PRAKTIKUM 14
GRAPH

Dosen Pengampu :
Ibu Entin Martiana Kusumaningtyas S.Kom, M.Kom

Disusun Oleh :

ADE HAFIS RABBANI
1 D3 IT A
3122500002

Politeknik Elektronika Negeri Surbaya
Program Studi Teknik Informatika

Mei 2023

D. PERCOBAAN

Percobaan 1 : Mendeklarasikan matriks Beban, Jalur dan Rute

- **Program**

```
#include <stdio.h>
#define N 5
#define M 1000
void Tampil(int data[N][N], char *judul)
{
  printf("%s = \n", judul);
  for (int i = 0; i < N; i++)
  {
    for (int j = 0; j < N; j++)
      if (data[i][j] >= M)
        printf("M ");
      else
        printf("%d ", data[i][j]);
    printf("\n");
  }
}
void main()
{
  int Beban[N][N] = {M, 1, 3, M, M,
                     M, M, 1, M, 5,
                     3, M, M, 2, M,
                     M, M, M, M, 1,
                     M, M, M, M, M};
  int Jalur[N][N] = {0, 1, 1, 0, 0,
                     0, 0, 1, 0, 1,
                     1, 0, 0, 1, 0,
                     0, 0, 0, 0, 1,
                     0, 0, 0, 0, 0};
  int Rute[N][N] = {M, 0, 0, M, M,
                    M, M, 0, M, 0,
                    0, M, M, 0, M,
                    M, M, M, M, 0,
                    M, M, M, M, M};
  Tampil(Beban, "Beban");
  Tampil(Jalur, "Jalur");
  Tampil(Rute, "Rute");
}
```

- **Output**

```
Beban =
M 1 3 M M
M M 1 M 5
3 M M 2 M
M M M M 1
M M M M M
Jalur =
0 1 1 0 0
0 0 1 0 1
1 0 0 1 0
0 0 0 0 1
0 0 0 0 0
Rute =
M 0 0 M M
M M 0 M 0
0 M M 0 M
M M M M 0
M M M M M
PS E:\Kuliah\Sems2\ASD\Praktikum13>
```

- **Analisa**

  Program di atas merupakan contoh kode dalam bahasa C yang menampilkan matriks 5x5 dengan menggunakan fungsi Tampil. Program ini memiliki tiga matriks yaitu Beban, Jalur, dan Rute.

  Matriks Beban berfungsi untuk menyimpan nilai beban antar titik dalam suatu jaringan. Nilai M digunakan sebagai penanda jika tidak ada koneksi antar titik. Nilai M ini setara dengan 1000 dalam program ini. Matriks ini digunakan untuk menampilkan beban antar titik dengan menggunakan fungsi Tampil.

  Matriks Jalur berfungsi untuk menyimpan informasi apakah terdapat jalur atau tidak antara dua titik. Nilai 1 menandakan adanya jalur, sedangkan nilai 0 menandakan tidak ada jalur. Matriks ini juga ditampilkan menggunakan fungsi Tampil.

  Matriks Rute digunakan untuk menyimpan jalur terpendek antara dua titik. Nilai M digunakan sebagai penanda jika belum ada jalur yang ditemukan. Matriks ini juga ditampilkan menggunakan fungsi Tampil.

  Fungsi Tampil digunakan untuk menampilkan matriks dengan judul yang diberikan. Jika nilai dalam matriks melebihi atau sama dengan M, maka akan ditampilkan karakter 'M'. Jika tidak, nilai tersebut akan ditampilkan sebagai bilangan integer.

Pada fungsi main, matriks Beban, Jalur, dan Rute diinisialisasi dengan nilai-nilai tertentu. Kemudian ketiga matriks tersebut ditampilkan menggunakan fungsi Tampil dengan judul yang sesuai.

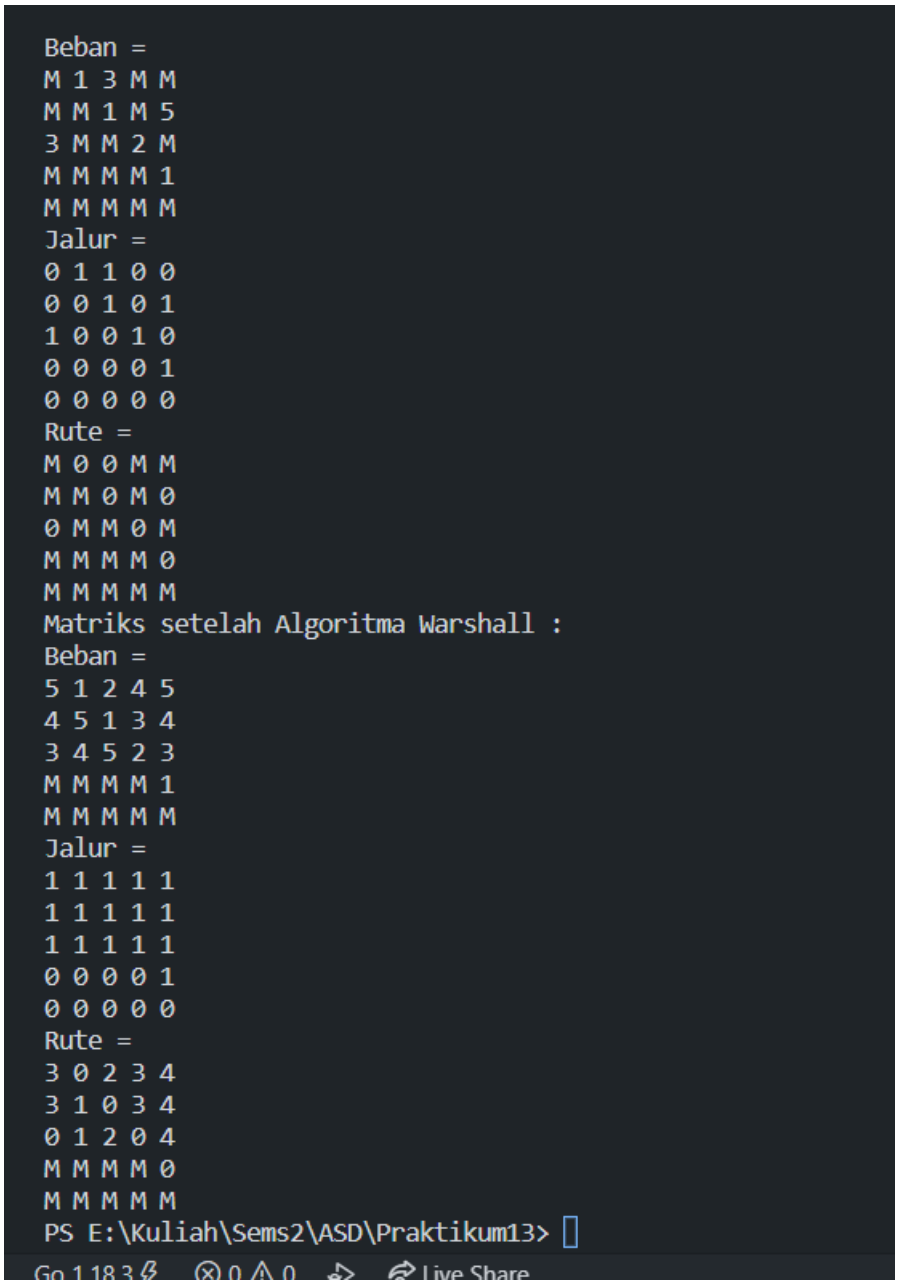
Percobaan 2 : Algoritma Warshall untuk pencarian jalur terpendek multipath

- **Program**

```
#include <stdio.h>
#define N 5
#define M 1000
void Tampil(int data[N][N], char *judul)
{
  printf("%s = \n", judul);
  for (int i = 0; i < N; i++)
  {
    for (int j = 0; j < N; j++)
      if (data[i][j] >= M)
        printf("M ");
      else
        printf("%d ", data[i][j]);
    printf("\n");
  }
}
void Warshall(int Q[N][N], int P[N][N], int R[N][N])
{
  for (int k = 0; k < N; k++)
    for (int i = 0; i < N; i++)
      for (int j = 0; j < N; j++)
      {
        P[i][j] = P[i][j] | (P[i][k] & P[k][j]);
        if ((Q[i][k] + Q[k][j]) < Q[i][j])
        {
          Q[i][j] = Q[i][k] + Q[k][j];
          if (R[k][j] == 0)
            R[i][j] = k + 1;
          else
            R[i][j] = R[k][j];
        }
      }
}
void main()
{
  int Beban[N][N] = {M, 1, 3, M, M,
                     M, M, 1, M, 5,
                     3, M, M, 2, M,
                     M, M, M, M, 1,
                     M, M, M, M, M};
  int Jalur[N][N] = {0, 1, 1, 0, 0,
                     0, 0, 1, 0, 1,
                     1, 0, 0, 1, 0,
```

```
                      0, 0, 0, 0, 1,
                      0, 0, 0, 0, 0};
  int Rute[N][N] = {M, 0, 0, M, M,
                    M, M, 0, M, 0,
                    0, M, M, 0, M,
                    M, M, M, M, 0,
                    M, M, M, M, M};
  Tampil(Beban, "Beban");
  Tampil(Jalur, "Jalur");
  Tampil(Rute, "Rute");
  Warshall(Beban, Jalur, Rute);
  printf("Matriks setelah Algoritma Warshall : \n");
  Tampil(Beban, "Beban");
  Tampil(Jalur, "Jalur");
  Tampil(Rute, "Rute");
}
```

- **Output**



- **Analisa**

  Program di atas merupakan contoh kode dalam bahasa C yang mengimplementasikan algoritma Warshall untuk mencari jalur terpendek dalam suatu graf. Program ini menggunakan fungsi Warshall untuk melakukan pemrosesan algoritma Warshall pada matriks Beban, Jalur, dan Rute.

  Fungsi Warshall mengimplementasikan algoritma Warshall dengan menggunakan tiga matriks sebagai argumen: Q untuk matriks beban, P untuk matriks jalur, dan R untuk matriks rute. Algoritma Warshall digunakan untuk memperbarui nilai-nilai dalam matriks Q, P, dan R

sehingga merepresentasikan jalur terpendek antara setiap pasang titik dalam graf.

Di dalam fungsi Warshall, terdapat tiga nested loop yang digunakan untuk mengiterasi semua pasangan titik dalam graf. Pada setiap iterasi, algoritma Warshall memperbarui nilai-nilai dalam matriks P, Q, dan R sesuai dengan aturan algoritma Warshall.

Fungsi main merupakan entry point program yang melakukan inisialisasi awal terhadap matriks Beban, Jalur, dan Rute. Kemudian, ketiga matriks tersebut ditampilkan menggunakan fungsi Tampil.

Setelah itu, program memanggil fungsi Warshall untuk melakukan pemrosesan algoritma Warshall pada matriks Beban, Jalur, dan Rute. Setelah pemrosesan selesai, hasilnya ditampilkan dengan memanggil fungsi Tampil kembali.

E. LATIHAN

1. Dari percobaan 2 diatas, tambahkan fungsi untuk menampilkan rute dari satu titik ke titik berikutnya berdasarkan matriks rute. Input berupa titik awal dan titik akhir. Gunakan struktur dan Stack untuk mencari rute seperti langkah – langkah dibawah ini.

    1. Rute 1-5?
    2. Ambil nilai di baris 1, kolom 5 = 4 ⟶ push
    3. Ambil nilai di baris 1, kolom 4 = 3 ⟶ push
    4. Ambil nilai di baris 1, kolom 3 = 2 ⟶ push
    5. Ambil nilai di baris 1, kolom 2 = 0 (stop) → pop sampai stack kosong
    6. Sehingga rutenya menjadi 1-2-3-4-5 dengan beban minimal 5.

    Jawab :

    - **Program**

```c
#include <stdio.h>
#include <stdlib.h>

#define N 5
#define M 1000


typedef struct{
    int Data[N];
    int top;
} Graph;

void initGraph(Graph *G){
    G->top = -1;
}

void push(Graph *G, int data){
    G->top++;
    G->Data[G->top] = data;
}

int pop(Graph *G){
    int temp = G->Data[G->top];
    G->top--;
    return temp;
}


int isEmpty(Graph *G){
```

```
    if(G->top == -1){
        return 1;
    }else{
        return 0;
    }
}

int isFull(Graph *G){
    if(G->top == N-1){
        return 1;
    }else{
        return 0;
    }
}


void Tampil(int data[N][N], char *judul)
{
  printf("%s = \n", judul);
  for (int i = 0; i < N; i++)
  {
    for (int j = 0; j < N; j++)
      if (data[i][j] >= M)
        printf("M ");
      else
        printf("%d ", data[i][j]);
    printf("\n");
  }
}
void Warshall(int Q[N][N], int P[N][N], int R[N][N])
{
  for (int k = 0; k < N; k++)
    for (int i = 0; i < N; i++)
      for (int j = 0; j < N; j++)
      {
        P[i][j] = P[i][j] | (P[i][k] & P[k][j]);
        if ((Q[i][k] + Q[k][j]) < Q[i][j])
        {
          Q[i][j] = Q[i][k] + Q[k][j];
          if (R[k][j] == 0)
            R[i][j] = k + 1;
          else
            R[i][j] = R[k][j];
        }
      }
}
```

```
void findRoute(int R[N][N], int Beban[N][N], int start,
int end){
    printf("Beban dari %d ke %d adalah : %d\n", start,
end, Beban[start-1][end-1]);
    Graph G;
    initGraph(&G);
    push(&G, end);
    int temp = R[start-1][end-1];
    while(temp != 0){
        push(&G, temp);
        temp = R[start-1][temp-1];
    }
    push(&G, start);
    printf("Rute : ");
    while(!isEmpty(&G)){
        printf("%d ", pop(&G));
    }
    printf("\n");
}


void main()
{
  int Beban[N][N] = {M, 1, 3, M, M,
                     M, M, 1, M, 5,
                     3, M, M, 2, M,
                     M, M, M, M, 1,
                     M, M, M, M, M};
  int Jalur[N][N] = {0, 1, 1, 0, 0,
                     0, 0, 1, 0, 1,
                     1, 0, 0, 1, 0,
                     0, 0, 0, 0, 1,
                     0, 0, 0, 0, 0};
  int Rute[N][N] = {M, 0, 0, M, M,
                    M, M, 0, M, 0,
                    0, M, M, 0, M,
                    M, M, M, M, 0,
                    M, M, M, M, M};
  Tampil(Beban, "Beban");
  Tampil(Jalur, "Jalur");
  Tampil(Rute, "Rute");
  Warshall(Beban, Jalur, Rute);
  printf("Matriks setelah Algoritma Warshall : \n");
  Tampil(Beban, "Beban");
```

```
  Tampil(Jalur, "Jalur");
  Tampil(Rute, "Rute");

    findRoute(Rute, Beban, 1, 5);
}
```

- **Output**

```
M 1 3 M M
M M 1 M 5
3 M M 2 M
M M M M 1
M M M M M
Jalur =
0 1 1 0 0
0 0 1 0 1
1 0 0 1 0
0 0 0 0 1
0 0 0 0 0
Rute =
M 0 0 M M
M M 0 M 0
0 M M 0 M
M M M M 0
M M M M M
Matriks setelah Algoritma Warshall :
Beban =
5 1 2 4 5
4 5 1 3 4
3 4 5 2 3
M M M M 1
M M M M M
Jalur =
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
0 0 0 0 1
0 0 0 0 0
Rute =
3 0 2 3 4
3 1 0 3 4
0 1 2 0 4
M M M M 0
M M M M M
Beban dari 1 ke 5 adalah : 5
Rute : 1 2 3 4 5
PS E:\Kuliah\Sems2\ASD\Praktikum13>
```

- **Analisa**

Kode tersebut mengimplementasikan algoritma Warshall untuk mencari jarak terpendek antara setiap pasangan titik dalam graf.

Fungsi `Warshall` menghitung matriks `Beban` yang menyimpan jarak terpendek antara setiap pasangan titik, matriks `Jalur` yang menyimpan informasi apakah ada jalur antara dua titik, dan matriks `Rute` yang menyimpan jalur terpendek antara setiap pasangan titik.

Fungsi `Tampil` digunakan untuk menampilkan isi matriks.

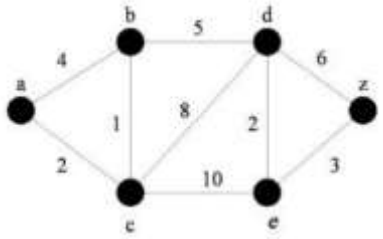
Fungsi `findRoute` digunakan untuk menemukan rute terpendek antara dua titik berdasarkan matriks `Rute` dan `Beban`.

Di dalam fungsi `main`, terdapat inisialisasi matriks `Beban`, `Jalur`, dan `Rute`, kemudian dipanggil fungsi `Warshall` untuk menghasilkan matriks yang diperbarui.

Setelah itu, fungsi `Tampil` digunakan untuk menampilkan matriks-matriks tersebut sebelum dan setelah algoritma Warshall.

Terakhir, fungsi `findRoute` dipanggil untuk menemukan rute terpendek dari titik 1 ke titik 5.

2. Berdasarkan *graph* dibawah ini, representasikan matriks, gunakan algoritma warshall untuk mencari rute terpendek dan rute seperti pada Latihan 1.



Jawab :

- **Program**

```
#include <stdio.h>
#include <stdlib.h>
#define M 1000
#define N 5
void cetak(int Mat[N][N], char *judul)
{
  int i, j;
  printf("%s: \n", judul);
  for (i = 0; i < N; i++)
  {
```

```
    for (j = 0; j < N; j++)
    {
      if (Mat[i][j] == M)
        printf("M ");
      else
        printf("%d ", Mat[i][j]);
    }
    printf("\n");
  }
  printf("\n");
}
void warshall(int Q[N][N], int P[N][N], int R[N][N])
{
  int i, j, k;
  for (k = 0; k < N; k++)
  {
    for (i = 0; i < N; i++)
    {
      for (j = 0; j < N; j++)
      {
        if (Q[i][k] + Q[k][j] < Q[i][j])
        {
          Q[i][j] = Q[i][k] + Q[k][j];
          if (R[k][j] == 0)
            R[i][j] = k + 1;
          else
            R[i][j] = R[k][j];
          P[i][j] = P[i][j] | (P[i][k] & P[k][j]);
        }
      }
    }
  }
}
int main()
{
  int Q[N][N] = {M, 4, 2, M, M, M,
                 M, M, 1, 5, M, M,
                 M, M, M, 8, 10, M,
                 M, M, M, M, 2, 6,
                 M, M, M, M, M, 3,
                 M, M, M, M, M, M};
  int P[N][N] = {0, 1, 1, 0, 0, 0,
                 0, 0, 1, 1, 0, 0,
                 0, 0, 0, 1, 1, 0,
                 0, 0, 0, 0, 1, 1,
                 0, 0, 0, 0, 0, 1,
```

```
                0, 0, 0, 0, 0, 0};
  int R[N][N] = {M, 0, 0, M, M, M,
                M, M, 0, 0, M, M,
                M, M, M, 0, 0, M,
                M, M, M, M, 0, 0,
                M, M, M, M, M, 0,
                M, M, M, M, M, M};
  printf("Matriks sebelum Warshall\n");
  cetak(Q, "Beban");
  cetak(P, "Jalur");
  cetak(R, "Rute");
  warshall(Q, P, R);
  printf("Matriks setelah Warshall\n");
  cetak(Q, "Beban");
  cetak(P, "Jalur");
  cetak(R, "Rute");
  return 0;
}
```

- **Output**

```
Jalur:
0 1 1 0 0
0 0 0 1 1
0 0 0 0 0
1 1 0 0 0
0 0 1 1 0

Rute:
M 0 0 M M
M M M 0 0
M M M M M
0 0 M M M
M M 0 0 M

Matriks setelah Warshall
Beban:
13 4 2 5 9
9 11 7 1 5
M M M M M
8 10 10 11 15
14 16 2 6 21

Jalur:
1 1 1 1 1
1 1 1 1 1
0 0 0 0 0
1 1 1 1 1
1 1 1 1 1

Rute:
4 0 0 2 2
4 4 5 0 0
M M M M M
0 0 1 2 2
4 4 0 0 2

PS E:\Kuliah\Sems2\ASD\Praktikum13>
```

- **Analisa**
- Kode mendefinisikan konstanta N sebagai ukuran matriks graf, dan M sebagai nilai tak terhingga yang digunakan untuk menyatakan ketiadaan jalur antara dua simpul.
- Terdapat fungsi cetak yang digunakan untuk mencetak matriks graf ke layar.
- Fungsi warshall mengimplementasikan algoritma Warshall untuk mencari jalur terpendek dan beban minimal. Algoritma ini menggunakan tiga matriks, yaitu matriks Beban (Q), matriks Jalur (P), dan matriks Rute (R). Matriks Beban menyimpan bobot lintasan antara dua simpul, matriks Jalur menyimpan informasi apakah ada jalur langsung antara dua simpul, dan matriks Rute

menyimpan simpul perantara pada jalur terpendek antara dua simpul.

- Fungsi main merupakan program utama yang melakukan inisialisasi matriks Beban, Jalur, dan Rute. Kemudian, matriks tersebut dicetak ke layar sebelum dilakukan algoritma Warshall. Selanjutnya, dilakukan pemanggilan fungsi warshall untuk menghitung jalur terpendek dan beban minimal. Setelah itu, matriks hasil perhitungan dicetak ke layar.
- Hasil akhir yang ditampilkan adalah matriks Beban, Jalur, dan Rute sebelum dan setelah dilakukan algoritma Warshall.

3. Rancanglah sendiri sebuah graph dan lakukan hal yang sama dengan Latihan 2.

Jawab :

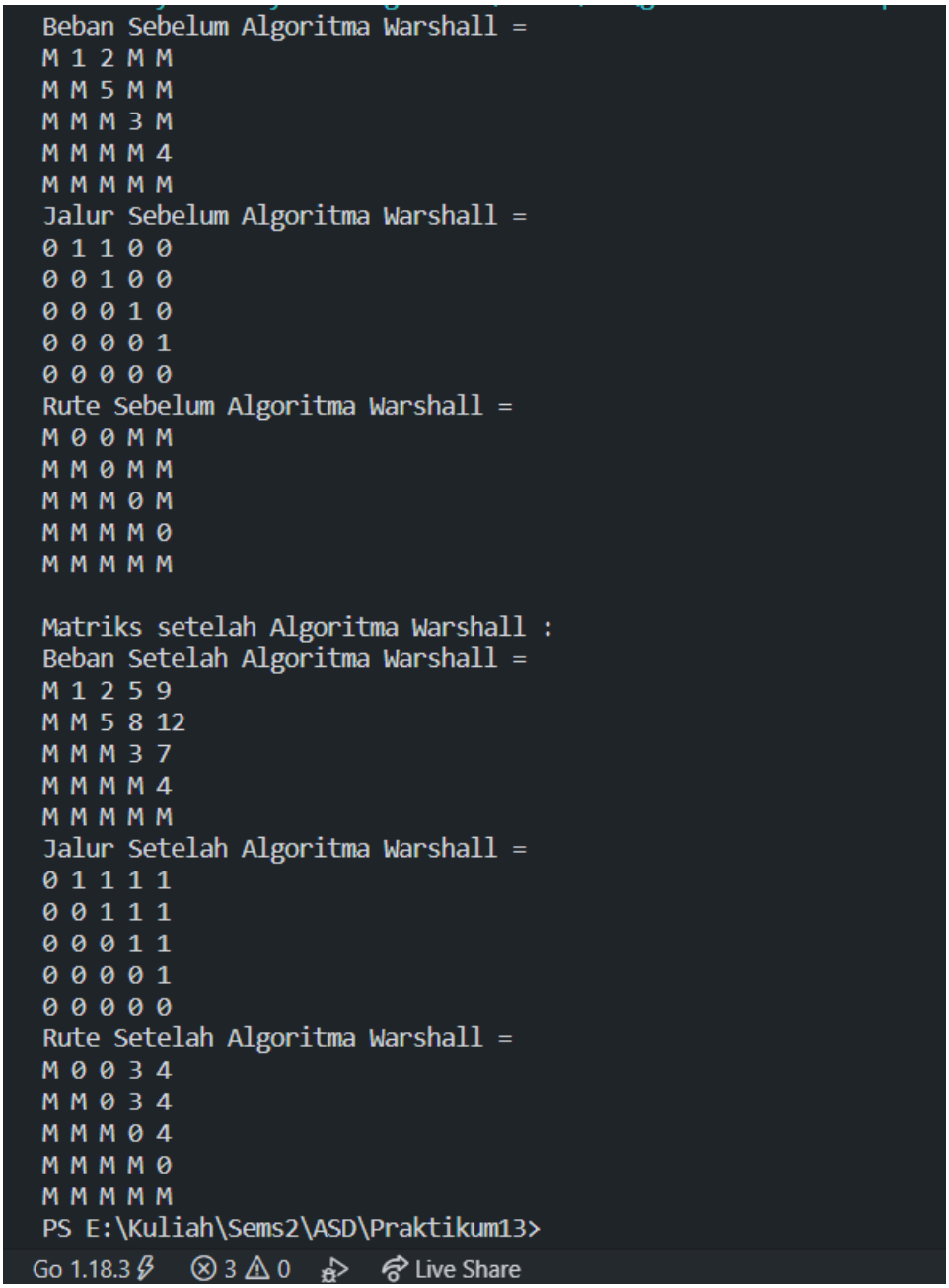
- **Program**

```c
#include <stdio.h>
#include <stdlib.h>

#define N 5
#define M 1000

void Tampil(int data[N][N], char *judul)
{
  printf("%s = \n", judul);
  for (int i = 0; i < N; i++)
  {
    for (int j = 0; j < N; j++)
      if (data[i][j] >= M)
        printf("M ");
      else
        printf("%d ", data[i][j]);
    printf("\n");
  }
}

void Warshall(int Q[N][N], int P[N][N], int R[N][N])
{
  for (int k = 0; k < N; k++)
    for (int i = 0; i < N; i++)
      for (int j = 0; j < N; j++)
      {
        P[i][j] = P[i][j] | (P[i][k] & P[k][j]);
        if ((Q[i][k] + Q[k][j]) < Q[i][j])
        {
```

```
          Q[i][j] = Q[i][k] + Q[k][j];
          if (R[k][j] == 0)
            R[i][j] = k + 1;
          else
            R[i][j] = R[k][j];
        }
      }
}

int main()
{
  int Beban[N][N] = {M, 1, 2, M, M,
                     M, M, 5, M, M,
                     M, M, M, 3, M,
                     M, M, M, M, 4,
                     M, M, M, M, M};
  int Jalur[N][N] = {0, 1, 1, 0, 0,
                     0, 0, 1, 0, 0,
                     0, 0, 0, 1, 0,
                     0, 0, 0, 0, 1,
                     0, 0, 0, 0, 0};
  int Rute[N][N] = {M, 0, 0, M, M,
                    M, M, 0, M, M,
                    M, M, M, 0, M,
                    M, M, M, M, 0,
                    M, M, M, M, M};
  Tampil(Beban, "Beban Sebelum Algoritma Warshall");
  Tampil(Jalur, "Jalur Sebelum Algoritma Warshall");
  Tampil(Rute, "Rute Sebelum Algoritma Warshall");

  Warshall(Beban, Jalur, Rute);
  printf("\nMatriks setelah Algoritma Warshall : \n");
  Tampil(Beban, "Beban Setelah Algoritma Warshall");
  Tampil(Jalur, "Jalur Setelah Algoritma Warshall");
  Tampil(Rute, "Rute Setelah Algoritma Warshall");

  return 0;
}
```

- **Output**
- Pada awalnya, kita mendefinisikan ukuran matriks N x N sebagai konstanta dengan nilai 5 dan M sebagai nilai tak terhingga yang digunakan untuk menyatakan jarak yang tidak terhubung antara dua titik.

- Terdapat fungsi Tampil yang digunakan untuk menampilkan matriks ke layar.
- Fungsi utama Warshall mengimplementasikan algoritma Warshall untuk mencari jalur terpendek dan beban minimal antara dua titik dalam graph. Algoritma Warshall menggunakan tiga matriks, yaitu matriks Beban, matriks Jalur, dan matriks Rute. Matriks Beban menyimpan bobot lintasan antara dua titik, matriks Jalur menyimpan informasi apakah ada jalur langsung antara dua titik, dan matriks Rute menyimpan titik perantara pada jalur terpendek antara dua titik.
- Fungsi main merupakan program utama yang melakukan inisialisasi matriks Beban, Jalur, dan Rute, kemudian menampilkan matriks tersebut sebelum dilakukan algoritma Warshall. Selanjutnya, dilakukan pemanggilan fungsi Warshall untuk menghitung jalur terpendek dan beban minimal. Setelah itu, matriks hasil perhitungan ditampilkan ke layar.
- Hasil akhir yang ditampilkan adalah matriks Beban, Jalur, dan Rute setelah dilakukan algoritma Warshall.


Beban Sebelum Algoritma Warshall =
M 1 2 M M
M M 5 M M
M M M 3 M
M M M M 4
M M M M M
Jalur Sebelum Algoritma Warshall =
0 1 1 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
0 0 0 0 0
Rute Sebelum Algoritma Warshall =
M 0 0 M M
M M 0 M M
M M M 0 M
M M M M 0
M M M M M
Matriks setelah Algoritma Warshall :
Beban Setelah Algoritma Warshall =
M 1 2 5 9
M M 5 8 12
M M M 3 7
M M M M 4
M M M M M
Jalur Setelah Algoritma Warshall =
0 1 1 1 1
0 0 1 1 1
0 0 0 1 1
0 0 0 0 1
0 0 0 0 0
Rute Setelah Algoritma Warshall =
M 0 0 3 4
M M 0 3 4
M M M 0 4
M M M M 0
M M M M M
PS E:\Kuliah\Sems2\ASD\Praktikum13>
Go 1.18.3
3 0
Live Share

- **Analisa**