

WORKSHEET – MODUL 7

Identitas mahasiswa:

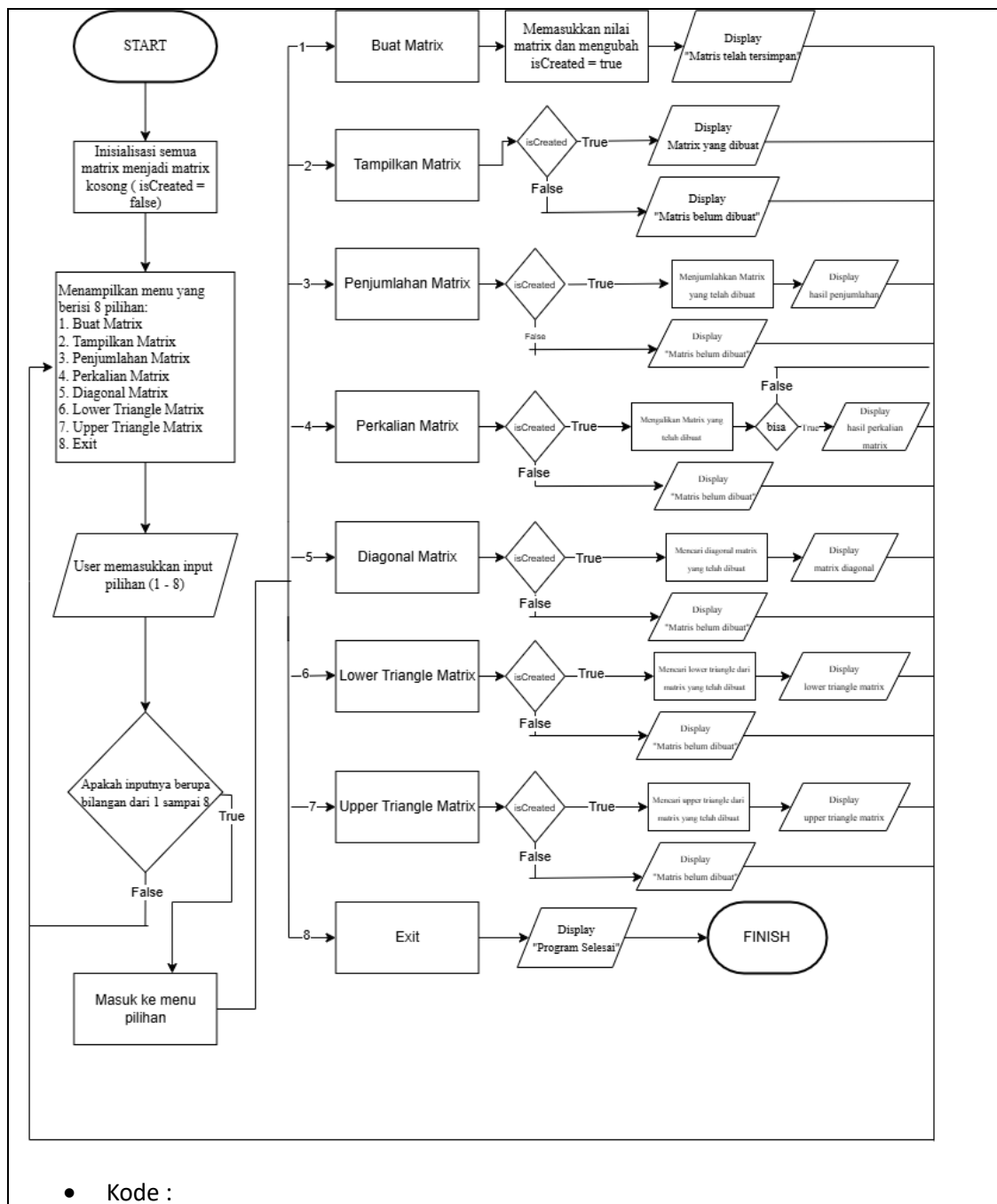
Nama : Ilham Yusuf Wi'am
NIM : 24/539979/TK/59890
Sesi praktikum: Senin B3

Intruksi: Kerjakan semua percobaan dan *check points* yang ada pada Modul 6 dan tuliskan jawabannya pada worksheet ini.

Lab 1.3 : TUGAS

1. Berikan flowchart, kode program dan hasil keluarannya

- Flowchart :



```

1  #include <iostream>
2  #include <vector>
3  #include <limits>
4  using namespace std;
5
6  struct Matrix {
7      int rows, cols;
8      vector<vector<int>> data;
9      string name;
10     bool isCreated = false;
11
12     void input() {
13         data.resize(rows, vector<int>(cols));
14         cout << "\nMasukkan nilai untuk " << name << ":\n";
15         for (int r = 0; r < rows; r++) {
16             for (int c = 0; c < cols; c++) {
17                 cout << "Nilai [" << r << "][" << c << "]: ";
18                 cin >> data[r][c];
19             }
20         }
21         isCreated = true;
22     }
23
24     void print() const {
25         if (!isCreated) {
26             cout << "Matriks belum dibuat.\n";
27             return;
28         }
29         cout << "\n" << name << " (" << rows << "x" << cols << "):\n";
30         for (const auto& row : data) {
31             for (int val : row)
32                 cout << val << "\t";
33             cout << '\n';
34         }
35     }
36 };
37
38 vector<Matrix> matrices(5);
39 Matrix resultAdd, resultMultiply;
40
41 bool areAllMatricesCreated() {
42     for (const auto& matrix : matrices)
43         if (!matrix.isCreated) return false;
44     return true;
45 }
46
47 bool isValidMatrixForOperation(int index) {
48     return index >= 0 && index < matrices.size() && matrices[index].isCreated;
49 }
50
51 void createMatrices() {
52     cout << "\n=== MEMBUAT MATRIKS ===\n";
53
54     int sizes[5][2] = {{3, 4}, {3, 4}, {3, 4}, {2, 3}, {2, 4}};
55     for (int i = 0; i < 5; ++i) {
56         matrices[i] = {sizes[i][0], sizes[i][1], {}};
57         matrices[i].input();
58     }
59     cout << "\nSemua matriks berhasil dibuat!\n";
60 }
61
62 void displayAllMatrices() {
63     cout << "\n=== MENAMPILKAN SEMUA MATRIKS ===\n";
64     if (!areAllMatricesCreated()) {
65         cout << "Beberapa matriks belum dibuat.\n";
66         return;
67     }
68     for (const auto& matrix : matrices) matrix.print();
69 }
70
71 void addMatrices() {
72     cout << "\n=== PENAMBAHAN MATRIKS ===\n";
73     int idx1, idx2;
74     cout << "Indeks matriks pertama (0-4): "; cin >> idx1;
75     cout << "Indeks matriks kedua (0-4): "; cin >> idx2;
76
77     if (!isValidMatrixForOperation(idx1) || !isValidMatrixForOperation(idx2)) {
78         cout << "Indeks tidak valid.\n"; return;
79     }
80
81     const auto& A = matrices[idx1];
82     const auto& B = matrices[idx2];
83
84     if (A.rows != B.rows || A.cols != B.cols) {
85         cout << "Dimensi tidak sesuai untuk penambahan.\n"; return;
86     }
87
88     resultAdd = {A.rows, A.cols, vector<vector<int>>(A.rows, vector<int>(A.cols))};
89     for (int i = 0; i < A.rows; ++i)
90         for (int j = 0; j < A.cols; ++j)
91             resultAdd.data[i][j] = A.data[i][j] + B.data[i][j];
92     cout << "Penambahan berhasil.\n";
93 }

```

```

95 void multiplyMatrices() {
96     cout << "\n==== PERKALIAN MATRIKS ====\n";
97     int idx1, idx2;
98     cout << "Indeks matriks pertama (0-4): "; cin >> idx1;
99     cout << "Indeks matriks kedua (0-4): "; cin >> idx2;
100
101     if (!isValidMatrixForOperation(idx1) || !isValidMatrixForOperation(idx2)) {
102         cout << "Indeks tidak valid.\n"; return;
103     }
104
105     const auto& A = matrices[idx1];
106     const auto& B = matrices[idx2];
107
108     if (A.cols != B.rows) {
109         cout << "Dimensi tidak sesuai untuk perkalian.\n"; return;
110     }
111
112     resultMultiply = {A.rows, B.cols, vector<vector<int>>(A.rows, vector<int>(B.cols, 0))}, "Hasil Perkalian", true);
113     for (int i = 0; i < A.rows; i++)
114         for (int j = 0; j < B.cols; j++)
115             for (int k = 0; k < A.cols; k++)
116                 resultMultiply.data[i][j] += A.data[i][k] * B.data[k][j];
117     cout << "Perkalian berhasil.\n";
118 }
119
120 void extractDiagonalMatrix(int idx) {
121     const auto& A = matrices[idx];
122     int n = min(A.rows, A.cols);
123     resultAdd = {n, n, vector<vector<int>>(n, vector<int>(n, 0))}, "Diagonal dari " + A.name, true);
124     for (int i = 0; i < n; i++) resultAdd.data[i][i] = A.data[i][i];
125     cout << "Diagonal matriks berhasil dibuat.\n";
126 }
127
128 void extractLowerTriangularMatrix(int idx) {
129     const auto& A = matrices[idx];
130     resultMultiply = {A.rows, A.cols, vector<vector<int>>(A.rows, vector<int>(A.cols, 0))}, "Segitiga Bawah dari " + A.name, true);
131     for (int i = 0; i < A.rows; i++)
132         for (int j = 0; j <= i && j < A.cols; j++)
133             resultMultiply.data[i][j] = A.data[i][j];
134     cout << "Segitiga bawah matriks berhasil dibuat.\n";
135 }
136
137 void extractUpperTriangularMatrix(int idx) {
138     const auto& A = matrices[idx];
139     resultAdd = {A.rows, A.cols, vector<vector<int>>(A.rows, vector<int>(A.cols, 0))}, "Segitiga Atas dari " + A.name, true);
140     for (int i = 0; i < A.rows; i++)
141         for (int j = i; j < A.cols; j++)
142             resultAdd.data[i][j] = A.data[i][j];
143     cout << "Segitiga atas matriks berhasil dibuat.\n";
144 }
145
146 void showMenu() {
147     cout << "\n==== MENU MATRIKS ====\n"
148         << "1. Buat matriks\n2. Tampilkan semua matriks\n"
149         << "3. Penambahan matriks\n4. Perkalian matriks\n"
150         << "5. Diagonal matriks\n6. Segitiga bawah\n"
151         << "7. Segitiga atas\n8. Keluar\n"
152         << "Pilihan: ";
153 }
154
155 int main() {
156     int choice;
157     while (true) {
158         showMenu();
159         cin >> choice;
160         cin.ignore(numeric_limits<streamsize>::max(), '\n');
161
162         if (choice == 8) break;
163         if (!areAllMatricesCreated() && choice != 1) {
164             cout << "Silakan buat semua matriks terlebih dahulu.\n";
165             continue;
166         }
167
168         switch (choice) {
169             case 1: createMatrices(); break;
170             case 2: displayAllMatrices(); break;
171             case 3: addMatrices(); resultAdd.print(); break;
172             case 4: multiplyMatrices(); resultMultiply.print(); break;
173             case 5: { int idx; cout << "Indeks matriks (0-4): "; cin >> idx;
174                     if (isValidMatrixForOperation(idx)) extractDiagonalMatrix(idx);
175                     resultAdd.print(); break; }
176             case 6: { int idx; cout << "Indeks matriks (0-4): "; cin >> idx;
177                     if (isValidMatrixForOperation(idx)) extractLowerTriangularMatrix(idx);
178                     resultMultiply.print(); break; }
179             case 7: { int idx; cout << "Indeks matriks (0-4): "; cin >> idx;
180                     if (isValidMatrixForOperation(idx)) extractUpperTriangularMatrix(idx);
181                     resultAdd.print(); break; }
182             default: cout << "Pilihan tidak valid.\n";
183         }
184     }
185     cout << "Terima kasih telah menggunakan program.\n";
186     return 0;
187 }
188

```

- Output :

Pilihan membuat matrix :

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 1

=== MEMBUAT MATRIKS ===

Masukkan nilai untuk Matrix 1 (3x4):
Nilai [0][0]: 1
Nilai [0][1]: 1
Nilai [0][2]: 2
Nilai [0][3]: 2
Nilai [1][0]: 3
Nilai [1][1]: 3
Nilai [1][2]: 3
Nilai [1][3]: 4
Nilai [2][0]: 4
Nilai [2][1]: 4
Nilai [2][2]: 5
Nilai [2][3]: 5

Masukkan nilai untuk Matrix 2 (3x4):
Nilai [0][0]: 1
Nilai [0][1]: 1
Nilai [0][2]: 1
Nilai [0][3]: 2
Nilai [1][0]: 2
Nilai [1][1]: 2
Nilai [1][2]: 3
Nilai [1][3]: 3
Nilai [2][0]: 4
Nilai [2][1]: 6
Nilai [2][2]: 6
Nilai [2][3]: 8

Masukkan nilai untuk Matrix 3 (3x4):
Nilai [0][0]: 1
Nilai [0][1]: 2
Nilai [0][2]: 3
Nilai [0][3]: 4
Nilai [1][0]: 5
Nilai [1][1]: 6
Nilai [1][2]: 7
Nilai [1][3]: 8
Nilai [2][0]: 9
Nilai [2][1]: 0
Nilai [2][2]: 1
Nilai [2][3]: 2

Masukkan nilai untuk Matrix 4 (2x3):
Nilai [0][0]: 1
Nilai [0][1]: 3
Nilai [0][2]: 4
Nilai [1][0]: 5
Nilai [1][1]: 7
Nilai [1][2]: 8

Masukkan nilai untuk Matrix 5 (2x4):
Nilai [0][0]: 2
Nilai [0][1]: 5
Nilai [0][2]: 7
Nilai [0][3]: 9
Nilai [1][0]: 1
Nilai [1][1]: 2
Nilai [1][2]: 3
Nilai [1][3]: 4

Semua matriks berhasil dibuat!

```

Menampilkan matrix :

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 2

=== MENAMPILKAN SEMUA MATRIKS ===

Matrix 1 (3x4) (3x4):
1      1      2      2
3      3      3      4
4      4      5      5

Matrix 2 (3x4) (3x4):
1      1      1      2
2      2      3      3
4      6      6      8

Matrix 3 (3x4) (3x4):
1      2      3      4
5      6      7      8
9      0      1      2

Matrix 4 (2x3) (2x3):
1      3      4
5      7      8

Matrix 5 (2x4) (2x4):
2      5      7      9
1      2      3      4

```

Penambahan Matrix ke 2 dan ke 3

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 3

=== PENAMBAHAN MATRIKS ===
Indeks matriks pertama (0-4): 1
Indeks matriks kedua (0-4): 2
Penambahan berhasil.

Hasil Penambahan (3x4):
2      3      4      6
7      8      10     11
13     6      7      10

```

Perkalian matrix ke 1 dan ke 4 (fail)

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 4

=== PERKALIAN MATRIKS ===
Indeks matriks pertama (0-4): 0
Indeks matriks kedua (0-4): 3
Dimensi tidak sesuai untuk perkalian.
Matriks belum dibuat.

```

Perkalian matrix ke 4 dan ke 1 (success)

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 4

=== PERKALIAN MATRIKS ===
Indeks matriks pertama (0-4): 3
Indeks matriks kedua (0-4): 0
Perkalian berhasil.

Hasil Perkalian (2x4):
26      26      31      34
58      58      71      78

```

Diagonal matrix ke 1

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 5
Indeks matriks (0-4): 0
Diagonal matriks berhasil dibuat.

Diagonal dari Matrix 1 (3x4) (3x3):
1      0      0
0      3      0
0      0      5

```

Segitiga bawah matrix ke 5

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 6
Indeks matriks (0-4): 4
Segitiga bawah matriks berhasil dibuat.

Segitiga Bawah dari Matrix 5 (2x4) (2x4):
2      0      0      0
1      2      0      0

```

Segitiga atas matrix ke-3

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 7
Indeks matriks (0-4): 2
Segitiga atas matriks berhasil dibuat.

Segitiga Atas dari Matrix 3 (3x4) (3x4):
1      2      3      4
0      6      7      8
0      0      1      2

```

Exit

```

===== MENU MATRIKS =====
1. Buat matriks
2. Tampilkan semua matriks
3. Penambahan matriks
4. Perkalian matriks
5. Diagonal matriks
6. Segitiga bawah
7. Segitiga atas
8. Keluar
Pilihan: 8
Terima kasih telah menggunakan program.

Praktikan@CDSR3-33 MSYS /c/iam
$

```