

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
“LAPORAN CUACA”

Disusun Oleh:

Rifqi Aditya

2511533002

Dosen Pengampu :

Dr. Wahyudi, S.T, M.T

Asisten Praktikum :

Rahmad Dwirizki Olders



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT karena atas rahmat dan karunia-Nya saya dapat menyelesaikan laporan praktikum mata kuliah **Algoritma dan Pemrograman** dengan judul "*Laporan Cuaca*".

Laporan ini disusun untuk memenuhi salah satu tugas praktikum sekaligus sebagai sarana pembelajaran dalam memahami dasar-dasar pemrograman menggunakan bahasa Java. Melalui praktikum ini, penulis dapat mempelajari penggunaan tipe data dasar seperti *integer*, *float*, *char*, dan *boolean*, serta bagaimana menampilkan informasi ke layar menggunakan perintah `System.out.println`.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah membantu dalam proses penyusunan laporan ini.

Akhir kata, semoga laporan ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi pembaca, khususnya dalam memperdalam pemahaman tentang pemrograman dasar dengan Java.

Padang, 26 September 2025

Penulis

Daftar Pustaka

BAB I	4
1.1 Latar Belakang	4
1.2 Tujuan.....	4
1.3 Manfaat	4
BAB II	5
2.1 Operator Aritmetika	5
2.2 Operator Penugasan (Assignment).....	6
2.3 Operator Relasional.....	7
2.4 Operator Logika	9
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman Java merupakan landasan penting dalam mempelajari algoritma. Untuk menguasainya, pemahaman mendalam tentang elemen dasar seperti tipe data sangatlah diperlukan. Praktikum ini bertujuan untuk memberikan pengalaman praktis dalam penggunaan tipe data di Java, meliputi *integer*, *float*, *char*, *double* dan *Boolean*.

1.2 Tujuan

1. Melatih keterampilan menampilkan output ke layar dengan format yang rapi.
2. Membiasakan penggunaan penginputan sebagai media tampilan informasi.
3. Menerapkan deklarasi variabel dengan berbagai tipe data dasar di Java.

1.3 Manfaat

1. Mahasiswa mampu membuat program sederhana.
2. Mahasiswa memahami penggunaan tipe data dasar dalam Java.
3. Mahasiswa dapat mengaitkan pembelajaran pemrograman dengan kasus nyata di dunia industri.

BAB II

PEMBAHASAN

Bab ini menguraikan secara rinci mengenai implementasi, langkah kerja, serta analisis hasil dari kode program yang telah dibuat. Pembahasan difokuskan pada empat jenis operator dasar dalam bahasa pemrograman Java, yaitu operator aritmetika, operator penugasan (*assignment*), operator relasional, dan operator logika.

2.1 Operator Aritmetika

Operator aritmetika adalah operator yang digunakan untuk melakukan operasi matematika dasar. Program OperatorAritmatika dibuat untuk mendemonstrasikan fungsionalitas dari operator-operator ini.

2.1.1 Uraian Kode Program

Kode program ini dirancang untuk menerima dua buah masukan bilangan bulat (integer) dari pengguna. Masukan tersebut disimpan dalam dua peubah (variabel) bernama A1 dan A2. Selanjutnya, program akan melakukan lima operasi aritmetika dasar: penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan sisa bagi atau modulus (%). Hasil dari setiap operasi disimpan dalam peubah hasil dan langsung ditampilkan ke layar monitor sesaat setelah perhitungan dilakukan. Program ini memanfaatkan kelas Scanner untuk menangani proses masukan dari papan tik (keyboard).

2.1.2 Langkah Kerja Program

1. Program memulai eksekusi pada metode main.
2. Tiga peubah integer dideklarasikan: A1, A2, dan hasil.
3. Sebuah objek Scanner dengan nama keyboard dibuat untuk membaca masukan dari pengguna.
4. Program menampilkan teks "Input angka-1: " dan menunggu pengguna memasukkan sebuah bilangan bulat. Nilai yang dimasukkan disimpan ke dalam peubah A1.
5. Program menampilkan teks "Input angka-2: " dan menunggu pengguna memasukkan bilangan bulat kedua. Nilai tersebut disimpan ke dalam peubah A2.

6. Program melakukan operasi penjumlahan ($A1 + A2$), menyimpan hasilnya ke hasil, dan menampilkannya.
7. Program melakukan operasi pengurangan ($A1 - A2$), menyimpan hasilnya ke hasil, dan menampilkannya.
8. Proses serupa diulang untuk operasi perkalian ($*$), pembagian ($/$), dan sisa bagi ($\%$).
9. Setelah semua operasi selesai, eksekusi program berakhir.

2.1.3 Analisis Hasil

Untuk menganalisis hasil, kita dapat menggunakan masukan data sebagai contoh: $A1 = 10$ dan $A2 = 3$.

- **Penjumlahan:** $10 + 3$ akan menghasilkan 13.
- **Pengurangan:** $10 - 3$ akan menghasilkan 7.
- **Perkalian:** $10 * 3$ akan menghasilkan 30.
- **Pembagian:** $10 / 3$ akan menghasilkan 3. Hasil ini sesuai dengan **teori pembagian integer (bilangan bulat)**. Dalam Java, ketika dua operand bertipe integer dibagi, hasilnya juga akan berupa integer, dan sisa desimalnya akan diabaikan (dipotong atau *truncated*), bukan dibulatkan.
- **Sisa Bagi (Modulus):** $10 \% 3$ akan menghasilkan 1. Operator modulus ($\%$) berfungsi untuk mencari sisa dari operasi pembagian integer. Dalam kasus ini, 10 dibagi 3 adalah 3 dengan sisa 1.

2.2 Operator Penugasan (Assignment)

Operator penugasan digunakan untuk memberikan atau menetapkan nilai kepada sebuah peubah. Program `OperatorAssignment.java` menunjukkan penggunaan operator penugasan dasar dan gabungan.

2.2.1 Uraian Kode Program

Program ini meminta dua masukan bilangan bulat dari pengguna yang disimpan dalam peubah $A1$ dan $A2$. Berbeda dari program sebelumnya, program ini mendemonstrasikan **operator penugasan gabungan** (*compound assignment operators*) seperti $+=$, $-=$, $*=$, $/=$, dan $\% =$. Operator ini melakukan operasi aritmetika dan penugasan dalam satu langkah. Peubah

A1 akan terus diperbarui nilainya setelah setiap operasi dan nilai terbaru tersebut akan digunakan pada operasi berikutnya.

2.2.2 Langkah Kerja Program

1. Program dimulai dan mendeklarasikan dua peubah integer A1 dan A2.
2. Objek Scanner dibuat untuk menerima masukan.
3. Pengguna memasukkan dua nilai integer untuk A1 dan A2.
4. Operasi $A1 += A2$; dieksekusi. Ini adalah bentuk ringkas dari $A1 = A1 + A2$; Nilai A1 yang baru kemudian ditampilkan.
5. Operasi $A1 -= A2$; dieksekusi menggunakan nilai A1 yang sudah diperbarui dari langkah sebelumnya. Hasilnya kembali disimpan di A1 dan ditampilkan.
6. Langkah serupa diulang secara berurutan untuk operator $*=$, $/=$, dan $\%=$. Setiap operasi selalu menggunakan nilai termutakhir dari peubah A1.

2.2.3 Analisis Hasil

Misal pengguna memasukkan $A1 = 15$ dan $A2 = 5$.

1. **$A1 += A2$** : A1 menjadi $15 + 5 = 20$. Keluaran: Penambahan : 20.
2. **$A1 -= A2$** : Menggunakan A1 yang baru, A1 menjadi $20 - 5 = 15$. Keluaran: Pengurangan : 15.
3. **$A1 *= A2$** : Menggunakan A1 yang baru, A1 menjadi $15 * 5 = 75$. Keluaran: Perkalian : 75.
4. **$A1 /= A2$** : Menggunakan A1 yang baru, A1 menjadi $75 / 5 = 15$. Keluaran: Pembagian : 15.
5. **$A1 \% = A2$** : Menggunakan A1 yang baru, A1 menjadi $15 \% 5 = 0$. Keluaran: Sisa Bagi : 0.

2.3 Operator Relasional

Operator relasional digunakan untuk membandingkan dua nilai dan menghasilkan sebuah nilai boolean (true atau false). Program OperatorRelasional.java menguji keenam operator relasional.

2.3.1 Uraian Kode Program

Program ini menerima dua masukan integer dari pengguna (A1 dan A2). Kemudian, program ini melakukan serangkaian perbandingan antara A1 dan A2 menggunakan operator: lebih besar dari (>), lebih kecil dari (<), lebih besar atau sama dengan (>=), lebih kecil atau sama dengan (<=), sama dengan (==), dan tidak sama dengan (!=). Hasil dari setiap perbandingan, yang selalu berupa nilai boolean, disimpan dalam peubah hasil dan ditampilkan ke layar.

2.3.2 Langkah Kerja Program

1. Program dimulai, mendeklarasikan peubah integer A1, A2, dan peubah boolean hasil.
2. Objek Scanner dibuat untuk masukan pengguna.
3. Pengguna memasukkan dua nilai integer.
4. Ekspresi $A1 > A2$ dievaluasi. Hasilnya (true atau false) disimpan ke hasil dan ditampilkan.
5. Ekspresi $A1 < A2$ dievaluasi. Hasilnya menimpa nilai sebelumnya di hasil dan ditampilkan.
6. Proses ini berlanjut untuk semua operator relasional lainnya (>=, <=, ==, !=).

2.3.3 Analisis Hasil

Misalkan pengguna memasukkan nilai $A1 = 7$ dan $A2 = 10$.

- **$A1 > A2$:** $7 > 10$ adalah false.
- **$A1 < A2$:** $7 < 10$ adalah true.
- **$A1 \geq A2$:** $7 \geq 10$ adalah false.
- **$A1 \leq A2$:** $7 \leq 10$ adalah true.
- **$A1 == A2$:** $7 == 10$ adalah false. Perlu dicatat, operator kesetaraan menggunakan dua tanda sama dengan (==) untuk membedakannya dari operator penugasan (=).
- **$A1 != A2$:** $7 != 10$ adalah true.

Sesuai **teori logika proposisional**, setiap ekspresi relasional akan menghasilkan satu dari dua kemungkinan nilai kebenaran. Operator ini merupakan fondasi untuk pengambilan keputusan dan kontrol alur dalam pemrograman, seperti pada struktur if-else dan perulangan (*looping*).

2.4 Operator Logika

Operator logika digunakan untuk mengombinasikan atau memodifikasi ekspresi boolean. Program `OperatorLogika.java` mendemonstrasikan tiga operator logika fundamental.

2.4.1 Uraian Kode Program

Program ini meminta pengguna untuk memasukkan dua nilai boolean (`true` atau `false`), yang disimpan dalam peubah `A1` dan `A2`. Kemudian, program melakukan tiga operasi logika: **Konjungsi (AND)** menggunakan operator `&&`, **Disjungsi (OR)** menggunakan operator `||`, dan **Negasi (NOT)** menggunakan operator `!`. Hasil dari setiap operasi logika disimpan dalam peubah boolean `c` dan ditampilkan ke layar.

2.4.2 Langkah Kerja Program

1. Program dimulai dan mendeklarasikan tiga peubah boolean `A1`, `A2`, dan `c`.
2. Objek `Scanner` dibuat.
3. Pengguna memasukkan dua nilai boolean (`true` atau `false`).
4. Operasi Konjungsi (`A1 && A2`) dilakukan. Hasilnya disimpan di `c` dan ditampilkan.
5. Operasi Disjungsi (`A1 || A2`) dilakukan. Hasilnya disimpan di `c` dan ditampilkan.
6. Operasi Negasi (`!A1`) dilakukan terhadap `A1`. Hasilnya disimpan di `c` dan ditampilkan.

2.4.3 Analisis Hasil

Sebagai contoh, asumsikan pengguna memasukkan `A1 = true` dan `A2 = false`.

- **Konjungsi (AND &&):** `c = true && false`; akan menghasilkan `false`. Sesuai dengan **tabel kebenaran AND**, hasilnya hanya akan `true` jika kedua operand bernilai `true`.
- **Disjungsi (OR ||):** `c = true || false`; akan menghasilkan `true`. Sesuai dengan **tabel kebenaran OR**, hasilnya akan `true` jika salah satu atau kedua operand bernilai `true`.
- **Negasi (NOT !):** `c = !A1`; atau `c = !true`; akan menghasilkan `false`. **Operator NOT** adalah operator uner yang membalikkan nilai kebenaran dari operandnya.

Operator logika ini sangat penting dalam membangun kondisi yang kompleks untuk mengontrol alur program secara presisi. Analisis menunjukkan bahwa keluaran program sepenuhnya konsisten dengan prinsip-prinsip dasar logika boolean.

BAB III

KESIMPULAN

Berdasarkan pembahasan dan analisis hasil dari program-program yang telah diimplementasikan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. **Operator Aritmetika** merupakan perangkat dasar untuk melakukan kalkulasi matematis. Implementasi dalam Java menunjukkan bahwa operator ini bekerja sesuai dengan aturan matematika universal, namun dengan kekhususan pada **pembagian integer** yang hasilnya akan selalu dibulatkan ke bawah (dipotong) tanpa menyertakan nilai desimal.
2. **Operator Penugasan**, khususnya operator penugasan gabungan (seperti += atau -=), menyediakan cara yang lebih ringkas dan efisien untuk melakukan operasi aritmetika sekaligus menetapkan hasilnya ke peubah yang sama. Praktikum ini mengonfirmasi bahwa nilai peubah bersifat dinamis dan akan terus diperbarui secara sekuensial pada setiap operasi.
3. **Operator Relasional** adalah pilar utama dalam membangun logika percabangan dan pengambilan keputusan dalam program. Setiap operasi perbandingan akan selalu menghasilkan nilai kebenaran (boolean), yaitu true atau false, yang menjadi dasar bagi komputer untuk menentukan alur eksekusi program.
4. **Operator Logika** (&&, ||, !) berfungsi untuk menggabungkan atau memodifikasi hasil dari ekspresi-ekspresi relasional. Penggunaannya memungkinkan pembentukan kondisi-kondisi yang kompleks dan bertingkat, yang sangat penting dalam penyelesaian masalah yang memerlukan logika rumit.

DAFTAR PUSTAKA

- [1] Deitel, P. J. (2018). *Java: How to Program, Early Objects*.
- [2] Horstmann, C. S. (2019). *Core Java Volume I--Fundamentals*.
- [3] Rosihan Ari Yuana, S. M. (2022). *Pemrograman Java*.