

VERSION 1.0

JULY, 2022



# PEMROGRAMAN LANJUT

## MODUL 4 - MODERN PROGRAMMING ENVIRONMENT

TIM PENYUSUN :

- WILDAN SUHARSO, S.KOM., M.KOM
- HANIA PRATIWI NINGRUM
- FILDZAH LATHIFAH

PRESENTED BY : LAB. TEKNIK INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## PEMROGRAMAN LANJUT

---

### CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu memahami IDE.
2. Mahasiswa mampu mengetahui fitur-fitur IDE.
3. Mahasiswa mampu melakukan/mengimplementasikan salah satu fitur IDE.

---

### KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/PC
- Netbeans/IntelliJ/Eclipse

---

### MATERI POKOK

#### 1. IDE

Pada tingkat yang lebih mendasar, IDE menyediakan antarmuka bagi pengguna untuk menulis kode, mengatur grup teks, dan mengotomatiskan redundansi pemrograman. IDE menggabungkan fungsi dari beberapa proses pemrograman menjadi satu. Beberapa IDE fokus pada bahasa pemrograman tertentu, tetapi banyak yang memiliki kemampuan lintas bahasa. Dalam hal kemampuan mengedit teks, IDE sering memiliki atau mengizinkan penyisipan kerangka kerja dan pustaka elemen untuk dibangun di atas kode tingkat dasar.

#### 2. Manfaat IDE

Ada sejumlah alasan untuk menggunakan IDE, yang sebagian besar berkisar pada pengembangan perangkat lunak. Platform ini memusatkan tiga alat utama yang digunakan oleh sebagian besar pengembang, yaitu : editor source code, debugger, dan compiler. Ini memungkinkan pengguna untuk menulis, menyempurnakan, dan memproses kode dalam satu lingkungan.

#### 3. Fitur Umum IDE a. Editor text

Hampir setiap IDE akan memiliki editor teks yang dirancang untuk menulis dan memanipulasi kode sumber. Beberapa alat mungkin memiliki komponen visual untuk menarik dan melepaskan komponen, tetapi sebagian besar memiliki antarmuka sederhana dengan penyorotan sintaksis khusus bahasa.

##### b. Debugger

Alat debugging membantu pengguna dalam mengidentifikasi dan memperbaiki kesalahan dalam kode sumber. Mereka sering mensimulasikan skenario dunia nyata untuk menguji fungsionalitas dan kinerja. Pemrogram dan insinyur perangkat lunak biasanya dapat menguji berbagai segmen kode dan mengidentifikasi kesalahan sebelum aplikasi dirilis.

**c. Penyusun**

Compiler adalah komponen yang menerjemahkan bahasa pemrograman ke dalam bentuk yang dapat diproses oleh mesin, seperti kode biner. Kode mesin dianalisis untuk memastikan akurasi. Compiler kemudian *memparsing* dan mengoptimalkan kode untuk mengoptimalkan kinerja.

**d. Pelengkapan kode**

Fitur kode lengkap membantu pemrogram dengan mengidentifikasi dan memasukkan komponen kode umum secara cerdas. Fitur-fitur ini menghemat waktu pengembang menulis kode dan mengurangi kemungkinan kesalahan ketik dan bug.

**e. Memprogram dukungan bahasa**

IDE biasanya khusus untuk satu bahasa pemrograman, meskipun beberapa juga menawarkan dukungan multi-bahasa. Dengan demikian, langkah pertama adalah mencari tahu bahasa yang akan kamu coding dan persempit daftar IDE prospektif kamu. Contohnya termasuk alat Ruby, Python, dan Java IDE.

**f. Integrasi dan plugin**

Dengan nama lingkungan pengembangan terintegrasi, tidak mengherankan bahwa integrasi perlu dipertimbangkan ketika melihat IDE. IDE kamu adalah portal pengembanganmu sendiri, sehingga dapat menggabungkan semua alat pengembanganmu yang lain akan meningkatkan alur kerja pengembangan dan produktivitas. Integrasi yang buruk dapat menyebabkan banyak masalah dan menyebabkan banyak sakit kepala, jadi pastikan kamu memahami seberapa baik IDE potensial cocok dengan ekosistem alat yang ada.

**g. Code search**

IDE dapat memberikan dukungan lanjutan untuk pencarian kode, antara lain : untuk menemukan deklarasi kelas dan fungsi, penggunaan, variabel dan bidang baca / tulis, dll. IDE dapat menggunakan berbagai jenis antarmuka pengguna untuk pencarian kode, misalnya form-based widgets dan natural-language based interfaces.

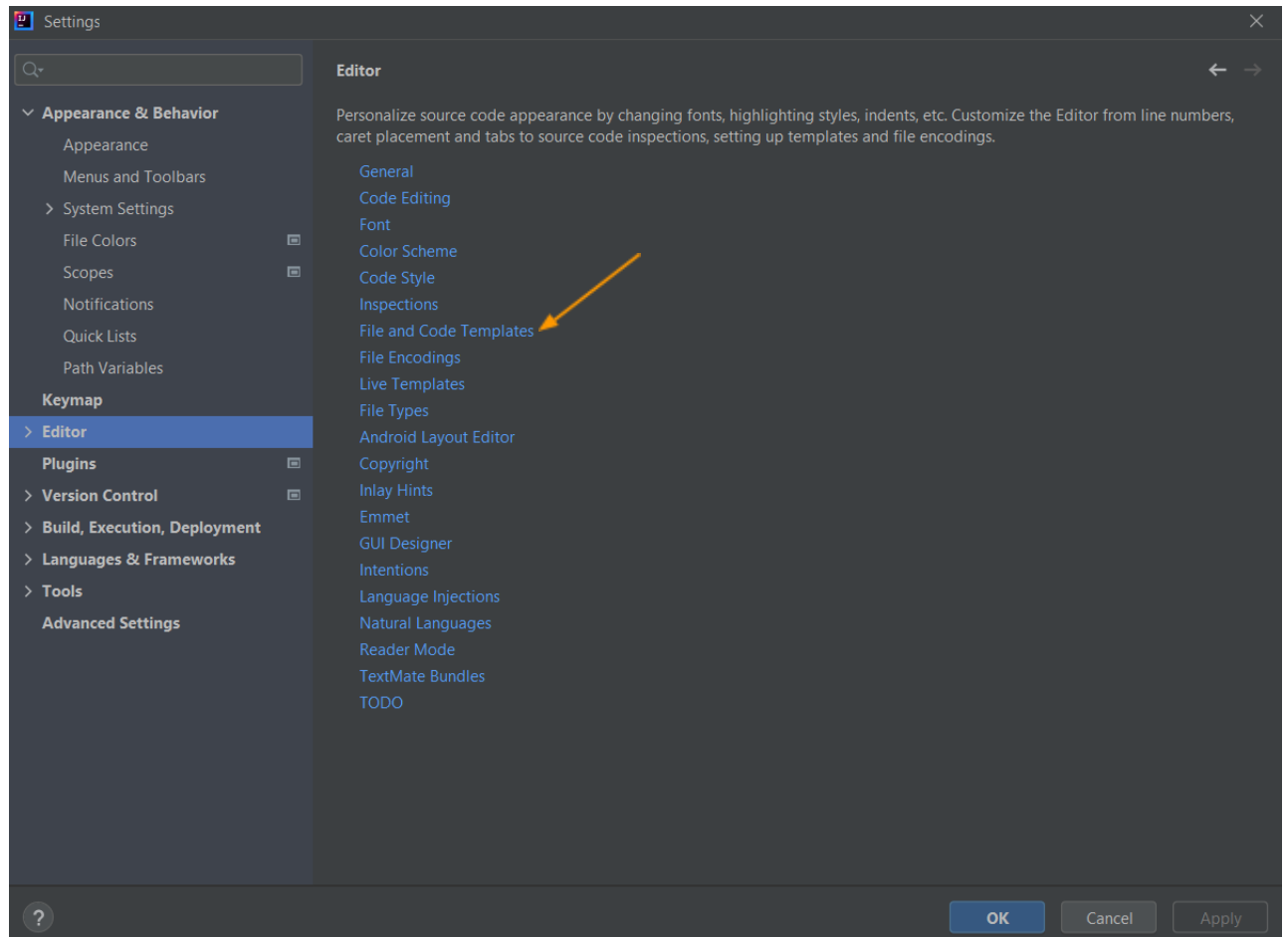
---

## MATERI PRAKTIKUM

Dalam pembuatan program java ada beberapa IDE (Integrated Development Editor) yang dapat digunakan untuk mengembangkan aplikasi java. Disini kita menggunakan IDE IntelliJ sebagai contohnya.

**1. Code Template**

Jika kamu ingin mencetak output 'System.out.println();' kamu bisa mengetikkan 'sout', kemudian klik tab pada keyboard. Maka secara otomatis IntelliJ akan mengubah 'sout' menjadi 'System.out.println();'. Atau jika kamu ingin setting sendiri, kamu bisa klik Ctrl+Alt+S atau dari File > Settings. Setelah itu akan muncul pop up berikut. Pilih Editor dan klik File and Code Templates. Jika kamu menggunakan IDE Netbeans, kamu bisa klik Tools > Option > Editor, kemudian klik tab Code Template.

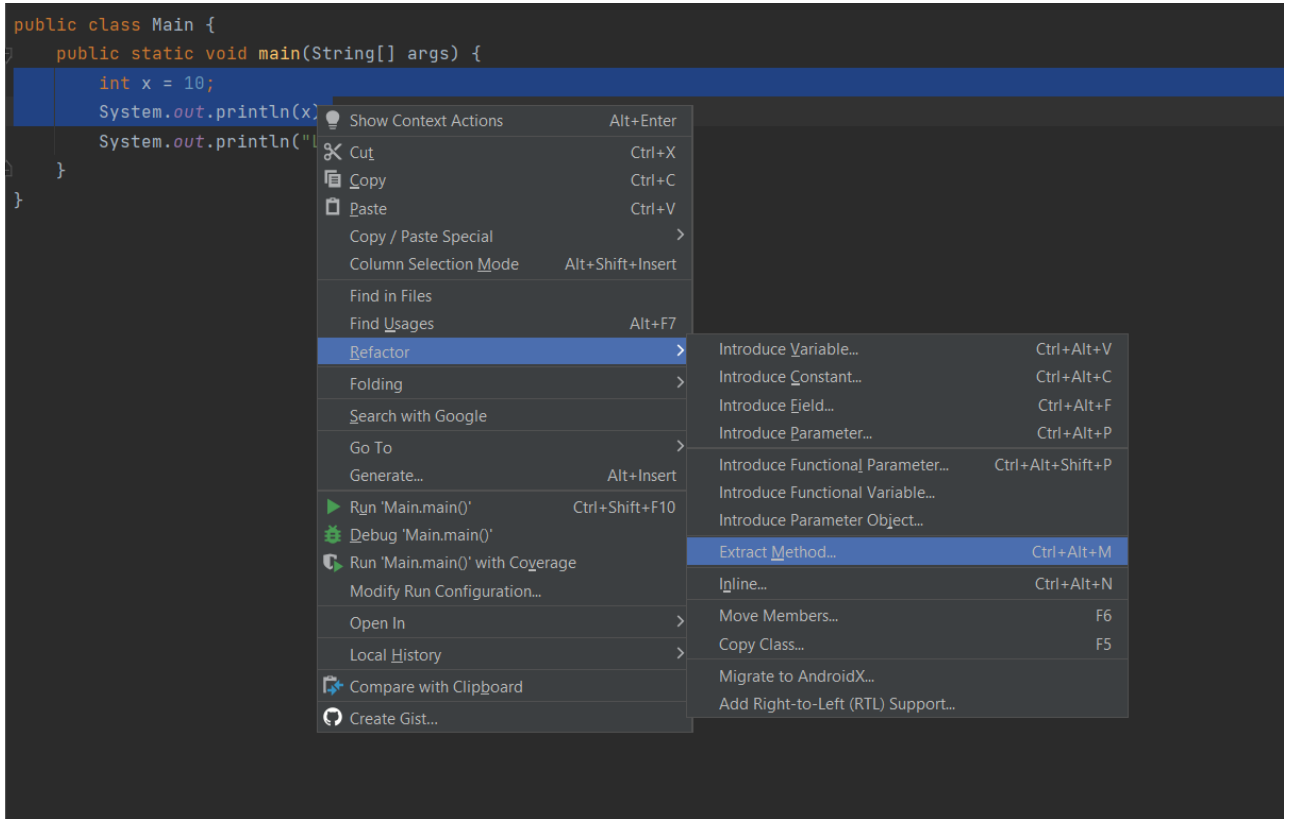


## 2. Mengganti Code Block Dengan Sebuah Method

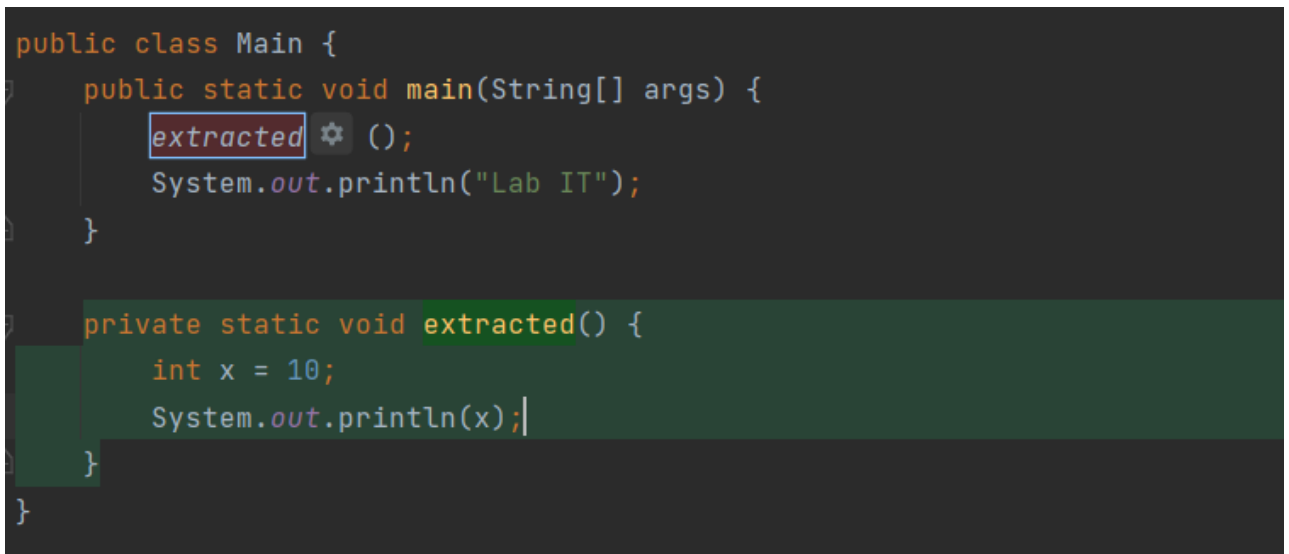
Cara ini mempermudah kita untuk memperjelas script yang kita buat. Misalkan kita memiliki code sebagai berikut.

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        System.out.println(x);
        System.out.println("Lab IT");
    }
}
```

Kemudian kita buat method dari baris `int x = 10` sampai `'System.out.println(x);'` dalam method, caranya adalah dengan block code yang ingin dijadikan 1 method lalu klik kanan > Refactor > Extract Method.



Kemudian source code akan menjadi seperti berikut.



'extracted' bisa kamu ganti dengan nama method yang kamu inginkan.

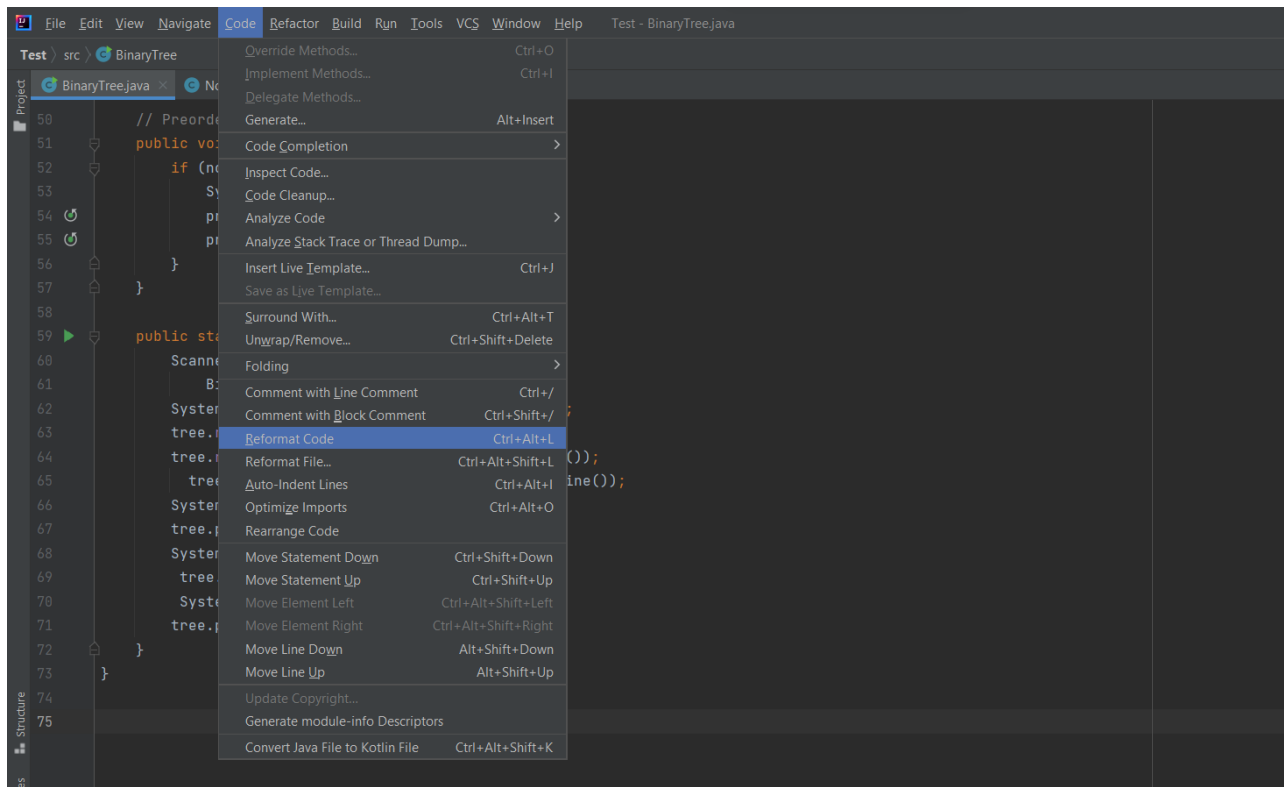
### 3. Merapikan Source Code

Pada IDE IntelliJ, kamu juga bisa merapikan source code yang kamu buat. Ini sangat berguna buat kamu yang memang suka ngoding tetapi pekerjaan kurang rapi. Kerapian dari

source code memang hal yang sepele, akan tetapi akan sangat memudahkan dalam pekerjaan kamu. Contohnya saja seperti source code berikut.

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    BinaryTree tree = new BinaryTree();
    System.out.println("Masukkan elemen tree : ");
    tree.root = new Node(input.nextLine());
    tree.root.left.left = new Node(input.nextLine());
    tree.root.left.right = new Node(input.nextLine());
    System.out.print("Pre order : ");
    tree.preOrder(tree.root);
    System.out.print("\nIn order : ");
    tree.inOrder(tree.root);
    System.out.print("\nPost order : ");
    tree.postOrder(tree.root);
}
```

Source code di atas adalah contoh source code yang tidak rapi, cara merapikan secara otomatis adalah dengan klik Code > Reformat Code.



Maka program akan menjadi lebih rapi seperti berikut.

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    BinaryTree tree = new BinaryTree();
    System.out.println("Masukkan elemen tree : ");
    tree.root = new Node(input.nextLine());
    tree.root.left.left = new Node(input.nextLine());
    tree.root.left.right = new Node(input.nextLine());
    System.out.print("Pre order : ");
    tree.preOrder(tree.root);
    System.out.print("\nIn order : ");
    tree.inOrder(tree.root);
    System.out.print("\nPost order : ");
    tree.postOrder(tree.root);
}
```

---

## LEMBAR KERJA

### TUGAS 1

Buatlah 3 code template. Code template ini bisa sebuah class, atau sebuah fungsi yang biasa anda gunakan di IDE (IntelliJ, Netbeans).

### TUGAS 2

Buatlah implementasi dari mengganti block code dengan sebuah method.

### TUGAS 3

Buatlah program konversi hari menggunakan code template dan extract method yang kalian kerjakan pada tugas 1 dan 2. Buatlah hingga menampilkan hasil output seperti berikut kemudian jelaskan cara penggunaannya kepada asisten.

```
Jumlah hari : 586
586 Hari = 1 tahun 7 bulan 11 hari

Process finished with exit code 0
|
```

Kerjakan dengan rapi ya, agar minim terjadi kesalahan syntax dan mudah dibaca.

---

**RUBRIK PENILAIAN**

<b>Kriteria</b>	<b>Nilai</b>
Tugas 1	10
Tugas 2	10
Tugas 3	40
Pemahaman	40
<b>Total</b>	<b>100</b>

Silahkan dikerjakan tanpa copas dan jangan lupa berdoa sebelum praktikum! Good Luck!!