

## Tubes Alpro 2018-1 :

### Complement class scheduler

#### I. Deskripsi singkat :

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. Dan mahasiswa **menentukan sendiri** cita-cita pekerjaan mereka.

Setiap pekerjaan akan 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.**

NB:

- 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, maka akan ditentukan juga jadwal waktu untuk mengadakan kelas di hari tersebut. Slot waktu untuk 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 waktu pelaksanaan tidak harus sama setiap minggu.  
Contoh : kelas kompetensi pemrograman berbasis object 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.
- Kelas akan berkaitan dengan daftar kompetensi.

Contoh: jika ada kompetensi Programming Skill Python, maka akan ada kelas Programming Skill Python

- Kompetensi akan memiliki daftar requirement yang sebelumnya harus dikuasai sebelum mengambil kelas kompetensi tersebut.

Contoh : Kompetensi Keamanan Jaringan memiliki prasyarat kompetensi Desain dan Manajemen Jaringan.

- Jika mahasiswa belum memiliki memenuhi requirement kompetensi, maka mahasiswa tersebut tidak boleh mengambil kelas kompetensi tersebut. Tetapi Mahasiswa bisa mengambil kelas kompetensi dan prasyaratnya secara bersamaan

Contoh : mahasiswa yang belum memiliki kompetensi Desain dan Manajemen Jaringan tidak bisa mengambil kelas Keamanan Jaringan secara langsung. Mahasiswa harus mengambil juga kelas Design dan Manajemen Jaringan dan Keamanan 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 dimana mahasiswa tidak dapat mengambil seluruh kelas kompetensi dari pekerjaan yang diinginkan. Misal karena kelas penuh, atau jadwal yang tidak tersedia.
- Defaultnya mahasiswa tidak dapat memilih dosen ajar ataupun pindah kelas. Tetapi meng-implementasi metode/fungsi memilih dosen ajar ataupun pindah kelas merupakan sebuah **bonus**.

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

Contoh :

- Data scientist: Statistical analysis, Data mining, Data processing, Programming skills(Prefer: R or python), Knowledge in SAS, Big data and Unstructured data -> maka kompetensi yang menjadi prioritas adalah statistical analysis diikuti oleh data mining, data processing, dst.
- Pierre, data mining, statistics, calculus -> maka dosen pierre memiliki kompetensi data mining yang menjadi prioritas diikuti statistics dan calculus.

## II. 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 bidang keahliannya.

Administrasi, yaitu pengguna yang dapat mengelola data pengguna dan data kelas,dll.

## III. 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.

Dosen dan mahasiswa dapat mengubah data diri (Nama,(NIM/NIP),Kompetensi saat ini), password akun individunya, sedangkan ID atau username tidak dapat diubah.

#username mahasiswa adalah: itb[NIM/NIP] contoh itb23518029

# password default adalah [NIM] contoh 23518029

3. Mengelola data kompetensi.

Admin dapat membuat, mengubah dan menghapus data kompetensi.

**/\*Contoh standar kompetensi\*/**

#M: tatap Muka saja, [P]: terdapat praktikum, {}: requirement, ~jumlah bobot

[P] Statistical analysis ~2, [P] Data mining {Database, Algorithm and Data Structure} ~3, [M] Data processing ~2, [P] Programming skills R ~2, [M] Knowledge in SAS ~2, [M] Big data and Unstructured data ~2.

4. Mengelola data pekerjaan.

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

Mahasiswa dapat memilih pekerjaan yang diinginkan.

**/\* Contoh data pekerjaan \*/**

Data scientist: Statistical analysis,Data mining, Data processing, Programming skills R -or- Programming skills python, Knowledge in SAS, 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.

Mahasiswa memasukkan **tepat tiga buah** pilihan pekerjaan yang diinginkan. Kemudian sistem akan menentukan pekerjaan yang paling cocok untuk ditempuh mahasiswa berdasarkan kompetensi yang paling banyak dipenuhi oleh mahasiswa tersebut.

Misal :

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

Mahasiswa A sudah memiliki kompetensi Statistical analysis, HTML5, requirement elicitation, Srs, dan project management.

**Jika optimasi 1** (pada point 6) yang dipilih, maka system akan memilih kelas - kelas yang memungkinkan mahasiswa untuk **memenuhi seluruh kompetensi dari pekerjaan yang dipilih**. Jika tidak ada pekerjaan yang mampu dipenuhi

kompetensinya, maka akan dipilih kombinasi kelas - kelas yang mampu memberikan **paling sedikit defisit kompetensi** untuk memenuhi syarat pekerjaan.

**Jika optimasi 2** (pada point 6) yang dipilih, maka system akan memilih kombinasi **bobot kelas terbanyak** yang dapat diambil dan **memberikan penghasilan** paling banyak bagi lembaga.

Dari hasil seleksi jadwal, didapatkan bahwa mahasiswa A memiliki potensi untuk mengambil beberapa mata kuliah yakni : Data mining, Data processing, Programming skills R, Knowledge in SAS, Big data, CSS3, PHP, JAVA, Responsive design, Laravel, dan data modeling.

Oleh karena itu,

Untuk pekerjaan data scientist, mahasiswa A kurang 1 kompetensi setelah mengambil kelas.

Untuk pekerjaan web developer, mahasiswa A memenuhi seluruh kompetensi setelah mengambil kelas.

Untuk pekerjaan system analyst, mahasiswa A kurang 2 kompetensi setelah mengambil kelas.

Untuk pekerjaan data scientist, maka ada total 5 kelas yang bisa diambil

Untuk pekerjaan web developer, maka ada total 5 kelas yang bisa diambil

Untuk pekerjaan system analyst, maka ada total 1 kelas yang bisa diambil

Sehingga untuk **optimasi jumlah mahasiswa (poin 6)** yang memenuhi kompetisi, maka mahasiswa A akan **dipilihkan jalur pekerjaan web developer**.

Untuk **optimasi keuntungan lembaga**, maka pekerjaan dengan **prioritas paling utama** dari dua pekerjaan yang bisa diambil mahasiswa A, yakni data scientist dan web developer yang diambil.

#### 5. Mengelola data hari libur.

Dosen dan mahasiswa mempunyai jadwal 'libur' sendiri-sendiri sehingga data masukan hanya berupa range tanggal yang menyatakan hari 'libur' mereka.

#### 6. Menghentikan periode pengisian data.

Admin dapat menghentikan periode dimana mahasiswa dan dosen bisa mengisi dan mengubah data pribadi, pengisian data kompetensi, dan data pekerjaan.

Setelah periode dihentikan maka sistem akan menghasilkan jadwal kelas. Jadwal dimulai sehari setelah admin menghentikan periode.

Ketika jadwal sudah terbentuk, maka mahasiswa dan dosen tidak bisa lagi menambah hari libur. Dosen masih bisa mengganti jadwal kelas ketika jadwal sudah dibentuk.

Admin akan memilih satu dari dua buah metode penjadwalan, sbb:

1. Optimasi jumlah mahasiswa yang memenuhi kompetensi pekerjaan.

Jadwal yang terbentuk harus memaksimalkan jumlah mahasiswa yang memenuhi seluruh kompetensi pekerjaan idamannya.

2. Optimasi penghasilan.

Jadwal yang terbentuk harus memaksimalkan jumlah penghasilan dari tagihan seluruh mahasiswa.

7. Melihat jadwal.

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

Mahasiswa dan dosen dapat melihat jadwalnya masing-masing.

8. Melihat biaya tagihan mahasiswa.

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

9. Membayar dan validasi pembayaran.

Mahasiswa dapat mengajukan pembayaran dengan memasukkan kode transaksi.

Admin dapat melakukan validasi dan melihat kode transaksi yang dimasukkan.

10. Melihat pemasukan dan tagihan.

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

11. Melihat daftar mahasiswa yang belum bayar.

Admin dapat melihat daftar mahasiswa yang belum bayar.

12. Melihat jumlah mahasiswa yang memenuhi kompetensi pekerjaan yang diharapkan.

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

13. Mengganti jadwal kelas

Setelah jadwal telah dibuat, jika dosen berhalangan pada satu hari maka sistem harus mencari jadwal pengganti secara otomatis dengan syarat seluruh mahasiswa peserta kelas kosong pada waktu tersebut. Dosen akan memasukkan tanggal hari ini dan tanggal kelas yang ingin diganti, setelah itu sistem akan menampilkan kelas tanggal kelas pengganti yang paling pertama tersedia dari tanggal hari ini (kecuali tanggal kelas awal yang ingin diganti).

## IV. Antarmuka Pengguna

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

1. Menu Login

#Login#

UserName : itb13515022

Password : \*\*\*\*\*

2. Menu Utama Admin

#Menu Utama Admin#

1. Kelola Akun
2. Kelola Kompetensi
3. Kelola Pekerjaan
4. Lihat Peserta Kelas & Dosen
5. Lihat Jadwal Kelas
6. Lihat Tagihan
7. Lihat Hasil Kelas Pendek
8. Hentikan Periode Isi Data
9. Ubah Password

Pilih Menu : <nomor menu>

3. Menu Utama Dosen

#Menu Utama Dosen#

1. Kelola Data Diri
2. Kelola Jadwal
3. Tambah Hari Libur
4. Ubah Password

Pilih Menu : <nomor menu>

4. Menu Utama Mahasiswa

#Menu Utama Mahasiswa#

1. Kelola Data Diri
2. Lihat Jadwal
3. Tambah Hari Libur
4. Lihat Riwayat Tagihan
5. Ubah Password

Pilih Menu : <nomor menu>

5. Menu Kelola Kompetensi Admin

#Kelola Kompetensi#			
No	Kompetensi	Prasyarat	Bobot
1	Keamanan Jaringan [P]	Design dan Manajemen Jaringan	2
2	Design Basis Data [M]	-	2
3	Data Mining [P]	Database, Algorithm and Data Structure	2

Ed\_<nomor kompetensi> : untuk edit kompetensi  
 Del\_<nomor kompetensi> : untuk hapus kompetensi  
 Add : untuk tambah kompetensi  
 0 : Kembali ke Menu Utama

Pilihan : <nomor menu>

#### 6. Tampilan Edit Kompetensi

#Kelola Kompetensi#			
No	Kompetensi	Prasyarat	Bobot
1	Keamanan Jaringan [P]	Design dan Manajemen Jaringan	2
2	Design Basis Data [M]	-	2
3	Data Mining [P]	Database, Algorithm and Data Structure	2

Ed\_<nomor kompetensi> : untuk edit kompetensi  
 Del\_<nomor kompetensi> : untuk hapus kompetensi  
 Add : untuk tambah kompetensi  
 0 : Kembali ke Menu Utama

Pilihan : **Ed\_3**

/\* Tekan enter jika ingin submit data \*/

Edit Kompetensi: **[P] Data mining {Algorithm and Data Structure}~3**

Apakah anda yakin untuk mengubah (y/n) ? **y**

Success : Data Berhasil Diubah

Kemudian akan menampilkan menu kelola kompetensi dengan data yang sudah diubah.

#Kelola Kompetensi#

No	Kompetensi	Prasyarat	Bobot
1	Keamanan Jaringan [P]	Design dan Manajemen Jaringan	2
2	Design Basis Data [M]	-	2
3	Data Mining [P]	Algorithm and Data Structure	3

Ed\_<nomor kompetensi> : untuk edit kompetensi

Del\_<nomor kompetensi> : untuk hapus kompetensi

Add : untuk tambah kompetensi

0 : Kembali ke Menu Utama

Pilihan : <nomor menu>

7. Menu Kelola Data Diri Mahasiswa

#Data Diri Mahasiswa#

1. Biodata
2. Kompetensi Saat Ini
3. Pekerjaan yang Diinginkan

0 : Kembali ke Menu Utama

Pilihan : 2

8. Menu Kompetensi Saat ini Mahasiswa



```

#Data Kompetensi Mahasiswa#
Kompetensi Saat Ini : Data Mining, Statistics, Calculus

1 : Edit Kompetensi
99 : Back
0 : Menu Utama

Pilihan : 1

Daftar Kompetensi yang tersedia di Database :
Data Mining, Statistics, Calculus, Python Programming, R Programming, PHP, Togaf Framework, COBIT,
ITIL, PMBOK 5, DMBOK, Javascript, Java Programming

/* tekan enter untuk submit */
Edit Kompetensi : Data Mining, Statistics, Python Programming

Apakah anda yakin untuk edit (y/n)? y

Success : data berhasil diedit
-----

#Data Kompetensi Mahasiswa#
Kompetensi Saat Ini : Data Mining, Statistics, Python Programming

1 : Edit Kompetensi
99 : Back
0 : Menu Utama

Pilihan : <Masukan Pilihan>

```

## 9. Menu Kelola Jadwal Dosen

```

#Jadwal Dosen#
Pilih Bulan :
1. Juni
2. Juli
3. Agustus

0 : Menu Awal

Pilihan : 1
-----
#Jadwal Dosen Bulan Agustus#

7 Agustus 2019
1. 08.00 - 10.00 : Python Programming (Tatap Muka)
2. 13.00 - 15.00 : Statistics

8 Agustus 2019
3. 13.00 - 15.00 : Python Programming (Praktikum)

Ed_<nomor jadwal> : untuk mengganti jadwal pada nomor tersebut ke hari lain yang tersedia
99 : Back
0 : Menu Awal

Pilihan : Ed_3

```

Pilihan : **Ed\_3**  
Success: Kelas Statistic diubah ke tanggal 25 Agustus 2019

---

#Jadwal Dosen#

Pilih Bulan :

1. Juni
2. Juli
3. Agustus

0 : Menu Awal

Pilihan : **1**

Pilihan : **Ed\_3**

Fail: Tidak Ada Hari Lain yang Tersedia untuk Mengganti Jadwal Kelas Statistic

---

#Jadwal Dosen Bulan Agustus#

7 Agustus 2019

1. 08.00 - 10.00 : Python Programming (Tatap Muka)
2. 13.00 - 15.00 : Statistics

8 Agustus 2019

3. 13.00 - 15.00 : Python Programming (Praktikum)

Ed\_<nomor jadwal> : untuk mengganti jadwal pada nomor tersebut ke hari lain yang tersedia

99 : Back

0 : Menu Awal

Pilihan : **<Masukan Pilihan>**

## 10. Menu Hentikan Periode Pengisian Data

#Hentikan Periode Isi Data#

Hentikan periode mengisi data pekerjaan yang diharapkan mahasiswa, kompetensi dosen / mahasiswa saat ini, dan pilihan pekerjaan mahasiswa. Kemudian pilih jenis optimasi yang digunakan untuk membuat jadwal.

1. Optimasi Mahasiswa yang Memenuhi Kompetensi
2. Optimasi Pemasukan ITB

0 : Menu Utama

Pilihan: **1**

Success : Jadwal Kelas, Peserta Kelas, dan Dosen Pengajar Kelas berhasil dibuat berdasarkan optimasi 1.

0 : Menu Utama

Pilihan: **0**

## V. 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.**

## VI. 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.

## VII. 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.mkymong.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.