

# Dasar-Dasar Pemrograman 2

## Tugas Pemrograman 03 Kehidupan Fasilkom



FAKULTAS  
ILMU  
KOMPUTER

Revision Number	Revision Date	Nature of Revision
1	05/05/21	<b>void Menyapa (class ElemenFasilkom):</b> method ini akan menyimpan orang yang disapa ke dalam array <b>telahMenyapa</b>  <b>perintah NEXT_DAY:</b> Jika suatu objek pada ElemenFasilkom telah menyapa <b>lebih dari atau sama dengan</b> setengah dari total elemenFasilkom (tidak termasuk diri sendiri)
2	05/05/21	<b>perintah ADD_MAHASISWA:</b> Dijamin nama mahasiswa unik dan belum pernah terdaftar pada <b>daftarElemenFasilkom</b> . <b>perintah ADD_DOSEN:</b> Dijamin nama dosen unik dan belum pernah terdaftar pada <b>daftarElemenFasilkom</b> <b>perintah ADD_ELEMEN_KANTIN:</b> Dijamin nama elemen kantin unik dan belum pernah terdaftar pada <b>daftarElemenFasilkom</b>
3	07/05/21	<b>class Dosen:</b> - atribut <b>MataKuliah mataKuliah</b>
4	11/05/21	<b>Perhitungan dari keramahan :</b> 3. Jika terdapat objek Mahasiswa menyapa objek Dosen dimana mereka terhubung dalam 1 mata kuliah yang sama, maka tingkat friendship dari dosen dan juga mahasiswa akan bertambah sebesar 2. <b>Hal tersebut juga berlaku ketika objek Dosen menyapa objek Mahasiswa</b>

5	15/05/21	<b>Perhitungan dari keramahan :</b> <b>5. Untuk aturan nomor 3 dan 4, perubahan nilai friendship dilakukan ketika perintah tersebut dijalankan, bukan pada saat perintah NEXT_DAY. Berbeda dengan aturan nomor 2 dimana perubahan nilai friendship hanya terjadi ketika pemanggilan NEXT_DAY.</b>
---	----------	--

Pada tugas pemrograman kali ini, Anda diminta untuk membuat sebuah program mengenai kehidupan di fasilkom, dimana terdapat elemen-elemen yang ada di fasilkom cseperti dosen, mahasiswa, dan elemen kantin. Selain itu, tiap elemen yang ada juga dapat menyapa satu sama lain. Juga terdapat Sistem akademik seperti pada **Tugas Pemrograman 2** dimana mahasiswa dapat melakukan add drop mata kuliah. Berikut adalah spesifikasi dari program :

#### 1. Class ElemenFasilkom

- a. Class ini merepresentasikan elemen-elemen yang ada pada Fasilkom. Pada fakultas ini, dijamin **elemen yang di dalamnya tidak akan melebihi 100 orang**.
- b. Atribut yang ada pada Class ElemenFasilkom :
  - i. int friendship: menunjukkan tingkat keramahan seseorang. Nilai dari friendship akan berubah **setiap harinya** sesuai dengan persyaratan yang ada di perintah NEXT\_DAY yang ada di bawah
  - ii. String tipe: menunjukkan tipe dari ElemenFasilkom, seperti Mahasiswa, ElemenKantin, atau Dosen
  - iii. String nama: menyimpan nama dari elemen fasilkom
  - iv. ElemenFasilkom[] telahMenyapa: menyimpan daftar dari ElemenFasilkom yang telah disapa
- c. Method yang perlu diimplementasikan :
  - i. void menyapa(ElemenFasilkom elemenFasilkom): method ini akan menyimpan orang yang disapa ke dalam array **telahMenyapa**. Setelah menyapa elemen fasilkom tersebut, cetak kalimat di bawah ini:

**[nama\_orang\_1] menyapa dengan [nama\_orang\_2]**

Pada method ini, perlu divalidasi apakah elemen fasilkom tersebut telah disapa sebelumnya atau tidak. Jika elemen fasilkom tersebut sudah pernah disapa, maka cetak peringatan di bawah ini :

**[DITOLAK] [nama\_orang\_1] telah menyapa [nama\_orang\_2] hari ini**

- ii. void membeliMakanan(ElemenFasilkom pembeli, ElemenFasilkom penjual, String makanan): Method ini akan mengecek apakah penjual

menjual makanan tersebut atau tidak. Jika ternyata penjual tidak menjual makanan tersebut, cetak peringatan :

**[DITOLAK] [nama\_penjual] tidak menjual [nama\_makanan]**

Sebaliknya, jika penjual tersebut ada menjual makanan tersebut cetak kalimat di bawah ini :

**[nama\_pembeli] berhasil membeli [nama\_makanan] seharga [harga\_makanan]**

- iii. void resetMenyapa(): method ini akan menghapus daftar orang yang telah disapa
- iv. String toString(): method ini akan mengembalikan nama dari ElemenFasilkom
- v. setter-getter yang merasa diperlukan

## 2. Class Dosen extends ElemenFasilkom

- a. Class ini merepresentasikan objek-objek Dosen yang ada di dalam ElemenFasilkom. Pada class ini, dosen hanya dapat mengajar **maksimal 1 mata kuliah saja**.
- b. Atribut yang ada pada class Dosen
  - i. MataKuliah mataKuliah: menyimpan mata kuliah yang sedang dipegang
- c. Method yang perlu diimplementasikan
  - i. void mengajarMataKuliah(MataKuliah mataKuliah): method ini akan mendaftarkan objek Dosen ke dalam mataKuliah. Method ini hanya akan mendaftarkan dosen ke suatu mata kuliah jika sebelumnya objek Dosen tidak mengajar mata kuliah apapun.

Jika sebelumnya tidak mengajar apapun, maka ketika method ini dipanggil cetak kalimat di bawah ini :

**[nama\_dosen] mengajar mata kuliah [nama\_matkul\_yang\_didaftar]**

Jika ternyata objek Dosen telah mengajar suatu mataKuliah, maka cetak peringatan ini :

**[DITOLAK] [nama\_dosen] sudah mengajar mata kuliah [nama\_matkul\_yang\_sedang\_dipegang]**

Jika dosen mendaftar ke mata kuliah yang sudah memiliki dosen pengajar, maka cetak peringatan

**[DITOLAK] [nama\_mata\_kuliah] sudah memiliki dosen pengajar**

note: Prioritas validasi untuk kasus di atas adalah apakah dosen sudah mengajar suatu mata kuliah, apakah matkul yang akan diajar sudah memiliki dosen pengajar

- ii. void dropMataKuliah(): method ini akan melepaskan mata kuliah yang sedang dipegang. Jika sebelumnya objek Dosen sedang memegang suatu mata kuliah, maka ketika method ini dipanggil cetak kalimat di bawah ini :

**[nama\_dosen] berhenti mengajar [nama\_matkul\_yang\_di\_drop]**

Jika ternyata objek Dosen tidak mengajar matkul apapun, maka cetak peringatan ini :

**[DITOLAK] [nama\_dosen] sedang tidak mengajar mata kuliah apapun**

- iii. setter-getter yang merasa diperlukan

### 3. Class Mahasiswa extends ElemenFasilkom

- a. Class ini merepresentasikan objek-objek Mahasiswa yang ada di dalam ElemenFasilkom. Setiap mahasiswa dijamin **tidak akan memiliki mata kuliah melebihi 10**.
- b. Atribut yang ada pada class Mahasiswa
  - i. MataKuliah[] daftarMataKuliah: menyimpan daftar dari mata kuliah yang diambil
  - ii. String tanggalLahir: menyimpan tanggal lahir dari mahasiswa
  - iii. String jurusan: menyimpan jurusan dari mahasiswa
  - iv. long npm : menyimpan nilai npm. Nilai NPM mengikuti dijamin **valid** sesuai dengan **format yang ada pada Tugas Pemrograman 1**.

19 02 22012003 1 7  
A B C D E

Untuk kode jurusan, dapat dilihat pada bagian B :

- Kode 01 → Ilmu Komputer
- Kode 02 → Sistem Informasi

- c. Method yang perlu diimplementasikan
  - i. void addMatkul(MataKuliah mataKuliah): method ini akan menambahkan mata kuliah ke dalam list daftarMataKuliah yang ada di dalam class Mahasiswa. Selain itu, mahasiswa juga akan didaftarkan ke dalam list daftarMahasiswa yang terdapat dalam class MataKuliah. Berbeda dengan Tugas Pemrograman 2, **mahasiswa dari semua**

**jurusan dapat mengambil mata kuliah apa saja.** Jika mata kuliah berhasil ditambahkan, maka cetak kalimat :

**[nama\_mahasiswa] berhasil menambahkan mata kuliah [nama\_matkul]**

Jika mata kuliah yang ingin ditambahkan ternyata sudah pernah terdaftar, maka cetak peringatan :

**[DITOLAK] [nama\_matkul] telah diambil sebelumnya**

Jika kapasitas dari mata kuliah yang akan ditambahkan ternyata telah penuh, maka cetak peringatan :

**[DITOLAK] [nama\_matkul] telah penuh kapasitasnya**

note : Prioritas validasi untuk kasus di atas adalah apakah sudah pernah terdaftar, apakah kapasitas sudah penuh

- ii. void dropMatkul(MataKuliah mataKuliah): method ini akan menghapus matakuliah dari list daftarMataKuliah yang ada dari class Mahasiswa. Selain itu, mahasiswa yang ada dari list daftarMahasiswa pada class MataKuliah juga akan dihapus. Jika mata kuliah berhasil dihapus, maka cetak kalimat :

**[nama\_mahasiswa] berhasil drop mata kuliah [nama\_matkul]**

Jika mata kuliah yang ingin dihapus ternyata belum pernah diambil, maka cetak peringatan :

**[DITOLAK] [nama\_matkul] belum pernah diambil**

- iii. String extractTanggalLahir(long npm): method ini akan mengembalikan tanggal lahir dari nilai npm
- iv. String extractJurusan(long npm): method ini akan mengembalikan jurusan dari nilai npm
- v. setter-getter yang merasa diperlukan

#### 4. Class ElemenKantin extends ElemenFasikom

- a. Class ini merepresentasikan objek-objek ElemenKantin yang ada di dalam ElemenFasikom. Setiap elemenKantin dijamin **tidak akan menjual makanan lebih dari 10.**
- b. Atribut yang ada pada Class ElemenKantin
  - i. Makanan[] daftarMakanan: berisi daftar dari makanan yang dijual.
- c. Method yang perlu diimplementasi
  - i. void setMakanan(String nama, long harga): method ini akan membuat objek Makanan dengan parameter nama dan harga dan memasukkan objek tersebut ke dalam daftarMakanan. Jika makanan belum pernah didaftarkan sebelumnya, maka cetak kalimat di bawah :

[nama\_elemen\_kantin] telah mendaftarkan makanan  
[nama\_makanan] dengan harga [harga\_makanan]

Jika makanan tersebut ternyata sudah pernah didaftarkan sebelumnya, cetak peringatan :

[DITOLAK] [nama\_makanan] sudah pernah terdaftar

- ii. setter-getter yang merasa diperlukan

#### 5. Class Makanan

- a. Class ini merepresentasikan objek-objek Makanan untuk ElemenKantin
- b. Atribut yang ada pada class Makanan
  - i. String nama: berisi nama dari makanan
  - ii. long harga: berisi harga dari makanan
- c. Method yang perlu diimplementasikan
  - i. String toString(): mengembalikan nama dari makanan
  - ii. setter-getter yang merasa diperlukan

#### 6. Class MataKuliah

- a. Class ini merepresentasikan objek MataKuliah
- b. Atribut yang ada pada class MataKuliah
  - i. String nama: berisi nama dari mata kuliah
  - ii. int kapasitas: berisi kapasitas dari mata kuliah
  - iii. Dosen dosen: menyimpan dosen pengajar dari mata kuliah tersebut
  - iv. Mahasiswa[] daftarMahasiswa: menyimpan daftar mahasiswa yang mengambil mata kuliah tersebut
- c. Method yang perlu diimplementasikan
  - i. void addMahasiswa(Mahasiswa mahasiswa): menambahkan objek mahasiswa ke dalam list daftarMahasiswa
  - ii. void dropMahasiswa(Mahasiswa mahasiswa): menghapus objek mahasiswa dari list daftarMahasiswa
  - iii. void addDosen(Dosen dosen): menambahkan objek dosen ke dalam mata kuliah
  - iv. void dropDosen(): menghapus objek dosen dari mata kuliah
  - v. String toString(): mengembalikan nama dari mata kuliah
  - vi. setter-getter yang merasa diperlukan

#### 7. Main.java

- a. Class ini merupakan program utama. Disini merupakan tempat dimana user memasukkan perintah dan menjalankan perintah tersebut ke class lain.
- b. Atribut yang ada pada Main.java

- i. ElemenFasilkom[] daftarElemenFasilkom: menyimpan daftar semua objek ElemenFasilkom (termasuk Dosen, Mahasiswa, dan ElemenKantin).
- ii. MataKuliah[] daftarMataKuliah: menyimpan daftar semua matkul yang ada
- c. Perintah-perintah yang ada pada class Main.java
  - i. ADD\_MAHASISWA [nama][npm]: Perintah ini akan membuat objek Mahasiswa baru dengan parameter nama dan npm serta menambahkan objek tersebut ke dalam daftarElemenFasilkom. Dijamin **nama mahasiswa unik dan belum pernah terdaftar pada daftarElemenFasilkom.**
  - ii. ADD\_DOSEN [nama]: Method ini akan membuat objek Dosen baru dengan parameter nama serta menambahkan objek tersebut ke dalam daftarElemenFasilkom. Dijamin **nama dosen unik dan belum pernah terdaftar pada daftarElemenFasilkom.**
  - iii. ADD\_ELEMEN\_KANTIN [nama]: Perintah ini akan membuat objek ElemenKantin dengan parameter nama serta menambahkan objek tersebut ke dalam daftarElemenFasilkom. Dijamin **nama elemen kantin unik dan belum pernah terdaftar pada daftarElemenFasilkom.**
  - iv. MENYAPA [objek1][objek2]: Perintah ini akan membuat interaksi antara objek1 dan objek2 serta berjalan secara 2 arah, artinya objek1 menyapa objek2 dan objek2 menyapa objek1. Dijamin **objek1 dan objek2 merupakan objek yang berada di dalam daftarElemenFasilkom.** Jika objek1 dan objek2 merupakan objek yang sama, maka cetak peringatan :

**[DITOLAK] Objek yang sama tidak bisa saling menyapa**

- v. ADD\_MAKANAN [objek][namaMakanan][harga]: Perintah ini menambahkan makanan dengan parameter namaMakanan dan harga ke dalam objek. Dijamin **objek berasal dari daftarElemenFasilkom.** Jika objek tersebut bukan merupakan bagian dari ElemenKantin, maka cetak peringatan :

**[DITOLAK] [nama\_objek] bukan merupakan elemen kantin**

- vi. MEMBELI\_MAKANAN [objek1][objek2][namaMakanan]: Perintah ini memiliki arti bahwa objek1 (pembeli) akan membeli makanan dari objek2 (penjual). Dijamin **objek1 dan objek2 berasal dari daftarElemenFasilkom dan namaMakanan merupakan makanan yang dijual oleh objek2.** Jika objek2 bukan berasal dari class ElemenKantin, maka cetak peringatan :

**[DITOLAK] Hanya elemen kantin yang dapat menjual makanan**

Jika objek2 berasal dari class ElemenKantin, tetapi objek1 dan objek2 merupakan objek yang sama, maka cetak peringatan :

**[DITOLAK] Elemen kantin tidak bisa membeli makanan sendiri**

- vii. CREATE\_MATKUL [nama][kapasitas]: Perintah ini akan membuat objek MataKuliah dengan parameter nama dan kapasitas dan menyimpannya ke list daftarMataKuliah
- viii. ADD\_MATKUL [objek][namaMataKuliah]: Perintah ini akan menambahkan mata kuliah ke dalam objek. Dijamin **objek berasal dari daftarElemenFasilkom dan namaMataKuliah berasal dari daftarMataKuliah**. Jika objek bukanlah mahasiswa, maka cetak peringatan :

**[DITOLAK] Hanya mahasiswa yang dapat menambahkan matkul**

- ix. DROP\_MATKUL [objek][namaMataKuliah]: Perintah ini akan menghapus mata kuliah dari objek. Dijamin **objek berasal dari daftarElemenFasilkom dan namaMataKuliah berasal dari daftarMataKuliah**. Jika objek bukanlah mahasiswa, maka cetak peringatan :

**[DITOLAK] Hanya mahasiswa yang dapat drop matkul**

- x. MENGAJAR\_MATKUL [objek][namaMataKuliah]: Perintah ini akan mendaftarkan objek ke dalam namaMataKuliah. Dijamin **objek berasal dari daftarElemenFasilkom dan namaMataKuliah berasal dari daftarMataKuliah**. Jika objek bukan berasal dari class Dosen, maka cetak peringatan :

**[DITOLAK] Hanya dosen yang dapat mengajar matkul**

- xi. BERHENTI\_MENGAJAR [objek]: Perintah ini akan menghapus mata kuliah yang sedang dipegang oleh objek Dosen. Dijamin **objek berasal dari daftarElemenFasilkom**. Jika objek tersebut bukan berasal dari class Dosen, maka cetak peringatan :

**[DITOLAK] Hanya dosen yang dapat berhenti mengajar**

- xii. RINGKASAN\_MAHASISWA [objek]: Perintah ini akan menampilkan ringkasan dari objek, yaitu **nama, tanggal lahir, jurusan, dan daftar mata kuliah** yang sedang diambil. Dijamin **objek berasal dari daftarElemenFasilkom**. Jika belum ada mata kuliah yang diambil mahasiswa, maka pada bagian daftar mata kuliah tampilkan :

**Belum ada mata kuliah yang diambil**

Jika objek bukan berasal dari class Mahasiswa, maka cetak peringatan :

**[DITOLAK] [nama objek] bukan merupakan seorang mahasiswa**



- xiii. RINGKASAN\_MATKUL [namaMataKuliah]: Perintah ini akan menampilkan ringkasan dari mata kuliah, yaitu nama mata kuliah, jumlah mahasiswa, kapasitas, dosen pengajar, dan daftar mahasiswa yang mengambil mata kuliah tersebut. Dijamin **objek berasal dari daftarMataKuliah**. Jika belum ada yang mendaftar mata kuliah tersebut, maka pada bagian daftar mahasiswa tampilkan :

**Belum ada mahasiswa yang mengambil mata kuliah ini**

- xiv. NEXT\_DAY: Perintah ini akan melakukan **reset terhadap array telahMenyapa** yang ada pada setiap objek pada ElemenFasilkom serta melakukan **kalkulasi apakah tingkat friendship pada setiap objek ElemenFasilkom akan bertambah atau berkurang**. Perhitungan dari keramahan adalah sebagai berikut :

1. Nilai dari keramahan hanya dapat bernilai antara 0 - 100, sehingga diperlukan **validasi agar nilai friendship tidak akan melewati batas tersebut**.
2. Jika suatu objek pada ElemenFasilkom telah menyapa **lebih dari atau sama dengan setengah dari** total elemenFasilkom (**tidak termasuk diri sendiri**), maka tingkat friendship akan bertambah sebesar 10. Sebaliknya, jika ElemenFasilkom **tidak menyapa lebih dari setengah dari** total elemenFasilkom, maka tingkat keramahan akan berkurang sebesar 5. Sebagai contoh, misalkan terdapat 3 ElemenFasilkom (Andi, Budi, dan Doni). Jika Andi menyapa Budi, maka Andi telah menyapa sebanyak 1 orang dan total ElemenFasilkom tanpa termasuk diri sendiri adalah 2 orang. Oleh karena itu, Andi akan mendapatkan nilai friendship sebesar 10 pada keesokan harinya.
3. Jika terdapat objek Mahasiswa menyapa objek Dosen dimana mereka terhubung dalam 1 mata kuliah yang sama, maka tingkat friendship dari dosen dan juga mahasiswa akan bertambah sebesar 2. Hal tersebut juga berlaku ketika objek Dosen menyapa objek Mahasiswa
4. Jika terdapat ElemenFasilkom yang membeli makanan dari ElemenKantin, maka tingkat friendship antara kedua objek tersebut akan bertambah sebesar 1.
5. Untuk aturan nomor 3 dan 4, perubahan nilai friendship dilakukan ketika perintah tersebut dijalankan, bukan pada saat perintah NEXT\_DAY. Berbeda dengan aturan nomor 2 dimana perubahan nilai friendship hanya terjadi ketika pemanggilan NEXT\_DAY.

Setelah itu, program akan mencetak kalimat di bawah ini

**Hari telah berakhir dan nilai friendship telah diupdate.**

Setelah mencetak kalimat di atas, program akan menampilkan peringkat dari nilai friendship dari nilai paling tinggi hingga nilai paling rendah. Jika terdapat 2 nilai friendship yang sama, maka peringkat akan diurutkan berdasarkan nama secara alphabetical. Misalkan Andi dan Budi memiliki nilai friendship yang sama, maka Andi akan memiliki peringkat yang lebih tinggi dari Budi. Dijamin saat menjalankan perintah `NEXT_DAY`, **terdapat minimal 1 objek pada daftarElemenFasilkom**.

- xv. `PROGRAM_END`: Perintah ini akan menghentikan program. Sebelum menghentikan program, perintah ini akan mencetak kalimat :

**Program telah berakhir. Berikut nilai terakhir dari friendship pada fasilkom**

diikuti peringkat dari nilai friendship sesuai ketentuan yang ada pada `NEXT_DAY`. Dijamin saat menjalankan perintah `PROGRAM_END`, **terdapat minimal 1 objek pada daftarElemenFasilkom**.

#### **Tugas :**

Pada tugas pemrograman kali ini, Anda diminta untuk mengimplementasikan requirements (class, methods, setter-getter, visibility modifier) yang diperlukan. Skeleton code untuk setiap class yang dibutuhkan sudah tersedia, silahkan lengkapi dan modifikasi class-class tersebut dengan menambahkan atribut dan method apabila dibutuhkan. Pastikan implementasi yang dibuat menyertakan teknik OOP, inheritance, polymorphism, serta abstract class/interface.

## **Contoh Masukan dan Keluaran**

```
ADD_MAHASISWA ALVIN 18020110200012
ALVIN berhasil ditambahkan
ADD_MAHASISWA LIKA 18010110200023
LIKA berhasil ditambahkan
ADD_MAHASISWA RYAAS 18012808200017
RYAAS berhasil ditambahkan
ADD_DOSEN BUDI
BUDI berhasil ditambahkan
ADD_ELEMEN_KANTIN KEVIN
KEVIN berhasil ditambahkan
ADD_MAKANAN ALVIN SAPI 10000
[DITOLAK] ALVIN bukan merupakan elemen kantin
ADD_MAKANAN KEVIN SAPI 10000
KEVIN telah mendaftarkan makanan SAPI dengan harga 10000
ADD_MAKANAN KEVIN SAPI 5000
[DITOLAK] SAPI sudah pernah terdaftar.
MEMBELI_MAKANAN KEVIN KEVIN SAPI
[DITOLAK] Elemen kantin tidak bisa membeli makanan sendiri
MEMBELI_MAKANAN ALVIN RYAAS SAPI
[DITOLAK] Hanya elemen kantin yang dapat menjual makanan
MEMBELI_MAKANAN ALVIN KEVIN AYAM
[DITOLAK] KEVIN tidak menjual AYAM
MEMBELI_MAKANAN ALVIN KEVIN SAPI
ALVIN berhasil membeli SAPI seharga 10000
CREATE_MATKUL DDP 10
DDP berhasil ditambahkan dengan kapasitas 10
RINGKASAN_MATKUL DDP
Nama mata kuliah: DDP
Jumlah mahasiswa: 0
Kapasitas: 10
Dosen pengajar: Belum ada
Daftar mahasiswa yang mengambil mata kuliah ini:
Belum ada mahasiswa yang mengambil mata kuliah ini
ADD_MATKUL KEVIN DDP
[DITOLAK] Hanya mahasiswa yang dapat menambahkan matkul
ADD_MATKUL ALVIN DDP
ALVIN berhasil menambahkan mata kuliah DDP
ADD_MATKUL LIKA DDP
LIKA berhasil menambahkan mata kuliah DDP
```

```
RINGKASAN_MATKUL DDP
Nama mata kuliah: DDP
Jumlah mahasiswa: 2
Kapasitas: 10
Dosen pengajar: Belum ada
Daftar mahasiswa yang mengambil mata kuliah ini:
1. ALVIN
2. LIKA
MENGAJAR_MATKUL KEVIN DDP
[DITOLAK] Hanya dosen yang dapat mengajar matkul
MENGAJAR_MATKUL BUDI DDP
BUDI mengajar mata kuliah DDP
RINGKASAN_MATKUL DDP
Nama mata kuliah: DDP
Jumlah mahasiswa: 2
Kapasitas: 10
Dosen pengajar: BUDI
Daftar mahasiswa yang mengambil mata kuliah ini:
1. ALVIN
2. LIKA
BERHENTI_MENGAJAR LIKA
[DITOLAK] Hanya dosen yang dapat berhenti mengajar
BERHENTI_MENGAJAR BUDI
BUDI berhenti mengajar DDP
RINGKASAN_MATKUL DDP
Nama mata kuliah: DDP
Jumlah mahasiswa: 2
Kapasitas: 10
Dosen pengajar: Belum ada
Daftar mahasiswa yang mengambil mata kuliah ini:
1. ALVIN
2. LIKA
CREATE_MATKUL POK 20
POK berhasil ditambahkan dengan kapasitas 20
ADD_MATKUL ALVIN POK
ALVIN berhasil menambahkan mata kuliah POK
RINGKASAN_MAHASISWA BUDI
[DITOLAK] BUDI bukan merupakan seorang mahasiswa
RINGKASAN_MAHASISWA ALVIN
Nama: ALVIN
Tanggal lahir: 1-10-2000
Jurusan: Sistem Informasi
Daftar Mata Kuliah:
1. DDP
2. POK
```

```
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(0)
2. BUDI(0)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
MENYAPA ALVIN LIKA
ALVIN menyapa dengan LIKA
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(0)
2. BUDI(0)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
MENYAPA ALVIN RYAAS
ALVIN menyapa dengan RYAAS
MENYAPA ALVIN LIKA
ALVIN menyapa dengan LIKA
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(10)
2. BUDI(0)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
MENYAPA ALVIN RYAAS
ALVIN menyapa dengan RYAAS
MENYAPA ALVIN BUDI
ALVIN menyapa dengan BUDI
MENYAPA KEVIN BUDI
KEVIN menyapa dengan BUDI
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(20)
2. BUDI(10)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(15)
2. BUDI(5)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
```

```

MEMBELI_MAKANAN ALVIN KEVIN SAPI
ALVIN berhasil membeli SAPI seharga 10000
MEMBELI_MAKANAN ALVIN KEVIN SAPI
ALVIN berhasil membeli SAPI seharga 10000
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(12)
2. BUDI(0)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
MENYAPA ALVIN LIKA
ALVIN menyapa dengan LIKA
MENYAPA ALVIN BUDI
ALVIN menyapa dengan BUDI
MENYAPA RYAAS BUDI
RYAAS menyapa dengan BUDI

NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(22)
2. BUDI(10)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
MENGAJAR_MATKUL BUDI POK
BUDI mengajar mata kuliah POK
MENYAPA BUDI ALVIN
BUDI menyapa dengan ALVIN
MENYAPA ALVIN KEVIN
ALVIN menyapa dengan KEVIN
NEXT_DAY
Hari telah berakhir dan nilai friendship telah diupdate
1. ALVIN(34)
2. BUDI(7)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)
PROGRAM_END
Program telah berakhir. Berikut nilai terakhir dari friendship pada Fasilkom :
1. ALVIN(34)
2. BUDI(7)
3. KEVIN(0)
4. LIKA(0)
5. RYAAS(0)

```