

PERANCANGAN SISTEM PENGGAJIAN PENGAJAR DENGAN MENGGUNAKAN FRAMEWORK ANGULAR DAN CODEIGNITER (STUDI KASUS MABIT NURUL FIKRI)

Anggi Fergian Pratama¹, Intan Hesti Indriana², Nur Hafifah Matondang³

Universitas Pembangunan Nasional "Veteran" Jakarta

Jl. RS. Fatmawati, Pondok Labu, Jakarta Selatan, 12450, Indonesia.

e-mail: anggyfergian56@gmail.com¹, aslihesti@gmail.com², nurhafifahmatondang@yahoo.com³

Abstrak— Sebagai organisasi pendidikan yang banyak melakukan kegiatan administrasi, Mabit Nurul Fikri belum mengimplementasi manajemen sistem informasi sehingga data-data yang ada belum saling terintegrasi. Tujuan penelitian ini adalah untuk membangun sistem informasi pengajian pengajar di Mabit Nurul Fikri guna meningkatkan pengelolaan administrasi yang efisien. Penulis menggunakan Waterfall sebagai metode perancangan sistem. Penulis juga menggunakan diagram UML untuk memodelkan sistem juga sebagai kerangka berfikir dalam perancangan berbasis objek. Penelitian ini menggunakan framework PIECES untuk menganalisis masalah pada sistem saat ini. Sistem informasi manajemen pengajian pengajar ini dibangun menggunakan CodeIgniter, Angular, dan MySQL. Ada 2 jenis aktor pada sistem ini yaitu Bendahara dan Kepala Departemen. Bendahara memiliki hak akses ke semua *use case* dalam sistem, sedangkan Kepala Departemen tidak. Hasil dari penelitian ini yaitu berupa dashboard internal Mabit Nurul Fikri yang dapat menyimpan data pengajar, presensi, dan insentif, menghitung gaji pengajar, serta mengunduh dokumen insentif dalam bentuk PDF.

Kata kunci: Sistem Informasi Manajemen, Pengajian, Dashboard, Waterfall

1. Pendahuluan

Sistem informasi sangat berkembang pesat akhir-akhir ini. Internet sebagai salah satu bagian dari sistem informasi sudah banyak dimanfaatkan oleh banyak orang. Salah satu peranan internet adalah sebagai *tools* dalam pengembangan perangkat lunak. Pengembangan perangkat lunak menghasilkan produk yang umumnya berbentuk web, aplikasi desktop, dan aplikasi mobile. Disisi lain pengembangan perangkat lunak juga merupakan serangkaian proses yang kompleks. Untuk mengatasi proses yang berulang, maka para pembembang perangkat lunak memutuskan untuk mengembangkan sebuah teknologi yang dinamakan *framework*. Menurut Mara Destiningrum (2017), *framework* merupakan sekumpulan instruksi yang didefinisikan didalam *class* maupun *function* yang bertujuan untuk memudahkan *developer* dalam memanggil program ini tanpa perlu menuliskan kembali syntax yang sama.

Salah satu *framework* yang digunakan untuk membangun tampilan antarmuka web adalah Angular. Angular merupakan sebuah *framework* JavaScript yang digunakan untuk membangun *Single Page Application* (SPA). Menurut Poetri Lestari (2018), SPA merupakan aplikasi web yang dapat diperbarui secara mandiri tanpa melakukan *refresh* atau *reload* sehingga seluruh halaman tidak perlu dimuat ulang setiap *event* atau tindakan yang dilakukan oleh user. Dengan hal ini, aplikasi yang menerapkan SPA memiliki 2 keuntungan utama yaitu mengurangi *bandwith* jaringan dan proses navigasi yang lebih cepat.

CodeIgniter juga merupakan salah satu *framework* yang digunakan dalam pengembangan web khususnya untuk membangun Restful API. CodeIgniter banyak digunakan untuk membangun *backend* yaitu sebuah layanan web yang erat kaitannya dengan server dan database. Dengan membangun Restful API, ini memungkinkan bagi layanan web satu dan lainnya saling berinteraksi dengan menggunakan protokol HTTP. CodeIgniter memiliki beberapa keunggulan diataranya yaitu memiliki struktur sederhana, ukuran kecil, dan dokumentasi yang lengkap.

Mabit Nurul Fikri sebagai salah satu penyelenggaraan bimbingan belajar saat ini belum menerapkan sistem pengajian pengajar secara online. Meskipun perhitungan sudah menggunakan aplikasi Excel, namun proses pencatatan data masih dilakukan secara konvensional yaitu dengan mengumpulkan berkas-berkas presensi pengajar yang dibuat oleh kepala departemen setiap mata pelajaran. Sehingga menyebabkan proses pembagian gaji yang memakan waktu cukup lama. Selain itu nominal gaji yang didapat oleh pengajar banyak yang tidak sesuai karena berkas-berkas presensi pengajar kerap kali tidak utuh dan bahkan hilang. Karena itu, diperlukan

sistem penggajian pengajar sehingga memudahkan pengurus organisasi dalam proses penggajian pengajar dan juga proses pencatatan presensi pengajar.

Berdasarkan survey yang dilakukan oleh situs Stackoverflow pada tahun 2020, Angular menjadi web *framework* paling populer nomor 3. Oleh karena itu, implementasi Angular sebagai *framework frontend*, Code Igniter sebagai *framework backend*, MySQL sebagai manajemen basisdata, dan *waterfall* sebagai metode pengembangan sistem menjadi langkah yang tepat untuk membangun sistem penggajian pengajar untuk menyelesaikan permasalahan pada Mabit Nurul Fikri.

Penelitian ini didasarkan pada aktivitas yang biasanya berlangsung di lapangan. Oleh karena itu fitur yang dikembangkan juga memiliki *requirement* yang disesuaikan dengan kebutuhan, salah satunya yaitu hak akses yang berbeda-beda pada setiap user yang ada di dalam sistem. Setidaknya ada jenis pengguna, yaitu bendahara dan kepala departemen. Seperti halnya dalam sistem yang berjalan, bendahara dan kepala departemen memiliki peran dan akses masing-masing dalam proses penggajian pengajar. Sistem penggajian pengajar nantinya memiliki beberapa fitur untuk memudahkan pengurus Mabit Nurul Fikri yaitu pengelolaan data-data pengajar dan pencetakan laporan.

2. Tinjauan Pustaka

2.1 Angular

Angular merupakan sebuah *framework* yang digunakan untuk membangun aplikasi mobile dan web. *Framework* ini dikembangkan oleh tim Google dan komunitas *open source* Angular. Angular dibangun menggunakan bahasa pemrograman TypeScript yang merupakan super set dari bahasa pemrograman JavaScript. Angular menggunakan konsep SPA (*single page application*) yang berarti komponen-komponen pada aplikasi akan di-*render* terlebih dahulu ke dalam *browser* sehingga pengguna tidak perlu melakukan *page load* untuk membuka halaman baru.

2.2 CodeIgniter

CodeIgniter merupakan sebuah *framework* PHP yang banyak digunakan untuk membangun aplikasi web. CodeIgniter menggunakan konsep MVC (*model-view-controller*) yang merupakan sebuah teknik pemrograman yang memisahkan antara *logic*, *data*, dan *presentation*. Karena hal ini, CodeIgniter cocok digunakan dalam membangun Restful API yaitu sebuah web service yang menggunakan protokol HTTP untuk pertukaran data. Restful API akan menjadi jembatan komunikasi data antara server dan client. Sehingga, server tidak lagi mengirimkan HTML sebagai response, melainkan hanya data.

2.3 Restful API

API merupakan sekumpulan perintah untuk mendefinisikan bagaimana aplikasi atau perangkat dapat terhubung dan berkomunikasi satu sama lain. Sedangkan, Restful API merupakan desain arsitektur dari API yang menggunakan protokol HTTP untuk mengakses data yang sering dimanfaatkan dalam pengembangan *web services*. Data ini kemudian dapat diakses melalui metode GET, POST, PUT, dan DELETE (Alexander Gillis, 2020).

3. Metode Penelitian

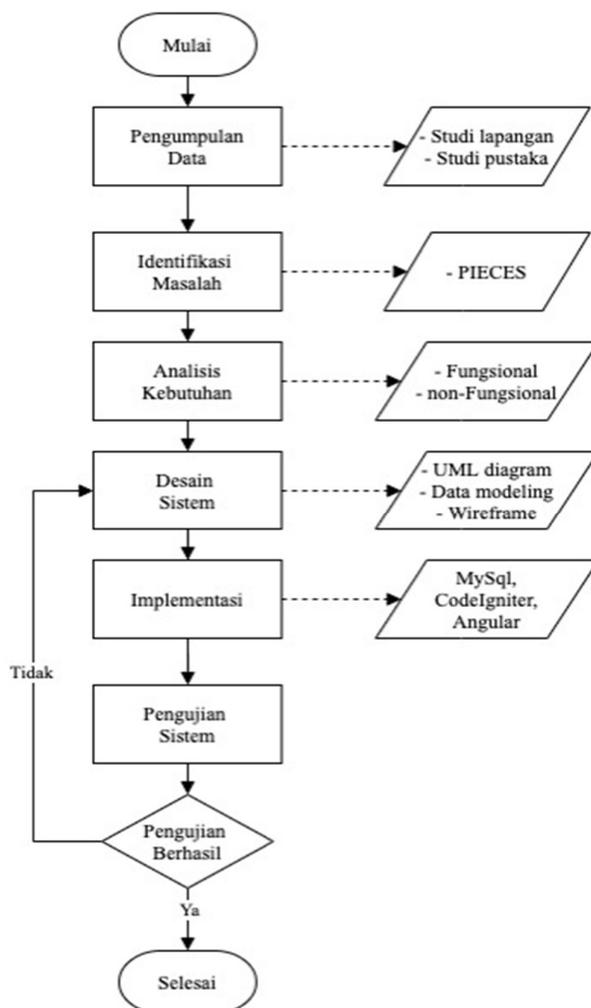
3.1 Metode Pengumpulan Data

Dalam penelitian ini ada 2 tahap untuk mengumpulkan data yaitu studi lapangan dan studi literatur. Studi lapangan yaitu meliputi pengamatan langsung dan sesi wawancara dengan stakeholder yang ada di Mabit Nurul Fikri. Sedangkan studi pustaka meliputi kegiatan mengumpulkan data dari buku, artikel, dan karya ilmiah.

3.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah *waterfall*. Tahap pertama adalah menganalisis kebutuhan sistem, baik kebutuhan fungsional maupun kebutuhan fungsional. Tahap kedua adalah mendesain sistem yang meliputi pemodelan database, pemodelan sistem menggunakan diagram UML, dan merancang wireframe. Tahap ketiga adalah implementasi atau pengkodean. Pada tahap ini, penulis merancang basis data berdasarkan diagram ERD, selain itu pada tahap ini juga dilakukan perancangan frontend dan backend. Tahap terakhir yaitu pengujian sistem menggunakan *black-box testing*.

3.3 Kerangka Berpikir



Gambar 1. Kerangka Berpikir

4. Hasil dan Pembahasan

4.1 Tinjauan Organisasi

Mabit Nurul Fikri merupakan organisasi yang bergerak di bidang pendidikan yang berada di bawah pengawasan Bimbingan dan Konsultasi Belajar Nurul Fikri. Adapun tujuan utama usaha Mabit Nurul Fikri adalah menyiarkan dakwah islam melalui jalur pendidikan dan membangun sumber daya muslim sesuai dengan perkembangan ilmu pengetahuan dan tuntutan zaman. Kegiatan utama Mabit Nurul Fikri adalah kegiatan belajar mengajar. Kepengurusan Mabit Nurul Fikri akan berganti setiap tahun ajaran baru dan anggotanya merupakan orang-orang yang menjadi peserta didik di tahun ajaran sebelumnya. Setiap tahunnya Mabit Nurul Fikri berhasil membantu peserta didiknya untuk masuk ke perguruan tinggi negeri.

4.2 Analisis Sistem Berjalan

Tabel 1. PIECES tabel

| No | Analisis Masalah | Sistem Lama | Sistem Baru |
|----|--------------------|--|--|
| 1 | <i>Performance</i> | Meskipun penghitungan sudah menggunakan Microsoft Excel, namun pengumpulan data masih dilakukan secara manual | Proses penghitungan dan pengumpulan data sudah terintegrasi |
| 2 | <i>Information</i> | Informasi data penggajian pengajar hanya bisa dilihat oleh bendahara | Kepala departemen memiliki akses melihat daftar insentif pengajar |
| 3 | <i>Economy</i> | Adanya biaya untuk mencetak tabel presensi dan diperlukan waktu yang lama untuk mengumpulkan data | Presensi dapat langsung di input ke dalam sistem dan juga dapat segera dicetak |
| 4 | <i>Control</i> | LPJ yang memuat data presensi pengajar yang dibuat oleh kepala departemen rentan hilang | Kepala departemen bisa menginput langsung data presensi di sistem yang baru |
| 5 | <i>Efficient</i> | Bendahara harus mengumpulkan semua LPJ dari setiap kepala departemen, kemudian mencocokkan dengan data pengajar | Bendahara hanya perlu untuk menyetujui presensi yang dibuat oleh kepala departemen untuk menghitung gaji pengajar |
| 6 | <i>Services</i> | Prosedur pengelolaan pengajar, pendataan presensi dan penghitungan gaji pengajar tidak terintegrasi satu sama lain | Semua prosedur yang ada terintegrasi satu dengan yang lain yang dapat membuat alur kerja menjadi lebih terstruktur |

4.3 Analisis Kebutuhan

a. Kebutuhan Fungsional

Tabel 2. Kebutuhan Fungsional

| No | Kode Fungsional | Deskripsi |
|----|-----------------|---|
| 1 | KF-01 | Sistem menyediakan fitur login untuk mengotorisasi user. |
| 2 | KF-02 | Sistem menyediakan fitur memasukkan data pengajar baru. |
| 3 | KF-03 | Sistem menyediakan fitur mengedit data pengajar. |
| 4 | KF-04 | Sistem menyediakan fitur melihat detail informasi pengajar. |
| 5 | KF-05 | Sistem menyediakan fitur memverifikasi data pengajar baru. |
| 6 | KF-06 | Sistem menyediakan fitur memasukkan data presensi baru. |
| 7 | KF-07 | Sistem menyediakan fitur mengedit data presensi. |
| 8 | KF-08 | Sistem menyediakan fitur melihat detail informasi presensi. |

| | | |
|----|-------|---|
| 9 | KF-09 | Sistem menyediakan fitur <i>me-accept</i> dan <i>me-reject</i> data presensi. |
| 10 | KF-10 | Sistem menyediakan fitur melihat detail informasi gaji pengajar. |
| 11 | KF-11 | Sistem menyediakan fitur mencetak laporan gaji semua pengajar. |

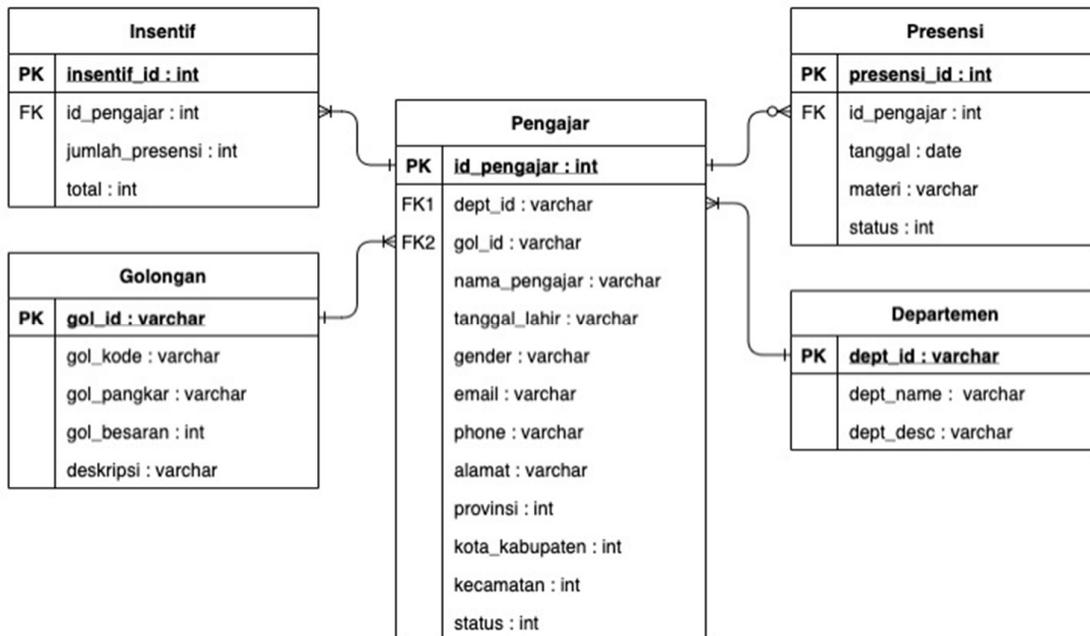
b. Kebutuhan non-Fungsional

Tabel 3. Kebutuhan Non-Fungsional

| No | Kode Non-Fungsional | Deskripsi |
|----|---------------------|---|
| 1 | KNF-01 | Aplikasi menyediakan tampilan antarmuka yang <i>user-friendly</i> dan <i>responsive</i> . |
| 2 | KNF-02 | Aplikasi menyediakan hak akses yang jelas, salah satunya adalah yang dapat <i>me-accept</i> dan <i>me-reject</i> data presensi hanya Bendahara. |
| 3 | KNF-03 | Aplikasi memiliki waktu respons yang baik |

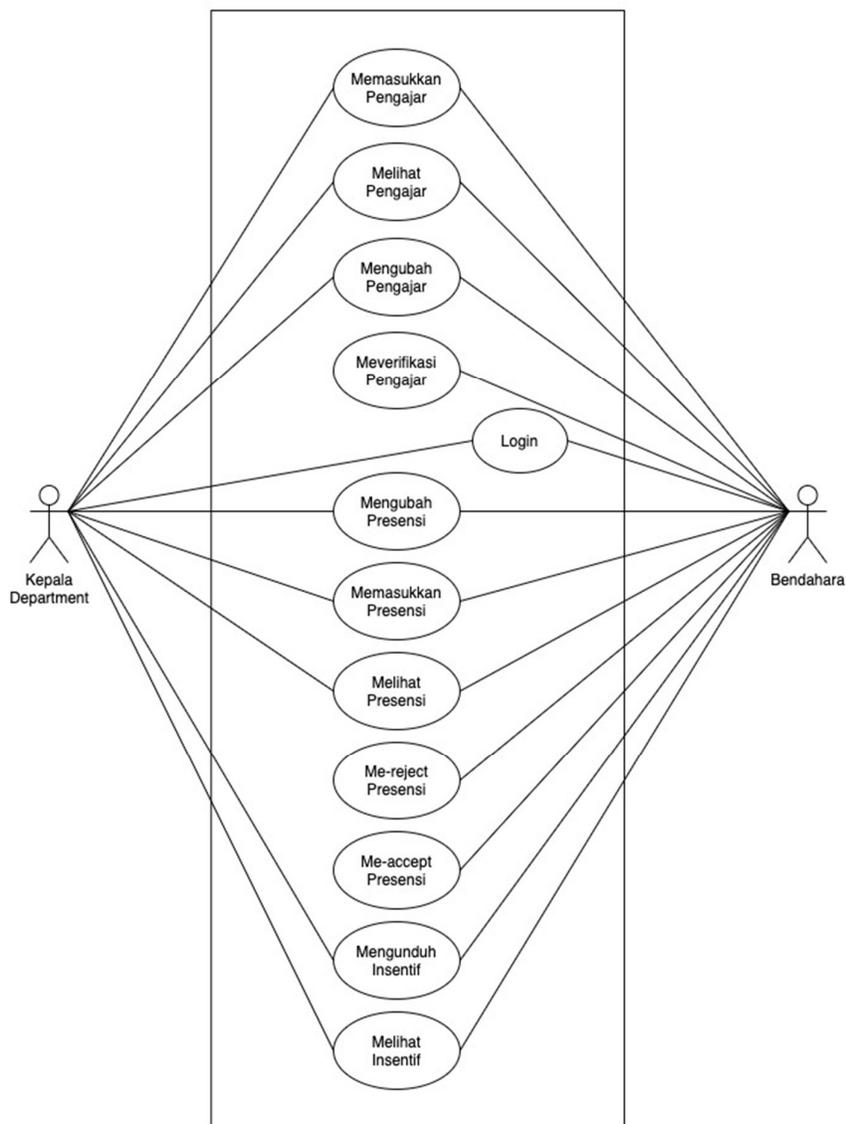
4.4 Pemodelan Sistem

Ada 5 tabel yang terlibat pada perancangan basis data sistem penggajian pengajar Mabit Nurul Fikri. Penulis menggunakan *entity relational diagram* untuk melakukan pemodelan data. Diagram ini digunakan untuk menggambarkan tabel yang akan dibangun beserta kardinalitasnya. Penulis juga menambahkan tipe data pada setiap atribut untuk membantu mendefinisikan tabel pada tahap pengkodean.



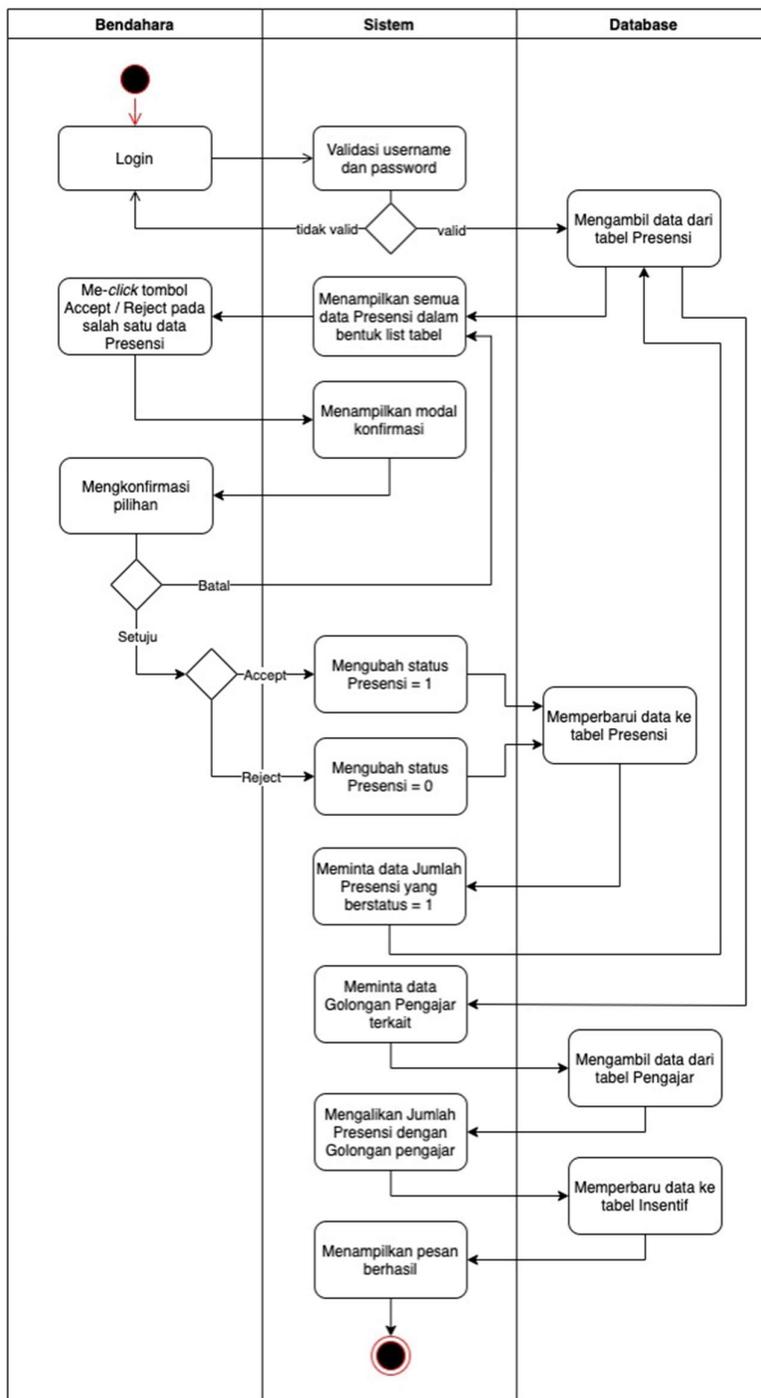
Gambar 2. Entity Relational Diagram

Dari hasil analisis kebutuhan fungsional dan non-fungsional, penulis melakukan pemodelan sistem menggunakan diagram UML. Berikut adalah *Use Case Diagram* yang menggambarkan aktor-aktor yang terlibat dengan kebutuhan-kebutuhan fungsional sistem penggajian pengajar:



Gambar 3. Use Case Diagram

Dari diagram diatas didapat bahwa ada 2 aktor yang terlibat dalam sistem penggajian pengajar. Diagram ini juga menggambarkan ada 12 use case yang akan dikembangkan. Setiap use case tersebut akan dikembangkan lagi ke dalam activity diagram. Pengelolaan presensi merupakan kebutuhan fungsional utama pada sistem penggajian pengajar Mabit Nurul Fikri. Berikut adalah activity diagram Me-accept Presensi dan Mereject Presensi yang memodelkan kebutuhan pengelolaan presensi:

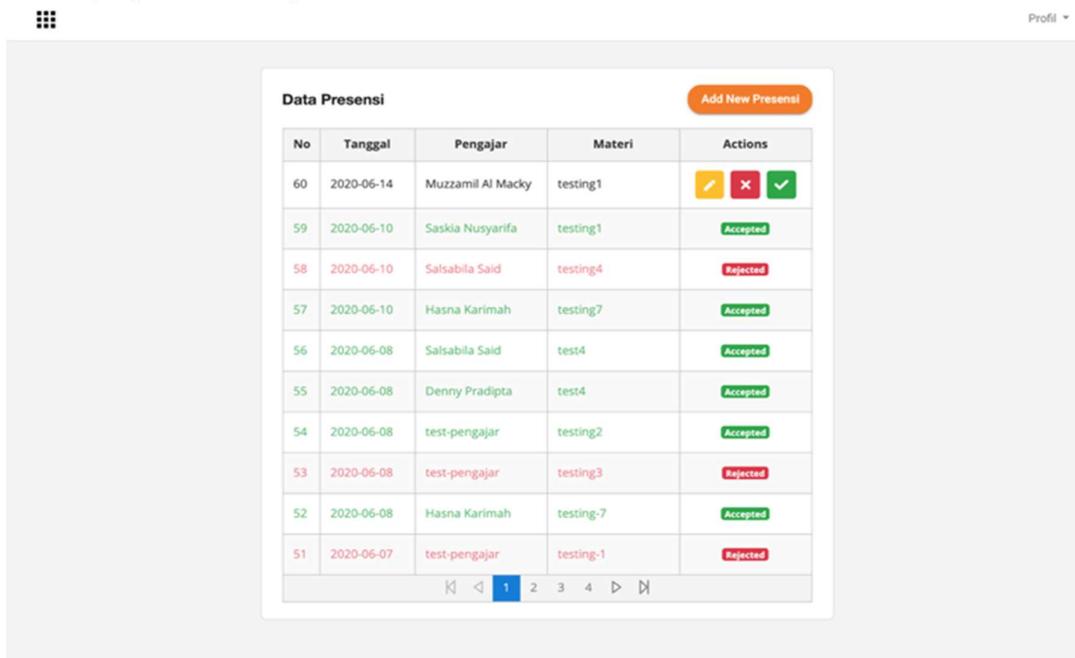


Gambar 4. Activity Diagram Me-accept Presensi dan Me-reject Presensi

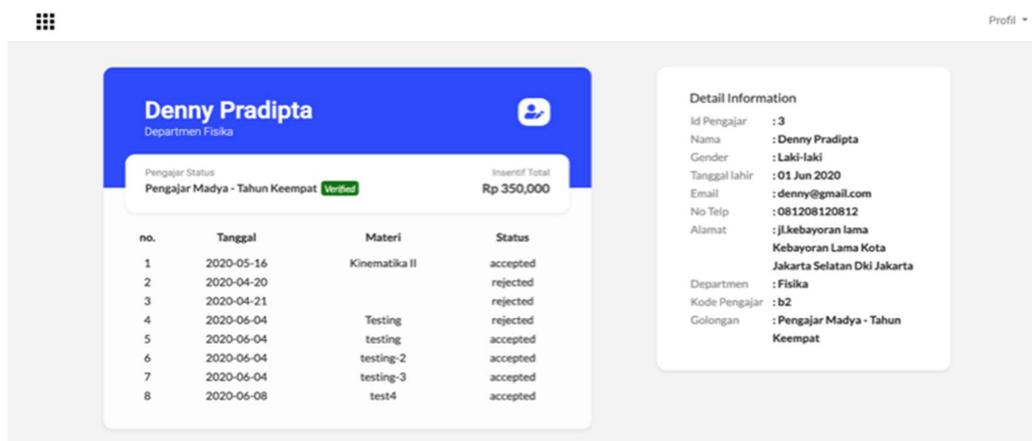
4.5 Implementasi

Pada tahap implementasi, ada 3 hal yang dilakukan. Pertama yaitu merancang tabel-tabel yang dibutuhkan pada basis data. Dari ERD yang ada, total ada 5 tabel yang akan dibuat. Penelitian menggunakan Mysql sebagai manajemen basis data. Hal kedua yang dilakukan adalah membuat Rest Api menggunakan framework CodeIgniter. Ada 2 kegiatan yang dilakukan pada perancangan Rest Api yaitu membuat model berisi *query* untuk manipulasi

data ke basis data dan membuat *controller* untuk menangani permintaan data dari sisi *frontend*. Hal ketiga yang dilakukan adalah perancangan *frontend* menggunakan Angular. Pada perancangan *frontend*, penulis menggunakan rest api yang telah dibuat untuk sebagai layanan untuk melakukan transfer data. Berikut adalah beberapa tampilan antarmuka yang telah dirancang:



Gambar 5. User Interface Daftar Pengajar



Gambar 6. User Interface Lihat Pengajar

4.6 Pengujian Sistem

Tahap pengujian sistem bertujuan untuk mengevaluasi sistem yang telah dibangun, apakah sesuai dengan hasil yang diharapkan atau tidak. Berikut adalah salah satu hasil pengujian yang penulis lakukan pada kebutuhan fungsional kelola pengajar:

Tabel 4. Pengujian Kelola Pengajar

| No | Skenario | Test Case | Hasil yang diharapkan | Hasil Pengujian | Kesimpulan |
|----|---|---|---|-----------------|------------|
| 1 | Membuka halaman form <i>Create Pengajar</i> | Klik tombol <i>Add New Presensi</i> | Sistem akan mengarahkan ke halaman form create pengajar | Sesuai Harapan | Valid |
| 2 | Memvalidasi input form jika kosong | Mengosongkan semua kolom input pada form <i>Create Pengajar</i> | Tombol Simpan tidak bisa diklik dan ada peringatan pada kolom yang belum diisi atau belum valid | Sesuai Harapan | Valid |
| 3 | Menyimpan data pengajar baru | Mengisi semua kolom input dengan data pengajar | Tombol Simpan bisa di klik, kemudian sistem mengarahkan ke halaman daftar pengajar, dan muncul pesan berhasil | Sesuai Harapan | Valid |

5. Kesimpulan

Dari penelitian yang telah dilakukan dapat diambil kesimpulan sistem informasi penggajian pengajar sebagai berikut:

1. Sistem penggajian pengajar memiliki 3 fitur utama yaitu penyimpanan data, penghitungan gaji, dan pencetakan laporan. Fitur-fitur ini diharapkan dapat membantu pengurus organisasi dalam hal meminimalisir hilangnya dokumen dan kesalahan dalam penghitungan keuangan.
2. Sistem penggajian pengajar juga telah mengimplementasi fitur *role-based user* dimana ada 2 aktor didalam sistem yaitu bendahara dan kepala departemen. Kedua aktor memiliki peran yang berbeda didalam sistem. Fitur ini membantu mengurangi pekerjaan administratif sehingga secara tidak langsung dapat memaksimalkan kinerja operasional di lapangan.

Referensi

- [1] Dennis, A., Wixom, B. H., & Roth, R. M. (2018). *Systems Analysis and Design*. John wiley & sons.
- [2] Destiningrum, M. (2017). *Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework CodeIgniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)*. Jurnal TeknoInfo.
- [3] Gillis, A. (2020). *Rest API (Restful API)*. Diakses via <https://searchapparchitecture.techtarget.com/definition/RESTful-API>
- [4] Lestari, P. (2018). *Pengembangan Single Page Application Pada Sistem Informasi Akademik*. ILKOM Jurnal Ilmiah.
- [5] Martin. (2018). *7 Basic Software Development Life Cycles (SDLC) Methodologies: Which One is the Best?*. Diakses via <https://www.cleverism.com/software-development-life-cycle-sdlc-methodologies/>
- [6] Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2016). *Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)*. Jurnal Ilmiah Teknologi Informasi Terapan, 1(3).
- [7] Trisianto, C. (2018, July). *Penggunaan metode waterfall untuk pengembangan sistem monitoring dan evaluasi pembangunan pedesaan*. In ESIT (Vol. 12, No. 1, pp. 8-22).