

## **Sistem Informasi Kemahasiswaan**

**Versi: 1.1 (20 Maret 2018)**

### **Update 1.1:**

- **Response Page pada fitur [2] dan [3] diperbolehkan menggunakan URL tidak seperti pada deskripsi**

### **Outline:**

- Pendahuluan
- Basis Data
- Fitur
- Tampilan
- Deliverables

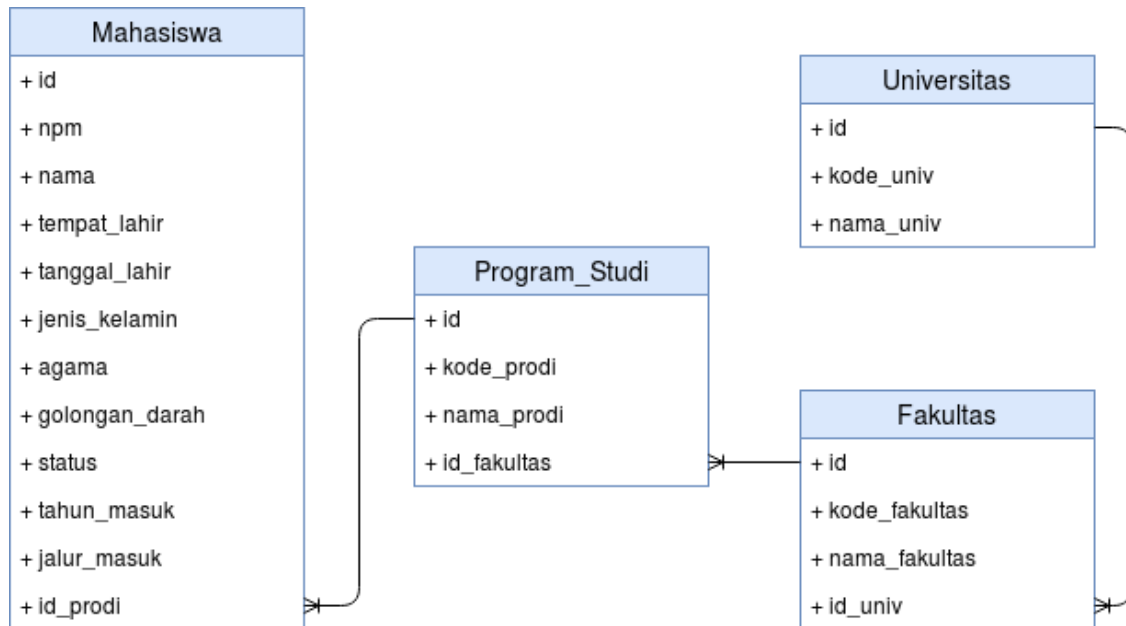
## **Pendahuluan**

Dengan banyaknya perguruan tinggi di Indonesia, dibutuhkan sistem yang dapat diandalkan dalam hal pendataan mahasiswa. Sistem ini harus dapat menangani banyak permintaan dalam satu waktu dan menanggapi permintaan dengan cepat.

Sistem ini bertujuan agar petugas dapat dengan mudah menambahkan data mahasiswa. Sementara mahasiswa dapat dengan mudah melakukan pengecekan terhadap data kemahasiswaan mereka.

Bapak Kemendikbud telah setuju untuk membuat sebuah sistem kemahasiswaan online. Anda diminta untuk mengembangkan sistem tersebut berdasarkan spesifikasi fitur yang telah diberikan.

## Basis Data



Gambar 1. ERD Sistem Pendataan Mahasiswa

Pada skema tersebut, terdapat empat macam tabel. Tabel tersebut adalah MAHASISWA, PROGRAM\_STUDI, FAKULTAS, dan UNIVERSITAS. Tabel UNIVERSITAS merepresentasikan Universitas yang ada di Indonesia. Tabel FAKULTAS merepresentasikan Fakultas yang ada di suatu Universitas, dan tabel PROGRAM\_STUDI merepresentasikan Program Studi yang ada di suatu Fakultas. Sedangkan tabel MAHASISWA menggambarkan seorang individu yang menjadi mahasiswa di suatu Universitas.

Seorang mahasiswa pasti berafiliasi dengan suatu Program Studi, sehingga setiap mahasiswa pasti tercatat dalam suatu Program Studi. Suatu Program Studi pasti terdapat dalam suatu Fakultas, begitu juga dengan suatu Fakultas yang pasti terdapat dalam suatu Universitas.

Untuk detail dari setiap atribut dapat dilihat pada dokumen *data dictionary* yang dilampirkan untuk tugas ini. Sementara itu, untuk melakukan *import database* dapat menggunakan *file si\_kemahasiswaan.sql* yang sudah dilampirkan juga untuk tugas ini.

## Fitur

### [1] Menampilkan Data Mahasiswa

**Initial Page:** /

**Form Request:** GET, /mahasiswa?npm={NPM}

**Response Page:** /mahasiswa?npm={NPM}

**Poin:**

## Sistem Informasi Kemahasiswaan

Selamat datang di Sistem Informasi Kemahasiswaan

### Lihat Data Mahasiswa Berdasarkan NPM

Masukan Nomor Pokok Mahasiswa

Menampilkan data seorang mahasiswa jika diberikan NPM mahasiswa tersebut. Pada halaman "/" terdapat sebuah form yang berisi input NPM dan tombol Lihat. Ketika tombol ini diklik, maka *request* akan dikirim. Kemudian aplikasi Anda perlu mengembalikan sebuah halaman, **"/mahasiswa?npm={NPM}"**, yang menampilkan data mahasiswa dengan npm tersebut ditambah data prodi dimana mahasiswa tersebut terdaftar. Berikut contoh halaman jika NPM ditemukan.

### Lihat Data Mahasiswa - 3101011405170001

NIK	3101011405170001
Nama	Heru Haryanto
Tempat/Tanggal Lahir	Jakarta 14-05-2017
Program Studi	PULAU TIDUNG
Fakultas	KEPULAUAN SERIBU SELATAN
Universitas	KABUPATEN KEPULAUAN SERIBU
Jenis Kelamin	O-
Agama	Islam
Golongan Darah	O-
Tahun Masuk	BELUM/TIDAK BEKERJA
Jalur Masuk	WNI
Status	Hidup <span>Lulus</span> <span>Non-Aktifkan</span>

## [2] Menambahkan Data Mahasiswa

Initial Page: /mahasiswa/tambah

Form Request: POST, /mahasiswa/tambah

Response Page: /mahasiswa/tambah (disarankan menggunakan URL ini tetapi boleh menggunakan yang lain)

Poin:

Menambahkan data mahasiswa dan menambahkan mahasiswa tersebut agar terdaftar pada suatu Program studi. Pada halaman “**mahasiswa/tambah**”, Anda perlu membuat *form* dengan ketentuan *input form field* yaitu seluruh kolom yang ada di tabel Mahasiswa, kecuali id dan NPM. Id merupakan *identifier* pada database yang *auto increment* sehingga tidak perlu di *input*, sedangkan NPM merupakan data yang di-generate berdasarkan input dari pengguna. Berikut merupakan ketentuan nomor NPM seluruh Universitas di Indonesia.

- NPM terdiri dari 12 digit.
- 2 digit awal merupakan 2 digit akhir dari tahun masuk mahasiswa.
- 3 digit setelahnya merupakan kode Universitas, dan 2 digit setelahnya merupakan kode Program Studi.
- 2 digit setelahnya merupakan jalur masuk mahasiswa.
- 3 digit terakhir adalah nomor berdasarkan urutan input data

Jalur masuk yang tersedia adalah sebagai berikut.

- Undangan Olimpiade : 53
- Undangan Reguler / SNMPTN : 54
- Undangan Paralel / PPKB : 55
- Ujian Tulis Bersama / SBMPTN : 57
- Ujian Tulis Mandiri : 62

Sebagai contoh, misalkan seorang perempuan pada tahun 2017 diterima di suatu Universitas dengan kode: 201, pada Program Studi dengan kode: 41 dan melalui jalur masuk ujian tulis mandiri. Maka perempuan tersebut mendapatkan nomor NPM 17 201 41 62 001. Kemudian jika ada seseorang yang juga diterima pada tahun yang sama, Universitas yang sama, program studi yang sama, serta jalur masuk yang sama, maka akan mendapatkan NPM 17 201 41 62 002.

\**whitespace* pada NPM hanya untuk mempermudah penjelasan.

SI Kemahasiswaan   Home   Tambah Mahasiswa   Kelulusan Mahasiswa   Cari Mahasiswa

### Tambah Mahasiswa

Nama Lengkap

Tempat Lahir

Tanggal Lahir

Jenis Kelamin

Laki-laki

Agama

Islam

Golongan Darah

A-

Tahun Masuk

Jalur Masuk

Undangan Olimpiade

Id Program Studi

Simpan

Ketika tombol simpan di-klik request POST akan dikirim. Jika operasi berhasil, redirect ke halaman “/mahasiswa/tambah” dan kali ini tampilkan pesan bahwa operasi penambahan mahasiswa berhasil dilakukan. Berikut contohnya.

SI Kemahasiswaan   Home   Tambah Mahasiswa   Kelulusan Mahasiswa   Cari Mahasiswa

### Sukses!

Mahasiswa dengan NPM 140971523412 berhasil ditambahkan

### [3] Mengubah Data Mahasiswa

**Initial Page:** /mahasiswa/ubah/{NPM}

**Form Request:** POST, /mahasiswa/ubah/{NPM}

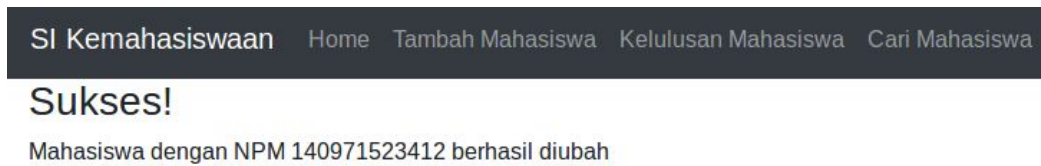
**Response Page:** /mahasiswa/ubah/{NPM} (disarankan menggunakan URL ini tetapi boleh menggunakan yang lain)

**Poin:**

Mengubah data mahasiswa yang sudah ada. *Field input form* pada halaman “/mahasiswa/ubah/{NPM}” harus sudah terisi dengan data sebelumnya saat halaman

“/mahasiswa/ubah/{NPM}” diakses. Jangan lupa aturan NPM. Jika terdapat data yang berubah yang mempengaruhi NPM maka NPM juga harus berubah.

Ketika tombol *submit* di-klik *request* POST akan dikirim. Jika operasi berhasil, kembali ke halaman “/mahasiswa/ubah/{NPM}” dan kali ini tampilkan juga pesan bahwa operasi berhasil dilakukan. Berikut contohnya.



#### [4] Menampilkan Presentase Kelulusan Mahasiswa pada Tahun dan Prodi Tertentu

**Initial Page:** /kelulusan

**Form Request:** GET, /kelulusan?thn={TAHUN}&prodi={ID\_PRODI}

**Response Page:** /kelulusan?thn={TAHUN}&prodi={ID\_PRODI}

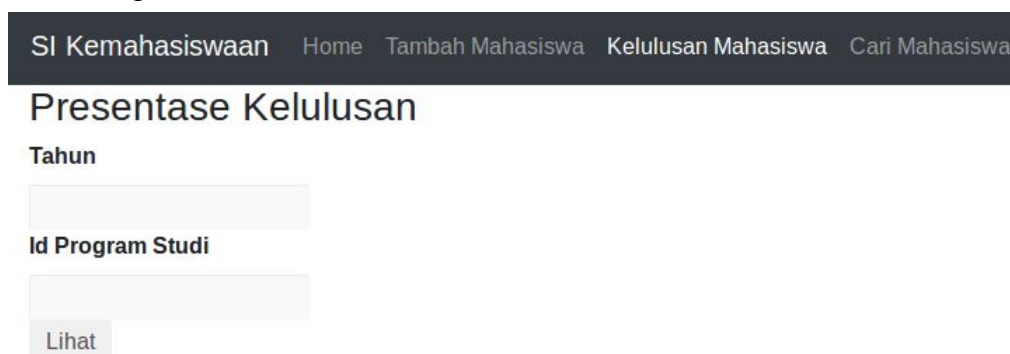
**Poin:**

Pada fitur ini Anda diminta untuk menampilkan presentase kelulusan mahasiswa pada tahun dan prodi tertentu. Pada halaman “/kelulusan” akan terdapat *field input* tahun dan program studi yang keduanya harus diisi. Ketika tombol “lihat” di-klik maka request POST akan dikirim. Jika operasi berhasil, tampilkan informasi presentase kelulusan tersebut beserta dengan jumlah mahasiswa masing-masing status.

Contohnya jika terdapat data

- Mahasiswa Ilmu Komunikasi Univ XYZ 2017 sebanyak 100 orang.
- 80 Mahasiswa telah lulus.
- 5 Mahasiswa telah DO.

Maka akan ditampilkan informasi kelulusan sebesar 80%.



Ketika tombol lihat di-klik request GET akan dikirim. Jika operasi berhasil, tampilkan pesan informasi kelulusan. Berikut contohnya.

SI Kemahasiswaan   Home   Tambah Mahasiswa   Kelulusan Mahasiswa   Cari Mahasiswa		
Presentase Kelulusan		
Tahun	2014	95%
Program Studi	Akuntansi	95 dari 100 Mahasiswa Telah Lulus
Fakultas	Fakultas Ekonomi dan Bisnis	
Universitas	Universitas Indonesia	

## [5] Menampilkan data Mahasiswa Berdasarkan Universitas, Fakultas, dan Program Studi Tertentu

**Initial Page:** /mahasiswa/cari

**Request Method:** GET,

/mahasiswa/cari?univ={ID\_UNIV}&fakultas={ID\_FAKULTAS}&prodi={ID\_PRODI}

**Response Page:**

/mahasiswa/cari?univ={ID\_UNIV}&fakultas={ID\_FAKULTAS}&prodi={ID\_PRODI}

**Poin:**

Fitur yang paling menantang. Pada fitur ini Anda diminta untuk menampilkan daftar semua mahasiswa sesuai dengan *filter* Universitas dan Fakultas dan Program Studi yang dipilih. *Filter* tersebut menggunakan operator AND, sehingga Program Studi yang dipilih pasti Program Studi yang berada di Fakultas yang dipilih. Fakultas yang dipilih pasti Fakultas yang berada di Universitas yang dipilih.

Untuk mempermudah, operasi pencarian ini dilakukan dengan empat tahap.

Tahapan:

1. Memilih Universitas pada halaman “/mahasiswa/cari”

Pada halaman ini terdapat *dropdown* Universitas di Indonesia dan tombol *submit*. Pilih salah satu Universitas kemudian klik *submit*. *Request* akan dikirim. Aplikasi Anda lalu mengembalikan halaman “/mahasiswa/cari?univ={ID\_UNIV}”.

SI Kemahasiswaan	Home	Tambah Mahasiswa	Kelulusan Mahasiswa	Cari Mahasiswa
------------------	------	------------------	---------------------	----------------

### Cari Mahasiswa

Universitas

Pilih Universitas ▼

Cari

- Memilih Fakultas pada halaman “/mahasiswa/cari?univ={ID\_UNIV}”

Pada halaman ini terdapat *dropdown* Fakultas di Universitas yang telah Anda pilih sebelumnya, *dropdown* Universitas yang *read-only* (tidak bisa diubah), dan tombol *submit*. Pilih salah satu Fakultas kemudian klik *submit*. *Request* akan dikirim. Aplikasi Anda lalu mengembalikan halaman

“/mahasiswa/cari?univ={ID\_UNIV}&fakultas={ID\_FAKULTAS}”.

SI Kemahasiswaan   Home   Tambah Mahasiswa   Kelulusan Mahasiswa   Cari Mahasiswa

Cari Mahasiswa

Universitas

Institut Teknologi Bandung ▾

Fakultas

Pilih Fakultas ▾

Cari

- Memilih Program Studi pada halaman “/mahasiswa/cari?univ={ID\_UNIV}&fakultas={ID\_FAKULTAS}”

Pada halaman ini terdapat *dropdown* Program Studi di Fakultas yang telah Anda pilih sebelumnya, *dropdown* Universitas yang *read-only*, *dropdown* Fakultas yang *read-only*, dan tombol *submit*. Pilih salah satu Program Studi kemudian klik *submit*. *Request* akan dikirim. Aplikasi Anda lalu mengembalikan halaman “/mahasiswa/cari?univ={ID\_UNIV}&fakultas={ID\_FAKULTAS}&prodi={ID\_PRODI}”.

SI Kemahasiswaan   Home   Tambah Mahasiswa   Kelulusan Mahasiswa   Cari Mahasiswa

Cari Mahasiswa

Universitas

Institut Teknologi Bandung ▾

Fakultas

Fakultas Seni Rupa dan Desain ▾

Program Studi

Pilih Program Studi ▾

Cari

- Menampilkan semua mahasiswa yang ada pada Program Studi yang dipilih

Pada halaman ini Anda perlu menampilkan semua mahasiswa yang terdaftar pada Program Studi yang dipilih, baik yang aktif, sudah lulus atau sudah *drop out*. Tampilkan data mahasiswa tersebut menggunakan *DataTables*. *DataTables*



mempermudah Anda untuk melakukan pagination. Anda dapat melakukan konfigurasi DataTables pada JavaScript halaman.

SI Kemahasiswaan						
Home   Tambah Mahasiswa   Kelulusan Mahasiswa   Cari Mahasiswa						
Lihat data Mahasiswa Program Studi Teknik Pertambangan Fakultas Teknik Pertambangan dan Perminyakan Institut Teknologi Bandung						
Show 10 entries		Search:				
No.	NPM	Nama	Program Studi	Tahun Masuk	Jalur Masuk	Detail Mahasiswa
1	142012155001	Mr. Rhett Strosin Jr.	Teknik Pertambangan	2014	Undangan Paralel/PPKB	Lihat detail
2	152012162002	Prof. Hailey Block	Teknik Pertambangan	2015	Ujian Tulis Mandiri	Lihat detail
3	142012154003	Sylvester Hessel	Teknik Pertambangan	2014	Undangan Reguler/SNMPTN	Lihat detail
4	172012162004	Henri O'Kon	Teknik Pertambangan	2017	Ujian Tulis Mandiri	Lihat detail
5	162012153005	Crawford Medhurst	Teknik Pertambangan	2016	Undangan Olimpiade	Lihat detail
6	152012153006	Dr. Dalton Block I	Teknik Pertambangan	2015	Undangan Olimpiade	Lihat detail
7	162012153007	Miss Mercedes Parisian	Teknik Pertambangan	2016	Undangan Olimpiade	Lihat detail
8	172012155008	Corene Stamm	Teknik Pertambangan	2017	Undangan Paralel/PPKB	Lihat detail
9	142012154009	Mariam Rath	Teknik Pertambangan	2014	Undangan Reguler/SNMPTN	Lihat detail
10	142012154010	Prof. Jack Donnelly MD	Teknik Pertambangan	2014	Undangan Reguler/SNMPTN	Lihat detail
Showing 1 to 10 of 98 entries			Previous 1 2 3 4 5 ... 10 Next			

Silakan *download* DataTables pada tautan <https://datatables.net/download/> dan lihat dokumentasi penggunaan DataTables pada tautan <https://datatables.net/manual/installation>.

## [6] **\*\*BONUS\*\*** Validasi Input Untuk Semua Form POST

**Fitur Terkait:** [2], [3], [4], [5], [6], [7]

**Poin:**

Melakukan validasi dari sisi backend untuk setiap form POST dan fitur cari. Terdapat beberapa varian validasi misalkan validasi jika input kosong, validasi jika format tidak benar untuk tanggal, validasi jika value tidak ada pada domain, dll.

## [7] **\*\*BONUS\*\*** Menampilkan Mahasiswa Paling Muda dan Paling Tua di Suatu Program Studi Pada Angkatan Tertentu

**Fitur Terkait:** -

**Poin:**

Menampilkan nama, tanggal lahir dan NPM mahasiswa paling muda dan paling tua di suatu program studi pada angkatan tertentu.

## [8] **\*\*BONUS\*\*** Menambahkan Error Page

**Fitur Terkait:** [1], [3], [4]

**Poin:**

Tampilkan error page jika data tidak ditemukan. Misalkan, jika NPM tidak ada, kombinasi NPM salah atau NPM tidak lengkap. Silakan Anda eksplorasi kemungkinan error lainnya. Tampilkan juga error page jika pengguna mengakses link yang tidak ada pada aplikasi.

## [9] **\*\*BONUS\*\*** Fitur-Fitur Lain yang Mendukung Aplikasi

**Fitur Terkait: -**

**Poin:**

Anda juga dibebaskan untuk menambahkan fitur-fitur lain dalam pengerjaan tugas ini. Fitur ini tentunya juga harus relevan dengan sistem yang Anda kembangkan. Contohnya adalah menampilkan mahasiswa termuda dan tertua dari setiap program studi yang ada. **Jelaskan konsep, manfaat, dan cara mengakses fitur lain ini pada file write-up.**

## Optimasi Database

Anda telah diberikan *database* yang sudah berisi data mahasiswa dan perguruan tinggi. Sayangnya *database* tersebut belum dalam kondisi optimal untuk dipakai sistem yang diakses banyak orang dalam waktu bersamaan.

Terapkan optimasi struktur dan properti dari *database* yang Anda dapatkan agar eksekusi *query* dapat lebih cepat. **Tuliskan optimasi apa saja yang Anda lakukan terhadap database tersebut pada file writeup.**

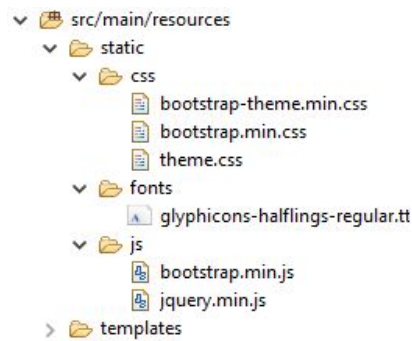
## Stress Testing

Bapak Kemendikbud menginginkan sistem ini dapat diakses tanpa batasan. Oleh karena itu, sistem perlu memiliki sifat *reliable*, dapat diandalkan. Anda perlu melakukan *stress testing* untuk memastikan sistem tetap responsif saat diakses oleh ribuan orang.

Gunakan Apache JMeter. Anda dipersilakan untuk bereksperimen dengan variabel-variabel yang digunakan misalkan *Number of Threads*, *Ramp-up Periode*, dll. **Jelaskan eksperimen Anda dan hasil serta analisis stress testing Anda, apakah cepat/lambat, reliable/not reliable, dan aspek lainnya pada write-up tugas.**

## Tampilan

Anda direkomendasikan menggunakan **framework Bootstrap, dan JQuery**. Anda boleh menggunakan framework/library CSS dan JS lainnya. Tampilan yang diberikan pada soal ini hanyalah contoh menggunakan *template* dasar bootstrap.



*Hanya contoh stuktur folder static. Sesuaikan dengan kebutuhan Anda*

Dalam project Spring Boot, data-data seperti CSS, Image, Javascript, dll. dimasukkan dalam folder **resources/static/**.

Anda dapat menambahkan fitur lain seperti menambahkan validasi di *front-end* atau menggunakan *template* lain sesuai dengan kreasi Anda. **Penambahan fitur tersebut akan dimasukkan dalam poin ‘Bonus Lain-lain’.** Jangan lupa jelaskan fitur bonus yang Anda kembangkan tersebut pada **write-up**.

## Deliverables

Deliverables untuk tugas ini adalah:

### 1. File Project

Buat sebuah project baru pada organization APAP-Ekstensi-2018 dengan format nama tugas1\_NPM, contoh tugas1\_1501234567. Push project Anda ke repository GitHub tersebut.

### 2. Write-up

Buat sebuah file write-up. Jelaskan proses Anda mengembangkan tugas ini. Cantumkan penjelasan Anda terhadap hal-hal yang kami minta dimasukkan pada write-up pada soal diatas. Jika Anda mengembangkan fitur lain selain fitur-fitur yang telah dijelaskan diatas, cantumkan pula penjelasan fitur tersebut.

Kami rekomendasikan Anda menggunakan format pdf agar dapat lebih leluasa dalam menuangkan penjelasan Anda dengan dukungan screenshot dan kode. Masukkan file writeup ke *folder project*. Pastikan file write-up juga di-push ke *repository*.

## Penilaian

Poin penilaian:

### 1. [70] Fitur Aplikasi

- [14] Tampilkan Data Mahasiswa Berdasarkan NPM
- [12] Menambahkan Mahasiswa Baru di Suatu Program Studi
- [12] Mengubah Data Mahasiswa

- d. [12] Menampilkan Presentase Kelulusan
  - e. [20] Tampilkan Data Mahasiswa Berdasarkan Universitas, Fakultas, dan Program Studi Tertentu
2. [20] Penerapan konsep MVC dan layering
3. [10] Optimasi Database dan Stress Testing
4. [20] **\*\*BONUS\*\***
  - a. [10] **\*\*BONUS\*\*** Validasi Input Untuk Semua Form POST
  - b. [3] **\*\*BONUS\*\*** Menampilkan Mahasiswa Paling Muda dan Paling Tua di Suatu Program Studi pada Angkatan Tertentu
  - c. [2] **\*\*BONUS\*\*** Menambahkan Error Page
  - d. [5] **\*\*BONUS\*\*** Fitur lain-lain yang mendukung aplikasi

Total nilai maksimal adalah 120.

## Tips

Berikut beberapa tips dari kami dalam mengerjakan tugas 1 ini:

- **Baca dan analisis soal sejak awal**  
Skala tugas 1 terlalu besar jika dikerjakan dalam waktu yang singkat dekat dengan deadline. Sebelum memulai *coding*, pastikan Anda paham betul dengan semua fitur dan deliverables yang diminta dari soal. Ekstrak dan lakukan analisis terhadap soal sejak awal.
- **Lakukan diskusi soal dengan teman Anda**  
Anda diperbolehkan melakukan diskusi soal dan fitur dengan teman Anda yang lain. Pastikan Anda mengerti dan paham. Namun diskusi soal hanya sebatas bertukar pikiran untuk memperdalam pemahaman Anda. Segala bentuk kecurangan yang dilakukan dalam tugas 1 akan tim asisten dan tim dosen tindak lanjuti.
- **Baca soal dengan teliti, pahami setiap penjelasan**  
Soal tugas kami rancang secara mendetail untuk mempermudah Anda dalam melakukan analisis dan memahami fitur yang diminta. Kesalahan sering terjadi karena Anda tidak membaca dengan teliti.
- **Ingat kembali konsep OOP dan beberapa Class/Interface pada Java**  
Beberapa *interface* cukup berguna bagi pengerjaan tugas ini misalkan *interface Comparable* untuk membandingkan antar Model Class. Kemudian konsep mengimplementasikan *interface* berarti *implements method-method* dalam *interface* tersebut ke *class* yang *implements interface* tersebut. Memahami kembali konsep OOP akan membantu Anda mengerjakan tugas lebih lancar.
- **Ingat kembali sintaks-sintaks HTML dan Thymeleaf**  
Beberapa fitur membutuhkan fitur HTML dan Thymeleaf yang cukup dalam. Kami rekomendasikan Anda untuk menyegarkan kembali ingatan Anda terhadap sintaks-sintaks HTML dan Thymeleaf dari tutorial-tutorial sebelumnya maupun dari

sumber lain. Misalkan, bagaimana mengirim GET *request* melalui *form*, bagaimana menggunakan if pada Thymeleaf, dll.

- **Gunakan sebagian data saja untuk tahap pengembangan**

Untuk mempercepat proses pengerjaan fitur dan kemudahan debugging, kami sarankan Anda untuk menggunakan sebagian data saja dari *database* saat pengembangan. Silakan kloning sebagian data dari *database* yang kami berikan, *generate* data, atau gunakan cara lain sesuai preferensi Anda. Setelah pengembangan, gunakan data sesungguhnya untuk mencoba fitur-fitur Anda juga untuk melakukan *stress testing*.

- **Bertanya di Scele, jangan kontak pribadi**

Tugas 1 ini berskala besar dan cukup lama pengerjaannya. Berdasarkan pengalaman, mahasiswa sering menemukan kebingungan atau pun pertanyaan seputar soal dan bertanya kepada asisten berulang kali. Gunakan Scele untuk bertanya. Dengan menggunakan Scele, asisten dapat memberikan jawaban yang lebih akurat karena kami semua mengetahui apa yang Anda tanyakan dan kami dapat melakukan diskusi mendalam terlebih dahulu. Selain itu, bertanya di Scele juga turut membantu teman-teman Anda yang mungkin memiliki permasalahan yang sama.

## **Deadline**

**25 Maret 2018, 23:59:59**

## **Penalti**

Penalti:

- **Keterlambatan**

Penalti sebesar -10 poin akan ditambahkan setiap 10 menit keterlambatan