

## Tugas Besar IF5021/Algoritma dan Pemrograman B

### Complement Class Scheduler

#### Deskripsi Soal

Diketahui lembaga pendidikan Alpro yang bekerja sama dengan ITB akan mengadakan beberapa kelas 'pelengkap' saat libur panjang Juni-Agustus (**12 minggu**). Kelas tersebut menyiapkan dan memantapkan mahasiswa akhir untuk menguasai beberapa skill programming dan pengetahuan yang dibutuhkan untuk memenuhi standar kompetensi pekerjaan yang diinginkan mahasiswa tersebut.

Diketahui bahwa mahasiswa memiliki **background pengetahuan yang berbeda-beda** tergantung pada mata kuliah yang mereka ambil sebelumnya. Mahasiswa **menentukan sendiri** cita-cita pekerjaan mereka.

Setiap pekerjaan memiliki kompetensi yang berbeda-beda yang perlu dimiliki mahasiswa untuk bisa berkarir di pekerjaan tersebut. Mahasiswa juga sudah memiliki daftar kompetensi yang mereka sudah kuasai.

Kelas akan diadakan apabila terdapat **1 dosen** yang memiliki kompetensi dan **minimal 10 mahasiswa dan maksimal 25 mahasiswa** yang memiliki minat di bidang tersebut. Kelas memiliki bobot masing-masing dan setiap bobot bernilai 1 jam tatap muka dan 1 jam praktikum (jika ada praktikum).

Biaya setiap kelas adalah **(500rb/mahasiswa \* bobot) untuk pertemuan tatap muka dan (750rb/mahasiswa\*bobot) untuk pertemuan tatap muka + praktikum**. Kelas diadakan selama **10 pertemuan + 10 praktikum (jika ada praktikum)** dan setiap kompetensi memiliki waktu ajar (jumlah hari) yang berbeda-beda.

**Praktikum diadakan setidaknya 1x ganti hari setelah pertemuan dan kurang dari 7 hari setelah pertemuan.**

Keterangan:

- Hari jadwal kelas akan dipilih dari tanggal 1 Juni sampai dengan tanggal 31 Agustus tahun 2019.
- Untuk pemilihan hari jadwal kelas, tidak perlu memperhatikan hari Sabtu dan Minggu ataupun hari libur. Tetapi jika kalian mampu memperhitungkan hari Sabtu dan Minggu dan hari libur nasional, maka akan ada **bonus** nilai.
- Selain memilih jadwal hari kelas, ditentukan juga jadwal jam untuk mengadakan kelas di hari tersebut. Slot jam perkuliahan yang disediakan oleh ITB adalah dari jam 07.00 s.d. 17.00 dengan catatan jam 12.00 s.d. 13.00 adalah waktu istirahat sehingga tidak boleh ada kegiatan perkuliahan.
- Hari dan jam pelaksanaan tidak harus sama setiap minggu.  
Contoh: kelas kompetensi pemrograman berorientasi objek pada minggu 1 diadakan di hari Selasa pukul 14.00-16.00, tetapi pada minggu 2 diadakan di hari Kamis pukul 13.00-15.00.
- Sebuah kelas hanya mengajarkan satu buah kompetensi, demikian pula, satu kompetensi hanya diajarkan di satu kelas.  
Contoh: kompetensi Programming skills (R dan Python) diajarkan di kelas Programming skills (R dan Python).
- Suatu kompetensi memiliki daftar requirement kompetensi lain yang harus dikuasai oleh peserta sebelum mengambil kelas kompetensi tersebut.  
Contoh: Kompetensi Keamanan Jaringan memiliki prasyarat kompetensi Desain dan Manajemen Jaringan.

- Jika mahasiswa belum memiliki memenuhi requirement kompetensi untuk suatu kelas, maka mahasiswa tersebut tidak boleh mengambil kelas kompetensi tersebut, kecuali kalau pada saat yang bersamaan dia juga mengambil kelas kompetensi prasyarat tersebut.  
Contoh: mahasiswa yang belum memiliki kompetensi Desain dan Manajemen Jaringan tidak bisa mengambil kelas Keamanan Jaringan, kecuali pada saat yang bersamaan dia juga mengambil kelas Desain dan Manajemen Jaringan. Jika tidak ada jadwal untuk kelas Design dan Manajemen Jaringan yang tersedia bagi mahasiswa, maka mahasiswa otomatis tidak dapat mengambil kelas Keamanan Jaringan.
- Dapat terjadi keadaan di mana mahasiswa tidak dapat mengambil seluruh kelas kompetensi dari pekerjaan yang diinginkan. Misal karena kelas penuh, atau jadwal yang tidak tersedia.
- Default-nya mahasiswa tidak dapat memilih dosen pengajar ataupun pindah kelas. **Bonus:** mengimplementasikan fitur memilih dosen ajar ataupun pindah kelas.

Dosen dan mahasiswa bisa mengikuti beberapa kelas sekaligus, tetapi tidak boleh memiliki jadwal kelas yang bentrok. Jika terdapat kelas yang bentrok, maka diprioritaskan kelas dengan urutan kompetensi pertama yang disimpan pada master data dosen dan pekerjaan.

Contoh:

- Pekerjaan Data scientist dengan urutan kompetensi: 1) Statistical analysis, 2) Data mining, 3) Data processing, 4) Programming skills (R dan Python), 5) Knowledge in SAS, 6) Big data and Unstructured data.
- Dosen dengan nama Pierre memiliki urutan kompetensi: 1) data mining, 2) statistical analysis, 3) Calculus.

## Deskripsi Role

Ada 3 macam pengguna:

- **Mahasiswa**, yaitu pengguna yang akan mengikuti kelas tersebut berdasarkan cita-cita pekerjaan yang ingin dicapai.
- **Dosen**, yaitu pengajar yang akan mengisi kelas sesuai dengan kompetensinya.
- **Admin**, yaitu pengguna yang dapat mengelola data pengguna, data kompetensi, data kelas, dll.

## Deskripsi Proses

Berikut adalah daftar proses dari sistem:

### 1. Login ke dalam sistem

Pengguna memasukkan username dan password untuk mengakses menu di dalam sistem. Setelah Login berhasil, tampilkan menu sesuai dengan peran pengguna.

### 2. Mengelola akun pengguna

Admin dapat membuat, mengubah dan menghapus akun dan data dosen dan mahasiswa.

Data dosen/mahasiswa terdiri atas: username, password, nama, NIM/NIP, daftar kompetensi awal (baik mahasiswa/dosen).

Dosen dan mahasiswa dapat mengubah data diri, yaitu: Nama, NIM/NIP, daftar kompetensi awal, dan password. username tidak dapat diubah.

Ketentuan username dan password:

- username pengguna harus dalam format sbb.: itb<NIM/NIP>. Contoh itb23518029
- password default adalah <NIM/NIP>, contoh 23518029

### 3. Mengelola data kompetensi

Admin dapat membuat, mengubah dan menghapus data kompetensi.

Data kompetensi terdiri atas: nama kompetensi, membutuhkan praktikum/tatap muka, requirement kompetensi, dan bobot.

Format penulisan data kompetensi pada input:

[<P/M>] <nama kompetensi> {<k1,k2,...,kn>} ~<bobot>

Keterangan:

- <P/M> bernilai P jika membutuhkan tatap muka dan praktikum; bernilai M jika hanya membutuhkan tatap muka.
- <k1,k2,...,kn> adalah daftar kompetensi awal yang harus dimiliki peserta sebelum mengambil kelas kompetensi yang berkaitan. Setiap kompetensi dipisahkan dengan koma. Daftar ini bisa kosong.

**Contoh format input kompetensi:**

- [P] Statistical analysis {} ~2 berarti kompetensi Statistical analysis membutuhkan tatap muka dan praktikum, tidak memiliki requirement kompetensi awal, dan bobot = 2.
- [M] Data processing {} ~2 berarti kompetensi Data processing membutuhkan hanya tatap muka, tidak memiliki requirement kompetensi awal, dan bobot = 2.
- [P] Data mining {Database,Algorithm and Data Structure} ~3 berarti kompetensi Data mining membutuhkan tatap muka dan praktikum, memiliki requirement kompetensi awal Database,Algorithm and Data Structure, dan bobot = 3.

#### 4. Mengelola data pekerjaan

Admin dapat membuat, mengubah dan menghapus data pekerjaan dan kompetensi.

Data pekerjaan terdiri atas: nama pekerjaan dan daftar kompetensi yang harus dimilikinya.

Format penulisan data pekerjaan pada input:

<nama pekerjaan> {<k1,k2,...,kn>}

Keterangan:

- <k1,k2,...,kn> adalah daftar kompetensi untuk pekerjaan ybs. Setiap kompetensi dipisahkan dengan koma. Urutan dalam daftar menunjukkan urutan prioritas kompetensi. Daftar ini tidak boleh kosong.

**Contoh format input pekerjaan:**

- Data scientist {Statistical analysis,Data mining,Data processing, Programming skills (R dan Python),Knowledge in SAS,Big data and Unstructured data} berarti pekerjaan Data scientist membutuhkan kompetensi dengan urutan: 1) Statistical analysis, 2) Data mining, 3) Data processing, 4) Programming skills (R dan Python), 5) Knowledge in SAS, 6) Big data and Unstructured data.
- Web developer {HTML5,CSS3,PHP,JAVA,SRS,Responsive design,Laravel}
- System analyst {requirement elicitation,project management,data modeling,SDLC,OO programming}

#### 5. Mengelola kesediaan jadwal dosen

Dosen memasukkan tanggal(-tanggal) di mana dia tidak dapat mengajar (cuti). Untuk tanggal yang lain, berarti dosen tersebut bisa mengajar.

#### 6. Membangkitkan jadwal kelas

Admin membangkitkan jadwal kelas berdasarkan data kelas (berdasarkan data kompetensi), data kesediaan dosen, dan aturan jumlah jam yang harus dipenuhi (lihat Deskripsi Soal). Untuk setiap kelas didefinisikan: tanggal, jam, dan dosen pengajar. Algoritma penjadwalan diserahkan pada Anda.

Prasyarat untuk melakukan langkah 6 adalah bahwa data-data berikut sudah diasumsikan lengkap:

- Data dosen dan ketersediaan jadwal
- Data kompetensi (kelas)
- Data pekerjaan

*Jika langkah 6 sudah pernah dilakukan, maka tidak bisa lagi dilakukan penambahan, pengubahan, dan penghapusan terhadap data dosen dan ketersediaan jadwalnya, kompetensi (kelas), dan pekerjaan.*

*Data Mahasiswa dapat ditambahkan/diubah/dihapus kapan pun.*

*Proses-proses berikut hanya dapat dilakukan jika butir 6 sudah dijalankan.*

## **7. Mengelola pilihan pekerjaan Mahasiswa**

Prasyarat: Jadwal kelas sudah terbentuk (butir 6 telah dilakukan).

Mahasiswa memasukkan **tepat tiga buah** pilihan pekerjaan yang diinginkan. Kemudian sistem akan memberikan kelas apa saja yang harus diambil untuk dapat memenuhi kompetensi yang dibutuhkan untuk pekerjaan-pekerjaan tersebut, termasuk mempertimbangkan kompetensi awal Mahasiswa

Misal:

Mahasiswa A memilih pekerjaan `data scientist`, `web developer`, dan `system analyst`

Mahasiswa A sudah memiliki kompetensi awal `Statistical analysis`, `HTML5`, `requirement elicitation`, `SRS`, dan `project management`.

Sistem akan memberikan daftar kelas berikut jadwal yang dapat diambil berdasarkan data pekerjaan yang dipilih dan kompetensi awal Mahasiswa. Berikut pilihan optimasi yang dapat dilakukan untuk menghasilkan daftar kelas dan jadwal:

- **Optimasi-1:** dihasilkan daftar kelas dan jadwal berdasarkan kelas-kelas yang memungkinkan mahasiswa untuk memenuhi seluruh kompetensi dari pekerjaan yang dipilih. Jika hal ini tidak mungkin dilakukan karena ada jadwal yang bentrok (lihat ketentuan pada Deskripsi Soal), maka akan dipilih kombinasi kelas-kelas yang mampu memberikan **paling sedikit defisit kompetensi** untuk memenuhi syarat pekerjaan.
- **Optimasi-2:** dihasilkan daftar kelas dan jadwal berdasarkan kombinasi bobot kelas terbanyak yang dapat diambil dan memberikan penghasilan paling banyak bagi lembaga (lihat kembali perhitungan biaya pada Deskripsi Soal).

Dengan demikian, Mahasiswa A memiliki potensi untuk mengambil beberapa kelas, yaitu: `Data mining`, `Data processing`, `Programming skills (R dan Python)`, `Knowledge in SAS`, `Big data and Unstructured data`, `CSS3`, `PHP`, `JAVA`, `Responsive design`, `Laravel`, `data modeling`, `SDLC`, `OO programming`.

## **8. Melihat jadwal**

Admin dapat melihat jadwal kelas yang terbentuk beserta Mahasiswa dan dosen yang terlibat.

Mahasiswa dan dosen dapat melihat jadwalnya masing-masing.

## **9. Melihat biaya tagihan mahasiswa.**

Admin dan mahasiswa dapat melihat biaya tagihan berdasarkan kelas yang diambil mahasiswa tersebut beserta status pembayaran (belum/sudah).

## **10. Membayar tagihan.**

Mahasiswa dapat mengajukan pembayaran dengan memasukkan kode transaksi dan tanggal pembayaran. Jika pembayaran telah dilakukan, status pembayaran menjadi lunas.

## **11. Melihat pemasukan dan tagihan.**

Admin dapat melihat total pemasukan dari seluruh tagihan mahasiswa seluruhnya, total tagihan yang sudah bayar dan yang belum bayar.

## **12. Melihat daftar mahasiswa yang belum bayar.**

Admin dapat melihat daftar mahasiswa yang belum bayar.

## **13. Melihat jumlah mahasiswa yang memenuhi kompetensi pekerjaan yang dia harapkan.**

Admin dapat melihat jumlah mahasiswa yang berhasil memenuhi seluruh persyaratan kompetensi untuk pekerjaan yang dipilih.

## **14. Mengganti jadwal kelas**

Dosen dapat memasukkan tanggal di mana dia berhalangan. Jika pada tanggal tersebut ternyata dia ada jadwal mengajar, sistem mencari jadwal pengganti secara otomatis di mana dosen

tersebut dapat mengajar. Harap diingat, hal ini mempengaruhi jadwal mahasiswa (pertimbangkan kemungkinan bentrok).

## Antarmuka Pengguna

Semua antarmuka sistem dibuat dalam command line interface. Tulisan tebal dan garis bawah adalah masukan pengguna. Bagian ini berupa contoh, silakan buat antar muka sesuai kreativitas masing-masing.

### 1. Login

```
#Login Sistem#

Username : itb23518029
Password : 23518029
```

### 2. Tambah Kompetensi

```
#Tambah Kompetensi#

Tambah Kompetensi : [P] Statistical analysis {} ~2
-----
Sukses : Kompetensi Berhasil Ditambahkan

1. Lihat Daftar Kompetensi
0. Menu Utama

Pilihan : 0
```

### 3. Tambah Data Pekerjaan

```
#Tambah Data Pekerjaan#

Tambah Pekerjaan : Data scientist {Statistical analysis, Data mining, Data
processing, Programming skills (R dan Python), Knowledge in SAS, Big
data and Unstructured data}
-----
Sukses : Pekerjaan Berhasil Ditambahkan

1. Lihat Daftar Pekerjaan
0. Menu Utama

Pilihan : 0
```

#### 4. Lihat Data Kompetensi

#Daftar Kompetensi#

No.	Kompetensi	Prasyarat	Bobot
1	Statistical Analysis [P]	-	2
2	Keamanan Jaringan [P]	Design Manajemen Jaringan [M]	2

Ed\_<nomor kompetensi> : untuk edit kompetensi

Del\_<nomor kompetensi> : untuk hapus kompetensi

Add : untuk tambah kompetensi

0 : Kembali ke Menu Utama

Pilihan : 0

#### 5. Lihat Data Pekerjaan

#Daftar Pekerjaan#

No.	Pekerjaan	Kompetensi
1	Data Scientist	Statistical Analysis, Data Mining, Data Processing, Programming Skills (R dan Python), Knowledge in SAS, Big Data and Unstructured Data
2	Web Developer	HTML5, CSS3, PHP, JAVA, SRS, Responsive Design, Laravel

Ed\_<nomor pekerjaan> : untuk edit pekerjaan

Del\_<nomor pekerjaan> : untuk hapus pekerjaan

Add : untuk tambah pekerjaan

0 : Kembali ke Menu Utama

Pilihan : 0

#### 6. Buat Jadwal Kelas

#Buat Jadwal#

Apakah anda yakin untuk membuat jadwal (y/n) ? y

---

Periode Pengisian data kompetensi, pekerjaan, dan hari kesediaan dosen dihentikan..  
Pembuatan Jadwal Sukses !

1. Lihat Jadwal Kelas

0. Kembali ke Menu Utama

Pilihan : 0

## Tugas

Implementasikan sistem sesuai dengan spesifikasi dan penjelasan yang telah diberikan. Nama fungsi/prosedur dibebaskan kecuali pada beberapa proses berikut:

1. Penambahan kompetensi (`tambahKompetensi`)
2. Ubah kompetensi (`ubahKompetensi`)
3. Hapus kompetensi (`hapusKompetensi`)
4. Lihat kompetensi (`lihatKompetensi`)
5. Penambahan pekerjaan (`tambahPekerjaan`)
6. Ubah pekerjaan (`ubahPekerjaan`)
7. Hapus pekerjaan (`hapusPekerjaan`)
8. Lihat pekerjaan (`lihatPekerjaan`)
9. Ubah data diri (`ubahDataIdv`)
10. Setting hari libur (`setLibur`)
11. Total pendapatan (`hitungPendapatan`)
12. Hentikan periode pengisian data (`doExecute`)
13. Mengisi pekerjaan yang diidamkan (`isiPekerjaan`)

Definisikan fungsi-fungsi pada proses lainnya dan pastikan keterlacakan antara proses dengan fungsi terdefinisi dengan jelas pada dokumentasi. Dokumentasi paling tidak harus mengandung Use Case Diagram, Class Diagram, dan Sequence Diagram.

## Asumsi

Jika terdapat ketidakjelasan atau ambiguitas pada soal, silahkan membuat asumsi sendiri dan asumsi tersebut disertakan pada forum tugas besar Olympia dengan alur validasi:

- Ketua kelompok menulis asumsi pada forum.
- Asisten akan mengecek forum pada jam tertentu.
- Asisten berpikir dan memutuskan untuk menambahkan asumsi pada file khusus yang linknya terdapat di olympia. Update akan dilakukan secepatnya dan silahkan cek pada hari Selasa jam 17.00, Jumat 19.00, Minggu 19.00.
- Seluruh kelompok wajib menggunakan asumsi yang sudah ditambahkan tersebut.

## Catatan Tambahan

1. Semua data yang dibuat dan dimanipulasi oleh sistem harus ***persistent***. Penyimpanan data dilakukan dalam file dengan format JSON. Pembacaan dan penulisan file **JSON dilakukan on-demand**, yaitu hanya pada saat diperlukan, bukan pada saat aplikasi dibuka dan ditutup. Library ini mampu secara otomatis memetakan sebuah data JSON menjadi Object Java, atau sebaliknya. Berikut tautan yang dapat dijadikan referensi: <https://github.com/FasterXML/jackson>  
<https://www.mkyong.com/java/jackson-2-convert-java-object-to-from-json/>
2. Buatlah struktur aplikasi Anda sedemikian rupa sehingga logik aplikasi dan antarmuka dapat dipisahkan dengan baik. Dengan demikian, suatu hari Anda dapat mengembangkan aplikasi ini menjadi aplikasi dengan antarmuka lain, seperti web atau mobile.
3. Gunakanlah repository **gitlab.informatika.org** untuk memudahkan tim Anda melakukan pelacakan perubahan dan menulis kode secara paralel.
4. **Asumsi:** aplikasi yang dibuat adalah aplikasi dengan sistem single-user, artinya pengguna dapat masuk satu per satu dengan menggunakan aplikasi yang sama secara offline. Tidak ada multiple users yang masuk secara bersamaan.

## Aturan Pengerjaan

1. Tugas dikerjakan mulai Kamis, 8 November 2017.
2. Tugas dikerjakan secara berkelompok. Setiap kelompok terdiri dari 4 orang.
3. Pembentukan kelompok bebas, tetapi pembagian pekerjaan harus tertulis di dokumen. Tulis kelompok dan anggota kelompok pada forum tugas besar.
4. Tugas dikerjakan dengan menggunakan bahasa Java.
5. Project di-*build* menggunakan Maven (<http://maven.apache.org/>)
6. Asumsi dapat ditambahkan sesuai bulir VI.
7. Pada jam praktikum mingguan (Kamis, 13.00-15.00), setiap kelompok wajib melakukan asistensi dengan asisten. Setiap kelompok harus membuat Catatan Asistensi (form dapat diunduh di situs Olympia) dan mengumpulkan deliverable sebagai hasil dari praktikum mingguan pada link yang akan diberikan di Olympia. Deliverable dikirim dalam 1 file .zip yang berisi:
  - a. Catatan Asistensi.
  - b. Activity Log mingguan setiap anggota kelompok.
  - c. Draft Laporan sebagai progress kemajuan pengerjaan tugas.
8. Aturan dan pengumuman tambahan yang belum tertulis di file ini akan diumumkan di forum tugas besar olympia.

## Jadwal Penting

Waktu	Kegiatan	Tempat
15 November 2018 pukul 13.00-14.40	Konsultasi Wajib dengan asisten	Lab Dasar Informatika IV
22 November 2018 pukul 13.00-14.40	Presentasi-1: Desain aplikasi	Lab Dasar Informatika IV
27 November 2018 pukul 10.00-11.40	Presentasi-2: Hasil Implementasi	Ruang 7608
29 November 2018 pukul 07.00	Deadline pengumpulan source code program dan laporan	
29 November 2018 pukul 13.00-14.40	Demo program	Lab Dasar Informatika IV