

```
1 # Laporan Modul 2: Dasar Pemrograman Java
2 **Mata Kuliah:** Praktikum Pemrograman Berorientasi
  Objek
3 **Nama:** DWI IDA RAHMADANI SITUORANG
4 **NIM:** 2024573010028
5 **Kelas:** TI 2E
6
7 ---Gunakan fitur "Export to PDF" atau "Print to PDF"
  dari editor tersebut.
8
9 Simpan file PDF tersebut, biasanya di folder yang
  sama dengan laporan.md atau di root folder laporan
  Anda.
10
11 ## 1. Abstrak
12 Laporan ini merangkum hasil implemntasi pratikum
  modul 2 tentang Dasar Pemograman Java.
13 tujuannya adalah mendemonstrasikan pemahaman dan
  penguasaan konsep fundamental java, meliputi
  variabel dan tipe data,
14 mekanisme input/output menggunakan class scanner,
  serta implementasi berbagai struktur kontrol
  percabangan
15 (if-else, switch) dan perulangan (for, ehile, do-
  while, dan nested loop).
16 setiap pratikum berhasil diujicoba dan dianalisis
  untuk memastikan program berjalan sesuai logika yang
  dirancang.
17
18 ---
19 ## 2. Praktikum
20 ### Praktikum 1 - Variabel dan Tipe Data
21 #### Dasar Teori
22 java membagi tipe data menjadi dua: primitif (int,
  double, boolean, char, dll.) yang menyimpan nilai
  aktual,
23 dan reference (String, Array, dll.).
24 yang menyimpan alamat memori ke objek. variabel harus
  dinamai sesuai aturan dan bersifat case-sensitive.
25 pratikum ini juga menguji oprasi aritmatika dan
  casting (konversi eksplisit) antar tipe data.
```

26 ##### Langkah Praktikum

27 1. membuat file VariabelDemo.java dan mendeklarasikan variabel dengan berbagai tipe data, termasuk string nama = "DWI IDA RAHMADANI SITUMORANG";.

28 2. melakukan oprasi aritmatika (+, -, \*, /, %) menggunakan variabel integer.

29 3. mendemonstarasikan cascing dari double ke int dan automatic promotion tipe data.

30 ##### Screenshoot Hasil

31 ![output program VariabelDemo](screenshots/P1\_VariabelDemo\_Output.png)

32 ##### Analisa dan Pembahasan

33 variabel data diri (nama, umur, tinggi) berhasil diinsialisasi. oprasi artimatika dasar menunjukkan bahwa hasil

34 pembagian int / int (misalnya 10/3) menghasilkan pembulatan ke bawah (floor) menjadi 3.

35 proses casting darai double (9.8) menjadi int (9) berhasil dilakukan, namun menunjukkan adanya kehilangan presisi,

36 membuktikan konversi eksplisit diperlukan saat mengubah tipe data dengan ukuran atau jangkauan yang lebih kecil.

37

38 ### Praktikum 2 - Input, Output dan Scanner

39 ##### Dasar Teori

40 Class scanner darai package java.ulti digunakan untuk menerima masukan dari pengguna.

41 method yang digunakan seperti nextLine() untuk membaca string lengkap dan nextInt() atau nextDouble() untuk membaca

42 nilai numerik.

43 ##### Langkah Praktikum

44 1. membuat file InputOutputDemon.java dan mengimpor java.ulti.scanner.

45 2. menggunakan scanner untuk membaca nama (nextLine() ), umur (nextInt() ), dan tinggi (nextDouble() ).

46 3. menampilkan kembali data yang dimasukkan dalam format yang terstruktur.

47 ##### Screenshoot Hasil

48 ![output program InputOutputDemon](screenshots/

48 P2\_InputOutputDemon\_Output.png)

49 ##### Analisa dan Pembahasan

50 program berhasil menjalankan fungsi input-output secara interaktif.

51 pengguna nextLine() untuk nama menjamin nama lengkap yang mengandung spasi dapat terbaca sempurna.

52 semua data yang diinputkan disimpan ke variabel dengan tipe data yang sesuai (string, int, double)

53 dan ditampilkan kembali, menunjukkan program dapat berintraksi dengan pengguna secara efektif.

54

55 ### Praktikum 3 - Program Penentuan Grade (if-else if-else)

56 ##### Dasar Teori

57 struktur percabangan if-else if-else digunakan untuk membuat keputusan multi-cabang.

58 program akan mengevaluasi kondisi secara berurutan, dan hanya blok kode pertama yang kondisinya true yang akan

59 dieksekusi, mengabaikan sisanya.

60 ##### Langkah Praktikum

61 1. membuat file GradeDemo.java dan menerima input nilai angka.

62 2. mengimplementasikan logika penilaian menggunakan if-else if-else untuk menetapkan grade ('A' hingga 'E').

63 ##### Screenshoot Hasil

64 ![output program GradeDemo](screenshots/P3\_GradeDemo.png)

65 ##### Analisa dan Pembahasan

66 logika pemrograman berjalan sesuai hierarki.

67 contohnya, jika nilai 88 dimasukkan, kondisi (nilai >=92) langsung bernilai true, menetapkan grade 'A',

68 dan program melompati pemeriksaan kondisi (nilai >=75), membuktikan keefektifan if-else if-else untuk penilaian berjenjang.

69

70 ### Praktikum 4 - Pemrograman Menu Dengan Switch

71 ##### Dasar Teori

72 struktur switch digunakan untuk memilih satu blok kode yang akan dieksekusi dari banyak pilihan case berdasarkan

```
73 nilai pasti sebuah variabel.
74 penggunaan keyword break sangat krusial untuk
    mencegah fall-through ke case berikutnya.
75 ##### Langkah Praktikum
76 1. membuat file MenuDemo.java.
77 2. menampilkan menu pilihan (1-4).
78 3. menggunakan switch untuk menjalankan kalkulasi
    luas (persegi, lingkaran, segitiga) berdasarkan
    pilihan.
79 ##### Screenshoot Hasil
80 ![output program MenuDemo](screenshots/P4_MenuDemo.
    png)
81 ##### Analisa dan Pembahasan
82
83 ### Praktikum 5 - Pemograman Nested if
84 ##### Dasar Teori
85 nested if adalah struktur if di dalam blok if
    lainnya, digunakan untuk implementasi logika
    keputusan yang lebih
86 kompleks dan berlapis, seringkali untuk validasi
    bertingkat.
87 ##### Langkah Praktikum
88 1. membuat file NestedIfDemo.java dan menerima input
    umur.
89 2. menggunakan if luar untuk memverifikasi (umur >=0
    ).
90 3. menggunakan if-else if-else bersarang di dalam if
    luar untuk mentukan katagori usia (bayi, balita,
    dst.).
91 ##### Screenshoot Hasil
92 ![output program NestedIfDemo](screenshots/
    P5_NestedIfDemo.png)
93 ##### Analisa dan Pembahasan
94 struktur if bersarang sangat efektif untuk validasi
    .
95 if luar berhasil memfilter input yang tidak valid (
    umur negatif), mengarahkan program langsung ke blok
    else luar.
96 hanya umur yang valid yang kemudian diuji oleh
    netsted if-else if-else untuk dikategorikan,
    menunjukkan struktur yang
```

```
97 logis dan aman.
98
99 ### Praktikum 6 - Perulangan For
100 #### Dasar Teori
101 for loop digunakan untuk perulangan di mana jumlah
    iterasi suda diketahui.
102 struktur terdiri dari inisialisasi, kondisi
    terminasi, dan increment/decrement.
103 #### Langkah Praktikum
104 1. membuat file ForLoopDemo.java.
105 2. mendemonstrasikan perulangan sederhana (1-10),
    perulangan dengan step (genap i+=2), countdown (i
    --), dan pembuatan tabel perkalian.
106 #### Screenshoot Hasil
107 ![output program ForLoopDemon](screenshots/
    P6_ForLoopDemon.png)
108 #### Analisa dan Pembahasan
109 for loop berhasil digunakan untuk semua contoh.
    penggunaan i += 2 pada contoh 2 dan i-- pada contoh
    3
110 menunjukkan kontrol penuh atas laju perulangan.
111 contoh 4 menunjukkan kemampuan for loop untuk
    menghasilkan output yang terstruktur (tabel
    perkalian)
112 melalui oprasi matematika di dalam blok perulangan.
113
114 ### Praktikum 7 - Perulangan While dan Do-While
115 #### Dasar Teori
116 while loop mengulang selama kondisi true, dan dengan
    pemeriksaan di awal.
117 do-while loop menjamin eksekusi blok kode minimal
    satu kali, karena kondsi diperiksa diakhir
    perulangan.
118 #### Langkah Praktikum
119 1. membuat file WhileLoopDemo.java.
120 2. mengimplementasikan permainan tebak angka
    menggunakan while loop yang trus berjalan hinga
    tebakan benar.
121 3. mengimplemnatsikan sistem menu menggunakan do-
    while loop yang menjamin menu tampilan minimal
    sekali.
```

```
122 ##### Screenshoot Hasil
123 ![output program WhileLoopDemo](screenshots/
    P7_WhileLoopDemon.png)
124 ##### Analisa dan Pembahasan
125 pada tebak angka, while loop efektif karena
    perulangan hanya berhenti setelah kondisi kemenangan
    tercapai.
126 untuk menu, do-while loop memastikan pengguna selalu
    melihat menu utama terlebih dahulu (eksekusi
    minimal satu kali)
127 sebelum kondisi keluar (pilihan !=3) diperiksa untuk
    navigasi menu.
128
129 ### Praktikum 8 - Nested Loop (perulangan bersarang)
130 ##### Dasar Teori
131 Nested Loop adalah perulangan di dalam perulangan
    lain, ideal untuk memproses pola dua dimensi,
132 di mana loop luar mengontrol baris dan loop dalam
    mengontrol kolom atau elemen baris.
133 ##### Langkah Praktikum
134 1. membuat file NestedLoopDemo.java.
135 2. mencetak pola bintang segitiga dan pola angka
    segitiga.
136 3. mencetak tabel perkalian 1-5.
137 ##### Screenshoot Hasil
138 ![output program NestedLoopDemo](screenshots/
    P8_NestedLoopDemo.png)
139 ##### Analisa dan Pembahasan
140 Pola berhasil dibuat menggunakan perulangan
    bersarang. Pada Pola Segitiga, loop dalam (j)
    dibatasi oleh nilai
141 iterasi loop luar (i), menghasilkan penambahan
    elemen di setiap baris.
142 Tabel Perkalian juga menunjukkan penggunaan nested
    loop secara efisien untuk menghasilkan matriks data
    terstruktur 5x5.
143
144 ---
145
146 ## 3. Kesimpulan
147
```

148 Praktikum Modul 2 tentang Dasar Pemrograman Java  
telah berhasil diselesaikan. Implementasi yang  
sukses dari variabel,  
149 tipe data, input/output (Scanner), percabangan (if-  
else, switch), dan perulangan (for, while, do-while  
, nested loop)  
150 menunjukkan pemahaman yang kuat tentang konsep  
pemrograman prosedural. Penguasaan struktur kontrol  
ini adalah fondasi  
151 penting untuk merancang alur logika program yang  
efisien dan interaktif.  
152  
153 ---  
154  
155 ## 5. Referensi  
156 1. Java Documentation: Control Flow Statements –  
[https://docs.oracle.com/javase/tutorial/java/  
nutsandbolts/flow/](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow/)  
157 2. Java Documentation: Variables and Data Types –  
[https://docs.oracle.com/javase/tutorial/java/  
nutsandbolts/datatypes.html](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html)  
158  
159 ---  
160