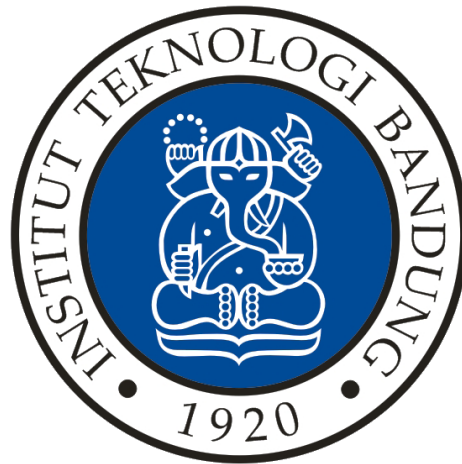


**LAPORAN TUGAS BONUS
IF2211 - STRATEGI ALGORITMA**

**Topik/Judul : Penyelesaian Shortest Path Menggunakan DP Prinsip
Optimalitas**



Nama	: Gilang Ardyamandala Al Assyifa
NIM	: 13515096
Kelas	: K03
Nama Dosen	: Rinaldi Munir
Dalam Bahasa	: Python
Deadline	: Selasa, 11 April 2017

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2017**

A. Program Sumber

File “spoptimalitas.py”

```
# Nama/NIM      : Gilang Ardyamandala Al Assyifa/13515096
# Nama file     : spoptimalitas.py
# Topik        : Penyelesaian Shortest Path Menggunakan DP Prinsip Optimalitas

from copy import deepcopy

# Definisi inf
inf = float('inf')

# Fungsi untuk mengkonversi string menjadi integer
def toInt(x):
    return int(x)

# Fungsi untuk menambah [path] dengan simpul baru
def tambah(lop, s):
    newlop = deepcopy(lop)
    for i in range(len(lop)):
        newlop[i].append(s)
    return newlop

## MAIN PROGRAM ##
# Baca dari file eksternal
nama = input('Input nama file : ')
filex = open(nama).read()    # reading from external file
filex = [item.split() for item in filex.split('\n')[:-1]]
dim = len(filex)             # Dimensi
graf = []                    # Inisiasi graf
for i in range(dim):
    tamp = []
    for j in range(dim):
        if (filex[i][j] != 'inf'):
            tamp.append((j, int(filex[i][j])))
    graf.append(tamp)

# Subjalur yang tersedia beserta costnya
jalur = []
jalur.append([0, [[0]]])
for i in range(dim-1):
    jalur.append([inf, [[0]]])

# Simpul yang sudah dikunjungi
kunjungi = [True]
for i in range(dim-1):
```

```

    kunjungi.append(False)

# SOLVING #
antrian = [0]      # Antrian simpul yang akan dikunjungi
while (len(antrian) != 0):
    # Kunjungi tetangganya seperti BFS
    subantrian = deepcopy(antrian)
    antrian = []
    while (len(subantrian) != 0):
        simpul = subantrian.pop(0)      # Simpul yang akan memulai perjalanan
        for sisi in graf[simpul]:      # Sisi yang bisa dilalui dari simpul
            if (kunjungi[sisi[0]] == False):
                if (jalur[simpul][0] + sisi[1] < jalur[sisi[0]][0]):      #
                    Jalur yang baru lebih singkat
                    jalur[sisi[0]][0] = jalur[simpul][0] + sisi[1]
                    jalur[sisi[0]][1] = tambah(jalur[simpul][1],
sisi[0])
                    elif (jalur[simpul][0] + sisi[1] == jalur[sisi[0]][0]):
# Jalur yang baru sama singkat
                    jalur[sisi[0]][1] += tambah(jalur[simpul][1],
sisi[0])
                    if (sisi[0] not in antrian) : antrian.append(sisi[0])      #
Masukkan ke antrian selanjutnya
            # Yang tadi sudah dikunjungi, ubah nilai kunjunginya
            for x in antrian:
                kunjungi[x] = True

# Mencetak hasilnya (Cost dan rutenya)
print('Cost : ', jalur[9][0])
print('Path')
for rute in jalur[9][1]:
    print([x+1 for x in rute])

```

B. Hasil *Running* Program (Input dan Output)

<p>1</p>	<p>Dari data uji slide</p> <p>inf 2 4 3 inf inf inf inf inf inf inf inf inf inf 7 4 6 inf inf inf inf inf inf inf 3 2 4 inf inf inf inf inf inf inf 4 1 5 inf inf inf inf inf inf inf inf inf inf 1 4 inf inf inf inf inf inf inf inf 6 3 inf inf inf inf inf inf inf inf 3 3 inf inf inf inf inf inf inf inf inf 3 inf inf inf inf inf inf inf inf 4 inf inf inf inf inf inf inf inf inf</p>	<pre>Input nama file : tc1.txt Cost : 11 Path [1, 3, 5, 8, 10] [1, 4, 5, 8, 10] [1, 4, 6, 9, 10]</pre>
<p>2</p>	<p>Data uji 1 yang dimodifikasi: 2-3 diberi sisi berbobot 1 4-2 diberi sisi berbobot 1</p> <p>inf 2 4 3 inf inf inf inf inf inf inf inf 1 inf 7 4 6 inf inf inf inf inf inf inf 3 2 4 inf inf inf inf 1 inf inf 4 1 5 inf inf inf inf inf inf inf inf inf inf 1 4 inf inf inf inf inf inf inf inf 6 3 inf inf inf inf inf inf inf inf 3 3 inf inf inf inf inf inf inf inf inf 3 inf inf inf inf inf inf inf inf 4 inf inf inf inf inf inf inf inf inf</p>	<pre>Input nama file : tc2.txt Cost : 11 Path [1, 3, 5, 8, 10] [1, 4, 5, 8, 10] [1, 4, 6, 9, 10]</pre>
<p>3</p>	<p>Data uji 1 dimodifikasi 7-10 diberi sisi berbobot 2</p> <p>inf 2 4 3 inf inf inf inf inf inf inf inf inf inf 7 4 6 inf inf inf inf inf inf inf 3 2 4 inf inf inf inf inf inf inf 4 1 5 inf inf inf inf inf inf inf inf inf inf 1 4 inf inf inf inf inf inf inf inf 6 3 inf inf inf inf inf inf inf inf 3 3 2 inf inf inf inf inf inf inf inf 3 inf inf inf inf inf inf inf inf 4 inf inf inf inf inf inf inf inf inf</p>	<pre>Input nama file : tc3.txt Cost : 10 Path [1, 2, 7, 10] [1, 3, 7, 10] [1, 4, 7, 10]</pre>