LAPORAN PRAKTIKUM PEMOGRAMAN ALGORITMA PEMOGRAMAN

CLASS SCANNER, OPERATOR ARITMATIKA, OPERATOR ASSIGNMENT, OPERATOR RELASIONAL, DAN OPERATOR LOGIKA

disusun Oleh:

Aisyah Najmiatul Fauziah

NIM 2511533030

Dosen Pengampu: DR. Wahyudi, S.T, M.T Asisten Praktikum: Rahmat DRO



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 Class Scanner, Operator Aritmatika, Operator Assignment, Operator Relasional, dan Operator Logika

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

# BAB II

# PEMBAHASAN

* 1. **Praktikum “OperatorAritmatika”**



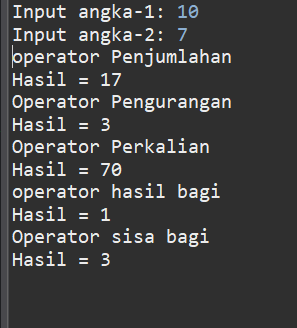
Gambar 2.1 Kode program praktikum OperatorAritmatika

Penggalan kode di atas merupakan kode program yang meminta 2 bilangan bulat, lalu melakukan operasi aritmatika dan di keluarkan melalui variabel “hasil”.

Adapun penjelasan tentang Langkah kerja program tersebut adalah sebagai berikut:

1. Menggunakan import java.util.scanner untuk mengakses kelas scanner yang ada di dalam paket java.util.
2. Menginisialisasi variabel A1, A2, dan hasil dalam tipe int.
3. Memasukkan alat baca input berupa Scanner yang sudah kita dapatkan aksesnya di Langkah nomor 1.
4. *System.out.print*(“..”) untuk menampilkan teks yang diinginkan, contohnya (“Input angka ke-1: “) maka teks yang di dalam kurung yang akan di cetak.
5. *keyboard.nextInt*() berfungsi membaca angka bulat yang diketik oleh pengguna, lalu masukkan ke dalam variabel A1.
6. Lalu menutup kelas scanner dengan *keyboard.close*().
7. Pada variabel “hasil” kita melakukan operator aritmatika antara A1 dan A2 yang di dalamnya telah disimpan nilai bilangan bulat.

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



Gambar 2.2 Output dari kode program OperatorAritmatika

* + 1. **Analisis**

• +, -, \* bekerja seperti biasa.

* + - * / (pembagian): karena tipe int, hasilnya dibulatkan ke bawah.
      * % (modulus): mengembalikan sisa pembagian.
      * \*, /, % memiliki prioritas lebih tinggi daripada +, -.
  1. **Praktikum “OperatorAssignment”**



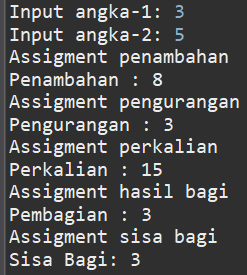
Gambar 2.3 Kode program praktikum OperatorAssignment

Penggalan kode program di atas merupakan kode program yang meminta 2 bilangan bulat, lalu melakukan operator assignment dan di keluarkan melalui variabel “A1”.

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

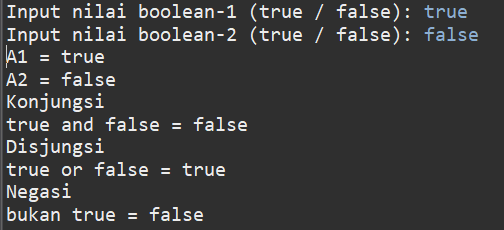
1. Menggunakan import java.util.scanner untuk mengakses kelas scanner yang ada di dalam paket java.util.
2. Menginisialisasi variabel A1 dan A2 dalam tipe int.
3. Memasukkan alat baca input berupa Scanner yang sudah kita dapatkan aksesnya di Langkah nomor 1.
4. System.out.print(“..”) untuk menampilkan teks yang diinginkan, contohnya (“Input angka ke-1: “) maka teks yang di dalam kurung yang akan di cetak.
5. *keyboard.nextInt*() berfungsi membaca angka bulat yang diketik oleh pengguna, lalu masukkan ke dalam variabel A1.
6. Lalu menutup kelas scanner dengan *keyboard.close*().
7. A1 +=,-=,\*=,/=,%= A2 merupakan bentuk operator assignment yang hasil nya akan ditampilkan berupa “A1 = …”.

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



Gambar 2.4 Output kode program OperatorAssignment

* + 1. **Analisis**
       - Variabel A1 dan A2 bertipe int, sehingga semua operasi akan menghasilkan bilangan bulat. Pembagian (/=) antara dua integer akan menghasilkan hasil pembulatan ke bawah (*floor division*).
  1. **Praktikum “OperatorLogika”**

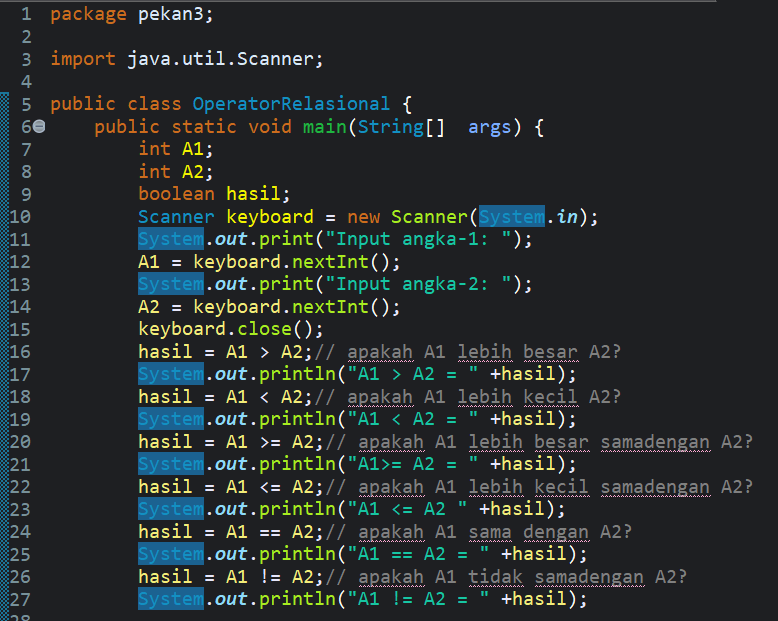


Gambar 2.6 Output kode program OperatorLogika

* + 1. **Analisis**

Kode ini mengajarkan cara kerja logika Boolean dalam pemrograman, yaitu bagaimana komputer mengambil keputusan berdasarkan kondisi benar atau salah.

* 1. **Praktikum “OperatorRelasional**

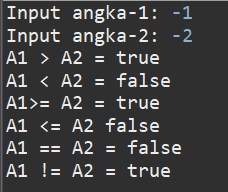


Gambar 2.7 kode program praktikum OperatorRelasional

Penggalan kode program di atas Program ini meminta dua angka bulat dari pengguna, lalu membandingkannya menggunakan operator relasional >, <, >=, <=, ==, dan !=.

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

1. Inisialisasi variabel A1 dan A2 dalam tipe int dan variabel hasil dalam tipe Boolean.
2. Pada baris kode 17-28 kita membandingkan dua angka menggunakan operator relasional. Setiap operasi menghasilkan nilai true atau false.
3. Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.8 dibawah ini.



Gambar 2.8 Output kode program OperatorRelasional

* + 1. **Analisis**

Setiap operator relasional memiliki definisi matematis yang jelas. Misalnya, a > b benar jika dan hanya jika nilai a lebih besar dari b.

# BAB III

# KESIMPULAN

## Kesimpulan

Praktikum mengenai Class Scanner, operator aritmatika, assignment, relasional, dan logika telah memperkuat pemahaman bahwa sebuah program Java yang interaktif dan logis dibangun dari integrasi komponen dasar: Scanner digunakan untuk menerima input dinamis dari pengguna; operator aritmatika (+, -, \*, /, %) memungkinkan perhitungan numerik; operator assignment (=, +=, -=, dll.) menyimpan dan memperbarui nilai variabel secara efisien; operator relasional (>, <, ==, dll.) membandingkan nilai dan menghasilkan kondisi boolean; serta operator logika (&&, ||, !) menggabungkan atau membalik kondisi untuk pengambilan keputusan yang kompleks. Kelima elemen ini bersama- sama membentuk fondasi logika pemrograman prosedural, memungkinkan program tidak hanya menjalankan instruksi statis, tetapi juga merespons input, melakukan perhitungan, dan mengambil keputusan berdasarkan kondisi—sehingga menjadi dasar penting dalam pengembangan perangkat lunak yang cerdas dan adaptif.

# DAFTAR PUSTAKA

1. Oracle Corporation, “Class Scanner,” Java Platform Standard Edition Documentation, 2024. [Online]. Available: https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/Scanner.htm l . [Accessed: 5 Apr. 2025].
2. Oracle Corporation, “Operators,” The Java™ Tutorials, 2024. [Online]. Available: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html

. [Accessed: 5 Apr. 2025].

1. R. W. Sebesta, Concepts of Programming Languages, 12th ed. Boston, MA, USA: Pearson, 2010