`', tambahkan method baru `hitungPerkalian()` untuk menghitung hasil perkalian kedua bilangan, serta method `hitungPembagian()` untuk menghitung hasil pembagiannya.
Petunjuk: di method `hitungPembagian()` gunakan tipe data `double`
2. Modifikasilah project '`MyProject`' sedemikian hingga bisa menampilkan hasil penjumlahan, pengurangan, perkalian, dan pembagian dari dua bilangan yaitu 100 dan 150.

Project 2

1. Buatlah package '`bangundatar`' di dalam project '`MyProject1`'
2. Buatlah class '`PersegiPanjang`' di dalam package '`bangundatar`'.
3. Tambahkan atribut '`panjang`' dan '`lebar`' di dalam class '`PersegiPanjang`', keduanya bertipe data `integer`.
4. Tambahkan method `hitungLuas()` dan `hitungKeliling()` yang masing-masing digunakan untuk menghitung luas dan keliling persegi panjang
5. Buatlah main class di dalam package '`bangundatar`' dengan nama '`BangunDatarMain`'

6. Melalui main class 'BangunDatarMain', gunakan class 'PersegiPanjang' yang telah dibuat untuk menghitung luas dan keliling persegi panjang jika diketahui panjangnya 25 dan lebarnya 38 satuan panjang.
7. Buatlah class 'Persegi', dan 'Lingkaran' di dalam package 'bangundatar'. Rancang dan definisikan atribut yang sesuai untuk masing-masing class, serta tuliskan method hitungLuas() dan hitungKeliling() nya.
8. Selanjutnya gunakan class 'Persegi' untuk menghitung luas dan keliling dari beberapa persegi berikut ini:
 - a. Sebuah persegi A jika diketahui sisi perseginya 10 satuan panjang
 - b. Sebuah persegi B, jika diketahui sisi perseginya 15 satuan panjang
9. Selanjutnya gunakan class 'Lingkaran' untuk menghitung luas dan keliling dari beberapa lingkaran berikut ini:
 - a. Sebuah lingkaran X jika diketahui jari-jarinya 25 satuan panjang
 - b. Sebuah lingkaran Z, jika diketahui jari-jarinya 37 satuan panjang

Project 3 (Pengayaan)

1. Buat package dengan nama 'bangunruang'
2. Buat 3 class di dalam package tersebut, masing-masing bernama 'Bola', 'Tabung', 'Kerucut'.
3. Untuk tiap-tiap class, definisikan atributnya
4. Untuk tiap-tiap class, buatlah method hitungVol() untuk menghitung volume, serta hitungLuasSelimut() untuk menghitung luas selimut benda ruang

Pengumpulan Project

- Buatlah repository di Github untuk mempush project hasil project 1, 2, dan 3
- Copy pastekan URL Reponya di bawah ini

URL Repo:

- Project 1
<https://github.com/fahirazahra/Java-Projects-PBO/tree/main/MyProject1/src/mypackage>
- Project 2
<https://github.com/fahirazahra/Java-Projects-PBO/tree/main/MyProject1/src/bangundatar>
- Project 3
<https://github.com/fahirazahra/Java-Projects-PBO/tree/main/MyProject1/src/bangunruang>