**LAPORAN RESMI 15.1**

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

GRAPH WARSHALL



Nama : Aqilah Akmalia Dewi

Kelas : 1 D4 IT B

NRP : 3120600046

PROGRAM STUDI D4 TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

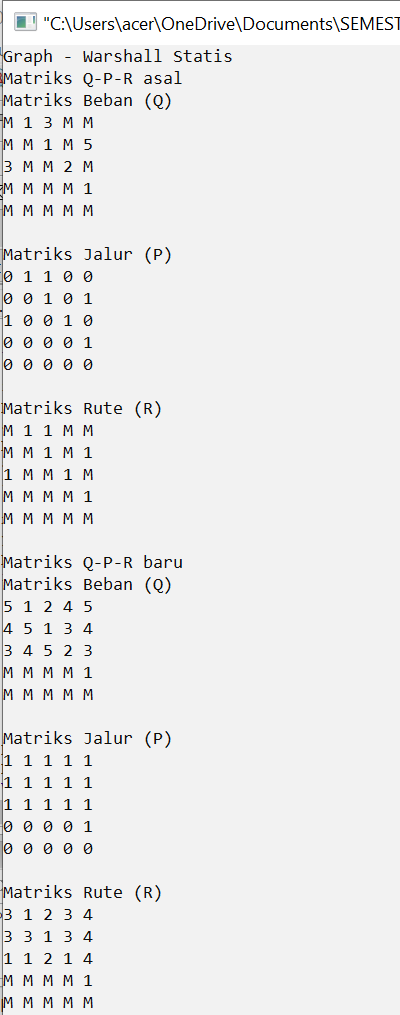
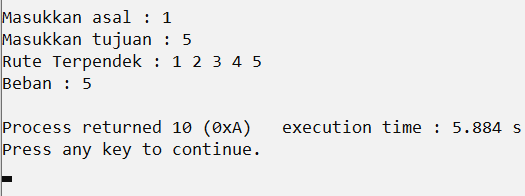
2021

# 1. WARSHALL STATIS

1. Listing Program

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define N 5  #define M 1000  #define MAX 30  typedef int itemType;  typedef struct{  itemType data[MAX];  int count;  }stack;  stack tumpukan;  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,1,1,M,M,  M,M,1,M,1,  1,M,M,1,M,  M,M,M,M,1,  M,M,M,M,M};  puts("Graph - Warshall Statis");  printf("Matriks Q-P-R asal\n");  tampil(Beban, "Matriks Beban (Q)");  tampil(Jalur, "Matriks Jalur (P)");  tampil(Rute, "Matriks Rute (R)");  warshall(Beban, Jalur, Rute);  printf("Matriks Q-P-R baru\n");  tampil(Beban, "Matriks Beban (Q)");  tampil(Jalur, "Matriks Jalur (P)");  tampil(Rute, "Matriks Rute (R)");  baca(Rute, Beban);  }  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");  }  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] == 1)  R[i][j] = k+1;  else  R[i][j] = R[k][j];  }  }  }  void baca(int R[N] [N], int Q[N] [N]){  int asal, tujuan, i, j, x, y;  inisialisasi(&tumpukan);  printf("Masukkan asal : ");  scanf("%d", &asal);  printf("Masukkan tujuan : ");  scanf("%d", &tujuan);  fflush(stdin);  x=asal-1;  y=tujuan-1;  push(tujuan,&tumpukan);  for(int i=x;i<=x;i++){  for(int j=y;j>=x;j--){  if(R[i][j]!=1){  push(R[i][j],&tumpukan);  }else{  break;  }  }  }  push(asal,&tumpukan);  printf("Rute Terpendek : ");  while(!kosong(&tumpukan)){  printf("%d ",pop(&tumpukan));  }  puts(" ");  printf("Beban : %d\n",Q[asal-1][tujuan-1]);  }  void inisialisasi (stack \*s){  s->count = 0;  }  int kosong (stack \*s){  if(s->count == 0)  return (1);  else  return (0);  }  int penuh(stack \*s){  if(s->count == MAX)  return (1);  else  return (0);  }  void push(itemType x, stack \*s){  if(penuh(s))  printf("Stack penuh, tidak bisa menyimpan data\n");  else{  s->data[s->count] = x;  s->count++;  }  }  itemType pop(stack \*s){  itemType x;  if(kosong(s)){  printf("Stack kosong, tidak bisa mengambil data\n");  }else{  --s->count;  x = s->data[s->count];  return x;  }  } |

1. Output

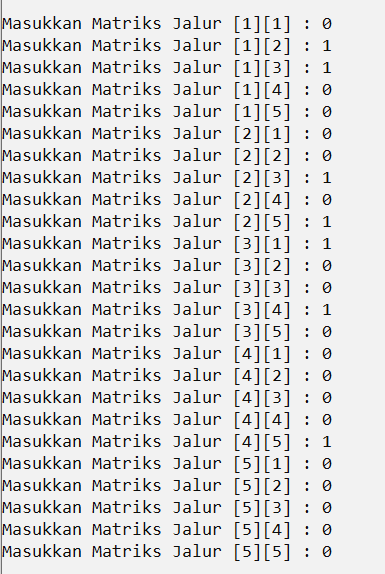


# 2. WARSHALL DINAMIS

1. Listing Program

|  |
| --- |
| #include<stdio.h>  #include <stdlib.h>  #define N 5  #define M 1000  #define MAX 30  typedef int itemType;  typedef struct{  itemType data[MAX];  int count;  }stack;  stack tumpukan;  void main(){  int i, j;  int Beban[N][N], Jalur[N][N], Rute[N][N];  puts("Graph - Warshall Dinamik\n");  puts("Matriks Q-P-R asal");  for(i=0; i<N; i++){  for(j=0; j<N; j++){  printf("Masukkan Matriks Beban [%d][%d] : ", i+1,j+1);  scanf("%d", &Beban[i][j]);  }  }  puts("");  for(i=0; i<N; i++){  for(j=0; j<N; j++){  printf("Masukkan Matriks Jalur [%d][%d] : ", i+1,j+1);  scanf("%d", &Jalur[i][j]);  }  }  puts("");  for(i=0; i<N; i++){  for(j=0; j<N; j++){  printf("Masukkan Matriks Rute [%d][%d] : ", i+1,j+1);  scanf("%d", &Rute[i][j]);  }  }  puts("");  tampil(Beban, "Matriks Beban (Q)");  tampil(Jalur, "Matriks Jalur (P)");  tampil(Rute, "Matriks Rute (R)");  warshall(Beban, Jalur, Rute);  printf("Matriks Q-P-R baru\n");  tampil(Beban, "Matriks Beban (Q)");  tampil(Jalur, "Matriks Jalur (P)");  tampil(Rute, "Matriks Rute (R)");  baca(Rute, Beban);  }  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");  }  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] == 1)  R[i][j] = k+1;  else  R[i][j] = R[k][j];  }  }  }  void baca(int R[N] [N], int Q[N] [N]){  int asal, tujuan, i, j, x, y;  inisialisasi(&tumpukan);  printf("Masukkan asal : ");  scanf("%d", &asal);  printf("Masukkan tujuan : ");  scanf("%d", &tujuan);  fflush(stdin);  x=asal-1;  y=tujuan-1;  push(tujuan,&tumpukan);  for(int i=x;i<=x;i++){  for(int j=y;j>=x;j--){  if(R[i][j]!=1){  push(R[i][j],&tumpukan);  }else{  break;  }  }  }  push(asal,&tumpukan);  printf("Rute Terpendek : ");  while(!kosong(&tumpukan)){  printf("%d ",pop(&tumpukan));  }  puts(" ");  printf("Beban : %d\n",Q[asal-1][tujuan-1]);  }  void inisialisasi (stack \*s){  s->count = 0;  }  int kosong (stack \*s){  if(s->count == 0)  return (1);  else  return (0);  }  int penuh(stack \*s){  if(s->count == MAX)  return (1);  else  return (0);  }  void push(itemType x, stack \*s){  if(penuh(s))  printf("Stack penuh, tidak bisa menyimpan data\n");  else{  s->data[s->count] = x;  s->count++;  }  }  itemType pop(stack \*s){  itemType x;  if(kosong(s)){  printf("Stack kosong, tidak bisa mengambil data\n");  }else{  --s->count;  x = s->data[s->count];  return x;  }  } |

1. Output

Ket : Saya mengganti inputan M dengan angka 10

