



JOBSHEET I

Programming Fundamentals Review

1. Learning Outcome

After completing this session, students must be able to:

1. Implement selection, looping, arrays, and functions in Java programming language.

2. Labs Activity

2.1 Selection

Times: 50 minutes

The material covered in this laboratory session has been explained actually in the Fundamentals Programming course, in semester 1. Therefore, no experimental steps will be conducted in this laboratory session. Answer the following questions:

2.1.1 Questions

1. Create a program to calculate the final grade of a student with the following conditions: 20% for assignments, 20% for quizzes, 30% for mid-term exams, and 40% for final exams. Each input grade should be within the range of 0 to 100. If the user inputs a value outside this range, the output will be "invalid value". Once the final grade is obtained, proceed with the grade conversion according to the following criteria:

Nilai Angka	Nilai Mutu		
	Nilai Huruf	Nilai Setara	Kualifikasi
$80 < N \leq 100$	A	4	Sangat Baik
$73 < N \leq 80$	B+	3,5	Lebih dari Baik
$65 < N \leq 73$	B	3	Baik
$60 < N \leq 65$	C+	2,5	Lebih dari Cukup
$50 < N \leq 60$	C	2	Cukup
$39 < N \leq 50$	D	1	Kurang
$N \leq 39$	E	0	Gagal

If the letter grades are A, B+, B, C+, or C, the student passes. If the letter grade is D or E, the student fails.

- The input consists of **assignments**, **quiz**, **mid-term exams**, and **final exams**.
- The program will output "**invalid grade**" if the entered grade is outside the specified range.
- The program will output the **final grade result**, **letter grade**, and **pass/fail status**.



Example of Program Output:

```

Program Menghitung Nilai Akhir
=====
Masukkan Nilai Tugas: 85
Masukkan Nilai Kuis: 90
Masukkan Nilai UTS: 120
Masukkan Nilai UAS: 70
=====
nilai tidak valid
=====
Mamluatuls-MacBook-Air:Praktikum

Program Menghitung Nilai Akhir
=====
Masukkan Nilai Tugas: 90
Masukkan Nilai Kuis: 40
Masukkan Nilai UTS: 75
Masukkan Nilai UAS: 85
=====
nilai akhir : 74.0
Nilai Huruf :B+
=====
SELAMAT ANDA LULUS

```

2.2 Looping

Times: 50 minutes

The material in this session has been explained in the Basic Programming course. So, in this session, there are no experimental steps will be carried out. Just directly answer the following questions:

2.2.1 Looping Questions:

1. Write a program that can output a series of numbers from **1** to **n** except numbers **6** and **10**, **odd numbers** will be printed with an asterisk "*", while **even numbers** will be printed the number itself, with **n** = the **last 2 digits** of your NIM. If **n<10** then add **10** (n+=10)

Example 1:

Input NIM: 2341720102 then n=12 (the last 2 digit is 02 then add 10)

OUTPUT: * 2 * 4 * * 8 * * 12

Example 2:

Input NIM: 2341720113 maka n=13

OUTPUT: * 2 * 4 * * 8 * * 12

Example of Program Output

```

Masukkan Nim :2341720102
=====
n : 12
* 2 * 4 * * 8 * * 12

```

2.3 Array

Times: 50 minutes

The material in this session has been explained in the Basic Programming course. So, in this session, there are no experimental steps that will be carried out. Just directly answer the following questions:

2.3.1 Array Questions:



1. Write a program to calculate the **IP Semester** for the courses you took last semester. The formula for calculating **IP Semester** as follows:

$$IP\ Semester = \frac{\sum_i (Nilai\ Setara_i * bobot\ SKS_i)}{\sum SKS}$$

Nilai setara follows the following rules:

Nilai Angka	Nilai Mutu		
	Nilai Huruf	Nilai Setara	Kualifikasi
80 < N ≤ 100	A	4	Sangat Baik
73 < N ≤ 80	B+	3,5	Lebih dari Baik
65 < N ≤ 73	B	3	Baik
60 < N ≤ 65	C+	2,5	Lebih dari Cukup
50 < N ≤ 60	C	2	Cukup
39 < N ≤ 50	D	1	Kurang
N ≤ 39	E	0	Gagal

Input from the program is the **courseName**, the **bobotSKS**, and the **letter grade** of the course.

Example for Program Output:

```

=====
Program Menghitung IP Semester
=====
masukkan nilai Angka untuk MK Pancasila: 75
masukkan nilai Angka untuk MK Konsep Teknologi Informasi: 85
masukkan nilai Angka untuk MK Critical Thinking dan Problem Solving: 70
masukkan nilai Angka untuk MK Matematika Dasar: 85
masukkan nilai Angka untuk MK Bahasa Inggris: 85
masukkan nilai Angka untuk MK Dasar Pemrograman: 62
masukkan nilai Angka untuk MK Praktikum Dasar Pemrograman: 62
masukkan nilai Angka untuk MK Keselamatan dan Kesehatan Kerja: 85
=====
hasil Konversi Nilai
=====
MK                               Nilai Angka   Nilai Huruf   Bobot Nilai
Pancasila                        75.00         B+            3.50
Konsep Teknologi Informasi       85.00         A             4.00
Critical Thinking dan Problem Solving 70.00         B             3.00
Matematika Dasar                 85.00         A             4.00
Bahasa Inggris                   85.00         A             4.00
Dasar Pemrograman                62.00         C+            2.50
Praktikum Dasar Pemrograman      62.00         C+            2.50
Keselamatan dan Kesehatan Kerja  85.00         A             4.00
=====
IP : 3.42

```

2.4 Function

Times: 50 minutes

The material in this session has been explained in the Basic Programming course. So, in this session, there are no experimental steps that will be carried out. Just directly answer the following questions:

2.4.1 Function Questions:

RoyalGarden is a flower shop with many branches. In everyday Flower Stock in every branch listed as follows:



Row = Branch, Column = Flower stock in a day

	Aglonema	Keladi	Alocasia	Mawar
RoyalGarden 1	10	5	15	7
RoyalGarden 2	6	11	9	12
RoyalGarden 3	2	10	10	5
RoyalGarden 4	5	7	12	9

The price of Aglonema =75.000, Keladi = 50.000, Alocasia =60.000, Mawar =10.000.

1. Create a function to display the income of each branch if all the flowers are sold out.
2. Create a function to find out the number of stock for each type of flower on the royalgarden branch 4. If there is additional information in the form of a reduction in stock because the flower dies. With details of Aglonema -1, Keladi -2, Alocasia -0, Mawar -5.

3. Assignments

Times: 100 minutes

1. Write a program to create two arrays with the contents as follows. The first array is a one-dimensional char array CODE[10], containing the car plate code. The second array, the two-dimensional array char CITY[10][12] contains the city name paired with the car plate code.

An illustration of the array display is as follows:

A	B	A	N	T	E	N						
B	J	A	K	A	R	T	A					
D	B	A	N	D	U	N	G					
E	C	I	R	E	B	O	N					
F	B	O	G	O	R							
G	P	E	K	A	L	O	N	G	A	N		
H	S	E	M	A	R	A	N	G				
L	S	U	R	A	B	A	Y	A				
N	M	A	L	A	N	G						
T	T	E	G	A	L							

When the user inputs the number plate code, the program will output the city name from the number plate code.

2. Create a program to calculate the **volume**, **surface area**, and **perimeter** (total edge length) of a cube. The program should include the following functions:
 - a) Display a menu and accept user input to choose the calculation type (volume, surface area, or perimeter).
 - b) Calculate and return the **volume** of the cube.



- c) Calculate and return the **surface area** of the cube.
- d) Calculate and return the **perimeter** (total edge length) of the cube.

Call these functions in the **main** function to execute the program.

3. Create a program that allows users to input data for **n** courses. For each course, the following information should be provided: Course Name (string), Credit Hours (SKS) (integer), Semester (integer), Lecture Day (string, e.g., "Monday", "Tuesday", etc.). Each piece of data (course name, credit hours, semester, and lecture day) should be stored in separate one-dimensional arrays. The program should be able to:
 - a. **Display Course Schedule** – The program allows users to view the course schedule based on their selection:
 - i. Display **all courses**.
 - ii. Display courses scheduled on a **specific day** (e.g., only Tuesday's courses).
 - iii. Display courses for a **specific semester**.
 - b. **Search for a Course** – The program enables users to search for a course by entering its name and then displays the relevant course details.