**LAPORAN PRAKTIKUM**

**STRUKTUR DATA**

**MODUL 8**

****

|  |
| --- |
| **Disusun Oleh:**  Nama: Juan Axl Ronaldio Zaka Putra  NIM: 220411100066  Kelas: IF 2D  **Dosen Pengampu:**  Nama: Hermawan, S.T., M.Kom  NIP: 197908282005011002  **Asisten Praktikum:**  Nama: Moh. Fadil Abdillah  NIM: 210411100142 |

**PRODI TEKNIK INFORMATIKA**

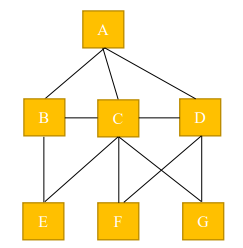
**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2023**

1. **Soal**

Buatlah program dari gambar graph di bawah ini!

****

**Jawaban**

**Code Program:**

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  void printAdjacencyMatrix(vector<vector<int>> &adjMatrix)  {  int n = adjMatrix.size();  for (int i = 0; i < n; i++)  {  cout << '[';  for (int j = 0; j < n; j++)  {  cout << adjMatrix[i][j] << " ";  }  cout << ']';  cout << endl;  }  }  int main()  {  int n = 7; // Jumlah simpul  vector<vector<int>> adjMatrix(n, vector<int>(n, 0));  // Mengatur sisi-sisi pada adjacency matrix  adjMatrix[0][1] = 1;  adjMatrix[0][2] = 1;  adjMatrix[0][3] = 1;  adjMatrix[1][0] = 1;  adjMatrix[1][2] = 1;  adjMatrix[1][4] = 1;  adjMatrix[2][0] = 1;  adjMatrix[2][1] = 1;  adjMatrix[2][3] = 1;  adjMatrix[2][4] = 1;  adjMatrix[2][5] = 1;  adjMatrix[2][6] = 1;  adjMatrix[3][0] = 1;  adjMatrix[3][2] = 1;  adjMatrix[3][5] = 1;  adjMatrix[3][6] = 1;  adjMatrix[4][1] = 1;  adjMatrix[4][2] = 1;  adjMatrix[5][2] = 1;  adjMatrix[5][3] = 1;  adjMatrix[6][2] = 1;  adjMatrix[6][3] = 1;  cout << "Adjacency matrix:" << endl;  printAdjacencyMatrix(adjMatrix);  return 0;  } |

**Penjelasan Code Program:**

Berdasarkan gambar data graph pada soal, maka dapat diubah kedalam tabel matriks sebagai berikut:

**Permisalan:**

**0 = A**

**1 = B**

**2 = C**

**3 = D**

**4 = E**

**5 = F**

**6 = G**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **1** | **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **1** | **1** | **1** | **1** |
| **1** | **0** | **1** | **0** | **0** | **1** | **1** |
| **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** | **0** | **0** | **0** |

Setelah didapatkan tabel tersebut, maka dapat mengatur sisi-sisi pada adjacency matriks dengan menginputkan nilai matriks pada index sesuai dengan posisi baris dan kolom pada tabel yang bernilai 1 dan pada index matriks tersebut juga diberi nilai 1.

**Hasil Running Program:**

