

## PEMBANGUNAN APLIKASI PENUGASAN & PENGELOLAAN NILAI PRAKTIKUM BERBASIS SAAS

### APPLICATION DEVELOPMENT OF ASSIGNMENT & MANAGEMENT SCORE PRACTICUM BASED ON SAAS

Rizky Ginanjar Pratama<sup>1</sup>, Fitri Susanti, S.T., M.T<sup>2</sup>

<sup>1</sup>Fakultas Ilmu Terapan, Univesitas Telkom

<sup>2</sup>Fakultas Ilmu Terapan, Universitas Telkom

<sup>1</sup>[rizkyginanjarpratama2@gmail.com](mailto:rizkyginanjarpratama2@gmail.com), <sup>2</sup>[fitri.susanti@tass.telkomuniversity.ac.id](mailto:fitri.susanti@tass.telkomuniversity.ac.id),

#### Abstrak

Laboratorium adalah tempat riset ilmiah, eksperimen, pengukuran ataupun pelatihan ilmiah dilakukan. Pada perguruan tinggi di Indonesia laboratorium biasanya dibuat untuk memungkinkan dilakukannya kegiatan praktikum dan sebagai penunjang dari pembelajaran. Oleh karena itu dibutuhkan aplikasi yang tidak hanya dapat digunakan oleh satu perguruan tinggi saja tetapi dapat digunakan oleh banyak perguruan tinggi. Di proyek akhir ini dibangun aplikasi yang dapat menunjang proses yang ada di laboratorium khususnya penugasan & penilaian praktikum yang dapat digunakan laboratorium perguruan tinggi di Indonesia. Pembuatan aplikasi yang diawali dengan menganalisis kebutuhan laboratorium pada perguruan tinggi di Tel-u serta menentukan metode dan tools yang cocok untuk digunakan. Aplikasi ini dibuat menggunakan *Framework CodeIgniter*, dan metode SDLC dengan model *waterfall* sebagai pengembangannya. Aplikasi ini berbasis SaaS sehingga dapat digunakan oleh banyak laboratorium perguruan tinggi tanpa perlu menginstalnya. Aplikasi ini dibuat dalam bahasa pemrograman PHP dan *database* MySQL. Hasil dari proyek akhir ini adalah aplikasi penugasan & penilaian praktikum berbasis saas.

**Kata kunci:** kata kunci – Aplikasi, Praktikum, Codeigniter, SaaS.

#### Abstract

*The laboratory is a place of scientific research, experiment, measurements nor scientific training done. At college in Indonesia laboratory usually made to allow the activities of lab work and as supporting of learning. Hence needed application that not only can be used by 1 college but can be used by many college. In the end of this project built an application that can support a process existing in a laboratory especially assignment & the assessment of lab work that can be used a laboratory college in Indonesia. Making an application that begins by analyzing the needs of laboratories at college in Tel-u and determine the methods and tools that are suitable for use. This application was made using CodeIgniter Framework, and methods of SDLC with the waterfall model as its development. This application SAAS-based because it can be used by many laboratories college without the need to install. This application is made in PHP language and MySQL database. Result from this project is assignment & management score practicum application based on saas.*

**Keywords:** keywords – Application, Practicum, Codeigniter, SaaS.

#### 1. Pendahuluan

##### 1.1 Latar Belakang

Laboratorium adalah tempat riset ilmiah, eksperimen, pengukuran ataupun pelatihan ilmiah dilakukan. Pada perguruan tinggi di Indonesia laboratorium biasanya dibuat untuk memungkinkan dilakukannya kegiatan praktikum dan sebagai penunjang dari pembelajaran. Pada saat ini perkembangan sistem informasi dapat mempengaruhi kinerja suatu Laboratorium, dengan adanya sistem informasi, dapat memberikan kemudahan serta memperkecil kesalahan dalam hal rekrutasi asisten praktikum, inventaris *hardware*, penjadwalan praktikum dan terutama dalam pemberian tugas dan pencatatan nilai hasil praktikum.

Proses penugasan dan penilaian praktikum merupakan salah satu aktivitas yang dilakukan laboratorium untuk memberikan materi dan mengukur sejauh mana kemampuan praktikan. Asprak akan memberi soal tugas pendahuluan, tes awal, jurnal dan tes akhir kepada seluruh praktikan. Setelah itu praktikan mengerjakan tugas yang diberikan oleh asprak, dan asprak akan menilai hasil tugas yang sudah dikerjakan oleh praktikan. Setelah tugas dinilai oleh asprak, hasil dari nilai tugas tersebut diberikan kepada dosen mata kuliah terkait. Rata-rata di setiap perguruan tinggi seluruh proses di atas dilakukan secara manual sehingga berlangsung kurang efektif dan menimbulkan beberapa masalah. Antara lain, pemberian soal tidak tercatat di satu tempat terpusat sehingga sulit untuk melihat histori soal yang telah diberikan. Banyaknya praktikan dalam suatu

mata kuliah dan hanya ditangani oleh beberapa asisten praktikum sehingga menyebabkan lamanya penilaian terhadap tugas yang diberikan. Sulitnya perekapan nilai yang ada karena dilakukan secara manual mengakibatkan lamanya pelaporan nilai praktikum kepada dosen mata kuliah terkait.

Permasalahan di atas rata-rata terjadi di hampir setiap perguruan tinggi yang ada. Oleh karena itu dibutuhkan aplikasi yang tidak hanya dapat digunakan pada salah satu perguruan tinggi saja, tetapi dibutuhkan aplikasi yang dapat digunakan dan sesuai dengan proses bisnis yang ada pada setiap perguruan tinggi. SaaS adalah salah satu layanan dari *Cloud Computing* yang berupa *software* atau aplikasi, di mana *user* dapat menggunakan aplikasi tersebut tanpa harus menginstalnya dan dapat mengaksesnya di manapun selama terhubung dengan internet. Sehingga penulis tertarik untuk melakukan pembangunan aplikasi berbasis Saas dengan mengambil judul "*Pembangunan Aplikasi Penugasan & Pengelolaan Nilai Praktikum Berbasis Saas*".

##### 1.2 Rumusan Masalah

Masalah yang akan di bahas dalam Proyek akhir ini adalah bagaimana membuat aplikasi yang dapat digunakan oleh banyak perguruan tinggi untuk proses praktikum yang memiliki fungsi :

1. Membantu asisten praktikum dalam melakukan pemberian tugas praktikum.

2. Membantu asisten praktikum dalam memeriksa hasil tugas praktikan.
3. Membuat laporan nilai tugas praktikum.
4. Membuat praktikan dapat mengakses histori tugas yang diberikan asisten praktikum.

**1.3 Tujuan**

Tujuan yang ingin dicapai dari Proyek Akhir ini adalah membuat aplikasi berbasis SaaS yang memiliki fungsionalitas :

1. Pemberian tugas praktikum.
2. Periksa tugas praktikan.
3. Cetak nilai tugas praktikum.
4. Akses histori tugas yang diberikan asisten praktikum.

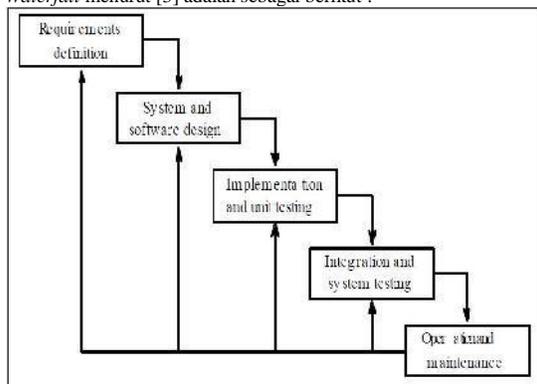
**1.4 Batasan Masalah**

Batasan masalah dalam proyek akhir ini adalah sebagai berikut :

1. Proyek akhir ini hanya membahas pembangunan aplikasi untuk proses penugasan & pengelolaan nilai tugas praktikum.
2. Proyek akhir ini hanya mengambil *sample* proses bisnis penugasan & pengelolaan nilai praktikum dari 3 fakultas di Universitas Telkom.
3. Proyek akhir ini hanya mencetak nilai tugas permodul.

**1.6 Metode Pengerjaan**

Menurut [1] *Sistem Development Life Cycle* (SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik). Metode yang digunakan dalam pengerjaan proyek akhir ini adalah dengan menggunakan *Waterfall* yang merupakan salah satu metode SDLC. Tahapan tahapan dari metode *Waterfall* menurut [3] adalah sebagai berikut :



Gambar 1-1 Grafik Metode Pengerjaan Waterfall

**1. Requirements Definition**

Pada tahap ini dilakukan pengumpulan data yang dilakukan beberapa metode, yaitu observasi dan studi pustaka.

- a. Observasi, mengadakan pengamatan secara langsung terhadap proses praktikum yang berlangsung di 3 laboratorium berbeda.
- b. Studi pustaka, mengumpulkan dan mempelajari dokumen- dokumen yang ada serta mencari buku-buku mengenai *SaaS* untuk dijadikan penunjang dalam melakukan penulisan proyek akhir ini.

**2. System and Software Design**

Pada tahap ini dilakukan perancangan aplikasi berdasarkan informasi data yang didapat dari hasil wawancara dengan asisten laboratorium sebelumnya. Desain tersebut dapat mengimplementasikan kebutuhan proses praktikum pada setiap laboratorium yang dijadikan *sample* pada proyek akhir ini.

**3. Implementation**

Berdasarkan desain yang telah dirancang, maka akan dilakukan pembangunan aplikasi secara langsung. Tahap ini adalah tahapan pengkodean atau penerjemahan desain kedalam bahasa yang dikenali oleh komputer.

**4. Testing**

Setelah proses desain dan *implementation* selesai, maka akan dilakukan pengujian atas aplikasi yang dibuat, untuk mengetahui adanya kesalahan atau tidak dalam aplikasi yang telah dibangun.

**2. Tinjauan Pustaka**

**2.1 Cloud Computing**

Menurut [2] *cloud computing* adalah sebuah model client-server, di mana *resources* seperti server, storage, *network* dan *software* dapat dipandang sebagai layanan yang dapat diakses oleh pengguna secara remote dan setiap saat. Pengguna dapat menikmati berbagai layanan yang disediakan oleh *provider cloud computing*, tanpa perlu terlalu banyak meminta bantuan teknis atau support dari pihak *provider*. Infrastruktur *cloud computing* seperti : server, storage, *network* dan berbagai *software* disebut "*cloud*".

**2.2 Layanan Cloud Computing**

Menurut [4], layanan utama yang disediakan oleh *cloud computing* dibagi menjadi 3 bagian :

1. IaaS (*Infrastructure as a Service*), kemampuan dalam menetapkan ketersediaan perangkat keras kepada konsumen meliputi: *processing, storage, networks and other fundamental computing resource*, Termasuk sistem operasi dan aplikasi.
2. PaaS (*Platform as a Service*), kemampuan dalam menyediakan layanan kepada konsumen untuk dapat membangun aplikasi yang mendukung kedalam infrastruktur *cloud computing* dengan menggunakan bahasa pemrograman sehingga aplikasi tersebut dapat berjalan pada platform yang telah disediakan.
3. SaaS (*Software as a Service*), kemampuan dalam menyediakan layanan yang ditujukan kepada konsumen untuk dapat menjalankan aplikasi di atas infrastruktur *cloud computing* yang telah disediakan.

**2.3 SaaS (Software as a service)**

Menurut [2] SaaS adalah salah satu layanan cloud computing di mana pengguna dapat menggunakan aplikasi namun tidak membuat aplikasi, tidak dapat mengontrol sistem operasi, *hardware*, dan *network*. Aplikasi dapat diakses via *Web-browser* atau *Web based interface*.

Menurut [4] SaaS memiliki beberapa karakteristik, karakteristik dari SaaS antara lain :

1. *Network based access*.
2. Manajemen aktivitas terpusat dan tersentralisasi.
3. Penyediaan aplikasi *one- to-many*, maksudnya satu *software* bisa diperuntukan banyak perusahaan (*multi-tenant*), termasuk arsitekturnya, harga, dan karakteristik manajemennya.
4. *Update-an software* secara terpusat, sehingga customer tidak perlu *download patches* untuk mendapatkan fitur - fitur *terupdate*.

#### 2.4 Manfaat Layanan SaaS

Keuntungan dari layanan ini adalah *user* tidak perlu membeli lisensi untuk mengakses aplikasi. *User* hanya membutuhkan perangkat (*hardware*) yang terhubung ke internet. SaaS mempermudah *user* dengan menghilangkan pembayaran *license software* dan pembayaran *upgrade software*. Sehingga meningkatkan produktivitas bagi *user*[5]. Beberapa keuntungan dan manfaat SaaS menurut [4]:

1. Administrasi lebih mudah,
2. Otomatis *update* dan *patch management*,
3. Kompatibilitas: Semua pengguna akan memiliki versi yang sama dari perangkat lunak,
4. Mudah kolaborasi,
5. Aksesibilitas secara *global*.

#### 2.5 Web Applications

Menurut [5] *Web Applications* adalah program yang menggunakan *HTTP* sebagai *protocol* komunikasi dan menyampaikan informasi berbasis Web kepada *user* dalam bentuk *HTML*. Sebelum adanya *webapps*, aplikasi dibangun dengan menggunakan teknologi *client-server* di mana instalasi aplikasi dibagi antara sisi pengguna dan *server*. Pada awal mula teknologi *web* diperkenalkan, situs *web* terdiri dari sambungan *sntarteks (hypertext link)* yang berupa teks dan sedikit gambar. *Protocol* yang digunakan adalah *HTTP (HyperText Transfer Protocol)*, *HTML (HyperText Markup Languages)* untuk *file* yang saling berhubungan dan menggunakan *URL (Uniform Resource Locator)* untuk mengakses alamat situs tersebut. Sejalan dengan perkembangan teknologi Bahasa pemrograman (*XML, Java, CSS, Javascript, PHP* dan *ASP.net*) dan teknologi *database*, situs mulai mengalami transisi menjadi situs yang lebih dinamis, dan dapat dibuat seperti aplikasi *desktop* tetapi dengan menggunakan teknologi *web* [5].

#### 2.6 Framework

*Framework* dapat diartikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal[6].

#### 2.7 Codeigniter(CI)

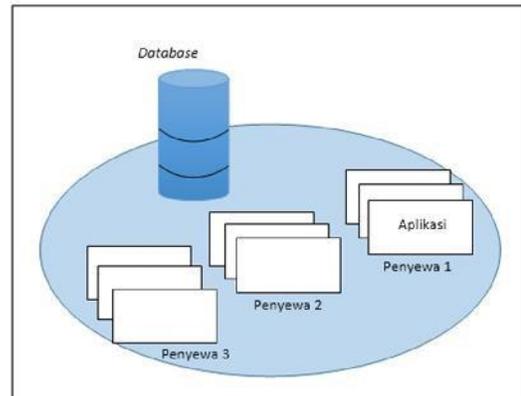
*Framework* pengembangan aplikasi dengan menggunakan *PHP*, suatu kerangka untuk bekerja atau membuat program dengan menggunakan *PHP* yang lebih sistematis [6]. Model *View Controller* merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *Small Talk*, *MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user* interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu *MVC* pattern dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template HTML*, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian *model*.
2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

### 3. Analisis dan Perancangan

#### 3.1 Gambaran Sistem

Berikut ini adalah rancangan aplikasi yang akan dibangun :



Gambar 3-1 Sistem yang akan dibangun

Penjelasan gambar di atas adalah penyewa 1, 2, dan 3 menggunakan aplikasi yang sama dan 1 *database* yang sama. Penyewa 1, 2, dan 3 akan memiliki tampilan aplikasi yang sama dengan alamat halaman yang berbeda.

Aplikasi ini akan dibangun sesuai dengan hasil *survey* wawancara di Fakultas Ilmu Elektro, Fakultas Rekayasa Industri, dan Fakultas Ilmu Terapan pada Universitas Telkom. Metode *survey* yang digunakan yaitu wawancara secara lisan kepada asisten praktikum di Fakultas Ilmu Terapan, asisten praktikum Fakultas Rekayasa Industri, asisten praktikum Fakultas Teknik Elektro.

#### 3.2 Deskripsi dan Fungsionalitas User

Aplikasi ini memiliki beberapa golongan *user* :

1. Superadmin / sebagai admin aplikasi

Superadmin berperan sebagai pemilik & pengendali utama aplikasi. Superadmin memiliki fungsionalitas: Konfirm registrasi *member* (admin fakultas), kelola data *member* (admin fakultas), serta pengembangan aplikasi.

2. *Member* (Admin Fakultas) / *User* sebagai admin fakultas

Admin fakultas merupakan pengguna/*user* yang mendaftarkan halaman fakultas pada aplikasi ini. *User* ini memiliki kendali penuh dengan halaman yang dibuatnya. Admin fakultas memiliki fungsionalitas : kelola data *asprak*, kelola data praktikan, kelola data dosen, kelola data *prodi*, kelola data mata kuliah, setting halaman *login* fakultas.

3. *User* (*asprak*) / *User* sebagai *asprak*

*User* ini merupakan *user* yang ditambahkan oleh admin fakultas. *User* ini memiliki fungsionalitas : kelola data tugas praktikum, periksa jawaban tugas praktikum, *input* nilai tugas praktikum.

4. *User* (*praktikan*)/*User* sebagai praktikan

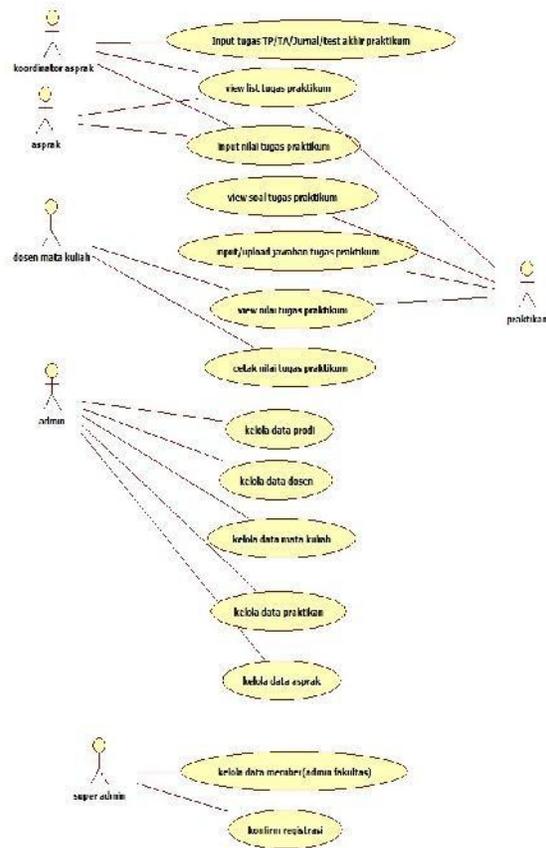
*User* ini merupakan *user* yang ditambahkan oleh admin fakultas. *User* ini memiliki fungsionalitas : lihat tugas praktikum, kerjakan tugas praktikum, *upload file* jawaban tugas praktikum, lihat nilai tugas praktikum.

5. User (Dosen)/User sebagai dosen

User ini merupakan user yang ditambahkan oleh admin fakultas. User ini memiliki fungsionalitas : lihat data tugas praktikum, lihat nilai tugas praktikum, cetak nilai tugas praktikum.

3.3 Use Case Diagram

Di bawah ini merupakan gambar use case diagram dari aplikasi :



Gambar 3-2 Use Case Diagram

3.4 Deskripsi Use Case Diagram

Berikut ini adalah tabel deskripsi use case diagram dari proyek akhir ini :

Tabel 3-1 Deskripsi Use Case Diagram

No	Nama use case	Keterangan
1	Input tugas TP/Jurnal/TA praktikum	Menambah data tugas tugas praktikum.
2	View list tugas praktikum	Melihat data tugas praktikum.
3	Input nilai tugas praktikum	Menginput nilai tugas praktikum.
4	View soal tugas praktikum	Melihat soal dari tugas praktikum.
5	Input/Upload jawaban tugas praktikum.	User memasukkan data jawaban dari tugas praktikum.

6	View nilai tugas praktikum	User Melihat data nilai tugas praktikum.
7	Cetak nilai tugas praktikum	User mencetak data nilai tugas praktikum kedalam format pdf.
8	Kelola data prodi	User dapat mengelola tambah, edit, hapus data prodi.
9	Kelola data dosen	User dapat mengelola tambah, edit, hapus data dosen.
10	Kelola data mata kuliah	User dapat mengelola tambah, edit, hapus data mata kuliah.
11	Kelola data praktikan	User dapat mengelola tambah, edit, hapus data praktikan.
12	Kelola data asprak	User dapat mengelola tambah, edit, hapus data asprak.
13	Kelola data member (admin fakultas)	User dapat mengelola tambah, edit, hapus data asprak.
14	Konfirm Registrasi	User dapat mengkonfirmasi data registrasi.

3.5 Kebutuhan Perangkat Keras dan Perangkat Lunak

Spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan dalam pembangunan aplikasi proyek akhir ini antara lain :

3.5.1 Kebutuhan Perangkat Keras

Spesifikasi perangkat keras (*hardware*) yang digunakan sebagai berikut:

Tabel 3-2 Kebutuhan Perangkat Keras

Perangkat Keras	Spesifikasi
Laptop	Acer Travelmate P342
Processor	Intel(R) Core(TM) i3-2370M CPU @2.40GHz(4 CPUs)
Harddisk	500 GB
Memory	2 GB
Monitor	14 inch

3.5.2 Kebutuhan Perangkat Lunak

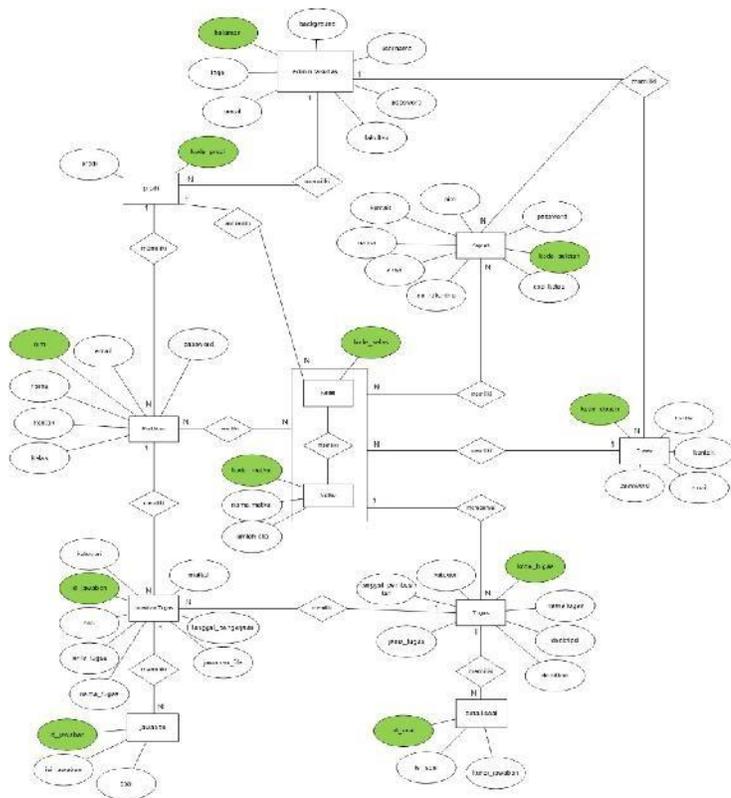
Spesifikasi perangkat lunak yang dibutuhkan untuk mendukung pembangunan aplikasi ini adalah sebagai berikut :

Tabel 3-3 Kebutuhan Perangkat Lunak

Kebutuhan	Spesifikasi
Sistem Operasi	Windows 7 Professional
Web Browser	Mozilla Firefox
Web Server	Apache
Database	MySQL
Teks Editor	Notepad++

### 3.6 Entity Relationship Diagram

Di bawah ini merupakan gambar dari Entity Relationship Diagram dari aplikasi :



Gambar 3-3 ERD

### Daftar Pustaka

- [1] Rosa A.S and M Shalahuddin, *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung, Indonesia: Modula, 2011.
- [2] Sofana Iwan, *Teori dan praktik cloud computing*. Bandung, Indonesia: Informatika Bandung, 2012.
- [3] Ian Sommerville, *Software Engineering*. United States: Addison-Wesley, 2011.
- [4] John W Rittinghouse and James F Ransome, *Cloud Computing Implementation, management, and security*. London: CRC Press, 2009.
- [5] Abdul Kadir, *Dasar Pemrograman WEB dengan ASP*. Yogyakarta, Indonesia: Andi, 2005.
- [6] Betha Sidik, *Framework Codeigniter*. Bandung, 2012.

