

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“OPERATOR ARITMATIKA”



disusun Oleh:

NAIRA RAMADHANI HALIL

2511533027

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T.

Asisten Praktikum:

JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “OPERATOR ARITMATIKA” dapat diselesaikan tepat waktu.

Laporan ini disusun untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran dalam memahami konsep dasar pemrograman khususnya operator aritmatika pada Bahasa Java.

Pada kesempatan ini, penulis menyampaikan terimakasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku dosen pengampu mata kuliah Algoritma dan pemrograman yang telah memberikan bimbingan dan arahan.
2. Bang Jovantri Immanuel Gulo selaku asisten praktikum kelas A yang telah membantu pelaksanaan praktikum.
3. Teman-teman mahasiswa yang telah membantu dan memberi dukungan serta berdiskusi Bersama dalam menyelesaikan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca dan menambah wawasan mengenai pemrograman Java.

i
DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI	3
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Kerja Praktikum.....	3
2.2 Analisis Hasil dan Pembahasan	7
BAB III PENUTUP.....	9
3.1 Kesimpulan	9

DAFTAR

PUSTAKA

.....

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, proses perhitungan merupakan salah satu kebutuhan dasar yang hampir selalu digunakan pada setiap aplikasi, baik sederhana maupun kompleks. Java sebagai bahasa pemrograman berorientasi objek menyediakan berbagai **operator aritmatika** yang berfungsi untuk melakukan operasi matematis seperti penjumlahan, pengurangan, perkalian, pembagian, hingga modulo. Operator-operator ini menjadi fondasi dalam pengolahan data numerik, logika program, maupun struktur kendali.

Penggunaan operator aritmatika sangat penting karena banyak algoritma dan proses komputasi membutuhkan manipulasi nilai numerik. Misalnya dalam membuat aplikasi perhitungan, simulasi, pengolahan data statistik, hingga pembuatan game. Dengan hadirnya operator aritmatika seperti $+$, $-$, $*$, $/$, dan $\%$, Java memungkinkan programmer melakukan operasi matematika secara efisien dan mudah dipahami.

Selain itu, operator aritmatika juga mendukung berbagai tipe data seperti `int`, `double`, `float`, dan `long`, sehingga dapat digunakan dalam berbagai kebutuhan program. Keberadaan operator ini membantu menyederhanakan perhitungan kompleks dan meningkatkan fleksibilitas dalam membangun aplikasi. Oleh karena itu, pemahaman yang baik tentang operator aritmatika menjadi dasar penting bagi siapa pun yang belajar Java, karena operator ini berperan dalam hampir semua aspek pemrograman, mulai dari logika dasar hingga pengembangan aplikasi berskala besar.

1.2 Tujuan Praktikum

1. Memahami fungsi dan jenis-jenis operator aritmatika yang tersedia dalam bahasa pemrograman Java, seperti penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan modulo (%).
2. Mampu mengimplementasikan operator aritmatika dalam program Java, terutama dalam melakukan perhitungan dasar pada berbagai tipe data numerik seperti int, double, dan float.
3. Melatih kemampuan dalam menyusun kode Java yang menggunakan operator aritmatika untuk menyelesaikan permasalahan sederhana, seperti perhitungan nilai, konversi satuan, atau operasi matematis lainnya.
4. Mengembangkan pemahaman mengenai prioritas dan urutan eksekusi operator (operator precedence) dalam Java agar dapat menghasilkan output yang benar sesuai logika program.
5. Mengetahui dan menguji penggunaan operator aritmatika dalam berbagai kasus praktis, termasuk operasi kombinasi, ekspresi kompleks, dan penggunaan operator dalam struktur kontrol.
6. Meningkatkan keterampilan debugging dengan mengidentifikasi dan memperbaiki kesalahan yang muncul saat menggunakan operator aritmatika dalam kode.

1.3 Manfaat Praktikum

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bekal awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.

BAB II PEMBAHASAN

2.1 Langkah Kerja Praktikum

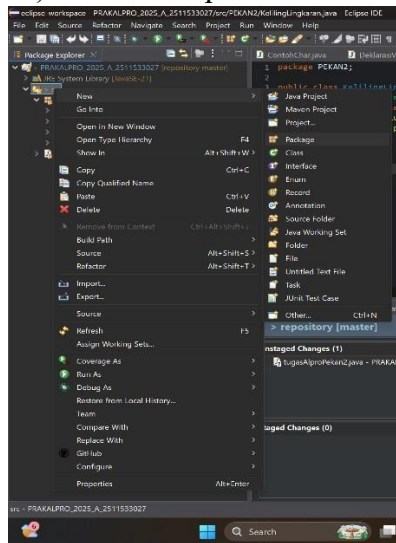
1. Persiapan Awal

a) Buka aplikasi Eclipse IDE di laptop atau di komputer.



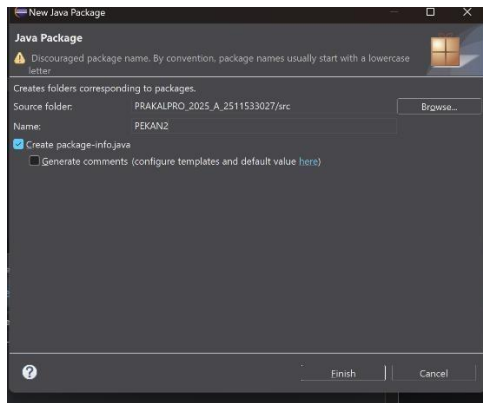
2. Membuat Package

a) Klik kanan pada folder **src**, lalu pilih **New**, terakhir klik **Package**.



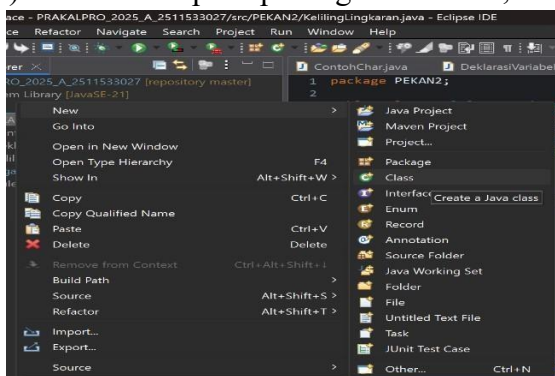
b)

Beri nama package **PEKAN8_2511533028**.



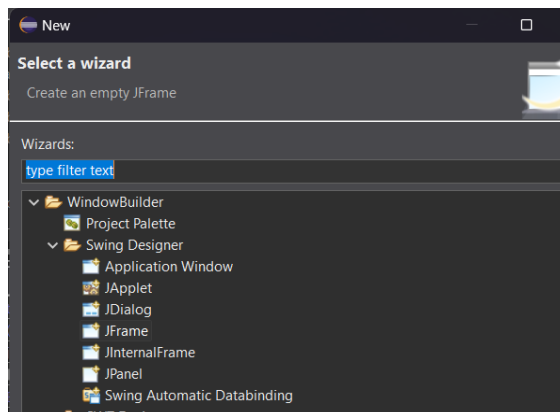
3. Membuat Class Baru untuk Setiap Percobaan

a) Klik kanan pada package **PEKAN8**, lalu pilih **New**, terakhir klik **Other**.



b) Pilih **JFrame** sesuai percobaan.

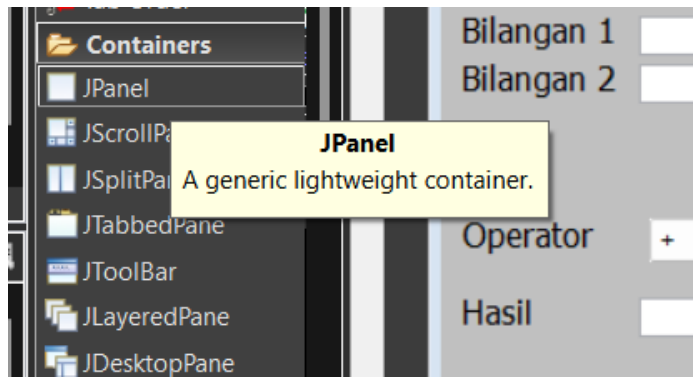
- OperatorAritmatika_2511533028



4. Menambah Collum Pada JFrame

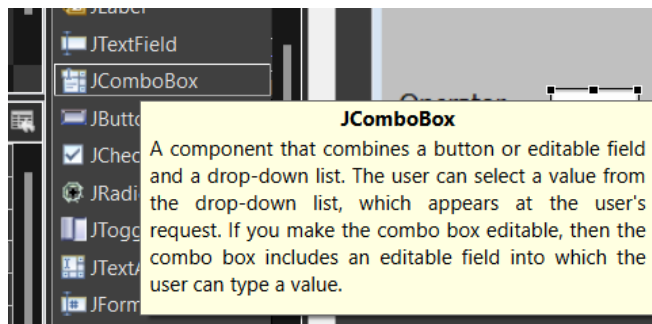
a) JPanel Pada JFrame

- Tambah JPanel sebanyak 3 buah, lalu ubah variabelnya menjadi **txtBill1**, **txtBil2**, dan **txtHasil** agar variabelnya dapat terbaca di codingan java.
- Editable dan Enabled diubah menjadi True
- Tampilan:



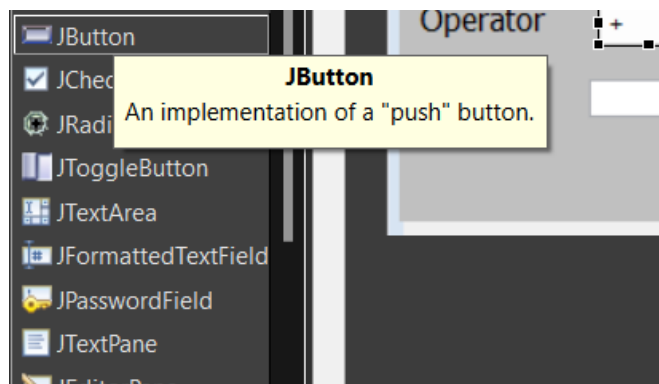
b) Menambahkan JComboBox pada JPanel

- Ubah nama variabelnya menjadi cbOperator, lalu editable menjadi false dan enabled true
- Pada frame model tambahkan logo +, -, *, /, %, masing-masing logo memiliki nilai tambah, kurang, kali, bagi, dan persenan.
- Tampilan:



c) Menambah JButton pada JFrame

- Tambahkan Text Button (Proses)
- Tekan 2 kali agar menghasilkan codingan pada operator aritmatika
- Tampilan:



d) Menambahkan codingan pada JButton

- Menambahkan **int** untuk menambahkan **txtBil1** dan **txtBil2** agar saat menginput angka dapat dibaca oleh program tersebut.
- Menambahkan nilai masing-masing model pada codingan jButton agar program dapat membaca penambahan, pengurangan, perkalian, pembagian, dan persen pada program yang akan di input.
- Sehingga menampilkan:

```

JButton btnNewButton = new JButton("Proses");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (txtBil1.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 1 harus diisi");
        } else if (txtBil2.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 2 harus diisi");
        } else {
            try {
                int a = Integer.parseInt(txtBil1.getText());
                int b = Integer.parseInt(txtBil2.getText());
                int c = cbOperator.getSelectedIndex();
                if (c == 0) {}
                if (c == 1) {}
                if (c == 2) {}
                if (c == 3) {}
                if (c == 4) {}
            } catch (NumberFormatException ex) {
                pesanError("Bilangan 1 dan Bilangan 2 harus angka");
            }
            txtHasil.setText(String.valueOf(txtHasil));
        }
    }
});
btnNewButton.setBounds(161, 152, 85, 21);
txthasil.add(btnNewButton);

```

e) Menambahkan Pesan Error pada Codingan

- Membuat dua pesan pada codingan untuk menampilkan input yang tidak valid atau error saat memasukkan angka pada program.
- Sehingga menampilkan:

```

27 private JTextField txthasil,
28
29 private void pesanPeringatan(String pesan) {
30     JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
31 }
32 private void pesanError(String pesan) {
33     JOptionPane.showConfirmDialog(this, pesan, "Error", JOptionPane.ERROR_MESSAGE);
34 }
35
36 /**
37  * Launch the application.
38  */

```

f) Codingan Hasil Program

- Setelah menambahkan collum hasil pada program, ubah codingan pada codingan button text agar terbaca pada input program.
- Sehingga menampilkan:

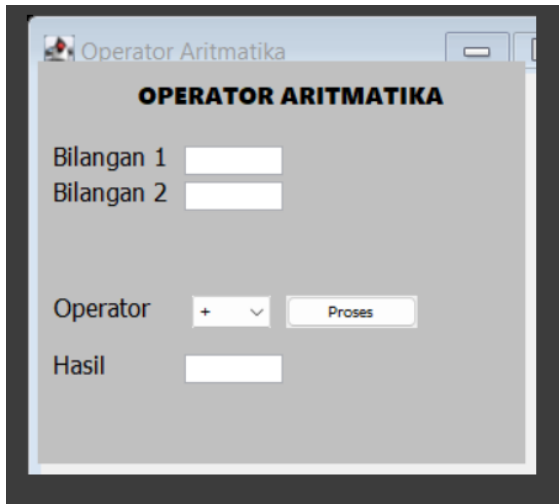
```

128         txtHasil.setText(String.valueOf(txtHasil));
129     }
130 });
131 btnNewButton.setBounds(161, 152, 85, 21);
132 txthasil.add(btnNewButton);
133
134 txthasil = new JTextField();
135 txthasil.setBounds(95, 190, 64, 19);
136 txthasil.add(txtHasil);
137 txthasil.setColumns(10);
138
139 }
140 }

```

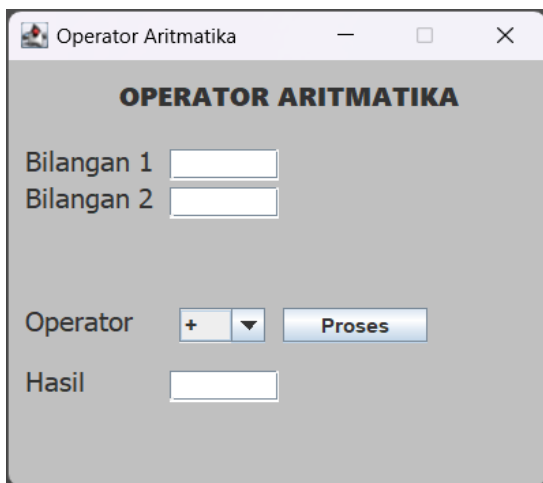
g) Tampilan Program

- Sehingga menampilkan:



5. Kompilasi dan Menjalankan Program

- a) Klik kanan pada file program, pilih **Run As**, terakhir klik **Java Application**.
- b) Lihat hasil output pada tab Console.



2.2 Analisis Hasil dan Pembahasan

Berdasarkan hasil praktikum mengenai penggunaan operator aritmatika dalam bahasa pemrograman Java, dapat dianalisis bahwa setiap operator bekerja sesuai dengan fungsi dasar matematisnya. Program yang dijalankan menunjukkan bahwa operator penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), dan modulo (%) menghasilkan output yang konsisten dengan konsep aritmatika yang berlaku.

Dari beberapa percobaan, terlihat bahwa tipe data yang digunakan sangat mempengaruhi hasil perhitungan. Misalnya, pada operasi pembagian menggunakan tipe data `int`, hasil yang diperoleh berupa bilangan bulat karena Java melakukan pembulatan terhadap hasil pembagian desimal. Namun, ketika tipe data diubah menjadi `double` atau `float`, hasil pembagian menampilkan angka desimal sesuai presisi. Hal ini menunjukkan bahwa pemilihan tipe data sangat penting untuk mendapatkan hasil perhitungan yang akurat.

Selain itu, percobaan terkait **operator precedence** memperlihatkan bahwa Java memproses operator dengan urutan prioritas tertentu. Operator perkalian dan pembagian diproses terlebih dahulu dibandingkan penjumlahan dan pengurangan. Ketika kurung digunakan, prioritas operasi berubah sesuai urutan yang ditetapkan programmer. Temuan ini menegaskan pentingnya penggunaan tanda kurung untuk menghindari kesalahan logika perhitungan dalam program.

Secara keseluruhan, hasil praktikum menunjukkan bahwa operator aritmatika dalam Java bekerja secara konsisten dan dapat digunakan sebagai dasar dalam pembuatan program yang membutuhkan proses perhitungan. Pemahaman terhadap fungsi operator, tipe data, serta prioritas eksekusi menjadi kunci dalam menghasilkan kode yang benar dan efisien. Praktikum ini juga membantu mahasiswa meningkatkan kemampuan debugging, karena beberapa kesalahan umum seperti pembagian integer, kurangnya tanda kurung, atau salah memilih tipe data dapat memengaruhi hasil akhir program.

BAB III PENUTUP

3.1 Kesimpulan

Dari hasil praktikum mengenai operator aritmatika dalam pemrograman Java, dapat disimpulkan bahwa pemahaman terhadap operator dasar seperti $+$, $-$, $*$, $/$, dan $\%$ sangat penting dalam proses perhitungan dan pengolahan data numerik. Percobaan yang dilakukan menunjukkan bahwa setiap operator bekerja sesuai fungsi matematisnya, dan hasil perhitungan sangat dipengaruhi oleh tipe data yang digunakan.

Selain itu, penguasaan terhadap **prioritas operator (operator precedence)** menjadi hal yang krusial untuk memastikan bahwa program menghasilkan output yang benar. Penggunaan tanda kurung juga terbukti membantu mengendalikan alur perhitungan agar sesuai dengan logika yang diinginkan. Praktikum ini menegaskan bahwa operator aritmatika merupakan dasar penting dalam penulisan program Java, serta menjadi fondasi untuk mempelajari konsep pemrograman yang lebih kompleks.

Secara keseluruhan, mahasiswa diharapkan mampu memahami, mengimplementasikan, serta menganalisis operator aritmatika dengan benar dalam berbagai situasi pemrograman.

DAFTAR PUSTAKA

- [1] Wahyudi, *Operator Aritmatika*, presentasi kuliah, Program Studi Informatika, Universitas Andalas, 2025.
- [2] Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program (10th Edition)*. Pearson Education.
- [3] Oracle. (2024). *The Java® Tutorials – Operators*. Oracle DocumentationGeeksforGeeks. “Loops in Java,” 2024.
- [Online] Tersedia di: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>
- [4] Schildt, H. (2018). *Java: The Complete Reference (11th Edition)*. McGraw-Hill Education.
- [5] Wahana Komputer. (2020). *Pemrograman Java untuk Pemula*. Andi Publisher.
- [6] Sujarwo, Y. (2019). *Belajar Java Dasar*. Informatika Bandung.