



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Selama masa menjalani kerja magang pada departemen IT di Universitas Multimedia Nusantara, dilakukan dengan mengambil jabatan sebagai *programmer*, dimana memiliki tanggungjawab dalam melakukan merancang aplikasi hingga sampai tahap pembangunan aplikasi dibahas arahan supervisi oleh Bapak Danryray selaku IT Development Coordinator, yang memiliki tugas mengatur dan membimbing dari mulai bertemu client, melakukan perancangan, dan pembangunan aplikasi di IT UMN.

#### **3.2 Tugas yang Dilakukan**

Dalam masa pelaksanaan, tanggung jawab yang dilakukan dalam pemenuhan kerja magang ini adalah sebagai berikut:

- Merancang alur sistem aplikasi.
- Membangun aplikasi dalam bentuk mobile.
- Membangun sistem *backend*, untuk dapat mengambil data yang diperlukan dari database, untuk disediakan pada aplikasi mobile.

### 3.3 Uraian Pelaksanaan

#### 3.3.1 Proses Pelaksanaan

Dalam pelaksanaannya, yang dilakukan dalam 8 minggu, dalam memenuhi kerja magang, berikut *timeline* kerja yang dilakukan selama masa magang.

Tabel 1 Timeline Kegiatan

Nama Kegiatan	Minggu pelaksanaan ke-							
	1	2	3	4	5	6	7	8
1. Pembahasan aplikasi dengan client	■							
2. Pembuatan Business Requirement	■							
3. Pembuatan Flowchart	■							
4. Pengenalan <i>Framework</i>	■	■	■	■	■			
5. Pembuatan ERD, dan Use Case				■	■			
6. Pembuatan Mockup aplikasi				■	■			
7. Pembuatan Functional Specification				■	■	■		
8. Pembuatan <i>Backend API</i> aplikasi						■	■	■
9. Pembuatan <i>Frontend</i> aplikasi							■	■

#### 3.3.2 Database dan Framework yang digunakan

Dalam minggu pertama sejak dimulai magang, waktu digunakan untuk mempelajari *framework* yang akan digunakan, sehingga ketika saatnya mulai melakukan masa pembangunan aplikasi telah siap dengan *framework* yang digunakan. Selama masa itu digunakan untuk mengenal Oracle Database, Yii2 PHP Framework, Ionic 4 Framework yang digunakan sebagai kebutuhan dasar dalam pemenuhan permintaan aplikasi berbasis mobile yang diajukan oleh *client*.

Oracle Database, adalah basis data yang digunakan oleh UMN dalam menunjang segala kebutuhan yang dalam penampungan seluruh transaksi data yang terjadi dalam lingkungan kerja UMN, dalam penggunaannya Oracle Database yang

digunakan adalah versi 12c ini dapat memenuhi segala kebutuhan *enterprise* dan pemenuhan kebutuhan akademik yang ada.

Yii2 adalah *framework* PHP yang digunakan dalam mengatur dan membangun sistem informasi yang baik, mudah dibangun, serta mudah digunakan, dalam dokumentasinya pun, sudah terdapat dalam bahasa Indonesia, yang membuat dalam masa pembelajaran *framework* menggunakan Yii2 menjadi lebih cepat ketika masuk ke dalam masa pembangunan aplikasi atau sistem informasi.

Ionic 4 adalah Javascript Framework yang membantu pengembang aplikasi agar dapat membangun aplikasi *mobile* dengan sensasi mirip dengan yang aslinya dengan memanfaatkan teknologi web seperti HTML, CSS, dan Javascript, yang juga terdapat Angular dalam *framework* ini. Penggunaan Ionic 4 ini juga memudahkan pengembang, karena aplikasi yang dibuat bukan hanya dapat menjadi aplikasi yang dibuka melalui *web browser*, melainkan juga dapat langsung langsung digunakan sebagai aplikasi untuk sistem operasi Android pada beragam merk *smartphone* terkini, maupun iOS pada *smartphone* keluaran perusahaan Apple.

### **3.3.3 Pembangunan Sistem**

#### **A. Gambaran Singkat Aplikasi**

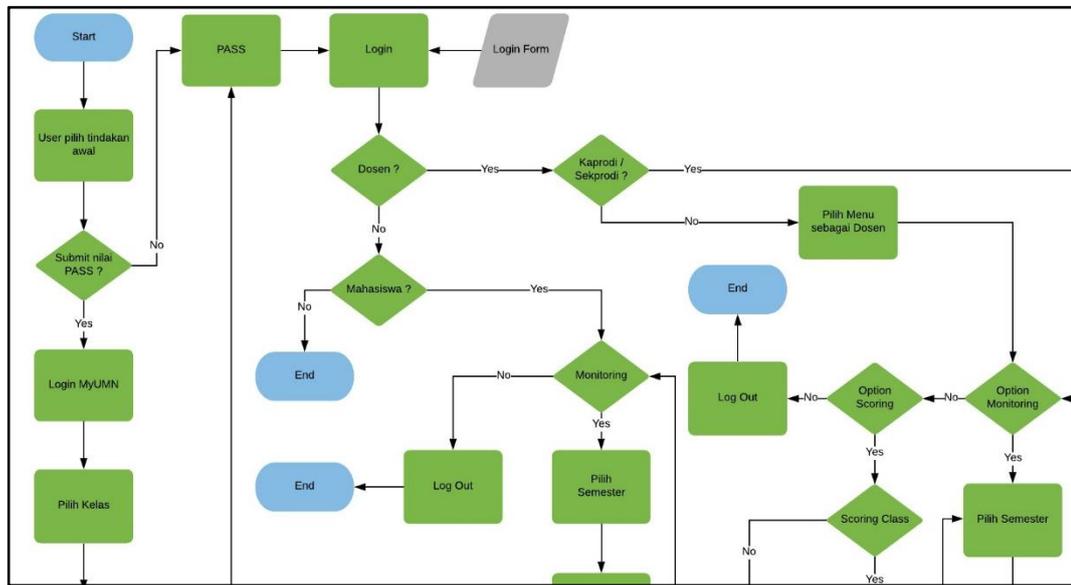
Pada aplikasi yang dibangun dibagi menjadi 2 sistem, pertama merupakan sistem yang digunakan untuk menyediakan dan mengirim data dari basis data yang diperlukan kepada aplikasi yang lainnya. Kedua, aplikasi yang dapat digunakan langsung oleh user dalam mengakses data yang diinginkan sesuai dengan data yang telah disediakan oleh aplikasi sebelumnya.

Aplikasi nantinya dapat diakses menggunakan SSO, setelah berhasil masuk menggunakan SSO, maka apabila *user* atau pengguna yang masuk merupakan kepala program studi yang disebut kaprodi, atau sekretaris program studi yang disebut sekprodi, maka akan dapat mengakses aplikasi dengan beragam fitur, yaitu mengakses master score, untuk dapat melihat, membuat, merubah, bahkan menghapusnya. Master score berguna sebagai penilaian yang akan diterima mahasiswa nantinya. Dalam membuat master score, diperlukan sebuah kategori, yang disebut *master category*, master category berguna untuk memisah – misahkan score berdasarkan kategori yang akan didaftarkan, sebagai kaprodi ataupun sekprodi, memiliki akses dapat melihat, membuat, mengubah, dan menghapus kategori yang telah dibuat, kaprodi dan sekprodi juga memiliki akses untuk dapat melihat seluruh nilai mahasiswa yang telah dibuat oleh seluruh dosen, untuk penilaian mahasiswa.

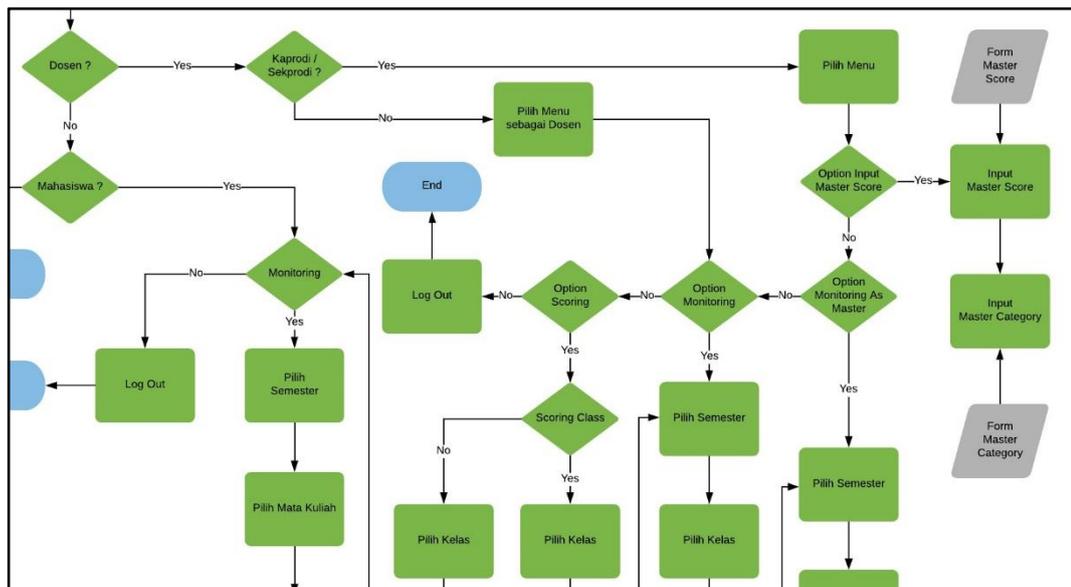
Kaprodi dan sekprodi memiliki akses yang sama seperti dosen biasa, yaitu *input score* atau *scoring*, dan melakukan *monitoring* atau pemantauan, hanya saja pada monitoring ini, tidak dapat mengakses seluruh kelas, melainkan mengakses ke detail penilaian yang hanya kelas yang diajarkannya saja.

Sedangkan pada mahasiswa, ketika mahasiswa masuk ke dalam aplikasi, langsung disajikan monitoring, terhadap kelas yang diambilnya. Dan mahasiswa juga memiliki fitur untuk mendapat notifikasi apabila mahasiswa mendapat pengurangan poin, atau *minus point*.

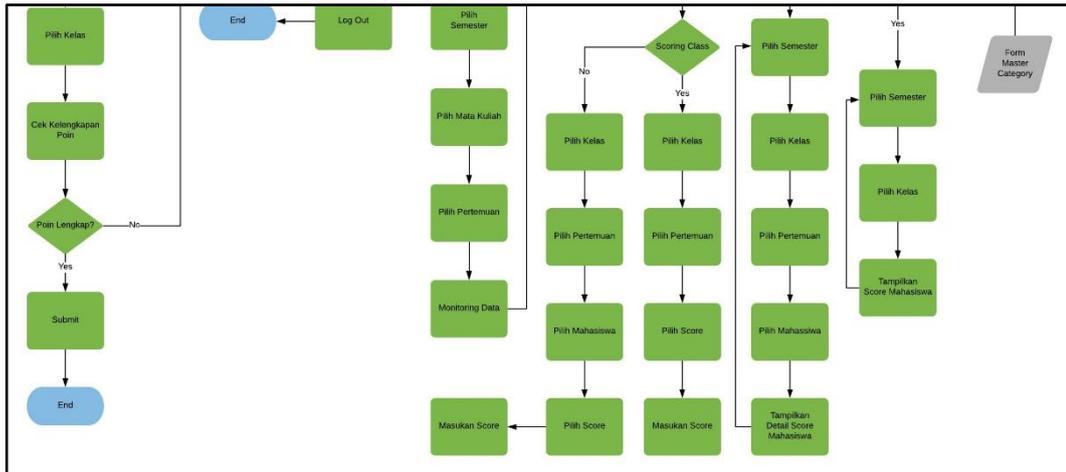
## B. Flowchart



Gambar 3. 1 Bagian Pertama Flowchart

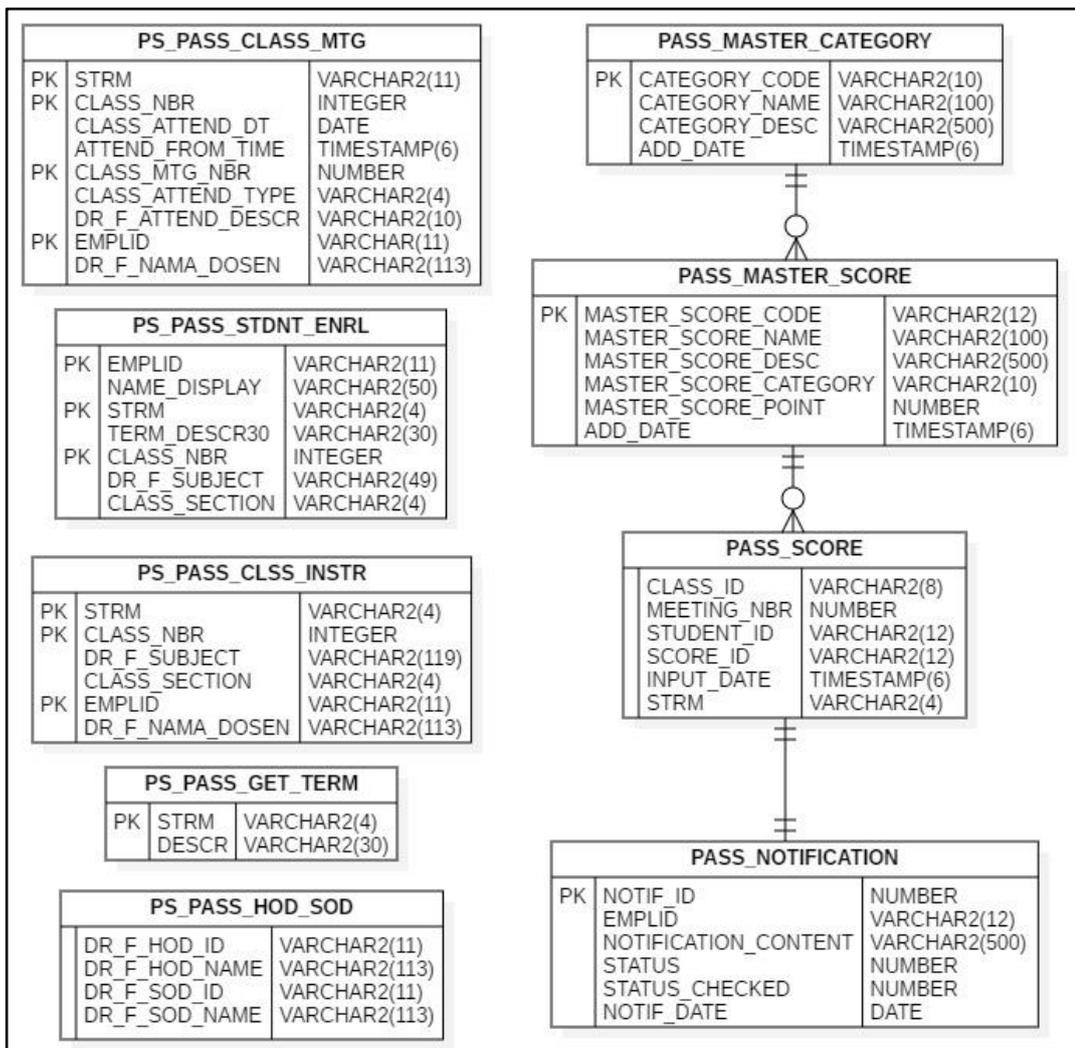


Gambar 3. 2 Bagian Kedua Flowchart



Gambar 3. 3 Bagian Ketiga Flowchart

### C. Struktur Tabel



Gambar 3. 4 Skema Table

## D. Aplikasi Back-End

Dalam menampilkan aplikasi backend, akan dijelaskan dengan menampilkan daftar API yang telah dibuat dan dapat diakses, pada bagian ini bermaksud untuk menjelaskan bagaimana API itu digunakan, mulai dari cara mengakses dengan *method* dan *body request* yang diperlukan hingga menuliskan response yang muncul nantinya. Pada tahap pengujian kali ini, API yang digunakan di *hosting* pada `api.umn.ac.id`, dan setiap api berada dalam 1 *base url* yang sama nantinya, yaitu pada `api.umn.ac.id/umn_pass/web`, sehingga pada nantinya hanya akan ditampilkan sisa potongan alamat yang digunakan.

### 1. /term

Method: GET

Digunakan untuk mengambil semester ajaran yang saat ini berlaku.

Response:

- a. STRM, Berisi format id untuk semester yang digunakan
- b. DESCR, Deskripsi full dari semester yang digunakan

### 2. /department

Method: POST

Digunakan untuk mengetahui dosen yang sedang login merupakan kaprodi atau sekprodi.

Request

- a. EMPLID, NIK yang akan diuji.

Response

- a. DR\_F\_HOD\_ID, NIK Kaprodi saat ini.
- b. DR\_F\_HOD\_NAME, Nama Kaprodi saat ini.

- c. DR\_F\_SOD\_ID, NIK Sekprodi saat ini.
- d. DR\_F\_SOD\_NAME, Nama Kaprodi saat ini.

3. /master-score

Method: POST

Digunakan untuk menampilkan seluruh master score.

Response

- a. MASTER\_SCORE\_CODE, ID untuk master score.
- b. MASTER\_SCORE\_NAME, Nama untuk master score
- c. MASTER\_SCORE\_DESC, Deskripsi untuk master score.
- d. MASTER\_SCORE\_CATEGORY, Kategori untuk master score.
- e. MASTER\_SCORE\_POINT, Besaran poin master score.

4. /master-score/create

Method: POST

Digunakan untuk menambahkan poin master score

Request

- a. MASTER\_SCORE\_NAME, Nama untuk master score.
- b. MASTER\_SCORE\_DESC, Deskripsi untuk master score.
- c. MASTER\_SCORE\_CATEGORY, Kategori untuk master score.
- d. MASTER\_SCORE\_POINT, Besaran poin untuk sebuah master score.

5. /master-score/get

Method: POST

Digunakan untuk mengambil data 1 master score.

Request

- a. MST\_SCR\_CODE, ID untuk master score.

Response

- a. MASTER\_SCORE\_NAME, Nama untuk master score
- b. MASTER\_SCORE\_DESC, Deskripsi untuk master score.
- c. MASTER\_SCORE\_CATEGORY, Kategori untuk master score.
- d. MASTER\_SCORE\_POINT, Besaran poin master score.

6. /master-score/edit

Method: POST

Digunakan untuk mengubah master score.

Request

- a. MST\_SCR\_CODE, ID untuk master score.
- b. MASTER\_SCORE\_NAME, Nama untuk master score
- c. MASTER\_SCORE\_DESC, Deskripsi untuk master score.
- d. MASTER\_SCORE\_CATEGORY, Kategori untuk master score.
- e. MASTER\_SCORE\_POINT, Besaran poin master score.

7. /master-score/delete

Method: POST

Digunakan untuk menghapus sebuah master score.

Request

- a. MST\_SCR\_CODE, ID untuk master score.

8. /master-category

Method: POST

Digunakan untuk menampilkan semua master category.

Response

- a. CATEGORY\_CODE, ID untuk master category.
- b. CATEGORY\_NAME, Nama untuk master category.
- c. CATEGORY\_DESC, Deskripsi untuk master category.
- d. ADD\_DATE, Waktu dibuatnya master category.

9. /master-category/create

Method: POST

Digunakan untuk membuat master category.

Request

- a. CATEGORY\_NAME, Nama untuk master category.
- b. CATEGORY\_DESC, Deskripsi untuk master category.

10. /master-category/edit

Method: POST

Digunakan untuk mengubah master category.

Request

- a. CATEGORY\_CODE, ID untuk master category.
- b. CATEGORY\_NAME, Nama untuk master category.
- c. CATEGORY\_DESC, Deskripsi untuk master category.

11. /master-category/edit

Method: POST

Digunakan untuk mengubah master category.

Request

- a. CATEGORY\_CODE, ID untuk master category.
- b. CATEGORY\_NAME, Nama untuk master category.
- c. CATEGORY\_DESC, Deskripsi untuk master category.\

12. /master-category/get

Method: POST

Digunakan untuk menampilkan satu buah master category.

Request

- a. MST\_CAT\_CODE, ID untuk master category.

Response

- a. MST\_CAT\_CODE, ID untuk master category.
- b. MST\_CAT\_NAME, Nama untuk master category.
- c. MST\_CAT\_DESC, Deskripsi untuk master category.
- d. ADD\_DATE, Waktu pembuatan master category.

13. /master-category/delete

Method: POST

Digunakan untuk menghapus sebuah master category.

Request

- a. CATEGORY\_CODE, ID untuk master category.

14. /meeting

Method: POST

Digunakan untuk melihat daftar pertemuan dari kelas yang dicari.

Request

- a. CLASS\_NBR, ID untuk kelas.
- b. STRM, Semester saat ini.

Response

- a. CLASS\_MTG\_NBR, ID nomor pertemuan kelas.
- b. DR\_F\_ATTEND\_DESCR, Deskripsi pertemuan kelas.

15. /student/search

Method: POST

Digunakan untuk mencari mahasiswa yang diajarkan oleh dosen yang diinginkan.

Request

- a. KEYWORD, Kata pencarian yang diinginkan.
- b. LECT\_ID, ID dosen yang digunakan.
- c. STRM, Semester yang diinginkan.

Response

- a. EMPLID, ID mahasiswa.
- b. NAME\_DISPLAY, Nama mahasiswa.

16. /subject

Method: POST

Digunakan untuk menampilkan seluruh kelas berdasarkan semester.

Request

- a. STRM, Semester yang diinginkan.

Response

- a. CLASS\_NBR, ID kelas yang ditampilkan.
- b. DR\_F\_SUBJECT, Nama kelas yang ditampilkan.
- c. CLASS\_SECTION, Section kelas yang ditampilkan.

17. /subject/search-class

Method: POST

Digunakan untuk mencari tahu kelas dari mahasiswa dan dosen.

Request

- a. lecturerId, ID dosen yang mencari.
- b. studentId, ID mahasiswa yang dicari.
- c. STRM, Semester yang diinginkan.

Response

- a. CLASS\_NBR, ID kelas yang ditampilkan.
- b. DR\_F\_SUBJECT, Nama kelas yang ditampilkan.
- c. CLASS\_SECTION, Section kelas yang ditampilkan.

18. /score/create

Method: POST

Digunakan untuk memasukan / melakukan input score.

Request

- a. CLASS\_NBR, ID kelas yang diinginkan.
- b. MTG\_NBR, ID pertemuan yang diinginkan.
- c. EMPLID, ID mahasiswa.
- d. SCORE\_ID, ID score yang didapat oleh mahasiswa.
- e. STRM, Semester pada saat melakukan scoring.

19. /score/class

Method: POST

Digunakan untuk melakukan input score terhadap satu kelas.

Request

- a. CLASS\_NBR, ID kelas yang diinginkan.
- b. MTG\_NBR, ID pertemuan yang diinginkan.

- c. SCORE\_ID, ID score yang didapat oleh mahasiswa
- d. STRM, Semester pada saat melakukan scoring.

20. /score/total-score

Method: POST

Digunakan untuk mendapatkan total score dari seorang mahasiswa.

Request

- a. CLASS\_NBR, ID kelas yang diinginkan.
- b. MEETING\_NBR, ID pertemuan yang diinginkan
- c. EMPLID, ID mahasiswa.
- d. STRM, Semester pada saat penilaian.

Response

- a. TOTAL\_SCORE\_POINT, Total keseluruhan nilai didapat.

21. /score/total-score-class

Method: POST

Digunakan untuk menampilkan seluruh penilaian dalam satu kelas.

Request

- a. CLASS\_NBR, ID kelas yang diinginkan.
- b. MTG\_NBR, ID pertemuan yang diinginkan.
- c. EMPLID, ID mahasiswa.
- d. SCORE\_ID, ID score didapat mahasiswa.
- e. STRM, Semester pada saat scoring.

Response

- a. STUDENT\_ID, ID mahasiswa pada kelas.
- b. TOTAL\_SCORE\_POINT, Total point score.

22. /score/delete

Method: POST

Digunakan untuk menghapus sebuah score yang telah dimasukkan.

Request

- a. ID, ID score yang diinginkan.
- b. STRM, Semester score yang diinginkan.

23. /monitoring/semester

Method: GET

Digunakan pada monitoring untuk melihat daftar seluruh semester.

Response

- a. STRM, Semester score yang diinginkan
- b. TERM\_DESCR30, Deskripsi dari semester.

24. /monitoring/lecturer-class

Method: POST

Digunakan pada monitoring untuk melihat kelas diajar oleh dosen.

Request

- a. STRM, Semester score yang diinginkan.
- b. STUDENT\_ID, ID dosen yang diinginkan.

Response

- a. CLASS\_NBR, ID kelas yang diajar.
- b. DR\_F\_SUBJECT, Nama kelas yang diajar.
- c. CLASS\_SECTION, Section kelas yang diajar.

25. /monitoring/student-class

Method: POST

Digunakan pada monitoring untuk melihat kelas diambil mahasiswa.

Request

- a. STRM, Semester score yang diinginkan.
- b. EMPLID, ID mahasiswa yang diinginkan.

Response

- a. CLASS\_NBR, ID kelas yang diambil.
- b. DR\_F\_SUBJECT, Nama kelas yang diambil.
- c. CLASS\_SECTION, Section kelas yang diambil.

26. /monitoring/student

Method: POST

Digunakan pada monitoring untuk mahasiswa dalam kelas dipilih.

Request

- a. STRM, Semester score yang diinginkan.
- b. EMPLID, ID mahasiswa yang diinginkan.

Response

- a. CLASS\_NBR, ID kelas yang diambil.
- b. DR\_F\_SUBJECT, Nama kelas yang diambil.
- c. CLASS\_SECTION, Section kelas yang diambil.

27. /monitoring/lecturer-student

Method: POST

Digunakan pada monitoring untuk daftar mahasiswa pada kelas tersebut.

Request

- a. STRM, Semester score yang diinginkan.
- b. CLASS\_NBR, ID kelas yang diinginkan.

Response

- a. EMPLID, ID mahasiswa dalam kelas.
- b. NAME\_DISPLAY, Nama mahasiswa dalam kelas.

28. /monitoring/score

Method: POST

Digunakan pada master monitoring untuk detail nilai mahasiswa.

Request

- a. STRM, Semester score yang diinginkan.
- b. CLASS\_ID, ID kelas.
- c. MEETING\_NBR, ID pertemuan kelas.
- d. STUDENT\_ID, ID mahasiswa yang diinginkan.

29. /monitoring/master-class

Method: POST

Digunakan pada master monitoring untuk melihat seluruh kelas.

Request

- a. STRM, Semester score yang diinginkan.

Response

- a. CLASS\_NBR, ID kelas yang diinginkan.
- b. DR\_F\_SUBJECT, Nama kelas yang diinginkan.
- c. CLASS\_SECTION, Section kelas yang diinginkan.

30. /monitoring/master-score

Method: POST

Digunakan pada master monitoring untuk melihat mahasiswa dan nilainya pada kelas.

Request

- a. STRM, Semester score yang diinginkan.
- b. CLASS\_NBR, ID kelas yang diinginkan.

Response

- a. NAME\_DISPLAY, Nama mahasiswa dalam kelas.
- b. SCORE, Total Score yang didapat mahasiswa.

31. /notification/view

Method: POST

Digunakan pada notifikasi untuk dapat melihat isi notifikasi.

Request

- a. EMPLID, ID mahasiswa yang diinginkan.

Response

- a. NOTIFICATION\_CONTENT, Nama mahasiswa dalam kelas.
- b. STATUS\_CHECKED, Status notifikasi sudah dibaca.
- c. NOTIF\_DATE, Waktu notifikasi masuk.

32. /notification/bad-grade

Method: POST

Digunakan untuk memasukan notifikasi ketika mahasiswa mendapat pengurangan nilai.

Request

- a. EMPLID, ID mahasiswa yang diinginkan.
- b. SUBJECT, Mata kuliah yang diinginkan.
- c. SCORING\_POINT, Poin tindakan dari master score.
- d. SCORING\_ACTION, Nama tindakan dari master score.

33. /notification/count

Method: POST

Digunakan untuk memberitahu jumlah notifikasi yang didapat.

Request

- a. EMPLID, ID mahasiswa yang diinginkan.

Response

- a. NTF, jumlah notifikasi yang diterima.

34. /notification/check

Method: POST

Digunakan untuk merubah status notifikasi telah dibaca.

Request

- a. EMPLID, ID mahasiswa yang diinginkan.

35. /notification/bad-grade-class

Method: POST

Digunakan untuk memasukkan notifikasi ketika pengurangan poin untuk satu kelas

Request

- a. STRM, Semester point dimasukkan.
- b. CLASS\_NBR, ID kelas yang diinginkan.
- c. SUBJECT, Mata kuliah yang diinginkan.
- d. SCORING\_POINT, Poin tindakan dari master score.
- e. SCORING\_ACTION, Nama tindakan dari master score.

36. /notification/clear

Method: POST

Digunakan untuk menghapus notifikasi yang telah dibaca.

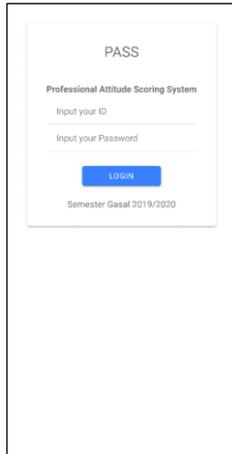
Request

- a. EMPLID, ID mahasiswa yang diinginkan.

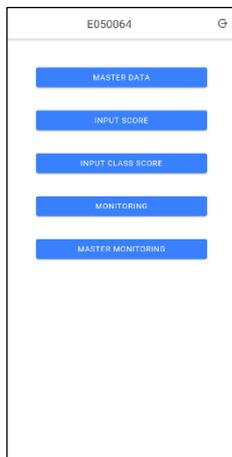
#### **E. Simulasi Aplikasi Front-End dan Pemanfaatan Back-End**

Pada pemanfaatan API akan dapat langsung ditampilkan dengan baik, apabila ditampilkan bersamaan dengan simulasi yang aplikasi yang digunakan user, sehingga API yang telah dibuat dapat terlihat bahwa dapat berjalan dengan baik.

Dalam simulasinya akan menjelaskan kapan dan bagaimana digunakannya API yang telah dibuat sebelumnya, menunjukkan kegunaan dan memaksimalkan semua API yang telah dibuat sehingga dapat berguna.



Gambar 3. 5 Halaman awal



Gambar 3. 6 Halaman Menu



Gambar 3. 7 Halaman Master Score

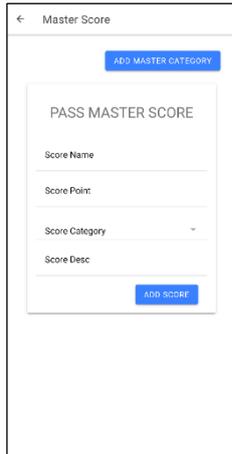
E.1 Pada saat aplikasi mulai dijalankan, maka akan membuka halaman utama untuk melakukan login, pada halaman utama terdapat pula ditampilkannya semester yang sedang berjalan saat ini.

API yang digunakan: 1. /term.

E.2 Setelah login, akan ada 3 jenis pengguna yang dapat login yaitu, kaprodi atau sekprodi, dosen, dan mahasiswa.

E.3 Dalam Master Score berguna untuk menampilkan daftar score yang berlaku.

API yang digunakan: /master-score

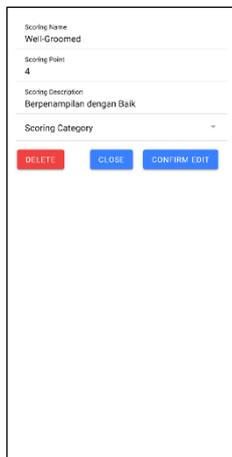


Gambar 3. 8 Tambah Master Score

E.4 Apabila ingin menambahkan Master Score maka dapat masuk Add Master Score, untuk dapat menambahkan daftar Master Score.

API yang digunakan:

1. /master-score/create
2. /master-category

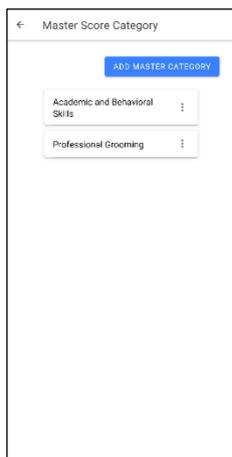


Gambar 3. 9 Ubah Score

E.5 Master Score dapat melakukan perubahan apabila telah dibuat, untuk menyesuaikan apabila ada kesalahan. Dan dapat menghapus apabila score tersebut, sudah tidak digunakan.

API yang digunakan:

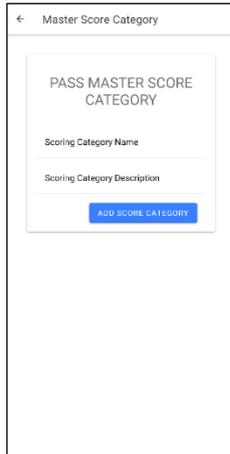
1. /master-score/delete
2. /master-score/edit
3. /master-category



Gambar 3. 10 Master Category

E.6 Master Category dapat menampilkan daftar category yang akan digunakan untuk pengelompokan score berdasarkan kategori.

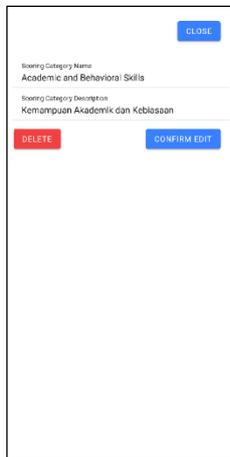
API yang digunakan: /master-category



Gambar 3. 11 Add Master Category

E.7 Daftar Master Category dapat ditambahkan untuk memperbanyak kategori yang diperlukan.

API yang digunakan: `/master-category/create`

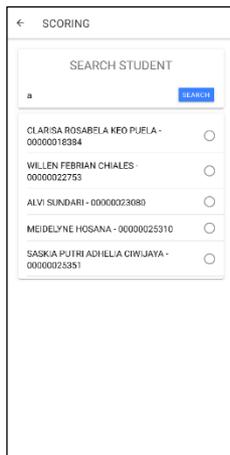


Gambar 3. 12 Edit Master Category

E.8 Master Category yang telah dibuat dapat diubah atau dihapus.

API yang digunakan:

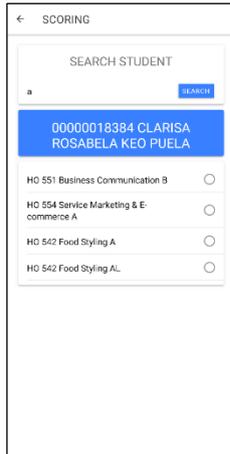
1. `/master-category/edit`
2. `/master-category/delete`



Gambar 3. 13 Input Score seach nama

E.9 Dosen dan Kaprodi atau sekprodi dapat input score hanya dapat mencari mahasiswa yang diajarkannya menggunakan nama atau nim mahasiswa tersebut sebagai kata kunci pencarian.

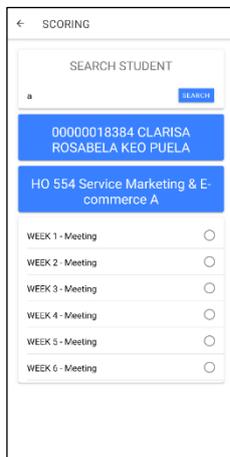
API yang digunakan: `/student/search`



Gambar 3. 14 Pilih Kelas

E.10 User sebagai dosen, setelah menginput nama, maka akan memilih dan daftar yang muncul dari kelas yang dimiliki oleh mahasiswa.

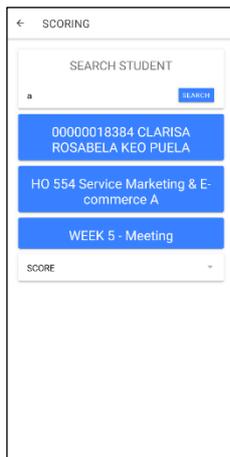
API yang digunakan: /subject/search-class/



Gambar 3. 15 Pilih Pertemuan

E.11 Ketika kelas sudah dipilih, dosen dapat memilih pertemuan yang dapat dimasukkan ketika pelanggaran dilakukan.

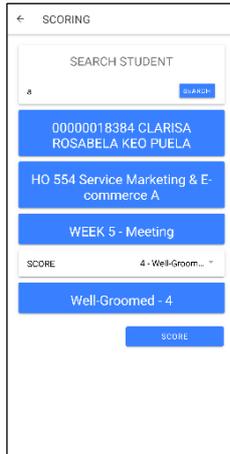
API yang digunakan: /meeting



Gambar 3. 16 Pilih Score

E.12 Setelah beberapa data yang diperlukan sudah diisi maka kita dapat memilih score apa saja yang dapat ditentukan untuk mahasiswa tersebut.

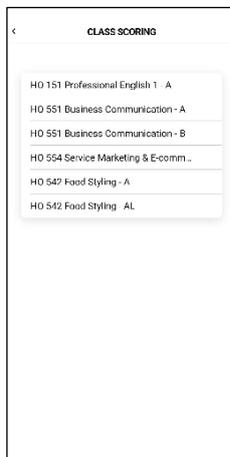
API yang digunakan: /master-score



Gambar 3. 17 Submit Score

E.13 Setelah memilih score yang dibutuhkan, dapat submit score yang dipilih sesuai

API yang digunakan: /score/create



Gambar 3. 18 Input Class Scoring

E.14 sebagai pengajar, dosen, kaprodi atau sekprodi, dapat memberikan penilaian langsung kepada 1 kelas yang sedang diajarnya pada semester itu.

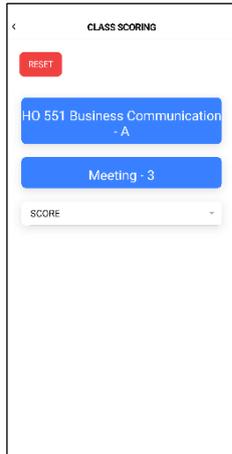
API yang digunakan: /lecturer/class



Gambar 3. 19 Pilih Pertemuan Scoring

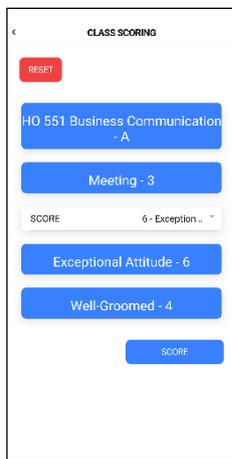
E.15 setelah melakukan pemilihan kelas, dosen memilih pertemuan yang sedang berjalan.

API yang digunakan: /meeting



Gambar 3. 20 Input Score untuk Kelas

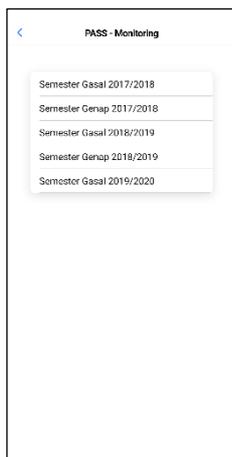
E.16 Setelah memilih kelas dan pertemuan yang dijalankan, dapat memasukkan score API yang digunakan: /master-score



Gambar 3. 21 Submit Score untuk kelas

E.17 Setelah semua score yang ingin ditentukan sudah dipilih, dapat melakukan submit penilaian yang nantinya akan dikenakan oleh seluruh mahasiswa pada kelas tersebut.

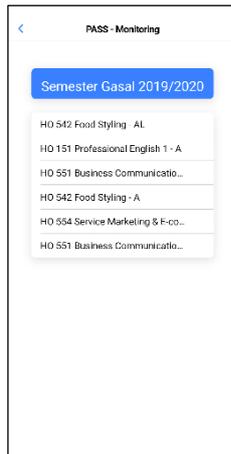
API yang digunakan: /score/create



Gambar 3. 22 Pilih Semester

E.18 Dosen, kaprodi atau sekprodi dapat melakukan monitoring, dengan pertama memasukkan semester yang diinginkan.

API yang digunakan: /monitoring/semester



Gambar 3. 23 Pilih Kelas

E.19 Dosen, kaprodi atau sekprodi, dapat memasukkan kelas berdasarkan semester yang dipilih.

API yang digunakan: /lecturer-class



Gambar 3. 24 Pilih Pertemuan

E.20 Setelah kelas dipilih, diperlukan untuk memilih pertemuan.

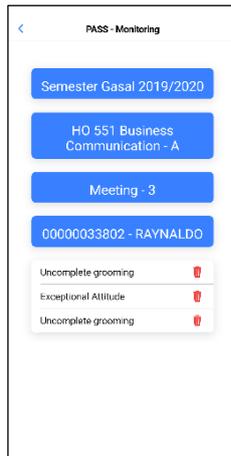
API yang digunakan: /meeting



Gambar 3. 25 Pilih Mahasiswa

E.21 Nama mahasiswa akan ditampilkan setelah pertemuan ditentukan.

API yang digunakan: /monitoring/lecturer-student



Gambar 3. 26 Tampilkan Score

E.22 Dosen dapat melihat score apa saja yang didapat mahasiswa dan dapat menghapusnya apabila score itu salah atau tidak tepat.

API yang digunakan:

1. /monitoring/score
2. /score/delete



Gambar 3. 27 Master Monitoring

E.23 Pada menu Master Monitoring yang terdapat pada kaprodi atau sekprodi, yang dimulai juga pada pencarian semester.

API yang digunakan: /monitoring/semester



Gambar 3. 28 Pilih Kelas

E.24 Setelah pemilihan semester, akan ditampilkan yang daftar kelas yang berlaku pada semester tersebut.

API yang digunakan: /monitoring/master-class

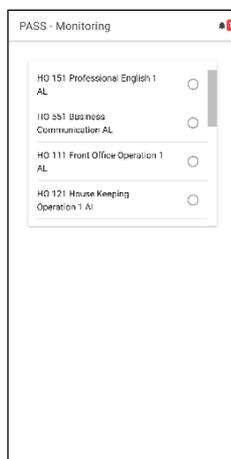


Gambar 3. 29 Tampil semua Mahasiswa

E.25 Setelah kelas berhasil dipilih maka akan ditampilkan mahasiswa dengan total akhir yang didapat pada mata kuliah tersebut.

API yang digunakan:

/monitoring/master-score

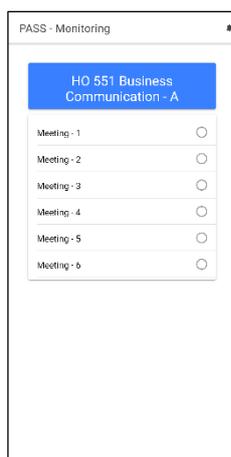


Gambar 3. 30 Student Monitoring

E.26 Mahasiswa dapat mengecek seputar nilai yang didapat olehnya hanya berdasarkan kelas yang diambil saat semester itu saja. Dan disertakan juga notifikasi apabila ada pengurangan poin.

API yang digunakan:

1. /monitoring/student-class
2. /notification/count



Gambar 3. 31 Pilih Pertemuan

E.27 Setelah mahasiswa memilih kelas, mahasiswa dapat memilih pertemuan

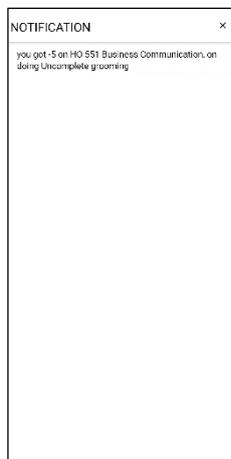
API yang digunakan: /meeting



Gambar 3. 32 Tampilkan Penilaian

E.28 Setelah kelas dipilih, maka akan ditampilkan score apa saja yang sudah didapat mahasiswa tersebut.

API yang digunakan: /monitoring/score



Gambar 3. 33 Notifikasi Mahasiswa

E.29 Mahasiswa dapat mengecek notifikasi untuk setiap pengurangan poin yang didapat mahasiswa.

API yang digunakan:

1. /notification/view
2. /notification/check

### 3.4 Kendala dan Solusi

#### 3.4.1 Kendala yang Ditemukan

Selama proses berjalannya magang, kendala yang umum yang sering muncul berada pada pengalaman dalam pengaplikasian framework yang masih baru, hal ini membuat pengembangan aplikasi banyak terhambat, terlebih lagi dengan komunitas yang masih belum terlalu aktif dan banyak perubahan versi dari *framework* yang digunakan, menyebabkan pada pembuatannya terkendala secara teknis.

### **3.4.2 Solusi atas Kendala yang Ditemukan**

Solusi atas kendala yang ditemukan adalah dengan belajar melalui kursus online berbayar untuk mempercepat pengembangan ilmu pengetahuan dan wawasan dalam pengaplikasian menggunakan *framework* Ionic tersebut. Adapun dalam penyelesaian masalah meminta bantuan dosen yang berwawasan dalam mata kuliah tersebut, atau bertanya solusi alternatif dalam penyelesaian kendala yang ada.