

LAPORAN PRAKTIKUM

PEMOGRAMAN ALGORITMA PEMOGRAMAN

TIPE DATA PRIMITIF, VARIABEL, DAN KONSTANTA

disusun Oleh:

Khaula Lathifa Firdausyi

NIM 2511531007

Dosen Pengampu: DR. Wahyudi, S.T, M.T

Asisten Praktikum: Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

## **KATA PENGANTAR**

Laporan praktikum ini disusun sebagai bentuk pertanggungjawaban atas pelaksanaan kegiatan praktikum mata kuliah Algoritma Pemograman yang membahas tentang Tipe Data Primitif, Variabel, dan Konstanta pada Bahasa Pemograman Java. Melalui laporan ini penulis dapat lebih memahami materi praktikum dan dengan penulisan laporan ini dapat melatih ketelitian, keteraturan, serta kemampuan menulis sesuai kaidah akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam pembelajaran selanjutnya.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat terbuka terhadap segala bentuk kritik dan saran yang bersifat membangun, demi perbaikan kualitas laporan maupun pemahaman penulis di laporan berikutnya.

Padang, 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
BAB I .....	1
PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Tujuan Praktikum.....	2
1.3    Manfaat Praktikum.....	2
BAB II.....	3
PEMBAHASAN .....	3
2.1    Praktikum “ContohChar” .....	3
2.2    Praktikum “DeklarasiVariabel” .....	5
2.3    Praktikum “KelilingLingkaran” .....	7
BAB III .....	9
KESIMPULAN.....	9
3.1    Kesimpulan .....	9
DAFTAR PUSTAKA .....	11

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Dalam dunia pemrograman, tipe data primitif, variabel, dan konstanta merupakan fondasi dasar yang harus dikuasai dalam mempelajari bahasa pemrograman Java. Ketiganya berperan sebagai elemen struktural utama dalam menyimpan, memanipulasi, dan mengelola data dalam sebuah program.

Tipe data primitif dalam Java adalah tipe data dasar yang secara langsung disediakan dan dikelola oleh bahasa Java, bukan berupa objek atau turunan dari kelas. Tipe data ini digunakan untuk menyimpan nilai sederhana seperti angka, karakter, dan nilai logika. Java memiliki delapan tipe data primitif, yaitu:

- *byte, short, int, long* → untuk bilangan bulat
- *float, double* → untuk bilangan desimal
- *char* → untuk karakter tunggal
- *boolean* → untuk nilai logika (*true/false*)

Setiap tipe memiliki ukuran memori tetap dan rentang nilai tertentu sehingga pemakaiannya disesuaikan dengan kebutuhan agar program bisa dijalankan dengan lancar dan tidak terjadi *overflow* atau kehilangan presisi.

Selain tipe data, variabel dalam Java adalah wadah atau tempat penyimpanan data yang memiliki nama sebagai identifikasi dan tipe data tertentu, yang nilainya dapat berubah selama program berjalan. Variabel digunakan untuk menyimpan informasi sementara yang akan diproses dalam program, seperti angka, teks, atau nilai logika.

Di sisi lain, konstanta (biasanya dideklarasikan dengan kata kunci *final*) adalah nilai tetap yang tidak dapat diubah setelah inisialisasi, berguna untuk menyimpan nilai-nilai penting seperti nilai  $\pi$ , nama aplikasi, atau batas maksimum system yang jika diubah dapat menyebabkan kesalahan logika atau inkonsistensi program.

## 1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini antara lain sebagai berikut:

1. Mengetahui fungsi dan pengaplikasian penggunaan tipe data primitif.
2. Mengetahui dan dapat mengaplikasikan variabel dalam bahasa Java.
3. Mengetahui dan dapat mengaplikasikan konstanta dalam bahasa Java.

## 1.3 Manfaat Praktikum

Manfaat dari praktikum ini antara lain sebagai berikut:

1. Mampu mengaplikasikan penggunaan tipe data primitif.
2. Mampu mengaplikasikan variabel dalam Bahasa Java.
3. Mampu mengaplikasikan konstanta dalam Bahasa Java.

## BAB II

### PEMBAHASAN

#### 2.1 Praktikum “ContohChar”

```

1 package pekan2;
2
3 public class ContohChar {
4     public static void main(String[] args) {
5         // Deklarasi variabel char
6         char huruf1 = 'A';
7         char huruf2 = 'B';
8         char angka = '7';
9         char simbol = '#';
10
11         // Menampilkan nilai variabel char
12         System.out.println("Contoh variabel char:");
13         System.out.println("Huruf pertama: " + huruf1);
14         System.out.println("Huruf kedua: " + huruf2);
15         System.out.println("Angka: " + angka);
16         System.out.println("Simbol: " + simbol);
17
18         // Operasi dengan char (berdasarkan kode Unicode/ASCII
19         char huruf3 = (char) (huruf1 + 1); // A (65) + 1 = B (66)
20         System.out.println("Huruf1 + 1 = " + huruf3);
21
22         // Char juga bisa disimpan dalam integer (ASCII/Unicode value)
23         int kodeHuruf = huruf1;
24         String biner1 = String.format("%8s", Integer.toBinaryString(kodeHuruf)).replace(' ', '0');
25         System.out.println("Kode ASCII dari " + huruf1 + " = " + kodeHuruf);
26         System.out.println("Kode BINER dari " + huruf1 + " = " + biner1);
27
28         // Menggunakan char menjadi string
29         String kata = "" + huruf1 + huruf2 + angka + simbol;
30         System.out.println("Gabungan char menjadi string " + kata);
31     }
32 }
33
34 }
35

```

Gambar 2.1 Kode program praktikum ContohChar

Penggalan kode di atas merupakan kode program yang memakai tipe data primitif berupa *char*; Tipe data *char* digunakan apabila kita memiliki tipe data karakter.

Adapun penjelasan tentang Langkah kerja program tersebut adalah sebagai berikut:

1. Pada baris kode 6-9 kita mendeklarasikan tipe data *char* yang digunakan untuk menyimpan satu karakter.
2. Pada baris kode 6-9 kita juga mempunyai “huruf1, huruf2, angka, simbol” sebagai variabel.

3. Jika kita ingin mencetak nilai masing-masing karakter gunakan “System.out.println()” seperti di baris kode 12-16
4. Pada baris kode 19-20 kita melakukan operasi penjumlahan menggunakan kode ASCII (*American Standard Code for Information Interchange*).
5. Pada baris kode 23-26 kita mengkonversi dari *char* ke *integer* dan biner.
6. Pada baris kode 29-30 kita menggabungkan semua nilai char menjadi *string* dengan menambahkan "" (string kosong), sehingga kita bisa menggabungkan nilai menjadi sebuah kalimat.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.2 dibawah ini.

```
Contoh variabel char:
Huruf pertama: A
Huruf kedua: B
Angka: 7
Simbol: #
Huruf1 + 1 = B
Kode ASCII dari A = 65
Kode BINER dari A = 01000001
Gabungan char menjadi string AB7#
```

Gambar 2.1 Output dari kode program ContohChar

### 2.1.1 Analisis

- Variabel umur dideklarasikan sebagai static sehingga dapat diakses dari fungsi main() tanpa instansiasi objek.
- Variabel lokal (kode, gaji, isDibawahUmur) harus diinisialisasi sebelum digunakan.
- Penamaan variabel menggunakan konvensi camelCase (isDibawahUmur), yang merupakan standar dalam Java.
- Tipe double digunakan untuk menyimpan nilai desimal dengan presisi tinggi.

## 2.2 Praktikum “DeklarasiVariabel”

```

1 package pekan2;
2
3 public class DeklarasiVariabel {
4     /* program java
5      *latihan
6      *tentang pendeklarasian variabel
7      */
8     static int umur=25; /* variabel dapat langsung diinisiasi */
9     public static void main(String[] args) {
10         int kode;
11         boolean isDibawahUmur; /* perhatikan penulisan nama variabel */
12         kode = 1234; /* pengisian variabel (assignment) */
13         double gaji; /* deklarasi variabel dapat dimana saja */
14         gaji = 5500000.23;
15         isDibawahUmur = true;
16         System.out.println("Status: "+isDibawahUmur);
17         System.out.println("kode: "+kode);
18         System.out.println("Umur: "+umur);
19         System.out.println("Gaji: "+gaji);
20     }
21 }
22 }

```

Gambar 2.3 Kode program praktikum DeklarasiVariabel

Penggalan kode program di atas menggunakan berbagai jenis variabel yang berfungsi menunjukkan cara mendeklarasikan dan menginisialisasi berbagai jenis variabel di Java (int, boolean, double).

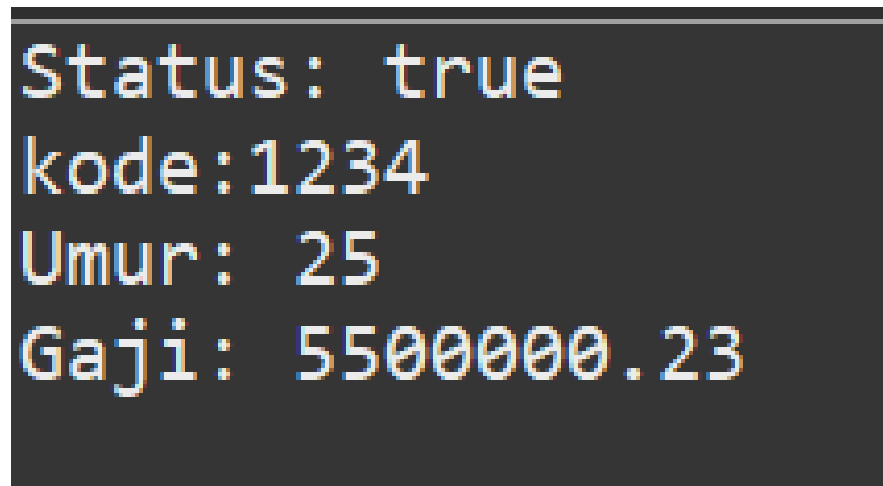
Adapun penjelasan Langkah-langkah kerja program tersebut adalah sebagai berikut:

1. Pada baris kode 8 kita mendeklarasikan nilai variabel “umur” yang di definisikan di luar main() dan langsung diisi nilai.
2. Inisiasi variabel yang berada dalam main():



- `int kode;` → deklarasi tanpa nilai awal.
  - `boolean isDibawahUmur;` → deklarasi variabel boolean (nilai hanya `true` atau `false`).
  - `kode = 1234;` → memberi nilai ke variabel `kode` (proses assignment).
  - `double gaji;` → deklarasi variabel desimal.
  - `gaji = 550000.23;` → isi nilai ke `gaji`.
  - `isDibawahUmur = true;` → isi nilai ke boolean.
3. Gunakan `System.out.println` untuk mencetak semua nilai variabel.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar dibawah ini.



```
Status: true
kode:1234
Umur: 25
Gaji: 5500000.23
```

Gambar 2.4 Output kode program DeklarasiVariabel

### 2.2.1 Analisis

- 2.2.1.1.1 Semua nilai ditampilkan sesuai assignment.
- 2.2.1.1.2 Tipe *double* menampilkan desimal dengan presisi penuh.
- 2.2.1.1.3 Tipe boolean ditampilkan sebagai teks "*true*" — bukan angka 1.
- 2.2.1.1.4 Variabel *static* berhasil diakses tanpa instansiasi.

## 2.3 Praktikum “KelilingLingkaran”

```

1 package pekan2;
2
3 public class KelilingLingkaran {
4
5     public static void main(String[] args) {
6         final double PI = 3.14; /* Definisi konstanta */
7         double radius = 30;    /* Deklarasi variabel */
8         System.out.println("Keliling = " + 2 * PI * radius);
9
10    }
11
12 }

```

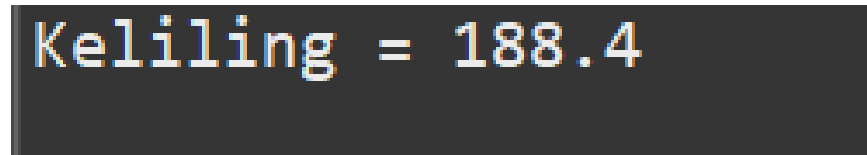
Gambar 2.5 kode program praktikum KelilingLingkaran

Penggalan kode program di atas menggunakan konstanta dan variabel berupa “*final double PI* dan *double radius*” untuk mendefinisikan nilai konstanta dan mendeklarasikan variabel.

Adapun penjelasan Langkah-langkah kerja program tersebut adalah sebagai berikut:

1. Pada baris kode program 6 kita mendefinisikan nilai konstanta
  - *final*: menandakan bahwa nilai ini tidak bisa diubah setelah diinisialisasi.
  - *double*: tipe data untuk angka desimal (dengan presisi tinggi).
  - *PI = 3.14*: nilai pendekatan  $\pi$  (pi).
2. Pada baris kode program 7 kita mendeklarasikan variabel
  - *double*: tipe data untuk menyimpan bilangan pecahan.
  - *radius*: nama variabel yang menyimpan jari-jari lingkaran.
  - *= 30*: nilai awal (dalam satuan cm, meter, dst., tergantung konteks).

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.6 dibawah ini.



```
Keliling = 188.4
```

Gambar 2.6 Output kode program KelilingLingkaran

### 2.3.1 Analisis

- Definisi konstanta → PI sebagai nilai tetap.
- Deklarasi variabel → radius sebagai input.
- Operasi matematis → perkalian dan perhitungan.
- Output hasil → menampilkan hasil perhitungan.

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Setelah melalui rangkaian kegiatan praktikum yang meliputi pengamatan, percobaan, dan analisis terhadap penggunaan tipe data primitif, deklarasi variabel, serta penerapan konstanta dalam kode program, penulis dapat menarik sejumlah kesimpulan penting. Praktikum ini tidak hanya memperkuat pemahaman teoritis, tetapi juga memberikan pengalaman langsung dalam mengimplementasikan konsep dasar pemrograman secara teknis. Hasil yang diperoleh menunjukkan relevansi dan urgensi penguasaan ketiga elemen dasar ini sebagai fondasi dalam pengembangan logika pemrograman yang efisien dan bebas kesalahan. Berdasarkan temuan dan analisis selama praktikum, berikut disampaikan kesimpulan secara sistematis:

1. Data Primitif merupakan tipe data dasar yang disediakan oleh bahasa pemrograman Java untuk menyimpan nilai sederhana seperti bilangan bulat (integer), bilangan pecahan (float/double), karakter (char), dan nilai logika (boolean). Pemahaman terhadap karakteristik masing-masing tipe data primitif sangat penting untuk memastikan efisiensi penggunaan memori dan ketepatan dalam operasi pemrograman.
2. Variabel berfungsi sebagai wadah penyimpanan data yang nilainya dapat berubah selama program berjalan. Deklarasi variabel harus memperhatikan tipe data, nama variabel yang deskriptif, dan inisialisasi awal agar program berjalan dengan baik dan bebas dari error. Variabel memungkinkan program bersifat dinamis dan interaktif sesuai kebutuhan logika aplikasi.
3. Konstanta adalah entitas yang menyimpan nilai tetap selama eksekusi program. Penggunaan konstanta sangat dianjurkan untuk

nilai-nilai yang tidak berubah seperti nilai matematis ( $\pi$ ), konfigurasi sistem, atau kode status tertentu. Konstanta meningkatkan keterbacaan kode dan memudahkan proses pemeliharaan program.

Praktikum ini berhasil memperkuat pemahaman tentang perbedaan mendasar antara variabel dan konstanta, serta pentingnya pemilihan tipe data primitif yang tepat dalam implementasi program. Kesalahan umum seperti salah deklarasi tipe data, tidak menginisialisasi variabel, atau mencoba mengubah nilai konstanta dapat menyebabkan runtime error atau logical error yang harus dihindari.

Secara keseluruhan, penguasaan konsep data primitif, variabel, dan konstanta merupakan fondasi penting dalam pembelajaran pemrograman, karena ketiga elemen ini menjadi dasar dalam pembuatan struktur data dan algoritma yang lebih kompleks.

## DAFTAR PUSTAKA

- [1] Oracle Corporation. (2023). The Java™ Tutorials: Learning the Java Language. Diakses dari <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/>
- [2] Horstmann, C. S. (2020). Core Java Volume I: Fundamentals (11th ed.). Prentice Hall.
- [3] Deitel, H. M., & Deitel, P. J. (2021). Java How to Program (11th ed.). Pearson Education.