

Laporan Tugas 1 Kelompok 10
Algoritma Pemrograman Lanjut



Kelompok 10
Disusun oleh:

Anugerah Fakhriza Reswara (2509106025)

Muhammad Riyadh Zuhair (2509106029)

Nasha Nuron Wahdah (2509106037)

Kelas (A'25)

PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

2025

DAFTAR ISI

DAFTAR ISI.....	2
BAB 1.....	1
PEMBAHASAN SOAL 1 - 5.....	1
1.1 Penjelasan SOAL 1.....	1
1.1.1 Pseudocode.....	2
1.1.2 Flowchart.....	4
1.1.3 Algoritma.....	5
1.1.4 Screenshot Program.....	6
Gambar 1.1.4.1 Percobaan pertama.....	6
Gambar 1.1.4.2 Percobaan kedua.....	6
1.2 PENJELASAN SOAL 2.....	7
1.2.1 Pseudocode.....	8
1.2.2 Flowchart.....	9
1.2.3 Algoritma.....	10
1.2.4 Screenshot Program.....	11
Gambar 1.2.4.1 Percobaan pertama.....	11
Gambar 1.2.4.2 Percobaan kedua.....	11
1.3 PENJELASAN SOAL 3.....	12
1.3.1 Pseudocode.....	12
1.3.2 Flowchart.....	13
1.3.3 Algoritma.....	14
1.3.4 Screenshot Program.....	15
Gambar 1.3.4.1 Keseluruhan Program.....	15
1.4 PENJELASAN SOAL 4.....	15
1.4.1 Pseudocode.....	16
1.4.2 Flowchart.....	17
1.4.3 Algoritma.....	18
1.4.4 Screenshot Program.....	18
Gambar 1.4.4.1 Percobaan pertama.....	18
Gambar 1.4.4.1 Percobaan kedua.....	18
1.5 PENJELASAN SOAL 5.....	19
1.5.1 Pseudocode.....	19
1.5.2 Flowchart.....	20
1.5.3 Algoritma.....	21

1.5.4 Screenshot Program.....	21
Gambar 1.4.4.1 Keseluruhan Program.....	21
BAB 2.....	22
PEMBAHASAN TUGAS A.....	22
2.1 PENJELASAN TUGAS A.....	22
2.1.1 Pseudocode.....	23
2.1.2 Flowchart.....	24
2.1.3 Algoritma.....	24
2.1.4 Screenshot Program.....	25
Gambar 2.1.4.1 Tidak bisa membentuk segitiga.....	25
Gambar 2.1.4.2 Segitiga Siku-Siku.....	25
Gambar 2.1.4.3 Segitiga Sama Sisi.....	25
Gambar 2.1.4.4 Segitiga Sama Kaki.....	26
Gambar 2.1.4.5 Segitiga Sembarang.....	26
BAB 3.....	27
PEMBAHASAN TUGAS B.....	27
3.1 PENJELASAN TUGAS B.....	27
3.1.1 Pseudocode.....	27
3.1.2 Flowchart.....	28
3.1.3 Algoritma.....	29
3.1.4 Screenshot Program.....	30
Gambar 3.1.4.1 Bukan triple pythagoras.....	30
Gambar 3.1.4.2 Triple pythagoras.....	30
Gambar 3.1.4.2 Triple pythagoras.....	30
BAB 4.....	31
PEMBAHASAN TUGAS C.....	31
4.1 PENJELASAN SOAL C.....	31
4.1.1 Pseudocode.....	31
4.1.2 Flowchart.....	32
4.1.3 Algoritma.....	33
4.1.4 Screenshot Program.....	33
Gambar 4.1.4.1 Percobaan pertama.....	33
Gambar 4.1.4.1 Percobaan kedua dengan pangkat negatif.....	33
Gambar 4.1.4.1 Percobaan kedua dengan bilangan negatif.....	33
4.2 PENJELASAN TUGAS ARRAY SATU DIMENSI.....	34
4.2.1 Pseudocode.....	34
4.2.2 Flowchart.....	35

4.2.3 Algoritma.....	36
4.2.4 Screenshot Program.....	36
Gambar 4.2.4.1 Pilihan barang.....	36
Gambar 4.2.4.2 Jika memilih semua.....	37
Gambar 4.2.4.3 Jika memilih Indomie dan Sosis.....	37
BAB 5.....	38
KESIMPULAN.....	38
5.1 FOKUS UTAMA DAN METODOLOGI LAPORAN.....	38
5.2 EKSPLORASI KONSEP REKURSI (STUDI KASUS: PERPANGKATAN).....	38
5.3 IMPLIKASI DAN KESIMPULAN UMUM.....	39
DAFTAR PUSTAKA.....	40

BAB 1

PEMBAHASAN SOAL 1 - 5

1.1 Penjelasan SOAL 1

Program ini dirancang untuk menerima input berupa matriks dua dimensi, kemudian melakukan analisis data sederhana untuk menemukan nilai terbesar, terkecil, dan rata-rata dari seluruh elemen yang dimasukkan.

1. Tujuan Program

- **Input Data:** Mengambil data angka dalam bentuk tabel (baris x kolom).
- **Analisis Data:** Menghitung **Maksimum**, **Minimum**, dan **Rata-rata**.
- **Output:** Menampilkan hasil analisis kepada pengguna.

2. Langkah-Langkah Logika (Algoritma)

Secara garis besar, program bekerja dalam empat tahap utama:

Tahap 1: Inisialisasi Dimensi

Program meminta pengguna menentukan ukuran matriks (berapa jumlah baris dan kolomnya). Ukuran ini akan menentukan seberapa banyak data yang perlu dimasukkan.

Tahap 2: Input Elemen Matriks

Program menggunakan perulangan bersarang (*nested loop*):

1. Loop pertama untuk **baris**.
2. Loop kedua untuk **kolom**.
Setiap posisi diisi dengan angka yang diketik oleh pengguna.

Tahap 3: Pemrosesan Data

Setelah data terkumpul, program menyisir kembali seluruh isi matriks untuk:

- **Mencari Max:** Membandingkan setiap angka. Jika ditemukan angka yang lebih besar dari maxVal saat ini, maka maxVal diperbarui.
- **Mencari Min:** Sama seperti mencari max, jika ditemukan angka yang lebih kecil dari minVal, maka minVal diperbarui.
- **Menjumlahkan Total:** Menambahkan setiap angka ke variabel total untuk keperluan menghitung rata-rata.

Tahap 4: Kalkulasi Akhir & Output

Program menghitung rata-rata dengan rumus:

$$\text{Rata-rata} = \frac{\text{Total Jumlah}}{\text{Jumlah Baris} \times \text{Jumlah Kolom}}$$

Lalu, semua hasil ditampilkan ke layar.

3. Komponen Penting dalam Kode

- matriks[baris][kolom]: Array 2D untuk menyimpan data.
- maxVal & minVal: Diinisialisasi dengan elemen pertama (matriks[0][0]) sebagai titik acuan awal perbandingan.
- double total: Menggunakan tipe data double agar hasil pembagian rata-rata bisa memiliki angka desimal yang akurat.

Bagian anggota kelompok yang mengerjakan tugas No. 1

Pseudocode & Algoritma: **Anugerah Fakhri Reswara (2509106025)**

Flowchart: **Nasha Nuron Wahdah (2509106037)**

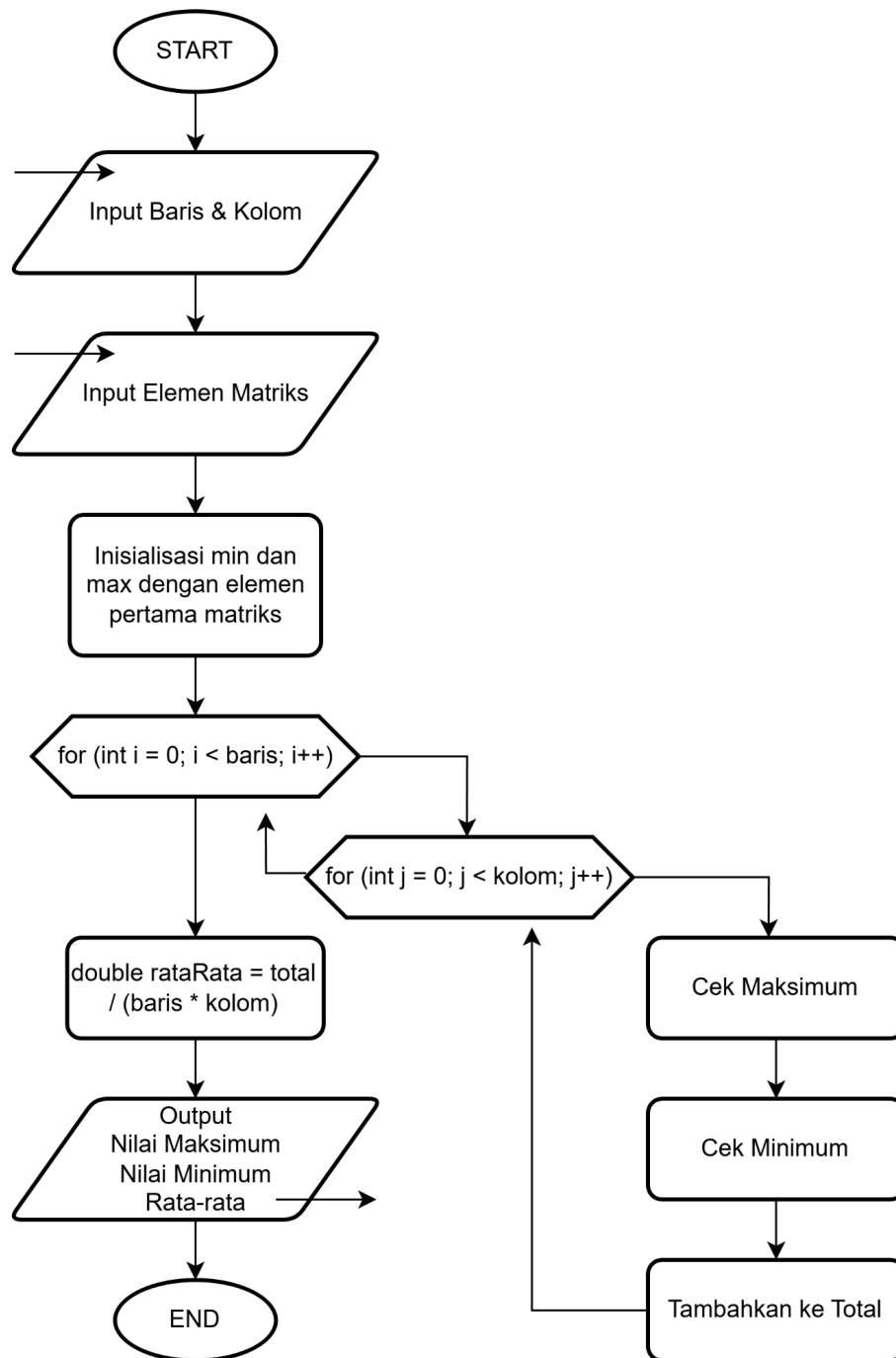
Code: **Muhammad Riyadh Zuhair (2509106029)**

1.1.1 Pseudocode

JUDUL: Statistik Matriks Dinamis
DEKLARASI: VAR baris, kolom, i, j : INTEGER VAR matriks[baris][kolom] : INTEGER VAR maxVal, minVal : INTEGER VAR total, rataRata : DOUBLE
DESKRIPSI: READ baris, kolom FOR i = 0 TO baris - 1 DO: FOR j = 0 TO kolom - 1 DO: READ matriks[i][j] ENDFOR ENDFOR SET maxVal = matriks[0][0] SET minVal = matriks[0][0] SET total = 0

```
FOR i = 0 TO baris - 1 DO:
  FOR j = 0 TO kolom - 1 DO:
    IF (matriks[i][j] > maxVal) THEN SET maxVal = matriks[i][j]
    IF (matriks[i][j] < minVal) THEN SET minVal = matriks[i][j]
    total = total + matriks[i][j]
  ENDFOR
ENDFOR
rataRata = total / (baris * kolom)
DISPLAY maxVal, minVal, rataRata
```

1.1.2 Flowchart



1.1.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Konfigurasi	INPUT baris dan INPUT kolom.
2	Input Matriks	FOR i, j lakukan INPUT matriks[i][j].
3	Penentuan Patokan	SET maxVal = matriks[0][0] dan minVal = matriks[0][0].
4	Analisis Data	FOR i, j lakukan perbandingan IF untuk mencari Max/Min dan hitung total += matriks[i][j].
5	Perhitungan	SET rataRata = total / (baris * kolom).
6	Penyajian	PRINT "Nilai Maksimum", "Nilai Minimum", "Rata-rata".
7	Selesai	RETURN 0 (Program berakhir).

1.1.4 Screenshot Program

```
=== Program Statistik Matriks ===
Masukkan jumlah baris: 2
Masukkan jumlah kolom: 3

Masukkan elemen matriks:
Matriks[0][0]: 6
Matriks[0][1]: 7
Matriks[0][2]: 8
Matriks[1][0]: 1
Matriks[1][1]: 5
Matriks[1][2]: 3

--- Hasil Analisis ---
Nilai Maksimum : 8
Nilai Minimum  : 1
Rata-rata      : 5
```

Gambar 1.1.4.1 Percobaan pertama

```
=== Program Statistik Matriks ===
Masukkan jumlah baris: 2
Masukkan jumlah kolom: 2

Masukkan elemen matriks:
Matriks[0][0]: -2
Matriks[0][1]: 10
Matriks[1][0]: 3
Matriks[1][1]: -33

--- Hasil Analisis ---
Nilai Maksimum : 10
Nilai Minimum  : -33
Rata-rata      : -5.5
```

Gambar 1.1.4.2 Percobaan kedua

1.2 PENJELASAN SOAL 2

Program ini dirancang untuk menghitung **invers dari sebuah matriks berukuran 2x2**. Secara sederhana, program akan meminta pengguna memasukkan empat angka sebagai elemen matriks, lalu menghitung kebalikannya menggunakan rumus matematika formal.

Rumus Utama yang Digunakan

Program bekerja berdasarkan rumus invers matriks berikut:

Jika matriks $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, maka inversnya adalah:

$$A^{-1} = \frac{1}{(ad - bc)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Langkah-Langkah Kerja Program

1. **Input Data:** Program meminta pengguna memasukkan nilai untuk elemen a,b,c, dan d.
2. **Menampilkan Matriks:** Program menampilkan kembali susunan matriks yang telah diinput agar pengguna bisa memeriksa datanya.
3. **Menghitung Determinan:** Program menghitung nilai determinan dengan rumus $(a \times d) - (b \times c)$.
4. **Percabangan (Kondisi):**
 - o **Jika Determinan = 0:** Program akan berhenti dan memberi tahu bahwa matriks tidak memiliki invers (disebut matriks singular).
 - o **Jika Determinan $\neq 0$:** Program lanjut menghitung elemen baru dengan menukar posisi a dan d, mengubah tanda b dan c (menjadi negatif), lalu membagi semuanya dengan hasil determinan.
5. **Output:** Program menampilkan hasil matriks yang sudah dibalik (invers).

Tabel Logika Perhitungan

Elemen	Proses Invers	Hasil Akhir
a	Ditukar dengan d , lalu bagi determinan	invers A
b	Diubah jadi negatif (-b), lalu bagi determinan	invers B
c	Diubah jadi negatif (-c), lalu bagi determinan	invers C
d	Ditukar dengan a , lalu bagi determinan	invers D

Bagian anggota kelompok yang mengerjakan tugas No. 2

Pseudocode & Algoritma: **Muhammad Riyadh Zuhair (2509106029)**

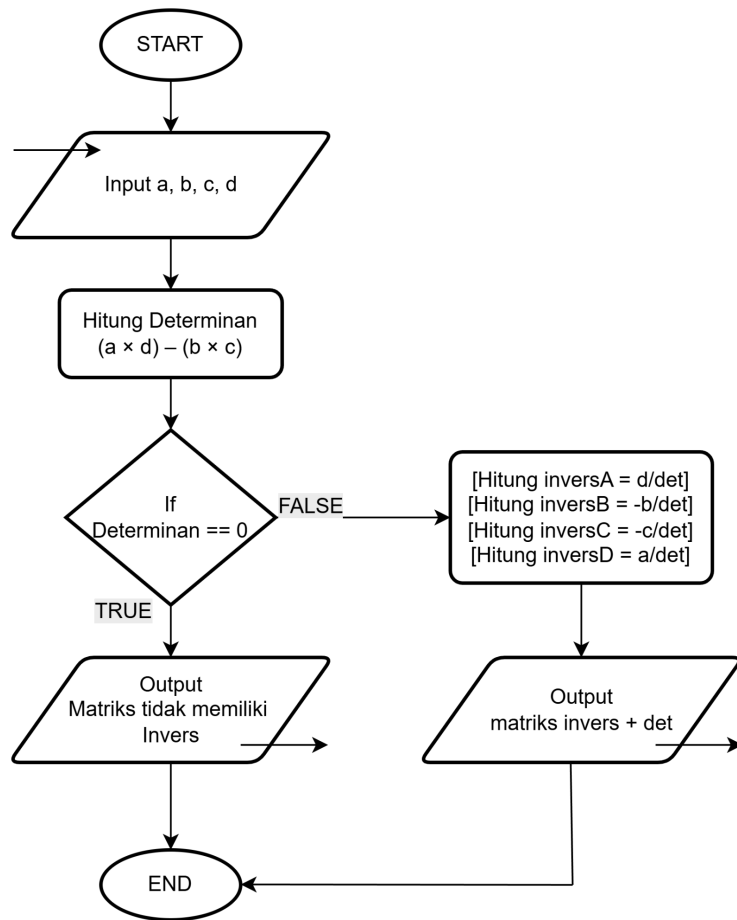
Flowchart: **Nasha Nuron Wahdah (2509106037)**

Code: **Anugerah Fakhri Reswara (2509106025)**

1.2.1 Pseudocode

JUDUL: Invers Matriks 2x2
DEKLARASI: VAR a, b, c, d, det : FLOAT VAR invA, invB, invC, invD : FLOAT
DESKRIPSI: READ a, b, c, d DISPLAY "Matriks Asli: " DISPLAY a, b DISPLAY c, d $det = (a * d) - (b * c)$ IF $det == 0$ THEN DISPLAY "Matriks Singular, Invers tidak ada" ELSE $invA = d / det$ $invB = -b / det$ $invC = -c / det$ $invD = a / det$ DISPLAY "Hasil Invers: " DISPLAY invA, invB DISPLAY invC, invD ENDIF

1.2.2 Flowchart



1.2.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Persiapan	PRINT "=== Program Invers Matriks 2x2 ==="
2	Input Data	INPUT a, b, c, d
3	Tampilan Awal	PRINT "Matriks Sebelum Invers:" PRINT a, b PRINT c, d
4	Hitung Determinan	SET determinan = (a * d) - (b * c)
5	Cek Kondisi	IF determinan == 0 THEN:
6	Kondisi Gagal	PRINT "Determinan nol! Matriks tidak memiliki invers."
7	Kondisi Berhasil	ELSE: (Jika determinan tidak nol, lakukan perhitungan berikut)
8	Hitung Elemen Baru	SET inversA = d / determinan SET inversB = -b / determinan SET inversC = -c / determinan SET inversD = a / determinan
9	Output Hasil	PRINT "Matriks Sesudah Invers", determinan PRINT inversA, inversB PRINT inversC, inversD
10	Selesai	ENDIF

1.2.4 Screenshot Program

```
=== Program Invers Matriks 2x2 ===
Masukkan elemen matriks A:
Elemen [0][0] (a): 3
Elemen [0][1] (b): -2
Elemen [1][0] (c): -1
Elemen [1][1] (d): 1

Matriks Sebelum Invers:
[ 3  -2 ]
[ -1  1 ]

Matriks Sesudah Invers (Determinan: 1):
[ 1  2 ]
[ 1  3 ]
```

Gambar 1.2.4.1 Percobaan pertama

```
=== Program Invers Matriks 2x2 ===
Masukkan elemen matriks A:
Elemen [0][0] (a): 6
Elemen [0][1] (b): -2
Elemen [1][0] (c): -1
Elemen [1][1] (d): 4

Matriks Sebelum Invers:
[ 6  -2 ]
[ -1  4 ]

Matriks Sesudah Invers (Determinan: 22):
[ 0.181818  0.0909091 ]
[ 0.0454545  0.272727 ]
```

Gambar 1.2.4.2 Percobaan kedua

1.3 PENJELASAN SOAL 3

Program ini adalah **Sistem Pengolah Nilai Matriks 3x3**. Tujuannya adalah untuk mengambil data angka yang sudah tersusun dalam bentuk tabel (baris dan kolom), menampilkannya ke layar, lalu secara otomatis menghitung total jumlah nilai serta nilai rata-ratanya.

A. Tabel Algoritma (Alur Logis)

Algoritma menjelaskan strategi penyelesaian masalah dengan bahasa manusia yang runtut.

No	Langkah	Penjelasan Detail
1	Persiapan	Menyiapkan data angka dalam bentuk tabel 3x3 (Matriks) dan variabel untuk menampung total nilai.
2	Penelusuran	Program "berjalan" menelusuri setiap baris, mulai dari baris pertama hingga terakhir.
3	Proses Data	Di setiap baris, program membaca setiap angka di kolomnya, menampilkannya, dan menambahkannya ke variabel total.
4	Kalkulasi	Setelah semua angka dijumlahkan, program membagi hasil total dengan jumlah seluruh elemen (9) untuk mendapat rata-rata.
5	Output	Menampilkan total nilai dan rata-rata dengan format dua angka di belakang koma (desimal).

Bagian anggota kelompok yang mengerjakan tugas No. 3

Pseudocode & Algoritma: **Nasha Nuron Wahdah (2509106037)**

Flowchart: **Anugerah Fakhri Reswara (2509106025)**

Code: **Muhammad Riyadh Zuhair (2509106029)**

1.3.1 Pseudocode

JUDUL: Hitung Rata-rata Matriks
DEKLARASI: VAR nilai[3][3] : INTEGER VAR total, rata_rata : DOUBLE

DESKRIPSI:

INPUT nilai matriks dengan {10, 20, ... , 90}

SET total = 0

FOR i = 0 TO 2 DO:

FOR j = 0 TO 2 DO:

total = total + nilai[i][j]

PRINT nilai[i][j]

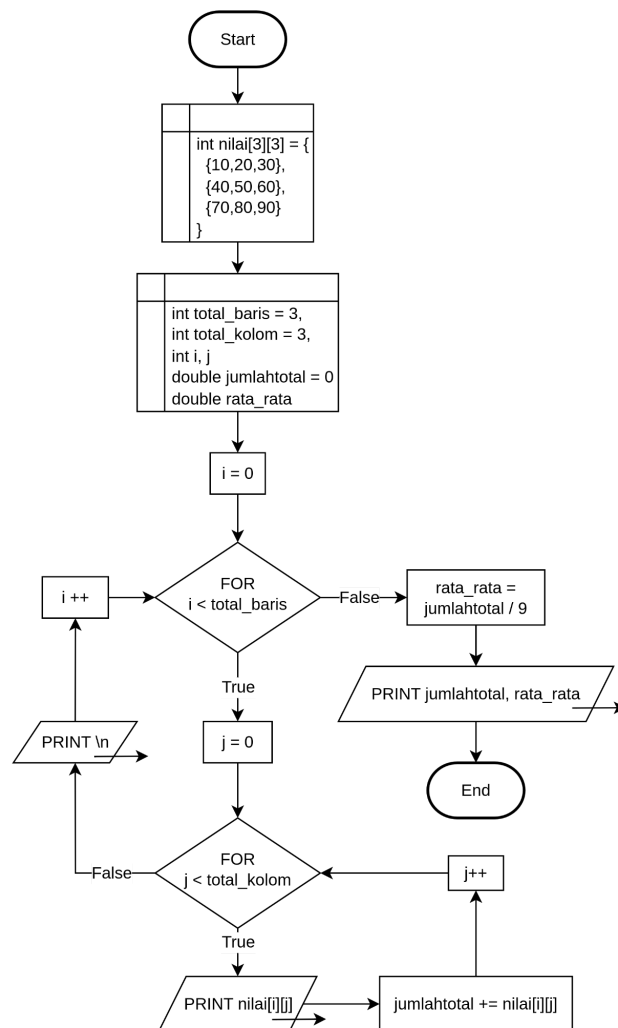
ENDFOR

ENDFOR

rata_rata = total / 9

PRINT total, rata_rata

1.3.2 Flowchart



1.3.3 Algoritma

No	Tahapan	Deskripsi Langkah
1	Persiapan	Menyiapkan sebuah matriks (array 2D) berukuran 3x3 yang sudah berisi angka-angka tertentu (10 hingga 90).
2	Inisialisasi	Menentukan variabel pembantu: total_baris (3), total_kolom (3), dan jumlahtotal untuk menampung jumlah angka.
3	Proses Penelusuran	Menggunakan perulangan bersarang (<i>nested loop</i>) untuk mengunjungi setiap baris dan setiap kolom satu per satu.
4	Cetak & Akumulasi	Di setiap sel matriks: 1. Tampilkan angka tersebut ke layar. 2. Tambahkan angka tersebut ke dalam variabel jumlahtotal.
5	Hitung Rata-rata	Setelah semua angka dijumlahkan, bagi jumlahtotal dengan total elemen (3 baris \times 3 kolom = 9).
6	Output Akhir	Tampilkan hasil akhir berupa Total Nilai dan Rata-rata dengan format 2 angka di belakang koma.
7	Selesai	Program berakhir.

1.3.4 Screenshot Program

```
=====array multidimensi=====
Isi array:
10      20      30
40      50      60
70      80      90
Total Nilai      : 450
Rata - rata      : 50.00
```

Gambar 1.3.4.1 Keseluruhan Program

1.4 PENJELASAN SOAL 4

Program ini bertujuan untuk menunjukkan cara kerja Pointer dalam C++. Program mengambil nilai input dari dua mahasiswa, lalu menyimpan alamat memori dari variabel tersebut ke dalam pointer. Hasil akhirnya adalah menampilkan alamat memori dan menjumlahkan nilai alamat tersebut.

Langkah-langkah Kerja:

- Input: Mengambil input nilai angka untuk mhs1 dan mhs2.
- Referencing: Menggunakan simbol & untuk mengambil alamat lokasi penyimpanan variabel di memori.
- Assignment: Menyimpan alamat tersebut ke dalam variabel p1 dan p2.
- Output: Menampilkan alamat memori tersebut ke layar dan hasil penjumlahan dari kedua alamat memori tersebut.

Bagian anggota kelompok yang mengerjakan tugas No. 4

Pseudocode & Algoritma: **Anugerah Fakhri Reswara (2509106025)**

Flowchart: **Anugerah Fakhri Reswara (2509106025)**

Code: **Nasha Nuron Wahdah (2509106037)**

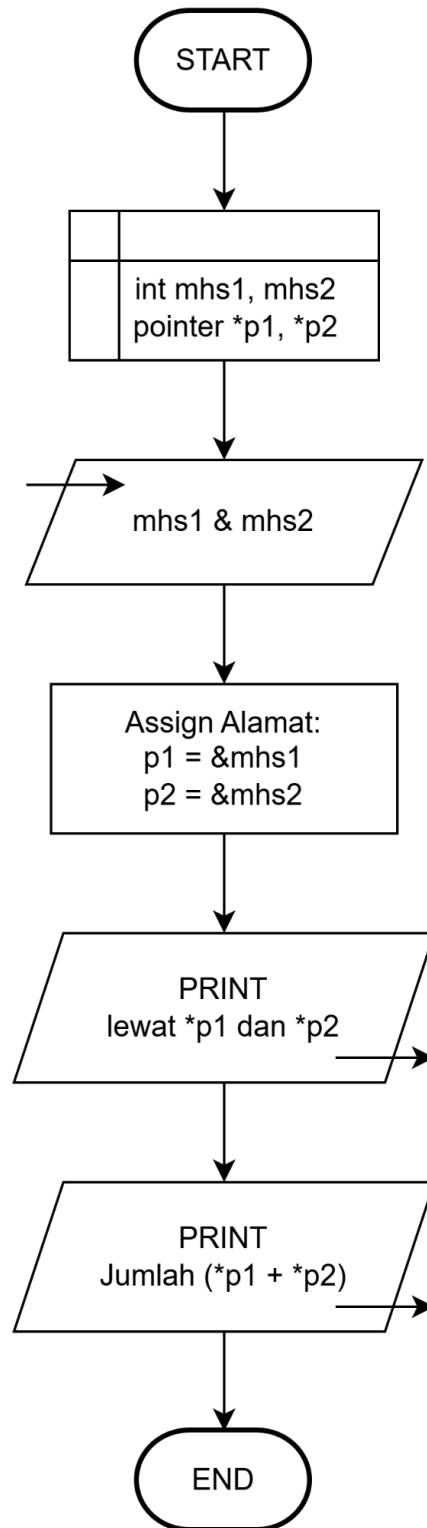
1.4.1 Pseudocode

JUDUL:
Pointer Alamat Mahasiswa

KAMUS/DEKLARASI:
VAR mhs1, mhs2 : INTEGER
VAR p1, p2 : INTEGER (Pointer Address)

DESKRIPSI:
READ mhs1
READ mhs2
SET p1 = ADDRESS OF mhs1
SET p2 = ADDRESS OF mhs2
DISPLAY "Alamat mhs1: ", p1
DISPLAY "Alamat mhs2: ", p2
DISPLAY "Jumlah Alamat: ", (p1 + p2)

1.4.2 Flowchart



1.4.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Persiapan	PRINT "Masukkan nilai mahasiswa..."
2	Input Data	INPUT mhs1, mhs2
3	Operasi Pointer	SET p1 = &mhs1 SET p2 = &mhs2
4	Tampilan Alamat	PRINT "Nilai mhs1 (via pointer): ", p1 PRINT "Nilai mhs2 (via pointer): ", p2
5	Kalkulasi Alamat	PRINT "Jumlah kedua nilai: ", (p1 + p2)
6	Selesai	RETURN 0

1.4.4 Screenshot Program

```
Masukkan nilai mahasiswa 1: 88
Masukkan nilai mahasiswa 2: 75

Nilai mahasiswa 1 (via pointer): 88
Nilai mahasiswa 2 (via pointer): 75
Jumlah kedua nilai: 163
```

Gambar 1.4.4.1 Percobaan pertama

```
Masukkan nilai mahasiswa 1: 55
Masukkan nilai mahasiswa 2: 22

Nilai mahasiswa 1 (via pointer): 55
Nilai mahasiswa 2 (via pointer): 22
Jumlah kedua nilai: 77
```

Gambar 1.4.4.1 Percobaan kedua

1.5 PENJELASAN SOAL 5

Program ini bertujuan untuk menghasilkan sebuah pola angka di dalam matriks berukuran 5x5. Program menggunakan logika koordinat (i sebagai baris dan j sebagai kolom) untuk menentukan angka mana yang harus dicetak.

Cara Kerja Program:

1. **Iterasi:** Program melakukan perulangan baris (i) dan kolom (j) sebanyak 5 kali.
2. **Kondisi Angka 9:** Jika posisi baris ditambah kolom kurang dari 4 ($i + j < n-1$), program mencetak angka **9** (area kiri atas).
3. **Kondisi Angka 0:** Jika posisi baris ditambah kolom sama dengan 4 ($i + j == n-1$), program mencetak angka **0** sebagai garis diagonal samping.
4. **Kondisi Angka 1:** Jika posisi baris ditambah kolom lebih dari 4, program mencetak angka **1** (area kanan bawah).

Bagian anggota kelompok yang mengerjakan tugas No. 5

Pseudocode & Algoritma: **Anugerah Fakhri Reswara (2509106025)**

Flowchart: **Anugerah Fakhri Reswara (2509106025)**

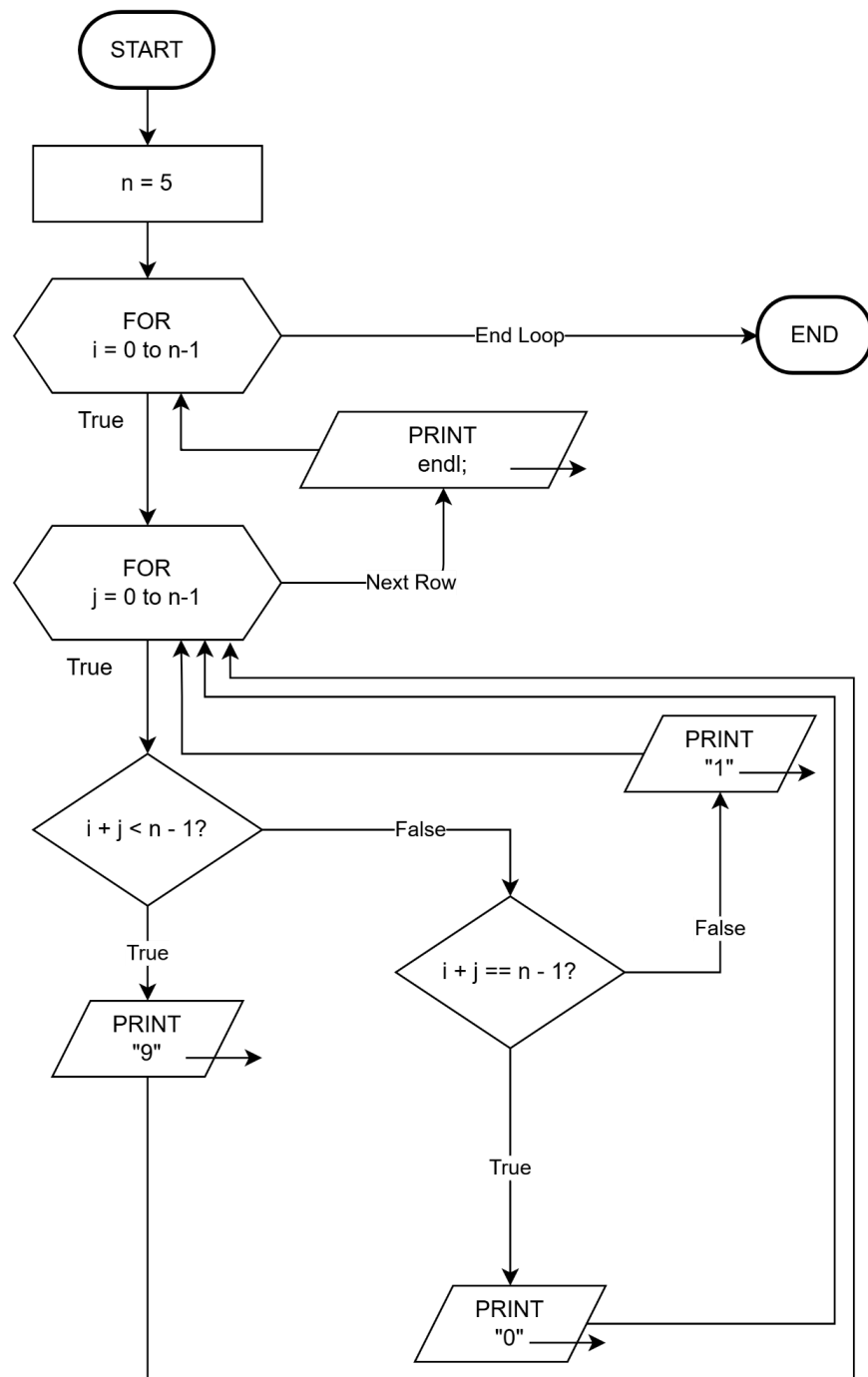
Code: **Anugerah Fakhri Reswara (2509106025)**

1.5.1 Pseudocode

JUDUL: Pola Matriks Segitiga 5x5
DEKLARASI: VAR n, i, j : INTEGER
DESKRIPSI: SET n = 5 PRINT "=== Program Pola Matriks Segitiga ===" FOR i = 0 TO n - 1 DO: FOR j = 0 TO n - 1 DO: IF (i + j < n - 1) THEN DISPLAY "9 " ELSE IF (i + j == n - 1) THEN DISPLAY "0 " ELSE DISPLAY "1 " ENDIF ENDFOR ENDFOR

PRINT NEWLINE
ENDFOR

1.5.2 Flowchart



1.5.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Persiapan	SET $n = 5$ dan tampilkan judul program.
2	Perulangan Baris	FOR $i = 0$ TO $n - 1$ untuk mengontrol baris matriks.
3	Perulangan Kolom	FOR $j = 0$ TO $n - 1$ untuk mengontrol kolom matriks.
4	Logika Segitiga Atas	IF $(i + j < n - 1)$ maka cetak angka 9.
5	Logika Diagonal	ELSE IF $(i + j == n - 1)$ maka cetak angka 0.
6	Logika Segitiga Bawah	ELSE (selain kondisi di atas) maka cetak angka 1.
7	Pindah Baris	Tampilkan endl setelah satu baris kolom selesai diproses.
8	Selesai	Program berakhir.

1.5.4 Screenshot Program

```
=== Program Pola Matriks Segitiga ===
9  9  9  9  0
9  9  9  0  1
9  9  0  1  1
9  0  1  1  1
0  1  1  1  1
```

Gambar 1.4.4.1 Keseluruhan Program

BAB 2

PEMBAHASAN TUGAS A

2.1 PENJELASAN TUGAS A

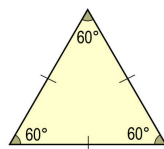
Program ini berfungsi untuk menentukan apakah tiga buah panjang sisi yang dimasukkan oleh pengguna dapat membentuk sebuah segitiga atau tidak. Jika valid, program akan mengklasifikasikan jenis segitiga tersebut berdasarkan panjang sisi-sisinya.

Langkah-langkah Kerja:

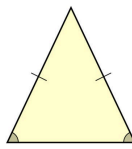
1. **Input Sisi:** Pengguna memasukkan tiga angka sebagai panjang sisi segitiga.
2. **Mencari Sisi Terpanjang:** Program mengidentifikasi mana dari ketiga sisi tersebut yang paling panjang.
3. **Validasi Segitiga:** Menggunakan aturan pertidaksamaan segitiga (jumlah dua sisi terpendek harus lebih besar dari sisi terpanjang). Jika tidak terpenuhi, program berhenti.
4. **Cek Jenis Siku-Siku:** Menggunakan Teorema Pythagoras ($a^2 + b^2 = c^2$) untuk mengecek apakah segitiga tersebut siku-siku.
5. **Cek Jenis Sama Sisi/Kaki:** Program mengecek kesamaan antar sisi untuk menentukan apakah segitiga tersebut memiliki sisi yang sama.
6. **Output:** Menampilkan hasil apakah segitiga bisa terbentuk dan apa jenisnya.

Jenis-jenis segitiga:

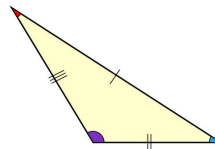
By lengths of sides



Equilateral

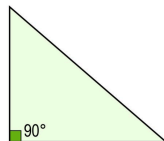


Isosceles

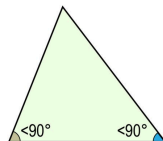


Scalene

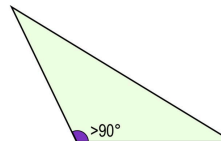
By internal angles



Right



Acute



Oblique

Bagian anggota kelompok yang mengerjakan tugas No. A

Pseudocode & Algoritma: **Anugerah Fakhriza Reswara (2509106025)**

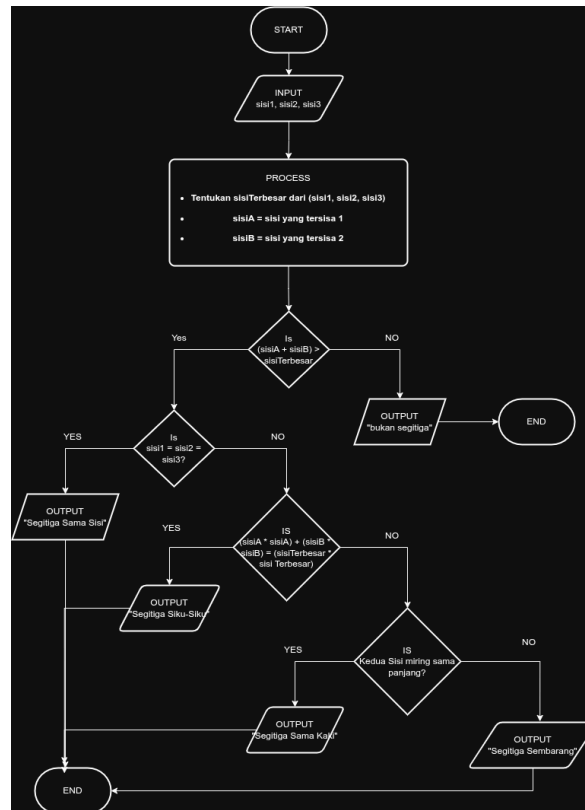
Flowchart: **Muhammad Riyadh Zuhair (2509106029)**

Code: **Nasha Nuron Wahdah (2509106037)**

2.1.1 Pseudocode

JUDUL: Pengecekan Jenis Segitiga
DEKLARASI: VAR sisi1, sisi2, sisi3 : INTEGER VAR sisiTerbesar, jumlahDuaSisi : INTEGER VAR sisiAlas, sisiTinggi : INTEGER
DESKRIPSI: READ sisi1, sisi2, sisi3 SET sisiTerbesar = sisi1 IF (sisi2 > sisiTerbesar) THEN SET sisiTerbesar = sisi2 IF (sisi3 > sisiTerbesar) THEN SET sisiTerbesar = sisi3 SET jumlahDuaSisi = (sisi1 + sisi2 + sisi3) - sisiTerbesar IF (jumlahDuaSisi <= sisiTerbesar) THEN DISPLAY "Tidak bisa membentuk segitiga" ELSE DISPLAY "Bisa membentuk segitiga" IF (sisiAlas ² + sisiTinggi ² == sisiTerbesar ²) THEN DISPLAY "Segitiga siku-siku" ELSE IF (sisi1 == sisi2 AND sisi2 == sisi3) THEN DISPLAY "Segitiga sama sisi" ELSE DISPLAY "Segitiga sembarang" ENDIF ENDIF

2.1.2 Flowchart



2.1.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Persiapan	INPUT sisi1, sisi2, sisi3.
2	Cari Maksimum	Tentukan sisiTerbesar di antara ketiga input.
3	Validasi	Hitung jumlahDuaSisi. Jika jumlahDuaSisi <= sisiTerbesar, maka bukan segitiga.
4	Identifikasi Posisi	Tentukan sisiAlas dan sisiTinggi dari dua sisi yang lebih kecil.
5	Cek Siku-Siku	Gunakan logika Pythagoras untuk mengecek jenis siku-siku.
6	Cek Kesamaan	Bandingkan antar sisi untuk menentukan jenis sama kaki/sisi atau sembarang.
7	Output	Tampilkan hasil validasi dan jenis segitiga ke layar.

8	Selesai	Program berakhir.
---	---------	-------------------

2.1.4 Screenshot Program

```

=== Program Pengecek Segitiga ===
Masukkan panjang sisi 1: 1
Masukkan panjang sisi 2: 1
Masukkan panjang sisi 3: 10

--- Hasil Analisis ---
Hasil: Tidak bisa membentuk segitiga.
Alasan: Jumlah dua sisi terpendek (2) tidak lebih besar dari sisi terpanjang (10).

```

Gambar 2.1.4.1 Tidak bisa membentuk segitiga

```

=== Program Pengecek Segitiga ===
Masukkan panjang sisi 1: 3
Masukkan panjang sisi 2: 4
Masukkan panjang sisi 3: 5

--- Hasil Analisis ---
Hasil: Bisa membentuk segitiga.
Jenis: Segitiga Siku-Siku

```

Gambar 2.1.4.2 Segitiga Siku-Siku

```

=== Program Pengecek Segitiga ===
Masukkan panjang sisi 1: 5
Masukkan panjang sisi 2: 5
Masukkan panjang sisi 3: 5

--- Hasil Analisis ---
Hasil: Bisa membentuk segitiga.
Jenis: Segitiga Sama Sisi

```

Gambar 2.1.4.3 Segitiga Sama Sisi

```
=== Program Pengecek Segitiga ===  
Masukkan panjang sisi 1: 5  
Masukkan panjang sisi 2: 5  
Masukkan panjang sisi 3: 8  
  
--- Hasil Analisis ---  
Hasil: Bisa membentuk segitiga.  
Jenis: Segitiga Sama Kaki
```

Gambar 2.1.4.4 Segitiga Sama Kaki

```
=== Program Pengecek Segitiga ===  
Masukkan panjang sisi 1: 10  
Masukkan panjang sisi 2: 7  
Masukkan panjang sisi 3: 12  
  
--- Hasil Analisis ---  
Hasil: Bisa membentuk segitiga.  
Jenis: Segitiga Sembarang
```

Gambar 2.1.4.5 Segitiga Sembarang

BAB 3

PEMBAHASAN TUGAS B

3.1 PENJELASAN TUGAS B

Program ini berfungsi untuk memvalidasi apakah tiga buah bilangan yang dimasukkan pengguna memenuhi syarat **Teorema Pythagoras**. Program secara otomatis mengurutkan angka dari yang terkecil hingga terbesar untuk memastikan sisi miring (sisi terpanjang) berada di posisi yang tepat sebelum dilakukan perhitungan.

Langkah-langkah Kerja:

1. **Input Data:** Pengguna memasukkan tiga bilangan acak (a, b, dan c).
2. **Pengurutan (Sorting):** Program mengurutkan ketiga bilangan tersebut agar sisi terpendek menjadi s1 dan s2, serta sisi terpanjang menjadi s3.
3. **Perhitungan Kuadrat:** Program menghitung jumlah kuadrat dua sisi terkecil ($s1^2 + s2^2$) dan membandingkannya dengan kuadrat sisi terpanjang ($s3^2$).
4. **Validasi:**
 - Jika hasil penjumlahan kuadrat dua sisi kecil **sama dengan** kuadrat sisi terbesar, maka dinyatakan sebagai **Triple Pythagoras**.
 - Jika **tidak sama**, maka dinyatakan **bukan Triple Pythagoras**.
5. **Output:** Menampilkan urutan sisi, proses perhitungan, dan kesimpulan akhirnya.

Bagian anggota kelompok yang mengerjakan tugas No. B

Pseudocode & Algoritma: **Nasha Nuron Wahdah (2509106037)**

Flowchart: **Anugerah Fakhri Reswara (2509106025)**

Code: **Muhammad Riyadh Zuhair (2509106029)**

3.1.1 Pseudocode

JUDUL: Pengecekan Triple Pythagoras
DEKLARASI: VAR a, b, c : INTEGER VAR s1, s2, s3 : LONG

VAR sisi[3] : ARRAY OF INTEGER

DESKRIPSI:

READ a, b, c

SET sisi = {a, b, c}

SORT(sisi) (Urutkan dari terkecil ke terbesar)

SET s1 = sisi[0]

SET s2 = sisi[1]

SET s3 = sisi[2]

IF ((s1 * s1) + (s2 * s2) == (s3 * s3)) THEN

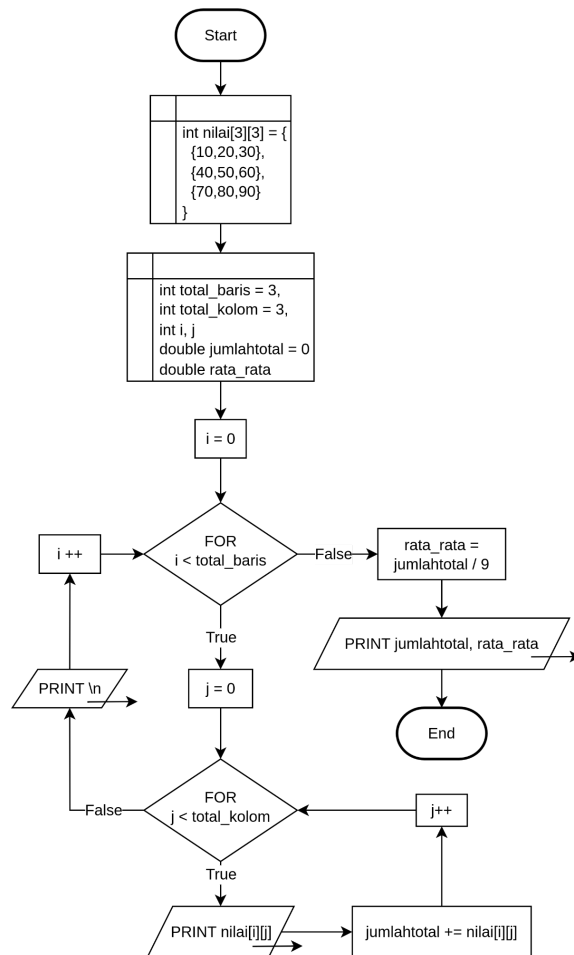
 DISPLAY "Kesimpulan: Hasil menunjukkan bilangan adalah triple pythagoras"

ELSE

 DISPLAY "Kesimpulan: Bukan triple pythagoras"

ENDIF

3.1.2 Flowchart



3.1.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Persiapan	Tampilkan judul program dan siapkan variabel input.
2	Input Data	INPUT a, b, c.
3	Pengurutan Sisi	Masukkan angka ke array dan jalankan fungsi sort agar sisi terpanjang ditemukan.
4	Identifikasi Sisi	Tetapkan s1, s2 sebagai sisi siku-siku dan s3 sebagai sisi miring.
5	Uji Pythagoras	Hitung apakah $s1^2 + s2^2$ memiliki nilai yang sama dengan $s3^2$.
6	Output Hasil	Tampilkan rincian perhitungan dan kesimpulan (Triple Pythagoras atau bukan).
7	Selesai	Program berakhir.

3.1.4 Screenshot Program

```
masukkan bilangan pertama (a): 8
masukkan bilangan kedua (b): 5
masukkan bilangan ketiga (c): 17

=== Hasil analisa ===
Urutan sisi: 5, 8, 17
Hasil: 89 Tidak sama dengan 289
kesimpulan: bukan triple pythagoras
```

Gambar 3.1.4.1 Bukan triple pythagoras

```
=== Pengecekan triple pytagoras oleh admin
masukkan bilangan pertama (a): 7
masukkan bilangan kedua (b): 24
masukkan bilangan ketiga (c): 25

=== Hasil analisa ===
Urutan sisi: 7, 24, 25
Hasil:  $7^2 + 24^2 = 25^2$ 
kesimpulan : Hasil menunjukkan bilangan adalah triple pytagoras
```

Gambar 3.1.4.2 Triple pythagoras

```
=== Pengecekan triple pytagoras oleh admin
masukkan bilangan pertama (a): 5
masukkan bilangan kedua (b): 12
masukkan bilangan ketiga (c): 13

=== Hasil analisa ===
Urutan sisi: 5, 12, 13
Hasil:  $5^2 + 12^2 = 13^2$ 
kesimpulan : Hasil menunjukkan bilangan adalah triple pytagoras
```

Gambar 3.1.4.2 Triple pythagoras

BAB 4

PEMBAHASAN TUGAS C

4.1 PENJELASAN SOAL C

Program ini digunakan untuk menghitung hasil pangkat dari suatu bilangan menggunakan teknik **rekursi**. Rekursi adalah metode di mana sebuah fungsi memanggil dirinya sendiri dengan nilai yang lebih kecil hingga mencapai kondisi berhenti (basis).

Langkah-langkah Kerja:

1. **Input Data:** Pengguna memasukkan angka dasar (bilangan) dan angka pemangkat (eksponen).
2. **Pemanggilan Fungsi:** Program memanggil fungsi pangkat().
3. **Logika Rekursif:**
 - **Basis:** Jika eksponen adalah 0, maka hasil adalah 1 (karena angka apa pun pangkat 0 adalah 1).
 - **Rekurens:** Jika eksponen lebih dari 0, program mengalikan bilangan dengan hasil dari pangkat(bilangan, eksponen - 1).
4. **Pengulangan:** Proses ini berulang terus menerus hingga eksponen menyentuh angka 0.
5. **Output:** Setelah tumpukan rekursi selesai, program menampilkan hasil akhir perpangkatan.

Bagian anggota kelompok yang mengerjakan tugas No. C

Pseudocode & Algoritma: **Anugerah Fakhri Reswara (2509106025)**

Flowchart: **Anugerah Fakhri Reswara (2509106025)**

Code: **Nasha Nuron Wahdah (2509106037)**

4.1.1 Pseudocode

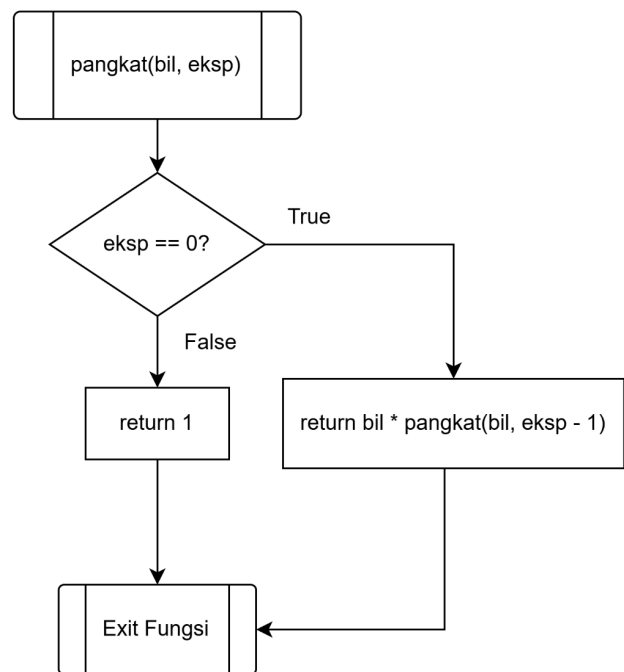
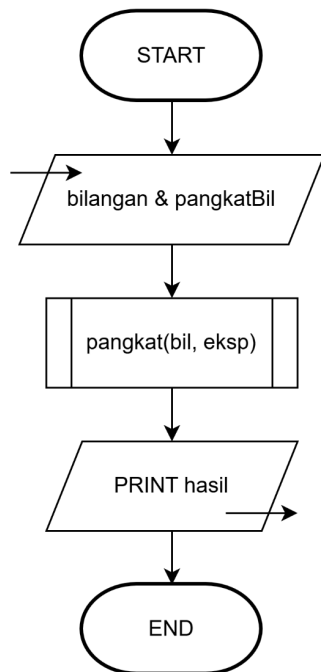
JUDUL: Perpangkatan Rekursif
DEKLARASI: VAR bilangan, pangkatBil, hasil : INTEGER

```

DESKRIPSI:
READ bilangan
READ pangkatBil
FUNCTION pangkat(bil, eks):
  IF (eks == 0) THEN
    RETURN 1
  ELSE RETURN
    bil * pangkat(bil, eks - 1)
  ENDIF
ENDFUNCTION
SET hasil = CALL pangkat(bilangan, pangkatBil)
DISPLAY bilangan, "^", pangkatBil, " = ", hasil

```

4.1.2 Flowchart



4.1.3 Algoritma

No	Tahapan	Instruksi / Pseudocode
1	Persiapan	Siapkan variabel untuk menampung input pengguna.
2	Input Data	INPUT bilangan dan INPUT pangkatBil.
3	Proses Rekursi	Jalankan fungsi pangkat() yang memanggil dirinya sendiri dengan mengurangi eksponen sebanyak 1 pada setiap tahap.
4	Kondisi Berhenti	Berhenti saat eksponen mencapai 0 dan kembalikan nilai 1 sebagai pengali terakhir.
5	Kalkulasi Akhir	Kalikan semua bilangan yang terkumpul dari tumpukan rekursi.
6	Output	Tampilkan hasil perhitungan perpangkatan ke layar.
7	Selesai	Program berakhir.

4.1.4 Screenshot Program

```
Masukkan bilangan: 2
Masukkan pangkat: 4
Hasil: 2^4 = 16
```

Gambar 4.1.4.1 Percobaan pertama

```
Masukkan bilangan: 5
Masukkan pangkat: -5
Maaf, program ini hanya mendukung pangkat bulat positif.
```

Gambar 4.1.4.1 Percobaan kedua dengan pangkat negatif

```
Masukkan bilangan: -4
Masukkan pangkat: 6
Hasil: -4^6 = 4096
```

Gambar 4.1.4.1 Percobaan kedua dengan bilangan negatif

4.2 PENJELASAN TUGAS ARRAY SATU DIMENSI

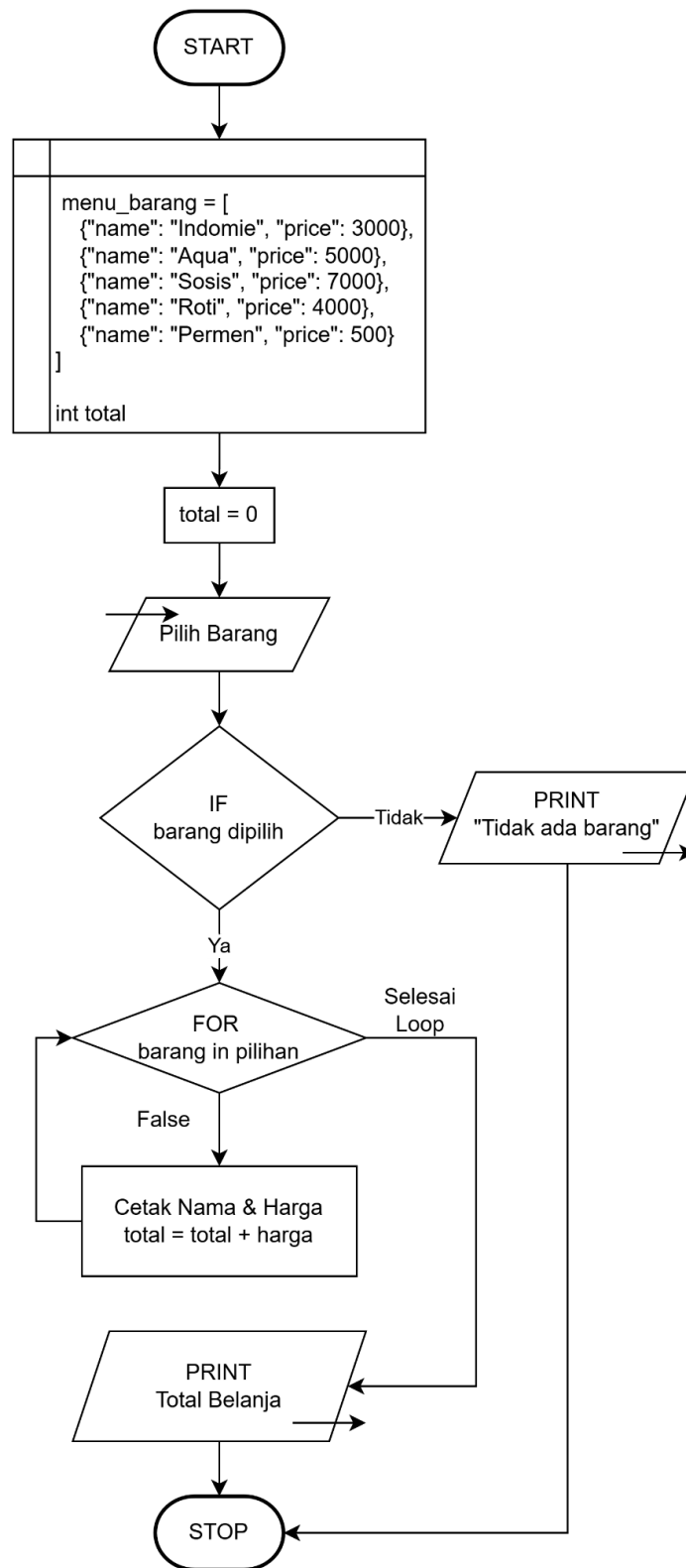
Program ini adalah sistem **Kasir Sederhana** yang memanfaatkan **Array (List)** untuk menyimpan data barang dan juga menggunakan library InquirerPy untuk mempercantik tampilan pemilihan.

1. **Penyimpanan Data:** Daftar barang disimpan dalam list of dictionaries (**menu_barang**).
2. **Interaksi:** Program menampilkan menu interaktif di mana pengguna bisa memilih banyak barang sekaligus menggunakan tombol spasi.
3. **Pemrosesan:** Program melakukan iterasi (perulangan) pada barang-barang yang dipilih untuk menghitung total harga.
4. **Output:** Menampilkan struk belanja yang berisi daftar barang yang dibeli dan total biayanya.

4.2.1 Pseudocode

JUDUL: Kasir Sederhana
DEKLARASI: ARRAY menu_barang INTEGER total ARRAY pilihan
DESKRIPSI: READ (InquirerPy) STORE in pilihan IF pilihan IS EMPTY THEN PRINT "Tidak ada barang" ELSE FOR EACH barang IN pilihan DO PRINT barang.name, barang.price total = total + barang.price END FOR PRINT total

4.2.2 Flowchart



4.2.3 Algoritma

No	Langkah-langkah	Keterangan
1	Mulai	Program dijalankan.
2	Mendefinisikan Daftar Barang	Membuat array berisi nama dan harga barang sebagai database sementara.
3	Menampilkan Menu Pilihan	Menampilkan daftar barang ke layar agar bisa dipilih oleh pengguna.
4	Menerima Input	Menangkap data barang apa saja yang dicentang oleh pengguna.
5	Cek Validasi	Jika tidak ada yang dipilih, langsung lompat ke langkah terakhir.
6	Perulangan (Looping)	Mengambil satu per satu barang dari daftar pilihan untuk diproses.
7	Akumulasi Harga	Menambahkan harga barang yang sedang diproses ke dalam variabel total.

4.2.4 Screenshot Program

```
? Pilih barang (Spasi untuk pilih, Enter untuk selesai):  
? ☐ Indomie - Rp3000  
  ☐ Aqua - Rp5000  
  ☐ Sosis - Rp7000  
  ☐ Roti - Rp4000  
  ☐ Permen - Rp500
```

Gambar 4.2.4.1 Pilihan barang


```
=== Struk Belanja ===  
- Indomie : Rp3000  
- Aqua : Rp5000  
- Sosis : Rp7000  
- Roti : Rp4000  
- Permen : Rp500  
-----  
Total belanja: Rp 19500  
Terima kasih sudah berbelanja!
```

Gambar 4.2.4.2 Jika memilih semua

```
=== Struk Belanja ===  
- Indomie : Rp3000  
- Sosis : Rp7000  
-----  
Total belanja: Rp 10000  
Terima kasih sudah berbelanja!
```

Gambar 4.2.4.3 Jika memilih Indomie dan Sosis

BAB 5

KESIMPULAN

Laporan ini merupakan hasil kerja seluruh anggota dari Kelompok 10, Program Studi Informatika Angkatan 2025, Universitas Mulawarman. Disusun oleh Anugerah Fakhri Reswara, Muhammad Riyadh Zuhair, dan Nasha Nuron Wahdah.

5.1 FOKUS UTAMA DAN METODOLOGI LAPORAN

Laporan ini mengkaji penyelesaian lima skenario soal algoritma yang berbeda. Metodologi yang diterapkan mencakup tiga pilar utama visualisasi dan dokumentasi logika:

- **Perancangan Logika Melalui Pseudocode:** Setiap soal diawali dengan penulisan pseudocode yang menggunakan standar notasi formal. Hal ini bertujuan untuk menjembatani antara bahasa manusia dengan sintaksis bahasa pemrograman, memastikan logika dasar tetap kokoh sebelum masuk ke tahap pengodean.
- **Visualisasi Flowchart yang Komprehensif:** Penggunaan diagram alir (flowchart) memberikan representasi visual yang jelas mengenai alur data, titik keputusan (*decision points*), dan proses iterasi. Ini membantu dalam mengidentifikasi potensi kebuntuan logika atau *deadlock* dalam program.
- **Analisis Implementasi Step-by-Step:** Dokumen tidak hanya memberikan hasil akhir, tetapi juga mendeskripsikan setiap tahapan instruksi. Penjelasan ini mencakup persiapan variabel, proses input/output, hingga mekanisme pemrosesan data di dalam memori.

5.2 EKSPLORASI KONSEP REKURSI (STUDI KASUS: PERPANGKATAN)

Dalam laporan ini kami membahas mengenai **Fungsi Rekursif**, khususnya pada implementasi operasi matematika perpangkatan (Soal 4). Analisis mendalamnya mencakup:

- **Mekanisme Base Case (Kondisi Terminasi):** Laporan menekankan pentingnya kondisi `if (eks == 0) then return 1`. Tanpa kondisi batas yang didefinisikan dengan tepat, fungsi rekursif akan menyebabkan penggunaan memori yang tidak terkendali (*stack overflow*).
- **Dinamika Langkah Rekurens:** Dijelaskan bagaimana fungsi memanggil dirinya sendiri dengan parameter yang terus mengecil. Ini menunjukkan konsep *divide and conquer* di mana masalah besar dipecah menjadi sub-masalah yang identik namun lebih sederhana.
- **Manajemen Tumpukan (Stack):** Melalui flowchart, laporan menggambarkan bagaimana setiap pemanggilan fungsi disimpan dalam tumpukan memori hingga mencapai *base case*, yang kemudian diikuti oleh proses pengembalian nilai secara berantai hingga ke pemanggil pertama.

5.3 IMPLIKASI DAN KESIMPULAN UMUM

Berdasarkan analisis menyeluruh terhadap konten laporan, beberapa poin kesimpulan penting dapat ditarik:

1. **Kemampuan Dekomposisi Masalah:** Mahasiswa telah menunjukkan kompetensi dalam melakukan dekomposisi masalah, yaitu memecah masalah komputasi yang kompleks menjadi unit-unit logis yang lebih kecil dan mudah dikelola melalui bantuan algoritma deskriptif.
2. **Ketangkasan Kontrol Alur:** Pemilihan struktur kendali yang tepat—baik itu percabangan (*branching*) maupun perulangan (*looping/recursion*)—menunjukkan pemahaman yang matang mengenai efisiensi alur eksekusi program.
3. **Standardisasi Dokumentasi Teknis:** Laporan ini memenuhi kriteria akademis yang tinggi dalam penyajian teknis informatika. Setiap bagian disusun secara kronologis mulai dari persiapan variabel, validasi input, pemrosesan inti, hingga presentasi hasil akhir kepada pengguna.
4. **Efisiensi Logika dan Keamanan Kode:** Solusi yang dirancang telah mempertimbangkan aspek keamanan logika, seperti pencegahan *infinite loop* pada fungsi rekursif, yang merupakan aspek krusial dalam pengembangan perangkat lunak skala besar.

Secara keseluruhan, laporan ini mencerminkan kesiapan mahasiswa informatika Universitas Mulawarman dalam menguasai konsep-konsep dasar desain algoritma. Pemahaman yang kuat pada tahap ini sangat krusial sebelum mereka melangkah ke implementasi menggunakan bahasa pemrograman tingkat tinggi yang memiliki kompleksitas manajemen memori dan struktur data yang lebih rumit.

DAFTAR PUSTAKA

- Cplusplus.com. (2024). *Algorithm Library - std::sort*. Diakses pada 19 Februari 2026 melalui pangkalan data standar dokumentasi C++, dari <https://cplusplus.com/reference/algorithm/sort/>
- GeeksforGeeks. (2023). *C++ If-else Statement*. Diakses pada 19 Februari 2026 melalui repositori teknik informatika, dari <https://www.geeksforgeeks.org/cpp/c-c-if-else-statement-with-examples/>
- GeeksforGeeks. (2024). *Multidimensional Arrays in C++*. Diakses pada 19 Februari 2026 melalui publikasi teknis daring, dari <https://www.geeksforgeeks.org/multidimensional-arrays-c-cpp/>
- GeeksforGeeks. (2024). *Pointers in C++ with Examples*. Diakses pada 19 Februari 2026 melalui modul instruksional struktur data, dari <https://www.geeksforgeeks.org/cpp/pointers-and-references-in-c/>
- Microsoft Learn. (2022). *Standard Library <iomanip>*. Microsoft Documentation. Diakses pada 19 Februari 2026 melalui portal dokumentasi resmi Microsoft, dari <https://learn.microsoft.com/en-us/cpp/standard-library/iomanip>
- Microsoft Learn. (2023). *Recursive Functions in C++*. Microsoft Documentation. Diakses pada 19 Februari 2026 melalui arsip teknis Microsoft, dari <https://learn.microsoft.com/en-us/cpp/c-language/recursive-functions?view=msvc-170>
- Microsoft Learn. (2023). *C++ Standard Library Reference*. Microsoft Documentation. Diakses pada 19 Februari 2026 melalui repositori otoritatif Microsoft, dari <https://learn.microsoft.com/en-us/cpp/standard-library/cpp-standard-library-reference>
- Programiz. (2024). *C++ Recursion (Recursive Function)*. Diakses pada 19 Februari 2026 melalui media edukasi pemrograman digital, dari <https://www.programiz.com/cpp-programming/recursion>
- Tutorialspoint. (2024). *C++ Multi-dimensional Arrays*. Diakses pada 19 Februari 2026 melalui pangkalan pengetahuan tutorial perangkat lunak, dari https://www.tutorialspoint.com/cplusplus/cpp_multi_dimensional_arrays.htm
- W3Schools. (n.d.). *C++ Pointers*. Diakses pada 19 Februari 2026 melalui platform edukasi komputasi global, dari https://www.w3schools.com/cpp/cpp_pointers.asp
- W3Schools. (n.d.). *C++ Recursion*. Diakses pada 19 Februari 2026 melalui sumber pustaka digital pemrograman, dari https://www.w3schools.com/cpp/cpp_functions_recursion.asp
- W3Schools. (n.d.). *C++ Conditions and If-Else*. Diakses pada 19 Februari 2026 melalui modul fundamental bahasa pemrograman, dari https://www.w3schools.com/cpp/cpp_conditions.asp