

Adrian Dwi Syachputra

Riki Maulana

Mutiara Cahaya

Rahma Lailatul Zahra

1. Java adalah bahas pemrograman yang bersifat *multiplatform* sehingga bisa berjalan diberbagai os yang memiliki interpreter *java virtual machine (JVM)* . kemudian java bisa berjalan diberbagai mesin yang telah memiliki *java runtime environment (JRE)*. Java merupakan bahasa pemrograman berbasis *object oriented* dan *compiler* java mempunyai kemampuan deteksi error lebih teliti dibandingkan dengan bahasa lain, serta memiliki *exception handling* yang cukup tangguh ketika program ada kesalahan saat dijalankan. Java juga bersifat *multithread* sehingga sangat baik digunakan pada *pemrograman terdistribusi*. kode java sangat kaya, karena mempunyai *identifier* yang bersifat *case sensitive* jadi anda harus teliti saat memprogram dengan java, karena kesalahan sedikit saja dianggap salah. Jelaskan maksud dari kata yang dicetak miring pada kalimat diatas !

Pertanyaan 1: Jelaskan arti kata-kata yang dicetak miring dalam konteks yang diberikan.

Jawaban 1:

- Java: Bahasa pemrograman yang multiplatform, berbasis objek, dan dapat mendeteksi kesalahan dengan kemampuan penanganan exception dan multithread.
- JVM (Java Virtual Machine): Interpreter yang memungkinkan Java untuk berjalan pada berbagai sistem operasi.

- JRE (Java Runtime Environment): Paket perangkat lunak yang mencakup JVM, library, dan komponen lainnya yang diperlukan untuk menjalankan program Java.
- Berbasis objek: Paradigma pemrograman yang berbasis pada konsep objek, yang dapat berisi data dan metode untuk memanipulasi data tersebut.
- Multiplatform: Bahasa pemrograman yang dapat berjalan pada berbagai sistem operasi.
- Compiler: Program yang mengkonversi kode sumber menjadi kode mesin atau bytecode.
- Penanganan exception: Mekanisme untuk menangani kesalahan runtime dalam Java.
- Multithread: Fitur dalam Java yang memungkinkan beberapa thread untuk berjalan secara bersamaan dalam satu program.
- Identifier: Nama yang diberikan kepada variabel, kelas, atau metode dalam Java.
- Case sensitive: Karakteristik Java yang memperlakukan huruf besar dan kecil sebagai karakter yang berbeda.

2. Java dibuat atau dituliskan dengan bantuan beberapa editor, baik itu yang bersifat IDE, console maupun berbasis text. Bisa anda sebutkan beberapa editor sesuai dengan golongan tersebut?

Jawaban

IDE (Integrated Development Environment) IDE adalah sebuah lingkungan pengembangan terintegrasi yang memungkinkan pengembang untuk menulis, mengkompilasi, dan menjalankan kode dalam satu aplikasi. Contoh IDE untuk Java adalah:

- Netbeans
- Eclipse
- JCreator
- IntelliJ IDEA

Console Console adalah sebuah editor yang berbasis teks dan digunakan untuk menulis kode Java. Contoh console untuk Java adalah:

- Command Prompt (Windows)
- Terminal (Linux/Mac)

Text Editor Text Editor adalah sebuah editor yang berbasis teks dan digunakan untuk menulis kode Java. Contoh text editor untuk Java adalah:

- Notepad++
- Sublime Text
- Atom
- Visual Studio Code

3. Untuk memulai membuat sebuah program sederhana harus diawali dengan penamaan class. Tulis beberapa aturan penamaan class dalam java

Jawaban

1. **Nama kelas harus dimulai dengan huruf** : Karakter pertama nama kelas harus berupa huruf, bukan angka atau karakter khusus.

2. **Nama kelas tidak dapat dimulai dengan angka** : Nama kelas tidak dapat dimulai dengan angka, seperti **123 MyClass**.
3. **Nama kelas tidak boleh mengandung spasi** : Nama kelas tidak boleh mengandung spasi, seperti **My Class**.
4. **Nama kelas tidak boleh mengandung karakter khusus** : Nama kelas tidak boleh mengandung karakter khusus, seperti **!, @, #, \$, dll.**
5. **Nama kelas peka huruf besar/kecil** : Nama kelas peka huruf besar/kecil, artinya **MyClass** dan **myclass** dianggap sebagai nama kelas yang berbeda.

Konvensi:

6. **Gunakan PascalCase** : Nama kelas harus menggunakan PascalCase, yang berarti huruf pertama setiap kata harus ditulis dengan huruf kapital, seperti **MyClassName**.
7. **Gunakan nama yang bermakna** : Nama kelas harus bermakna dan deskriptif, yang menunjukkan tujuan atau fungsionalitas kelas.
8. **Hindari singkatan** : Hindari penggunaan singkatan dalam nama kelas, kecuali jika singkatan tersebut dikenal dan diterima secara luas.
9. **Hindari penggunaan garis bawah** : Hindari penggunaan garis bawah dalam nama kelas, kecuali jika merupakan bagian dari konvensi yang diakui secara luas.

Contoh nama kelas yang valid:

- **MyClassName**
- **BankAccount**
- **Calculator**
- **HelloWorld**

4. Gambarkan fase perograman java dari mulai kode dituliskan hingga proses running!

Jawaban

Fase 1: Penulisan Kode (Coding)

- Programmer menulis kode Java menggunakan editor teks atau IDE (Integrated Development Environment) seperti NetBeans, Eclipse, atau IntelliJ IDEA.
- Kode Java ditulis dalam file dengan ekstensi **.java**.

Fase 2: Kompilasi (Compilation)

- Kode Java dikompilasi menggunakan compiler Java seperti **javac**.
- Compiler Java mengkonversi kode Java menjadi bytecode yang dapat dijalankan oleh JVM (Java Virtual Machine).
- Bytecode disimpan dalam file dengan ekstensi **.class**.

Fase 3: Loading

- JVM memuat bytecode dari file **.class** ke dalam memori.
- JVM melakukan verifikasi bytecode untuk memastikan bahwa kode tersebut valid dan aman untuk dijalankan.

Fase 4: Linking

- JVM melakukan linking, yaitu menghubungkan bytecode dengan library dan kelas lainnya yang dibutuhkan.
- JVM memastikan bahwa semua kelas dan metode yang dibutuhkan telah diinisialisasi dan siap untuk dijalankan.

Fase 5: Initialization

- JVM melakukan inisialisasi kelas dan objek yang dibutuhkan.
- JVM memastikan bahwa semua variabel dan konstanta telah diinisialisasi dengan nilai awal.

Fase 6: Running

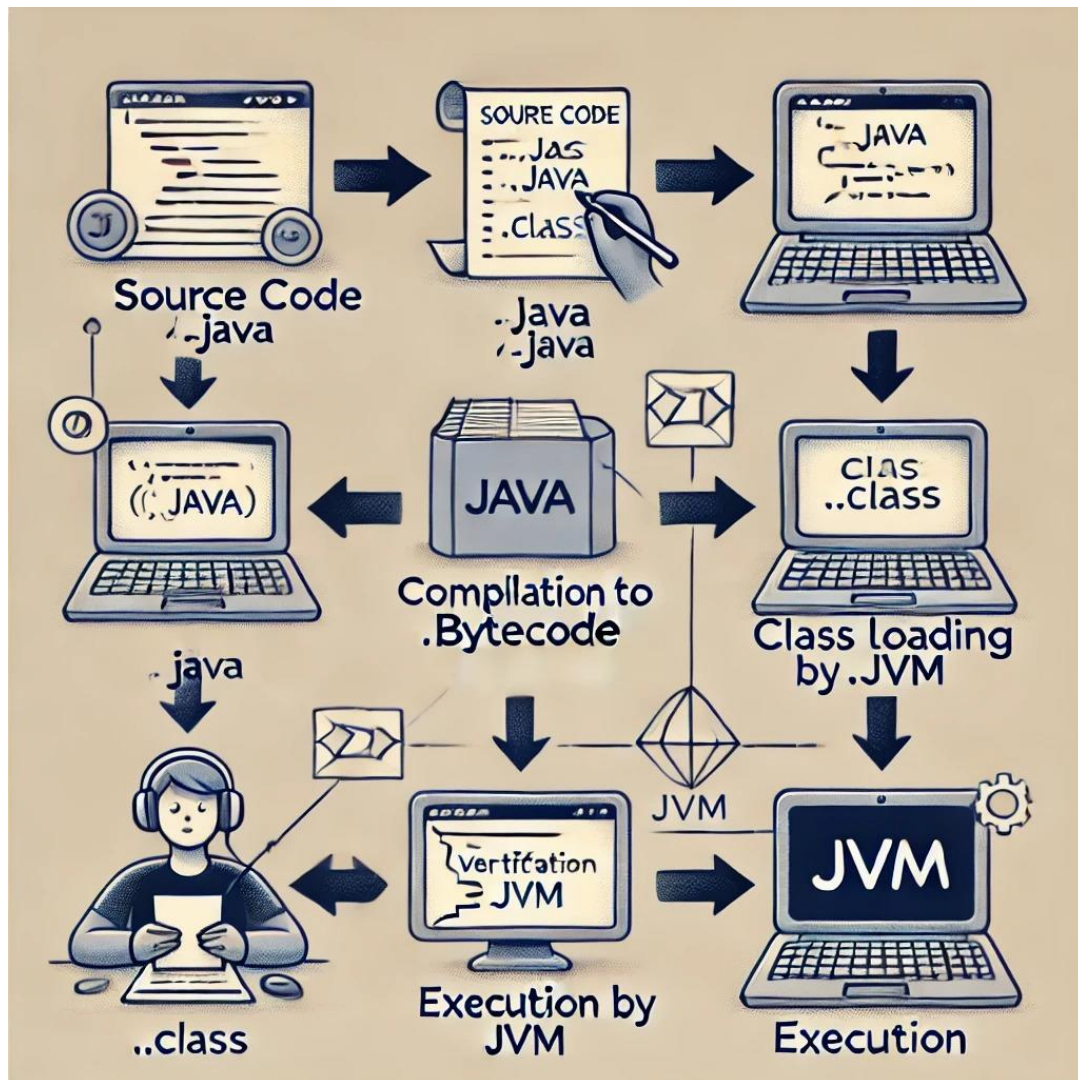
- JVM menjalankan bytecode yang telah dikompilasi dan diinisialisasi.
- Program Java mulai berjalan dan melakukan tugas-tugas yang telah ditentukan.
- JVM melakukan pengelolaan memori dan sumber daya lainnya untuk memastikan bahwa program berjalan dengan lancar.

Fase 7: Execution

- Program Java melakukan tugas-tugas yang telah ditentukan, seperti mengolah data, melakukan perhitungan, dan menampilkan output.
- JVM melakukan pengelolaan exception dan error handling untuk memastikan bahwa program berjalan dengan lancar.

Fase 8: Termination

- Program Java selesai berjalan dan JVM melakukan penghentian program.
- JVM melakukan pengelolaan sumber daya dan memori untuk memastikan bahwa program telah dihentikan dengan benar.



5. Untuk memberikan informasi kode program yang dibuat atau dokumentasi kode, kita bisa menambahkan suatu komentar program, dimana tulisan yang berada pada komentar tersebut akan diabaikan oleh compiler. silahkan tunjukkan aturan penulisan komentar program java, baik itu untuk satu baris atau banyak baris komentar.

Jawaban

1. Komentar Satu Baris:

- Menggunakan //.

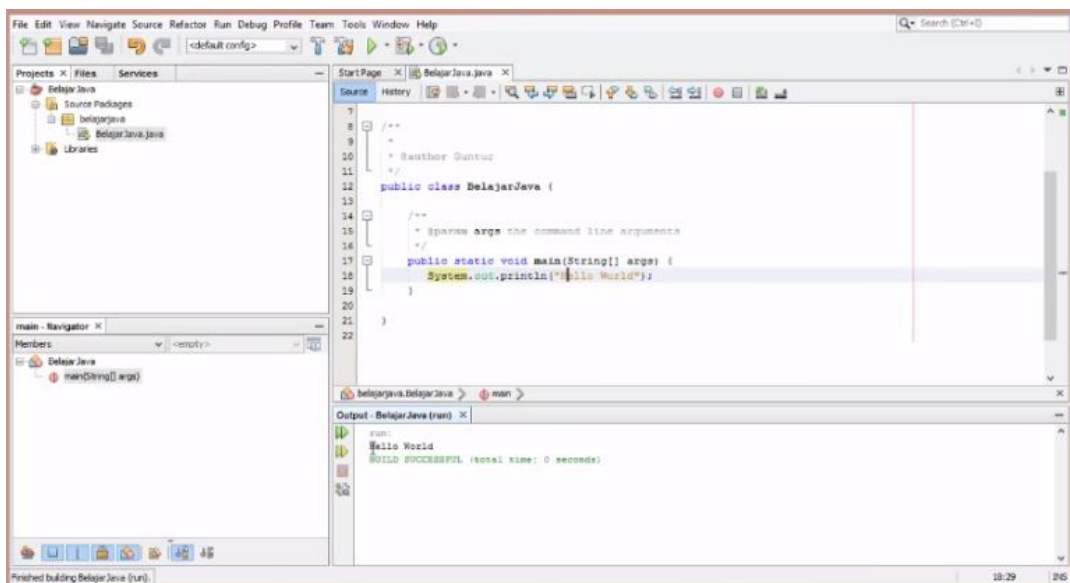
- Semua teks setelah // pada baris yang sama dianggap sebagai komentar.

2. Komentar Banyak Baris:

- Menggunakan /* */.
- Semua teks di antara /* dan */ dianggap sebagai komentar, dan bisa mencakup beberapa baris.

3. Komentar Javadoc:

- Menggunakan /** */.
- Digunakan untuk mendokumentasikan kode secara formal, biasanya untuk kelas, metode, atau paket, dan dapat menghasilkan dokumentasi otomatis.



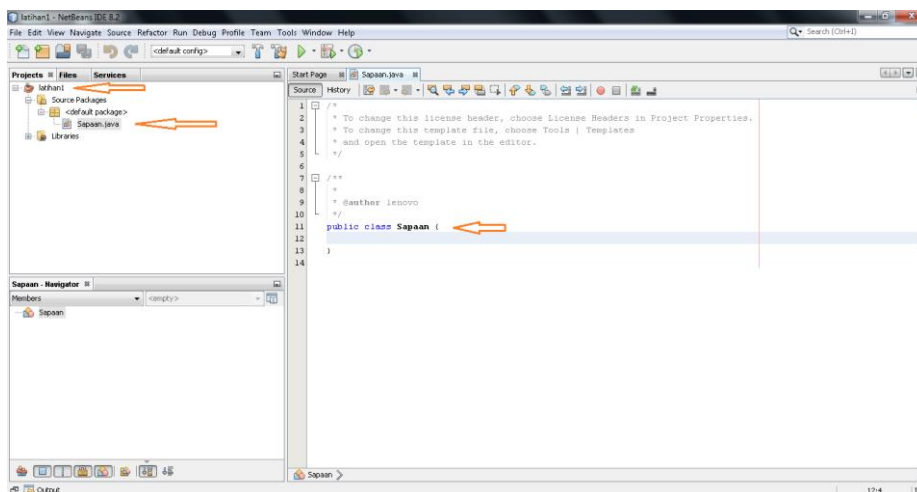
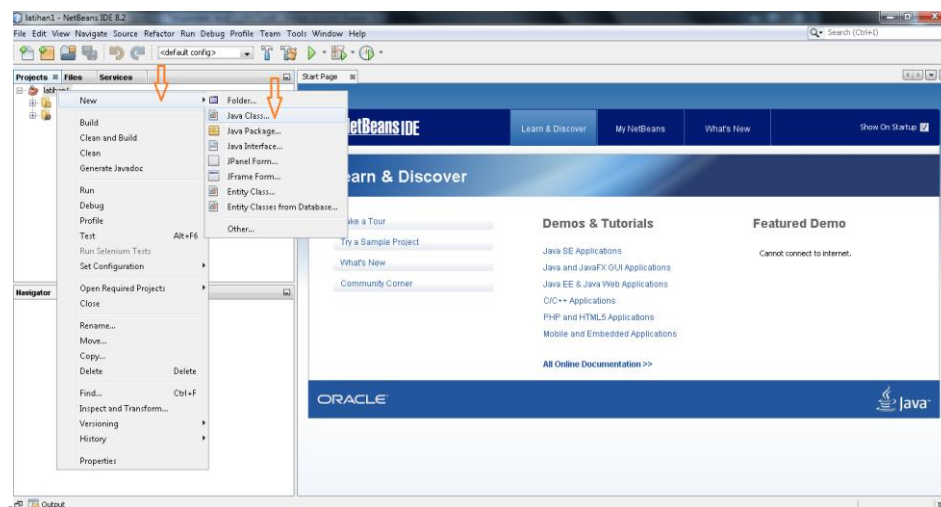
// This is C++ Style or Single line comment (komentar C++)

6. Pada praktikum daspro kali ini kita akan memanfaatkan editor yang bernama netbeans, silahkan anda cari tutorial tentang tahapan membuat projek dan kelas di netbeans. Kemudian anda tulis ulang (Dilarang kopi paste) setelah itu anda tulis kembali dan sertakan gambar tahapannya.

Jawaban

Untuk membuat proyek baru di NetBeans, ikuti langkah-langkah berikut:

1. Buka NetBeans dan klik menu "File".
2. Pilih "Proyek Baru" dari menu tarik-turun.
3. Pada kotak dialog "Proyek Baru", pilih "Java" sebagai jenis proyek dan klik "Berikutnya".
4. Pilih nama proyek, lokasi proyek, dan tetapkan versi Java. Klik "Finish"



7. Sebutkan contoh dari java literals! Yang meliputi Integer Literals, Floating-Point Literals, Boolean Literals, Character Literals dan String Literals!

Jawaban

Integer Literals: Nilai bilangan bulat.

Contoh: 10, -25, 0, 100000

Floating-Point Literals: Nilai bilangan desimal.

Contoh: 3.14, -0.001, 2.5e2 (yang sama dengan 250.0), 7.0F (menunjukkan float, bukan double)

Boolean Literals: Nilai logika.

Contoh: true, false

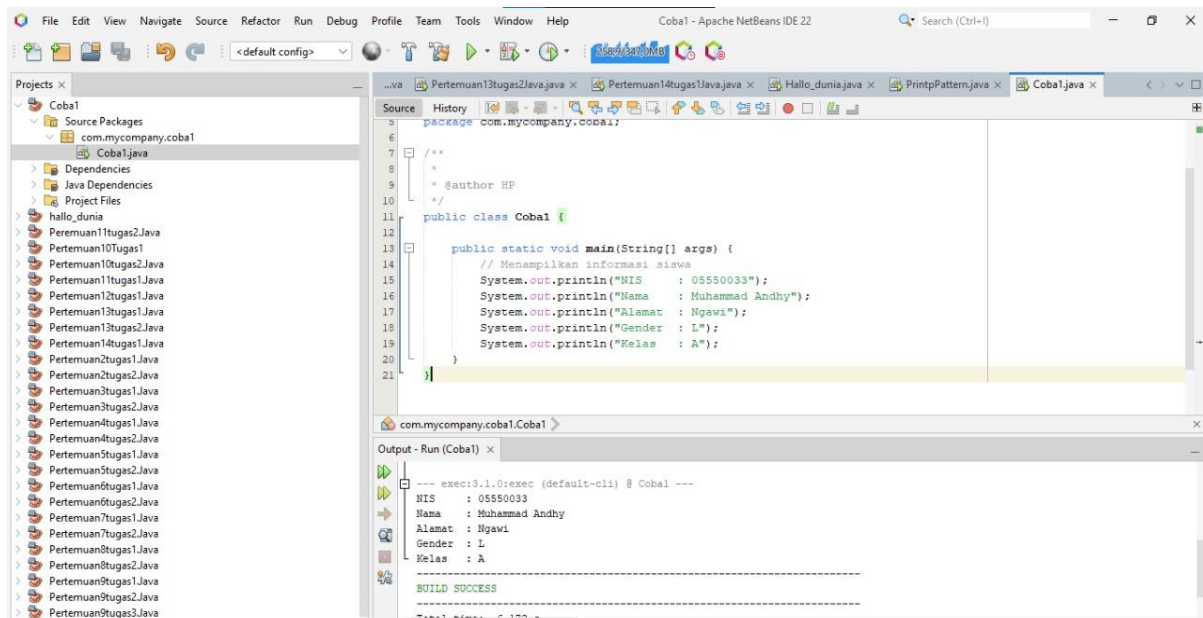
Character Literals: Karakter tunggal yang dikelilingi oleh tanda kutip tunggal.

Contoh: 'a', '1', '#', '\n' (newline)

String Literals: Urutan karakter yang dikelilingi oleh tanda kutip ganda.

Contoh: "Hello, World!", "123", "Java Programming", "Line1\nLine2" (mengandung newline di dalam string)

8.



Nama file: File ini disimpan dengan nama Coba1.java karena nama file Java harus sesuai dengan nama class utama.

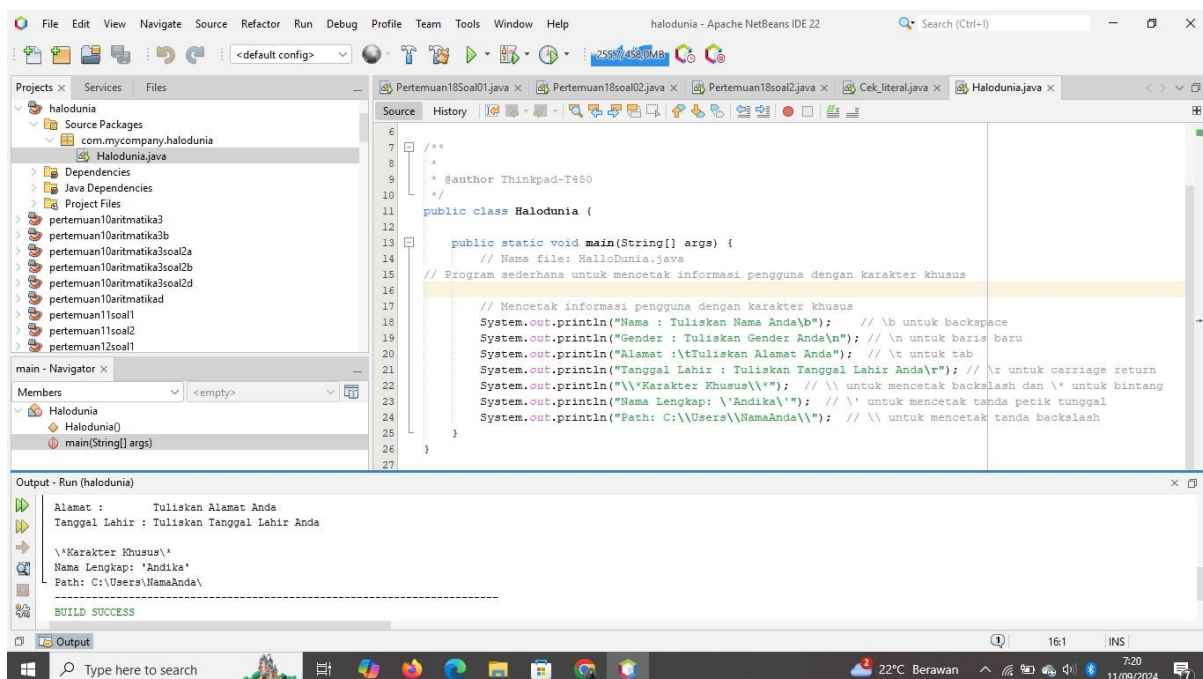
Ekstensi file: .java, karena ini adalah file Java.

Fungsi dari main class: Fungsi main di dalam class Coba1 digunakan sebagai titik awal eksekusi program. Potongan kode main class adalah:

Pemahaman tentang kode: Kode ini membuat sebuah class bernama Coba1 yang menampilkan informasi siswa ke layar ketika program dijalankan.

Komentar tambahan: Program ini dibuat oleh kelompok kami sebagai contoh sederhana untuk menampilkan informasi menggunakan bahasa pemrograman Java.

PRAKTIKUM 1



penjelasan tentang program di atas:

penjelasan tentang program di atas:

1. ***Struktur Dasar***: Program Java terdiri dari kelas `HalloDunia` dan method `main()` sebagai titik awal eksekusi.

2. ***Fungsi `System.out.println()`***: Digunakan untuk mencetak teks ke konsol.

3. ***Karakter Khusus***:

- `\b`: Menghapus satu karakter (backspace).

- `\n`: Membuat baris baru.

- `\t`: Menambahkan tab (spasi lebar).

- `\r`: Mengembalikan kursor ke awal baris.

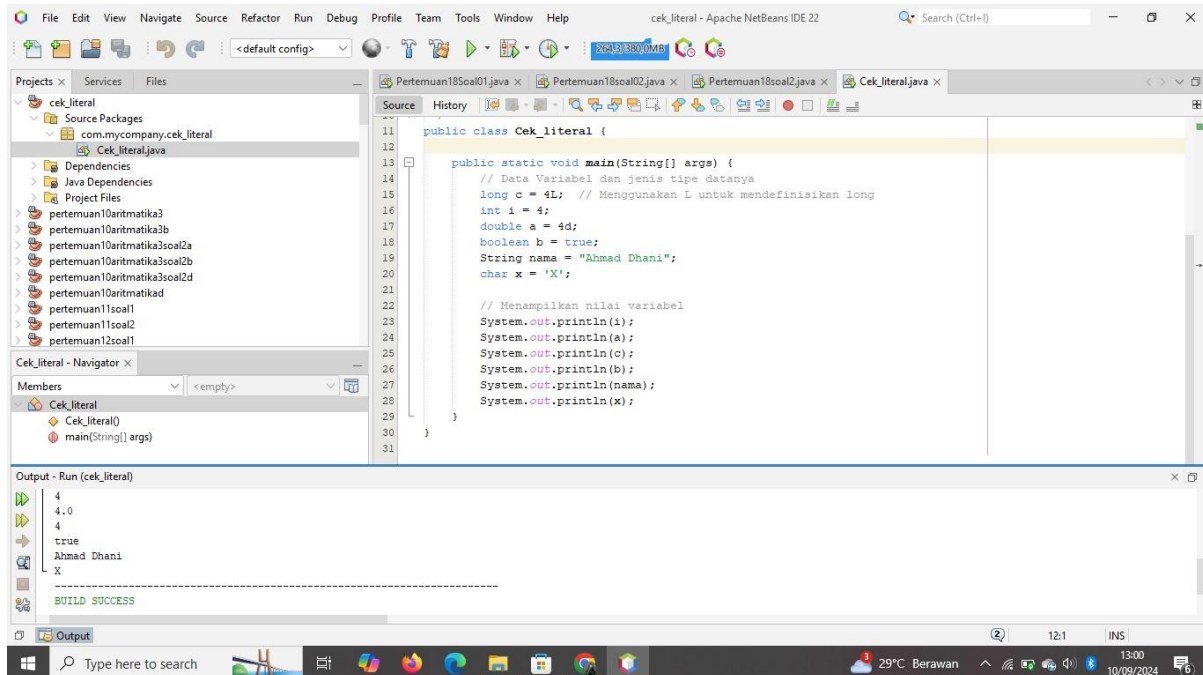
- `\\`: Mencetak backslash (`\`).

- `\'`: Mencetak petik tunggal (`'`).

4. Tujuan: Karakter khusus digunakan untuk mengatur format teks yang dicetak, seperti membuat baris baru, tab, atau mencetak karakter khusus seperti backslash dan petik tunggal.

5. Kesimpulan: Program ini menunjukkan bagaimana memanipulasi output teks menggunakan karakter khusus untuk menghasilkan format yang lebih terstruktur dan rapi di Java.

PRAKTIKUM 2



Gambar yang Anda kirimkan menunjukkan kode Java sederhana yang digunakan untuk memeriksa dan menampilkan nilai literal dari berbagai jenis data di dalam Java.

Berikut penjelasan mengapa Anda mengetik kode ini:

1. Deklarasi Variabel:

- **`long c = 4L;`**: Menggunakan literal `L` untuk memastikan bahwa nilai `4` dianggap sebagai tipe data `long`.
- **`int i = 4;`**: Literal bilangan bulat biasa, tipe data `int`.
- **`double a = 4.0;`**: Literal bilangan desimal, tipe data `double`.
- **`boolean b = true;`**: Literal boolean (`true` atau `false`), tipe data `boolean`.

- **`**String nama = "Ahmad Dhani";**`**: Literal string, digunakan untuk menyimpan teks, tipe data `String`.

- **`**char x = 'X';**`**: Literal karakter tunggal, tipe data `char`.

2. Menampilkan Nilai:

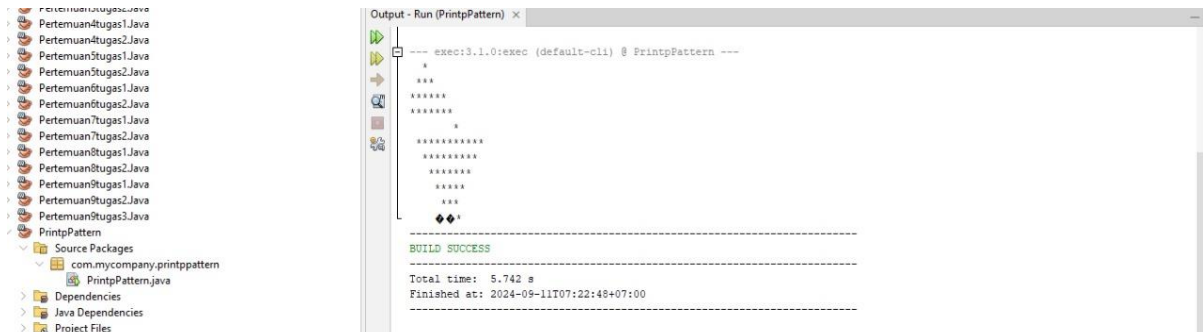
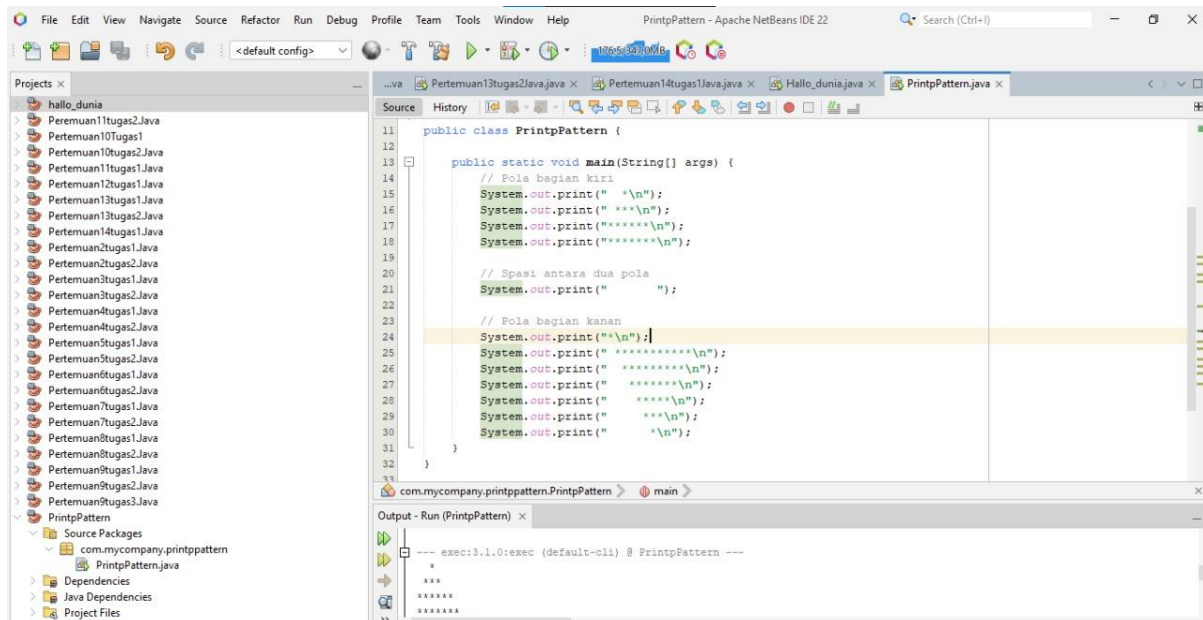
- Setiap variabel yang dideklarasikan dicetak ke konsol menggunakan `System.out.println()`. Ini menunjukkan bagaimana literal berbeda diinterpretasikan oleh Java.

3. Tujuan Pembelajaran:

- Memahami bagaimana berbagai jenis literal bekerja di Java.
- Mengenal tipe data dasar seperti `int`, `long`, `double`, `boolean`, `String`, dan `char`.
- Mengetahui cara mendefinisikan dan menampilkan nilai variabel dalam Java.

Kode ini sangat penting bagi pemula untuk memahami konsep dasar tipe data dan literal di dalam pemrograman Java.

PRAKTIKUM 3



Kode Java

Kode ini mencetak pola menggunakan `System.out.print` untuk mencetak setiap baris pola secara terpisah. Karakter spasi (" ") dan bintang ("*") digunakan untuk membentuk dua pola: segitiga normal di kiri dan segitiga terbalik di kanan.

Bagian Kode

Deklarasi Kelas dan Metode Utama

(PrintPattern) adalah kelas utama, dan (main) adalah metode yang dijalankan saat program dieksekusi.

Pola Bagian Kiri

Mencetak segitiga bertambah dengan menambahkan lebih banyak (*) di setiap baris.

Spasi Antar Pola

Mencetak spasi untuk memisahkan pola kiri dan kanan.

Pola Bagian Kanan

Mencetak segitiga terbalik dengan jumlah (*) yang berkurang di setiap baris.

Inti Kode

Kode menggunakan kombinasi spasi dan bintang untuk menghasilkan pola sesuai dengan gambar, dan (\n) untuk berpindah ke baris baru setelah setiap cetakan.

KESIMPULAN

Tentu! Berikut adalah revisinya:

Hal yang Diperoleh:

1. **Pemahaman tentang Tipe Data:**

- Kami mempelajari berbagai tipe data dasar di Java seperti `int`, `long`, `double`, `boolean`, `String`, dan `char`.

- Kami memahami bagaimana literal atau nilai konstan diwakili dalam berbagai tipe data tersebut.

2. ****Penulisan Kode Java****:

- Kami mempraktikkan cara mendeklarasikan variabel dan memberikan nilai (literal) kepada variabel tersebut.

- Kami belajar menggunakan `System.out.println()` untuk menampilkan nilai variabel di konsol.

3. ****Konversi dan Penggunaan Literal****:

- Kami melihat bagaimana literal angka seperti `4`, `4.0`, dan `4L` diperlakukan secara berbeda tergantung pada tipe data yang digunakan (`int`, `double`, `long`).

- Kami juga melihat perbedaan antara tipe data teks (`String`) dan karakter tunggal (`char`).

4. ****Pemahaman Sintaks Java****:

- Kami mengamati dan mempraktikkan aturan sintaksis dasar dalam Java, seperti penggunaan titik koma `;` di akhir pernyataan dan penggunaan tanda kutip untuk literal `String` dan `char`.

Kesimpulan:

- ****Pentingnya Memilih Tipe Data yang Tepat****:

- Setiap tipe data di Java memiliki fungsi dan batasannya masing-masing. Memilih tipe data yang tepat sangat penting untuk memastikan bahwa program kami bekerja secara efisien dan sesuai dengan yang diharapkan.

- ****Pemahaman Literal dan Variabel****:

- Literal adalah nilai tetap yang diberikan langsung pada variabel. Memahami bagaimana berbagai jenis literal bekerja di Java sangat penting untuk menghindari kesalahan tipe data.

- ****Dasar Pemrograman Java****:

- Praktikum ini memberikan dasar yang kuat dalam memahami dasar-dasar pemrograman Java, yang akan menjadi fondasi untuk mempelajari konsep yang lebih kompleks di masa depan.

Dengan memahami hal-hal ini, kami akan lebih siap untuk mengembangkan kemampuan pemrograman kami dalam Java dan bahasa pemrograman lainnya.

