**LAPORAN**

**PRAKTIKUM ANALISIS ALGORITMA**

**TUGAS 05**



**Disusun Oleh:**

Risyad Pangestu (140810170003)

Dimas Satria Prakoso (140810170007)

Syaina Nur Fauziyah (140810170025)

Imron Madani (140810170061)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**5SUMEDANG**

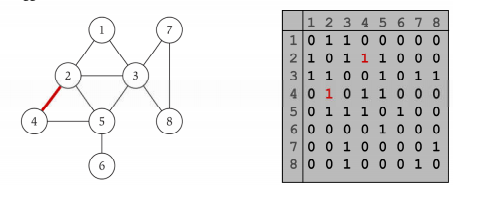
**2019**

**LAPORAN**

**PRAKTIKUM ANALISIS ALGORITMA**

**TUGAS 05**

1. Dengan menggunakan undirected graph dan adjacency matrix berikut, buatlah koding programmnya menggunakan bahasa C++



**Program:**

/\*

\* C++ Program to Implement Adjacency Matrix

\*/

#include <iostream>

#include <cstdlib>

using namespace std;

#define MAX 20

/\*

\* Adjacency Matrix Class

\*/

class AdjacencyMatrix

{

private:

int n;

int \*\*adj;

bool \*visited;

public:

AdjacencyMatrix(int n)

{

this->n = n;

visited = new bool [n];

adj = new int\* [n];

for (int i = 0; i < n; i++)

{

adj[i] = new int [n];

for(int j = 0; j < n; j++)

{

adj[i][j] = 0;

}

}

}

/\*

\* Adding Edge to Graph

\*/

void add\_edge(int origin, int destin)

{

if( origin > n || destin > n || origin < 0 || destin < 0)

{

cout<<"Invalid edge!\n";

}

else

{

adj[origin - 1][destin - 1] = 1;

}

}

/\*

\* Print the graph

\*/

void display()

{

int i,j;

for(i = 0;i < n;i++)

{

for(j = 0; j < n; j++)

cout<<adj[i][j]<<" ";

cout<<endl;

}

}

};

/\*

\* Main

\*/

int main()

{

int nodes, max\_edges, origin, destin;

cout<<"Enter number of nodes: ";

cin>>nodes;

AdjacencyMatrix am(nodes);

max\_edges = nodes \* (nodes - 1);

for (int i = 0; i < max\_edges; i++)

{

cout<<"Enter edge (-1 -1 to exit): ";

cin>>origin>>destin;

if((origin == -1) && (destin == -1))

break;

am.add\_edge(origin, destin);

}

am.display();

return 0;

}

**Output:**

