

# **LAPORAN TUGAS KECIL**

Penyelesaian Permasalahan Pengambilan Mata Kuliah dengan  
*Algoritma Topological Sort*

**Dibuat dalam rangka:  
Tugas Kecil 2 IF-2211 Strategi Algoritma**

**Oleh:  
Fadel Ananda Dotty - 13519146**

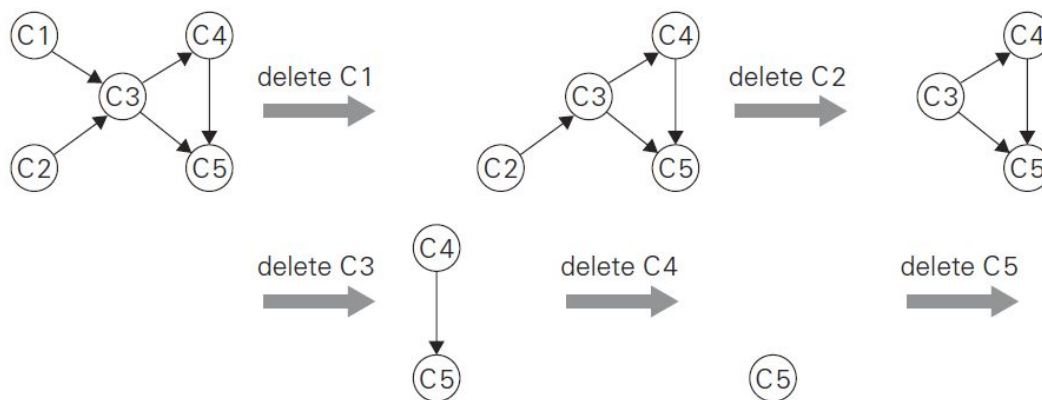


**PROGRAM STUDI TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
BANDUNG  
2020**

## I. Algoritma Topological Sort

Permasalahan yang akan diselesaikan menggunakan tipe algoritma *decrease and conquer* adalah persoalan pengambilan mata kuliah. Mata kuliah yang diambil dapat memiliki prasyarat dari mata kuliah lain maupun tidak. Solusi yang paling cocok diterapkan untuk permasalahan ini adalah dengan menggunakan pendekatan *decrease and conquer* yaitu algoritma *topological sorting*.

Algoritma *topological sorting* merupakan algoritma yang hanya dapat diterapkan pada *Directed Acyclic Graph* (DAG). DAG merupakan graf yang memiliki simpul berarah dan tidak memiliki siklus di dalamnya. Algoritma ini memiliki pendekatan cara kerja yaitu menghitung semua derajat masuk setiap simpul, lalu memilih simpul yang memiliki derajat 0 kemudian dihilangkan beserta semua busur yang keluar dari simpul tersebut, kemudian mengurangi derajat simpul yang berhubungan dengan simpul tersebut. Langkah ini kemudian dilakukan berulang kali sampai tidak ada graf yang tersisa. Algoritma ini juga dapat diterapkan pada semua pohon karena pada pohon, tidak ada siklus di dalamnya.



The solution obtained is C1, C2, C3, C4, C5

**FIGURE 4.8** Illustration of the source-removal algorithm for the **topological** sorting problem. On each iteration, a vertex with no incoming edges is deleted from the digraph.

Gambar 1.1 Ilustrasi Algoritma *Topological Sorting*

Algoritma *topological sort* memiliki kaitan dengan pendekatan *decrease and conquer* karena algoritma ini mengecek apakah terdapat simpul yang memiliki derajat masuk berjumlah nol kemudian menghapus simpul tersebut beserta semua busur yang keluar dari simpul tersebut, kemudian mengurangi derajat simpul yang berhubungan dengan simpul tersebut. Simpul berderajat nol tersebut kemudian akan dimasukkan ke himpunan solusi. Algoritma ini kemudian

akan mencari lagi apakah ada simpul berderajat nol sampai tidak ada lagi simpul yang belum dicek. Langkah ini dilakukan terus menerus. Algoritma ini menggunakan pendekatan *decrease and conquer* yang membagi permasalahan besar menjadi permasalahan kecil dengan cara mengurangi simpul yang memiliki derajat nol, kemudian mencari solusi dari permasalahan kecil tersebut yang kemudian akan digabungkan menjadi solusi permasalahan.

## II. Source Code Program dalam Python

```
# Nama : Fadel Ananda Dotty
# NIM : 13519146
# Kelas : K-03

# Asumsi test case sudah DAG
filename = input("Masukkan nama file: ")
try:
    listMatkul = open("../test/"+filename, "r").readlines() #membuka
file dan memasukkannya ke array listMatkul

    vertex = [] #berisi semua simpul unik yang ada pada soal
    connectedTo = [] #berisi simpul yang bertetangga dengan simpul di
list vertex
    numberOfConnectedVertex = [] #berisi derajat masuk simpul2 yang ada

    for i in range(len(listMatkul)): #mencari semua vertex dari text
file dan dimasukkan ke dalam vertex
        stored = []
        listMatkul[i] = listMatkul[i].strip().replace('.', ' ').replace('
', ' ').rsplit(",")
        for j in range(len(listMatkul[i])):
            if(j!=0):
                stored.append(listMatkul[i][j])
            else:
                vertex.append(listMatkul[i][j])
        numberOfConnectedVertex.append(len(stored))
        connectedTo.append(stored)

print("-----
-----\n")
```

```

    print("Simpul-simpul yang ada pada graf: ", vertex)
    for i in range(len(vertex)):
        print("Simpul lain yang masuk ke dalam simpul ", vertex[i], "
:", connectedTo[i])
        print("")

print("-----
-----\n")

    print("Jumlah derajat masuk masing2 simpul:",
numberOfConnectedVertex)
    print()

print("-----
-----")

    print("Matkul yang dapat diambil :")
    semester=0 #semester

    #Topological Sort
    while(len(vertex) !=0):
        solution = [] #berisi solusi dari permasalahan
        for i in range(len(numberOfConnectedVertex)): #cari yang derajat
0 masukkan ke list solusi sementara
            if(numberOfConnectedVertex[i] == 0):
                solution.append(vertex[i])

        for i in range(len(connectedTo)):
            try:
                connectedTo.remove([]) #remove list connectedTo yang
isinya kosong
            except ValueError: #error handling jika tidak ada elemen
1

        for element in connectedTo: #menghapus dari list connectedTo
yang sama dengan list solusi sementara
            for i in range(len(solution)):
                try:
                    element.remove(solution[i])
                except ValueError: #error handling jika tidak ada elemen

```

```

1

for i in range(len(solution)): #menghapus dari list connectedTo
    try:
        vertex.remove(solution[i])
    except ValueError: #error handling jika tidak ada elemen
        1

for i in range(len(numberOfConnectedVertex)): #mengurangi indeks
dengan elemen sama dengan 0
    try:
        numberOfConnectedVertex.remove(0)
    except ValueError: #error handling if gaada elemen
        1

for i in range(len(numberOfConnectedVertex)): #assign list
numberOfConnectedVertex dengan panjang list connectedTo yang baru
    numberOfConnectedVertex[i] = len(connectedTo[i])

print("Semester", semester+1, "mahasiswa dapat mengambil matkul:
", solution) #solusi
semester += 1

except:
    print("File tidak ditemukan")

```

Source code yang dibuat memiliki alur yaitu pertama kali membaca file eksternal, kemudian membuat tiga buah list yaitu list yang berisi simpul-simpul utama yang ada pada permasalahan, list yang berisi list simpul-simpul yang masuk kepada simpul utama, kemudian list yang berisi jumlah derajat masuk dari simpul-simpul utama. Kemudian program akan mengeksekusi algoritma *topological sort* selama jumlah elemen pada list simpul utama tidak sama dengan nol. Algoritma topological sort yang dipakai menggunakan alur yaitu membuat list yang berisi solusi sementara dari permasalahan per loop while. List solusi tersebut akan berisi simpul yang memiliki derajat masuk berjumlah nol. Kemudian dari list solusi tersebut, akan dilakukan penghapusan pada list simpul-simpul utama dan elemen pada list di dalam list simpul-simpul yang bersesuaian dengan simpul utama tersebut termasuk apabila terdapat list yang berisi himpunan kosong. Kemudian jumlah derajat masuk akan disesuaikan dengan list simpul-simpul bersesuaian yang baru termasuk menghapus apabila elemen list tersebut ada yang nol. Kemudian list solusi akan ditampilkan ke

layar dan program akan mengeksekusi loop while sampai jumlah list simpul utama sama dengan nol.

### III. Hasil Test Case Program

#### Test Case 1

```
test_case > ≡ graph1.txt

1      C1, C3.
2      C2, C1, C4.
3      C3.
4      C4, C1, C3.
5      C5, C2, C4.
```

#### Hasil Test Case 1

```
fadelananda@DESKTOP-M0MIBGF:/mnt/h/ITB Semester 4/Tucil 2/strategimatkul$ python3 strategimatkul.py
Masukkan nama file: graph1.txt
-----
Simpul-simpul yang ada pada graf: ['C1', 'C2', 'C3', 'C4', 'C5']
Simpul lain yang masuk ke dalam simpul C1 : ['C3']
Simpul lain yang masuk ke dalam simpul C2 : ['C1', 'C4']
Simpul lain yang masuk ke dalam simpul C3 : []
Simpul lain yang masuk ke dalam simpul C4 : ['C1', 'C3']
Simpul lain yang masuk ke dalam simpul C5 : ['C2', 'C4']
-----
Jumlah derajat masuk masing2 simpul: [1, 2, 0, 2, 2]
-----
Matkul yang dapat diambil :
Semester 1 mahasiswa dapat mengambil matkul: ['C3']
Semester 2 mahasiswa dapat mengambil matkul: ['C1']
Semester 3 mahasiswa dapat mengambil matkul: ['C4']
Semester 4 mahasiswa dapat mengambil matkul: ['C2']
Semester 5 mahasiswa dapat mengambil matkul: ['C5']
```

## Test Case 2

```
test_case > ≡ graph2.txt
```

```
1    MA1201, MA1101.
2    FI1202, FI1101.
3    IF1210, KU1102.
4    KU1202, KU1102.
5    KI1002, KU1011.
6    EL1200, FI1101.
7    KU1102.
8    MA1101.
9    FI1101.
10   KU1011.
```

## Hasil Test Case 2

```
Fadelananda@DESKTOP-M0MIBGF:/mnt/h/ITB Semester 4/Tucil 2/strategimatkul$ python3 strategimatkul.py
Masukkan nama file: graph2.txt
```

```
-----
Simpul-simpul yang ada pada graf: ['MA1201', 'FI1202', 'IF1210', 'KU1202', 'KI1002', 'EL1200', 'KU1102', 'MA1101', 'FI1101', 'KU1011']
Simpul lain yang masuk ke dalam simpul MA1201 : ['MA1101']
Simpul lain yang masuk ke dalam simpul FI1202 : ['FI1101']
Simpul lain yang masuk ke dalam simpul IF1210 : ['KU1102']
Simpul lain yang masuk ke dalam simpul KU1202 : ['KU1102']
Simpul lain yang masuk ke dalam simpul KI1002 : ['KU1011']
Simpul lain yang masuk ke dalam simpul EL1200 : ['FI1101']
Simpul lain yang masuk ke dalam simpul KU1102 : []
Simpul lain yang masuk ke dalam simpul MA1101 : []
Simpul lain yang masuk ke dalam simpul FI1101 : []
Simpul lain yang masuk ke dalam simpul KU1011 : []
-----
```

```
Djumlah derajat masuk masing2 simpul: [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
```

```
-----
Matkul yang dapat diambil :
```

```
Semester 1 mahasiswa dapat mengambil matkul: ['KU1102', 'MA1101', 'FI1101', 'KU1011']
Semester 2 mahasiswa dapat mengambil matkul: ['MA1201', 'FI1202', 'IF1210', 'KU1202', 'KI1002', 'EL1200']
```



### Test Case 3

```
test_case > ≡ graph3.txt
```

```
1    IF1102, IF1011.
2    IF1103.
3    IF1105.
4    IF1107.
5    IF1108, IF1103, IF1107.
6    IF1109, IF1108, IF1011.
7    IF1110, IF1103, IF1011.
8    IF1011, IF1105, IF1107.
```

### Hasil Test Case 3

```
fadelananda@DESKTOP-M0MIBGF:/mnt/h/ITB Semester 4/Tucil 2/strategimatkul$ python3 strategimatkul.py
Masukkan nama file: graph3.txt
```

```
-----
Simpul-simpul yang ada pada graf: ['IF1102', 'IF1103', 'IF1105', 'IF1107', 'IF1108', 'IF1109', 'IF1110', 'IF1011']
Simpul lain yang masuk ke dalam simpul IF1102 : ['IF1011']
Simpul lain yang masuk ke dalam simpul IF1103 : []
Simpul lain yang masuk ke dalam simpul IF1105 : []
Simpul lain yang masuk ke dalam simpul IF1107 : []
Simpul lain yang masuk ke dalam simpul IF1108 : ['IF1103', 'IF1107']
Simpul lain yang masuk ke dalam simpul IF1109 : ['IF1108', 'IF1011']
Simpul lain yang masuk ke dalam simpul IF1110 : ['IF1103', 'IF1011']
Simpul lain yang masuk ke dalam simpul IF1011 : ['IF1105', 'IF1107']
-----
```

```
Jumlah derajat masuk masing2 simpul: [1, 0, 0, 0, 2, 2, 2, 2]
```

```
-----
Matkul yang dapat diambil :
```

```
Semester 1 mahasiswa dapat mengambil matkul: ['IF1103', 'IF1105', 'IF1107']
Semester 2 mahasiswa dapat mengambil matkul: ['IF1108', 'IF1011']
Semester 3 mahasiswa dapat mengambil matkul: ['IF1102', 'IF1109', 'IF1110']
```

```
Matkul yang dapat diambil :
```

```
Semester 1 mahasiswa dapat mengambil matkul: ['IF1103', 'IF1105', 'IF1107']
Semester 2 mahasiswa dapat mengambil matkul: ['IF1108', 'IF1011']
Semester 3 mahasiswa dapat mengambil matkul: ['IF1102', 'IF1109', 'IF1110']
```



## Test Case 4

```
test_case > ≡ graph4.txt

1   FisDas1.
2   Kalkulus1.
3   PengKom.
4   Orkom, DasPro, Kalkulus2.
5   OS, Orkom.
6   DasPro, PengKom.
7   AlstrukDat, DasPro.
8   PBO, AlstrukDat.
9   Kalkulus2, Kalkulus1.
10  FisDas2, FisDas1.
11  MatDis, Kalkulus2.
12  AlGeo, Kalkulus2.
13  ProbStat, MatDis, AlGeo.
```

## Hasil Test Case 4

```
Fadelananda@DESKTOP-MQW1BGF:/mnt/hy118 Semester 4/Tucl1 2/strategimatkul$ python3 strategimatkul.py
Masukkan nama file: graph4.txt
-----
Simpul-simpul yang ada pada graf: ['FisDas1', 'Kalkulus1', 'PengKom', 'Orkom', 'OS', 'DasPro', 'AlstrukDat', 'PBO', 'Kalkulus2', 'FisDas2', 'MatDis', 'AlGeo', 'ProbStat']
Simpul lain yang masuk ke dalam simpul FisDas1 : []
Simpul lain yang masuk ke dalam simpul Kalkulus1 : []
Simpul lain yang masuk ke dalam simpul PengKom : []
Simpul lain yang masuk ke dalam simpul Orkom : ['DasPro', 'Kalkulus2']
Simpul lain yang masuk ke dalam simpul OS : ['Orkom']
Simpul lain yang masuk ke dalam simpul DasPro : ['PengKom']
Simpul lain yang masuk ke dalam simpul AlstrukDat : ['DasPro']
Simpul lain yang masuk ke dalam simpul PBO : ['AlstrukDat']
Simpul lain yang masuk ke dalam simpul Kalkulus2 : ['Kalkulus1']
Simpul lain yang masuk ke dalam simpul FisDas2 : ['FisDas1']
Simpul lain yang masuk ke dalam simpul MatDis : ['Kalkulus2']
Simpul lain yang masuk ke dalam simpul AlGeo : ['Kalkulus2']
Simpul lain yang masuk ke dalam simpul ProbStat : ['MatDis', 'AlGeo']
-----
Jumlah derajat masuk masing2 simpul: [0, 0, 0, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2]
-----
Matkul yang dapat diambil :
Semester 1 mahasiswa dapat mengambil matkul: ['FisDas1', 'Kalkulus1', 'PengKom']
Semester 2 mahasiswa dapat mengambil matkul: ['DasPro', 'Kalkulus2', 'FisDas2']
Semester 3 mahasiswa dapat mengambil matkul: ['Orkom', 'AlstrukDat', 'MatDis', 'AlGeo']
Semester 4 mahasiswa dapat mengambil matkul: ['OS', 'PBO', 'ProbStat']
```

Matkul yang dapat diambil :

Semester 1 mahasiswa dapat mengambil matkul: ['FisDas1', 'Kalkulus1', 'PengKom']

Semester 2 mahasiswa dapat mengambil matkul: ['DasPro', 'Kalkulus2', 'FisDas2']

Semester 3 mahasiswa dapat mengambil matkul: ['Orkom', 'AlstrukDat', 'MatDis', 'AlGeo']

Semester 4 mahasiswa dapat mengambil matkul: ['OS', 'PB0', 'ProbStat']

## Test Case 5

```
test_case > ≡ graph5.txt
```

```
1    MA1201.
```

```
2    FI1201.
```

```
3    IF2110.
```

```
4    IF2121, IF2120, IF2110.
```

```
5    IF2211, IF2120, MA1201.
```

```
6    IF2120, MA1201.
```

```
7    IF4020, IF2120, IF2110.
```

## Hasil Test Case 5

```
fadelananda@DESKTOP-M0MIBGF:/mnt/h/IITB Semester 4/Tucil 2/strategimatkul$ python3 strategimatkul.py
```

Masukkan nama file: graph5.txt

```
-----
Simpul-simpul yang ada pada graf: ['MA1201', 'FI1201', 'IF2110', 'IF2121', 'IF2211', 'IF2120', 'IF4020']
Simpul lain yang masuk ke dalam simpul MA1201 : []
Simpul lain yang masuk ke dalam simpul FI1201 : []
Simpul lain yang masuk ke dalam simpul IF2110 : []
Simpul lain yang masuk ke dalam simpul IF2121 : ['IF2120', 'IF2110']
Simpul lain yang masuk ke dalam simpul IF2211 : ['IF2120', 'MA1201']
Simpul lain yang masuk ke dalam simpul IF2120 : ['MA1201']
Simpul lain yang masuk ke dalam simpul IF4020 : ['IF2120', 'IF2110']
-----
```

```
Dumlah derajat masuk masing2 simpul: [0, 0, 0, 2, 2, 1, 2]
```

Matkul yang dapat diambil :

Semester 1 mahasiswa dapat mengambil matkul: ['MA1201', 'FI1201', 'IF2110']

Semester 2 mahasiswa dapat mengambil matkul: ['IF2120']

Semester 3 mahasiswa dapat mengambil matkul: ['IF2121', 'IF2211', 'IF4020']

## Test Case 6

```
test_case > ≡ graph6.txt
```

```
1    IF11.
2    IF12.
3    IF13.
4    IF14.
5    IF21, IF11, IF12.
6    IF22, IF21, IF11.
7    IF23, IF13, IF14.
8    IF33, IF22, IF23.
9    IF44, IF11.
10   IF55, IF44.
```

## Hasil Test Case 6

```
fadelananda@DESKTOP-M0M1BGF:/mnt/h/ITB Semester 4/Tucil 2/strategimatkul$ python3 strategimatkul.py
```

```
Masukkan nama file: graph6.txt
```

```
-----
Simpul-simpul yang ada pada graf: ['IF11', 'IF12', 'IF13', 'IF14', 'IF21', 'IF22', 'IF23', 'IF33', 'IF44', 'IF55']
Simpul lain yang masuk ke dalam simpul IF11 : []
Simpul lain yang masuk ke dalam simpul IF12 : []
Simpul lain yang masuk ke dalam simpul IF13 : []
Simpul lain yang masuk ke dalam simpul IF14 : []
Simpul lain yang masuk ke dalam simpul IF21 : ['IF11', 'IF12']
Simpul lain yang masuk ke dalam simpul IF22 : ['IF21', 'IF11']
Simpul lain yang masuk ke dalam simpul IF23 : ['IF13', 'IF14']
Simpul lain yang masuk ke dalam simpul IF33 : ['IF22', 'IF23']
Simpul lain yang masuk ke dalam simpul IF44 : ['IF11']
Simpul lain yang masuk ke dalam simpul IF55 : ['IF44']
-----
```

```
Dumlah derajat masuk masing2 simpul: [0, 0, 0, 0, 2, 2, 2, 2, 1, 1]
```

```
-----
Matkul yang dapat diambil :
```

```
Semester 1 mahasiswa dapat mengambil matkul: ['IF11', 'IF12', 'IF13', 'IF14']
Semester 2 mahasiswa dapat mengambil matkul: ['IF21', 'IF23', 'IF44']
Semester 3 mahasiswa dapat mengambil matkul: ['IF22', 'IF55']
Semester 4 mahasiswa dapat mengambil matkul: ['IF33']
```

## Test Case 7

```
test_case > ≡ graph7.txt
1   Biologi1.
2   Kimia1.
3   FisikaDasar1.
4   MatematikaDasar1.
5   Kimia2, Kimia1.
6   Biologi2, Biologi1, Kimia2.
7   Fisika2, FisikaDasar1.
8   MatematikaDasar2, MatematikaDasar1.
9   PRD, MatematikaDasar1, FisikaDasar1.
10  RekayasaPerangkatLunak, Fisika2, MatematikaDasar2, PRD.
11  RekayasaGenetika, Biologi2, PRD.
```

## Hasil Test Case 7

```
Fadelananda@DESKTOP-MQW1BGF:/mnt/h/ITS Semester 4/Tucl1 2/strategimatkul$ python3 strategimatkul.py
Masukkan nama file: graph7.txt
-----
Simpul-simpul yang ada pada graf: ['Biologi1', 'Kimia1', 'FisikaDasar1', 'MatematikaDasar1', 'Kimia2', 'Biologi2', 'Fisika2', 'MatematikaDasar2', 'PRD', 'RekayasaPerangkatLunak', 'RekayasaGenetika']
Simpul lain yang masuk ke dalam simpul Biologi1 : []
Simpul lain yang masuk ke dalam simpul Kimia1 : []
Simpul lain yang masuk ke dalam simpul FisikaDasar1 : []
Simpul lain yang masuk ke dalam simpul MatematikaDasar1 : []
Simpul lain yang masuk ke dalam simpul Kimia2 : ['Kimia1']
Simpul lain yang masuk ke dalam simpul Biologi2 : ['Biologi1', 'Kimia2']
Simpul lain yang masuk ke dalam simpul Fisika2 : ['FisikaDasar1']
Simpul lain yang masuk ke dalam simpul MatematikaDasar2 : ['MatematikaDasar1']
Simpul lain yang masuk ke dalam simpul PRD : ['MatematikaDasar1', 'FisikaDasar1']
Simpul lain yang masuk ke dalam simpul RekayasaPerangkatLunak : ['Fisika2', 'MatematikaDasar2', 'PRD']
Simpul lain yang masuk ke dalam simpul RekayasaGenetika : ['Biologi2', 'PRD']
-----
Dumlah derajat masuk masing2 simpul: [0, 0, 0, 0, 1, 2, 1, 1, 2, 3, 2]
-----
Matkul yang dapat diambil :
Semester 1 mahasiswa dapat mengambil matkul: ['Biologi1', 'Kimia1', 'FisikaDasar1', 'MatematikaDasar1']
Semester 2 mahasiswa dapat mengambil matkul: ['Kimia2', 'Fisika2', 'MatematikaDasar2', 'PRD']
Semester 3 mahasiswa dapat mengambil matkul: ['Biologi2', 'RekayasaPerangkatLunak']
Semester 4 mahasiswa dapat mengambil matkul: ['RekayasaGenetika']
```

```
Matkul yang dapat diambil :
Semester 1 mahasiswa dapat mengambil matkul: ['Biologi1', 'Kimia1', 'FisikaDasar1', 'MatematikaDasar1']
Semester 2 mahasiswa dapat mengambil matkul: ['Kimia2', 'Fisika2', 'MatematikaDasar2', 'PRD']
Semester 3 mahasiswa dapat mengambil matkul: ['Biologi2', 'RekayasaPerangkatLunak']
Semester 4 mahasiswa dapat mengambil matkul: ['RekayasaGenetika']
```



## Test Case 8

```
test_case > ≡ graph8.txt

1    4421, 3421, 2221, 1221.
2    2221, 1111, 1221.
3    1111.
4    3421, 2781.
5    2781.
6    1221, 1001, 1002, 1003.
7    1001.
8    1002.
9    1003.
```

## Hasil Test Case 8

```
fadelananda@DESKTOP-M0MIBGF:/mnt/h/ITB Semester 4/tucil 2/strategimatkul$ python3 strategimatkul.py
Masukkan nama file: graph8.txt
-----
Simpul-simpul yang ada pada graf: ['4421', '2221', '1111', '3421', '2781', '1221', '1001', '1002', '1003']
Simpul lain yang masuk ke dalam simpul 4421 : ['3421', '2221', '1221']
Simpul lain yang masuk ke dalam simpul 2221 : ['1111', '1221']
Simpul lain yang masuk ke dalam simpul 1111 : []
Simpul lain yang masuk ke dalam simpul 3421 : ['2781']
Simpul lain yang masuk ke dalam simpul 2781 : []
Simpul lain yang masuk ke dalam simpul 1221 : ['1001', '1002', '1003']
Simpul lain yang masuk ke dalam simpul 1001 : []
Simpul lain yang masuk ke dalam simpul 1002 : []
Simpul lain yang masuk ke dalam simpul 1003 : []
-----
Jumlah derajat masuk masing2 simpul: [3, 2, 0, 1, 0, 3, 0, 0, 0]
-----
Matkul yang dapat diambil :
Semester 1 mahasiswa dapat mengambil matkul: ['1111', '2781', '1001', '1002', '1003']
Semester 2 mahasiswa dapat mengambil matkul: ['3421', '1221']
Semester 3 mahasiswa dapat mengambil matkul: ['2221']
Semester 4 mahasiswa dapat mengambil matkul: ['4421']
```

#### IV. Link

<https://github.com/fadelananda/tucil2>

#### V. Checklist

Poin	Ya	Tidak
1. Program berhasil dikompilasi	V	
2. Program berhasil <i>running</i>	V	
3. Program dapat menerima berkas input dan menuliskan output	V	
4. Luaran sudah benar untuk semua kasus input	V	