Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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:

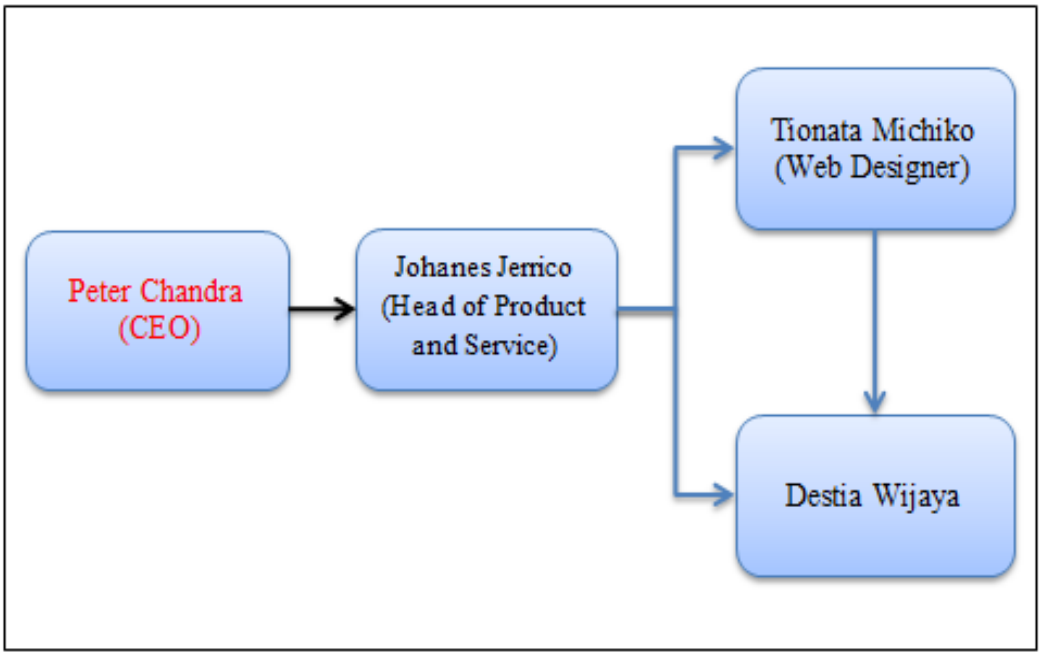
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di perusahaan Moonlay Technologies, peserta magang dipimpin oleh Bapak Johanes Jerrico Wijaya selaku *Head of Product dan Service*. Kedudukan peserta magang adalah sebagai *web programmer* untuk membantu mengembangkan produk-produk berbasis *web* yang ada di dalam situs Loyalti Express. Koordinasi yang dilakukan oleh peserta magang adalah bekerja dalam satu tim untuk menyelesaikan beberapa sub modul yang telah diberikan dan dijelaskan sesuai dengan *requirement* yang diperlukan oleh Bapak Johanes selama beberapa waktu yang telah ditentukan (*deadline*). Pekerjaan masing-masing peserta magang akan di *review* setiap hari untuk memastikan kemajuan kerja dari pekerjaan peserta magang.



Gambar 3.1 Koordinasi Kerja Magang

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama kerja magang adalah mengembangkan produk *web* situs Loyalti Express. Produk yang dikerjakan peserta magang adalah implementasi desain antarmuka *broadcast e-mail*. Produk tersebut dikembangkan dengan tujuan memudahkan *user* dalam mengirim pesan ke banyak orang dalam waktu bersamaan.

Dalam pengerjaannya banyak tahap-tahap yang harus dilewati peserta magang. Berikut cara kerja peserta magang:

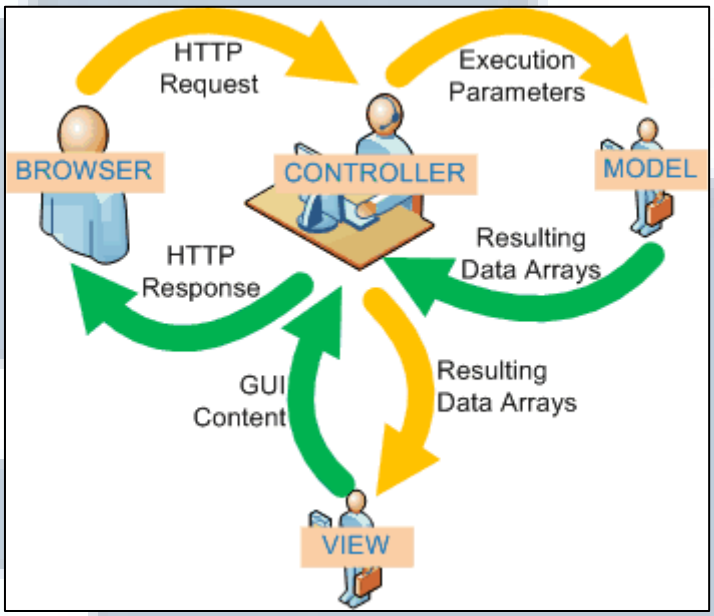
1. Peserta magang menerima rancangan *user interface* dari *designer* yang telah disetujui oleh pembimbing lapangan,
2. Peserta magang mencatat segala *requirements* yang diperlukan dari pembimbing lapangan,
3. Peserta magang melakukan studi literatur terhadap *tools-tools* yang diperlukan tapi belum diketahui dirinya,
4. Peserta magang memulai tahap pengerjaan proyek sesuai *requirements*,
5. Peserta magang mengumpulkan hasil pekerjaan ke pembimbing lapangan. Jika terdapat requirement yang tidak sesuai peserta magang harus segera mengubahnya.

Dalam pembuatan modul ini, peserta magang tidak diizinkan untuk mengakses *database* perusahaan dan metode yang digunakan adalah metode MVC 4. Dalam hal mendesain tampilan *web* sesuai dengan rancangan, peserta magang menggunakan CSS dan beberapa javascript untuk memenuhi spesifikasi dari *requirement* yang ada.

MVC atau *Model-View-Controller* adalah suatu konsep yang cukup populer dalam pembangunan aplikasi *web* dengan cara memisahkan data, tampilan, dan cara prosesnya. Komponen-komponen dalam MVC dipaparkan sebagai berikut.

1. Model, mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. *View* adalah bagian yang mengatur tampilan ke pengguna. *View* berupa halaman web yang terdiri dari rancangan HTML.

3. *Controller* merupakan bagian yang menjembatani model dan *view*. *Controller* berisi skrip-skrip PHP yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.



Gambar 3.2 Struktur MVC

(http://www.scribd.com/doc/92237164/MVC-Framework)

3.3 Requirement yang Diperlukan

Selama magang, peserta magang diberikan beberapa *requirements* yang diperlukan oleh perusahaan dari pembimbing lapangan seperti berikut:

- Pengguna dapat melihat daftar email dalam bentuk *slickgrid*,
- Pengguna dapat mencari *email* yang diinginkan,
- Pengguna dapat mengubah foto,
- Pengguna dapat mengubah informasi lama dengan yang baru.

3.4 Studi Literatur

Studi literatur yang dilakukan selama periode magang (8 minggu). Hal-hal yang menjadi sumber studi literatur diperoleh di internet mengenai tutorial dan demo mengenai HTML, CSS, Javascript, MVC 4, dan *SlickGrid*.

3.5 Uraian Pelaksanaan Kerja Magang

Uraian pelaksanaan kerja magang ini dibagi menjadi lima, yaitu proses pelaksanaan, perancangan sistem, implementasi, kendala yang ditemukan, dan solusi atas kendala yang ditemukan. Berikut penjabaran dari uraian pelaksanaan kerja magang.

3.5.1 Proses Pelaksanaan

Ada beberapa hal yang dilakukan pada saat melaksanakan kerja magang, dari mulai pengenalan framework, desain antarmuka, dan implementasi javascript. Berikut masing-masing penjelasannya.

Table 3.1 Jadwal Pengerjaan Kerja Magang

No	Kegiatan	Minggu							
		1	2	3	4	5	6	7	8
1	Pengenalan *framework* MVC 4								
2	Analisa *user requirement*								
3	Pembuatan *design interface*								
4	Pembangunan sistem								
5	*Functional testing*								
6	Perbaikan *functional testing*								
7	*Integration* dan perbaikan								
8	Bimbingan								

Tahapan pelaksanaan kerja magang yang dilakukan oleh peserta magang selama periode magang di Perusahaan Moonlay Technologies diawali dengan pengenalan dan instalasi beberapa peranti lunak yang digunakan untuk pengembangan sistem. Berikut alur kerja yang dikerjakan oleh peserta magang:

Minggu ke-1

1. Instalasi Visual Studio 2012 yang dilengkapi dengan MVC 4
2. Pengenalan rekan kerja dan MVC 4

Minggu ke-2

1. Analisa *user requirement*
2. Pengenalan *SlickGrid*

Minggu ke-3

1. Pembuatan *design interface* halaman awal sesuai dengan desain yang diterima dari *web designer*
2. Implementasi javascript *SlickGrid* ke dalam sistem yang dibuat

Minggu ke-4

1. Pengenalan javascript *easy-tab*
2. Implementasi javascript *easy-tab* ke dalam sistem yang dibuat

Minggu ke-5

1. Perbaikan bug javascript *SlickGrid* dan *easy-tab*
2. Pembuatan desain halaman *setting*

Minggu ke-6

1. Pengenalan javascript *pop-up*
2. Implementasi javascript *pop-up* ke dalam sistem yang dibuat
3. Pengenalan javascript *change-picture*
4. Implementasi *change-picture* ke dalam halaman *setting*

Minggu ke-7

1. Pengumpulan tugas ke Bapak Johanes Jerrico untuk dicek *user requirement* terpenuhi / tidak
2. Melakukan perbaikan terhadap hasil kerja

Minggu ke-8

Final Testing

3.5.2 Perancangan Sistem

Perancangan sistem yang dilakukan peserta magang meliputi pengembangan program, *use case* diagram, *activity* diagram, *sequence* diagram, serta rancangan tampilan antar muka. Berikut masing-masing penjelasannya

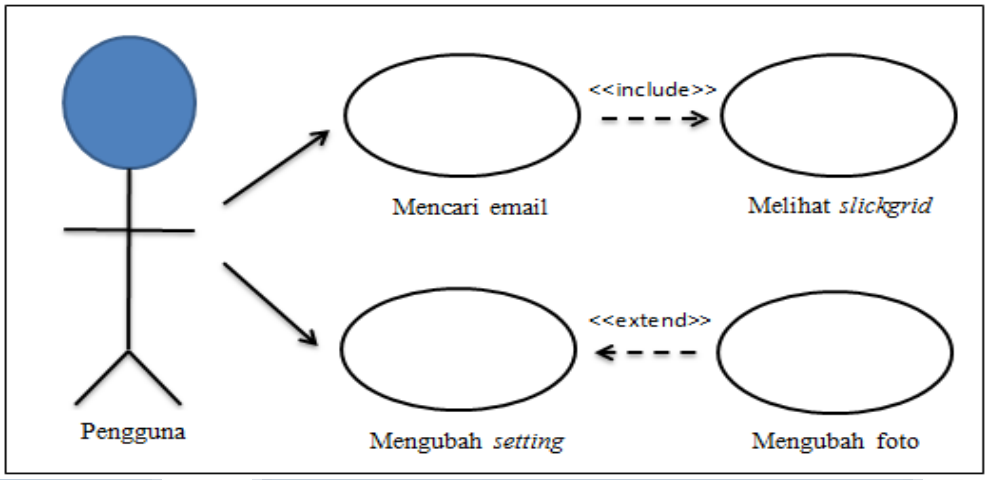
a. Metode Pengembangan Program

Kerja magang dilakukan sesuai dengan pengembangan SDLC (*System Development Life Cycle*). Keuntungan dari SDLC adalah meningkatkan produktivitas, meningkatkan kualitas, memudahkan pengembangan, mengurangi maintenance, dan meningkatkan fleksibilitas (Marzoor, 2012).

Dengan menggunakan SDLC, seluruh analisis dan rancangan aplikasi dilakukan terlebih dahulu sehingga pengerjaan proyek dapat lebih terorganisir dan dikontrol.

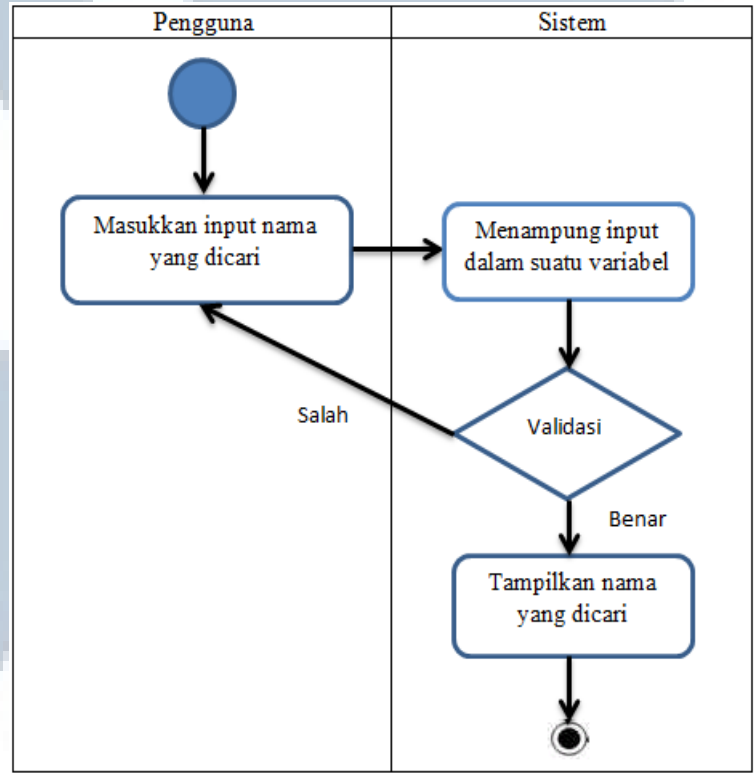
b. Use Case Diagram

Use Case Diagram menggambarkan interaksi yang terjadi antara aktor atau pengguna dengan komponen-komponen yang ada di dalam sistem.



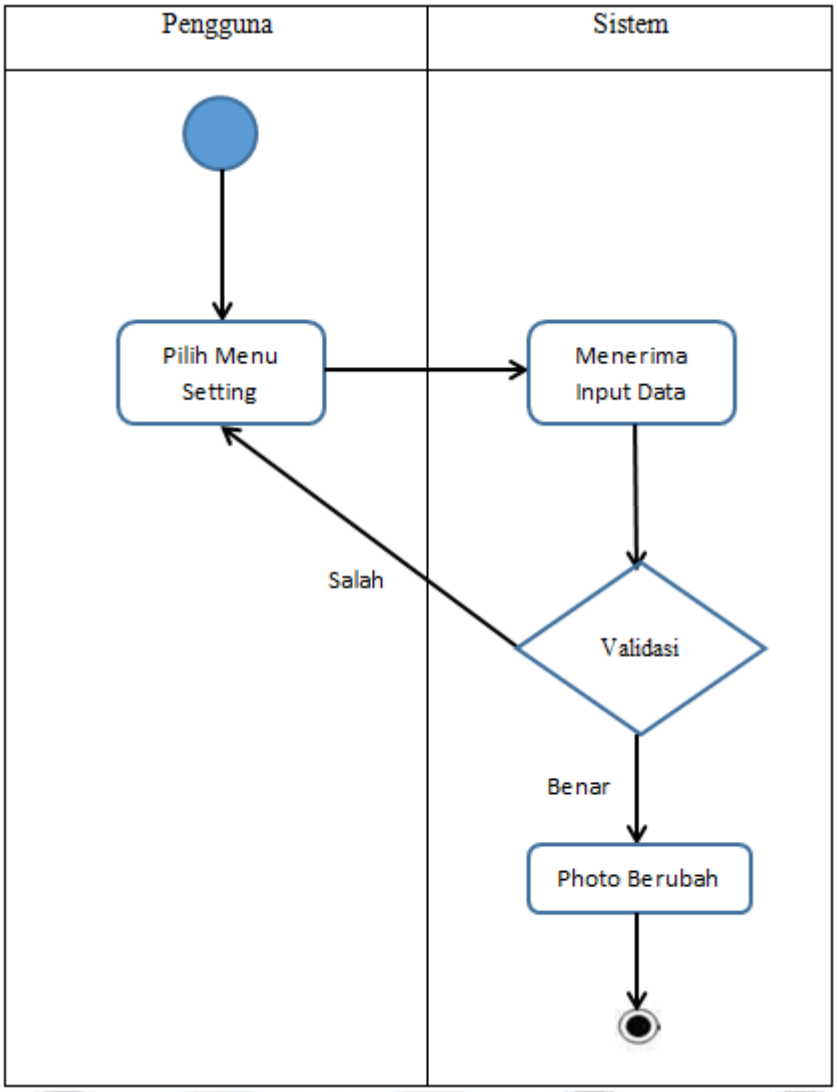
Gambar 3.3 *Use Case Diagram*

c. Activity Diagram



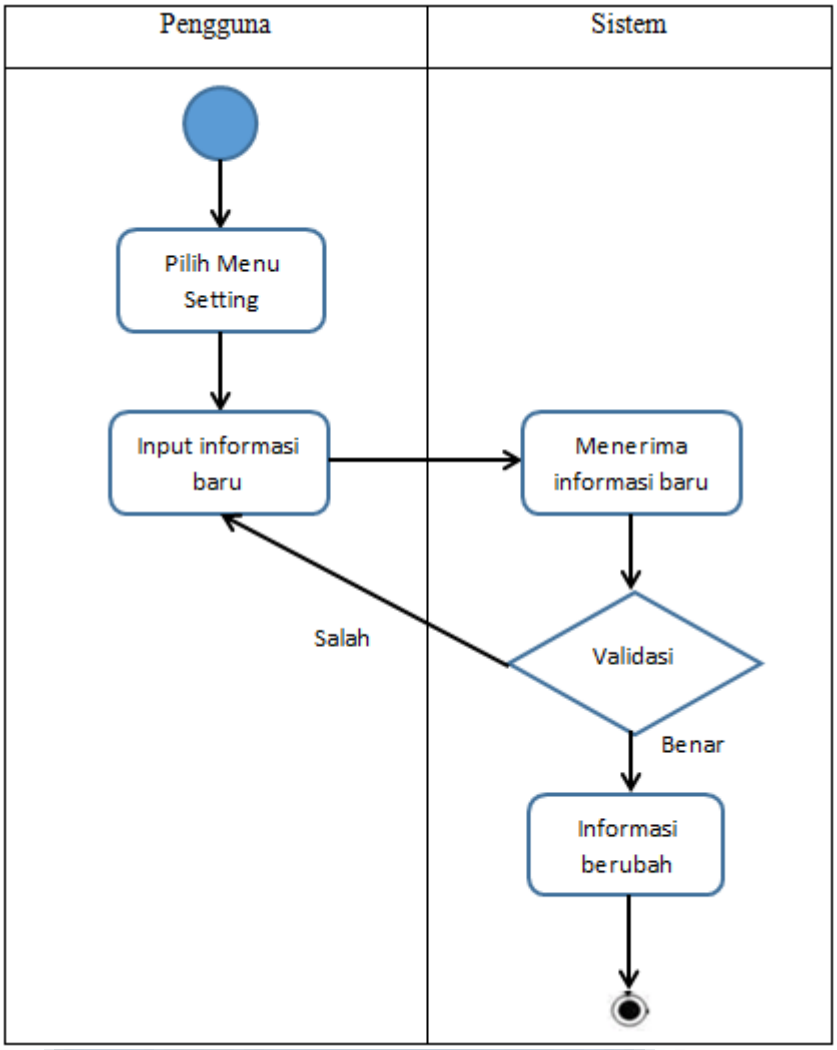
Gambar 3.4 *Activity Diagram* Mencari Email

Gambar 3.4 menjelaskan bahwa setelah pengguna menginput nama *template* yang ingin dicari, sistem akan mencocokkan data yang diinput dengan data yang ada di dalam *slickgrid*. Setelah data yang dicari ditemukan maka sistem akan menampilkannya.



Gambar 3.5 *Activity Diagram Setting* Mengubah Foto

Gambar 3.5 menjelaskan bahwa untuk mengganti foto di dalam *user profile*, pengguna harus memilih terlebih dahulu foto baru untuk menggantikan foto lama. Setelah selesai memilih, sistem akan mengecek (validasi) data yang diinput bertipe (JPG, GIF, PNG, TIF). Jika salah, pengguna harus memilih kembali foto dengan tipe yang ditentukan, dan jika benar foto yang lama akan digantikan dengan yang baru setelah *button Save* diklik.



Gambar 3.6 *Activity Diagram Setting* Mengubah Informasi

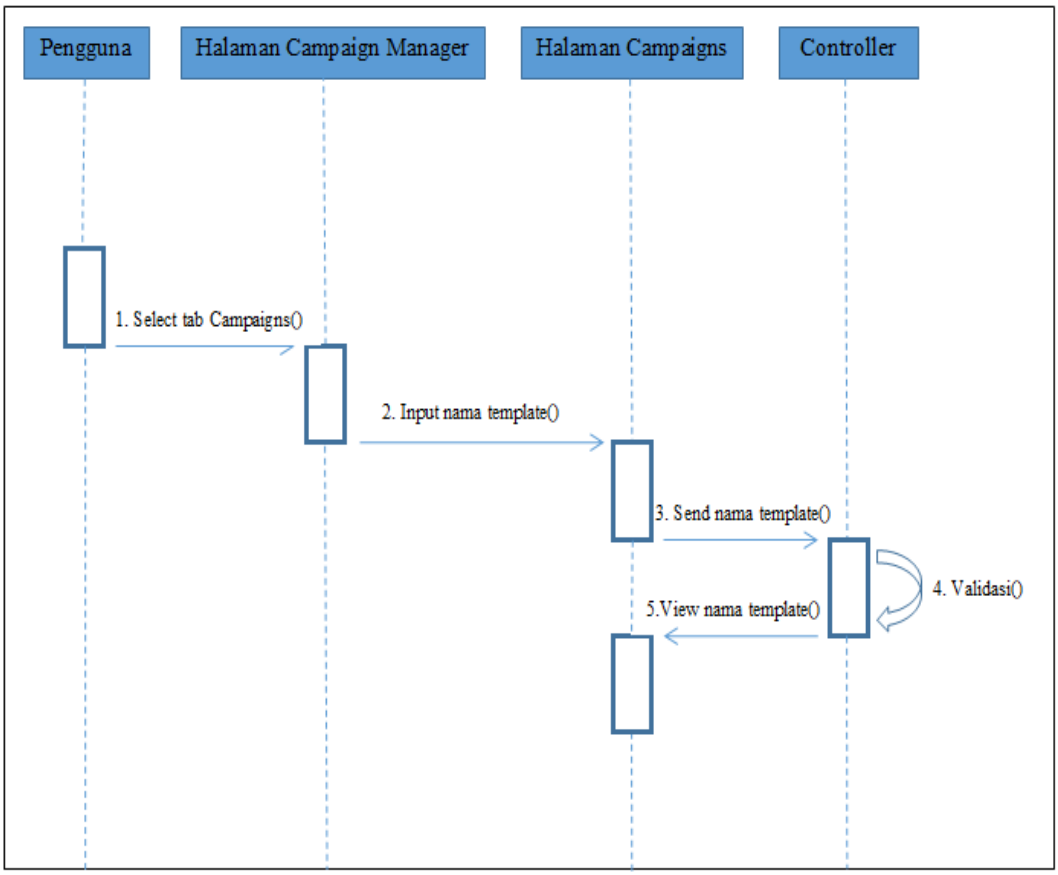
Gambar 3.6 menjelaskan bahwa untuk mengganti informasi lama dengan informasi baru, pengguna harus menginput informasi baru dalam halaman *setting* terlebih dahulu. Setelah pengguna yakin akan informasi baru yang dimasukkan, klik *button Save* untuk menyimpan semua informasi baru itu dan informasi lama akan secara otomatis diganti. Akan tetapi jika pengguna ingin membatalkan penggantian informasi lama dengan yang baru, pengguna dapat menekan *button Cancel*.

d. Sequence Diagram

Sequence diagram menampilkan interaksi antar objek yang ada dalam suatu sistem. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah

event untuk menghasilkan output tertentu. Berikut skenario *sequence diagram* mencari *email*.

1. Pengguna memilih tab Campaigns di halaman Campaign Manager
2. Pengguna menginput nama *template* di halaman Campaigns
3. Input nama *template* dikirim ke *Controller*
4. Controller validasi input nama *template*
5. Jika berhasil, *Controller* menampilkan nama *template* yang dicari di tab Report berbentuk *slickgrid* dalam halaman Campaigns



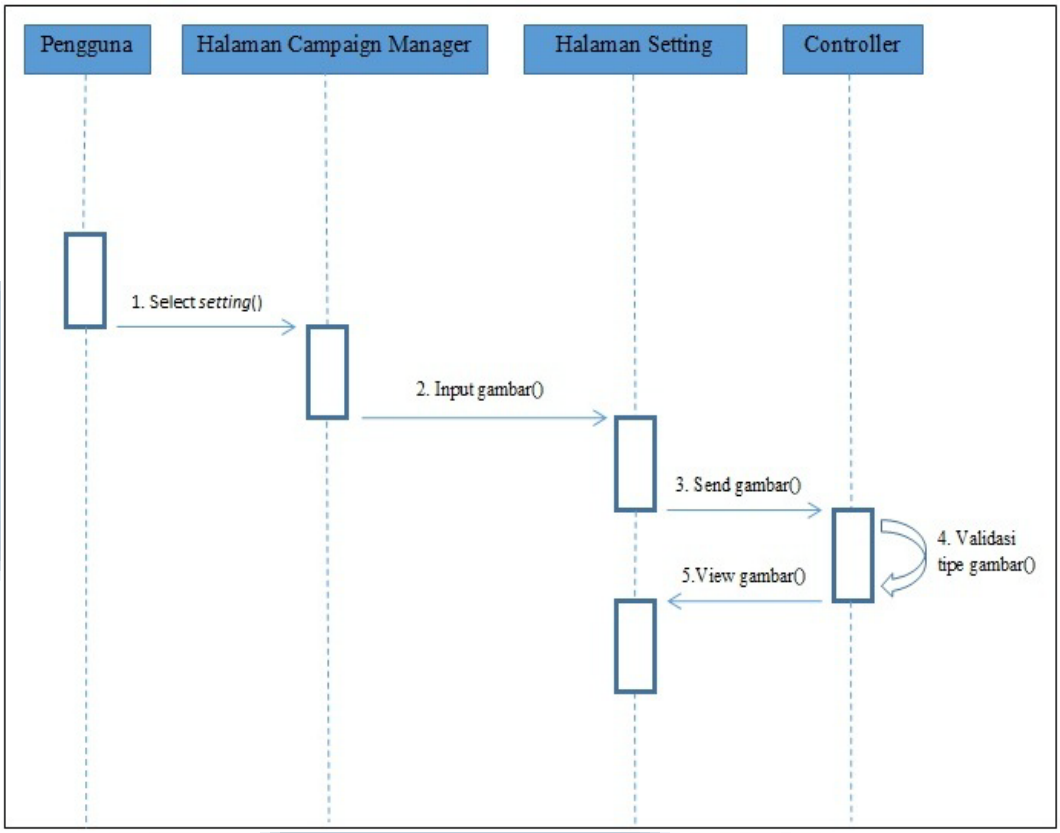
Gambar 3.7 *Sequence Diagram* Mencari Email

Gambar 3.7 menggambarkan *sequence diagram* mencari email pada saat pengguna ingin melihat *email* tertentu dalam bentuk *slickgrid* yang terdapat dalam halaman Campaigns bagian tab Report.

Sequence diagram berikutnya adalah *sequence diagram* mengubah foto. Berikut skenario *sequence diagram* mengubah foto.

1. Pengguna memilih *setting* di halaman Campaign Manager

2. Pengguna memilih gambar baru yang diinginkan
3. Gambar yang dipilih dikirim ke *Controller*
4. *Controller* validasi tipe input gambar
5. Jika berhasil, *Controller* menampilkan gambar baru tersebut dalam halaman Campaign Manager



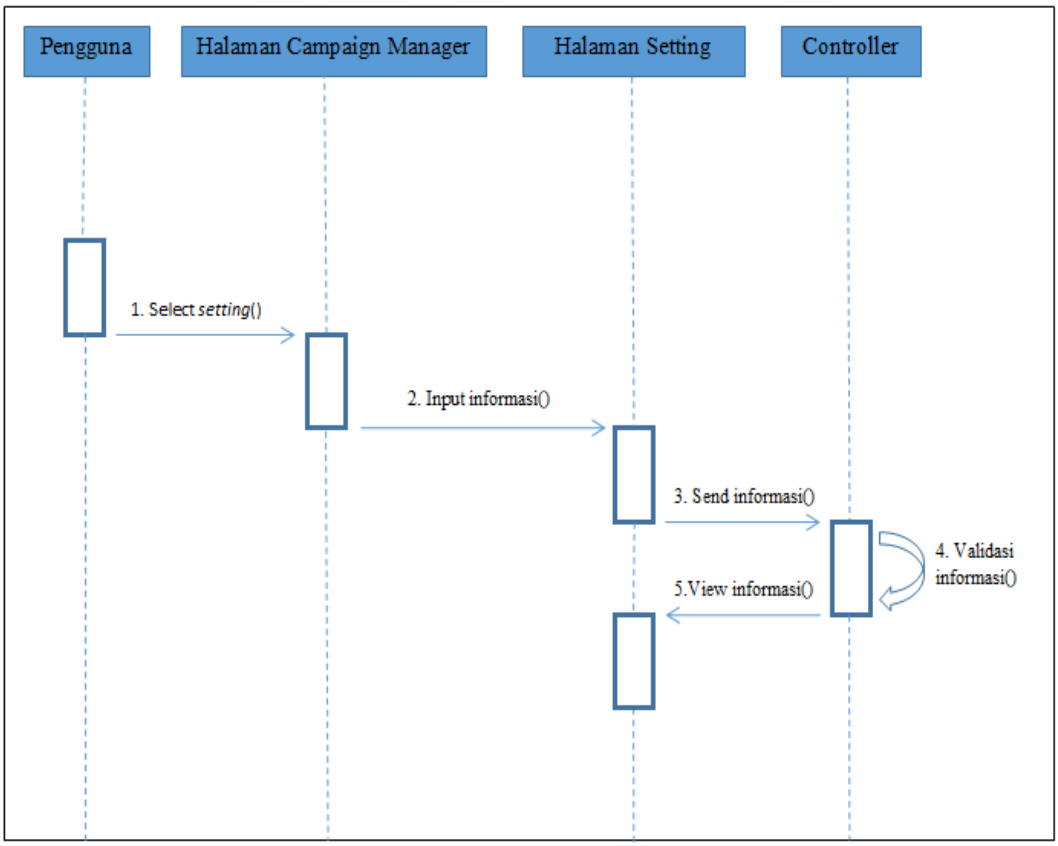
Gambar 3.8 *Sequence Diagram* Mengubah Foto

Gambar 3.8 menjelaskan *sequence* diagram pada saat pengguna ingin mengubah gambar atau foto *user profile* yang lama dengan yang baru dimana dalam halaman *setting* terdapat validasi untuk tipe input gambar yang dimasukkan pengguna.

Sequence diagram yang terakhir adalah *sequence diagram* mengubah informasi. Berikut skenario *sequence diagram* mengubah informasi.

1. Pengguna memilih *setting* di halaman Campaign Manager
2. Pengguna menginput informasi baru yang diinginkan
3. Informasi baru dikirim ke *Controller*
4. *Controller* validasi informasi baru

5. Jika berhasil, *Controller* menampilkan informasi baru tersebut dalam halaman Campaign Manager

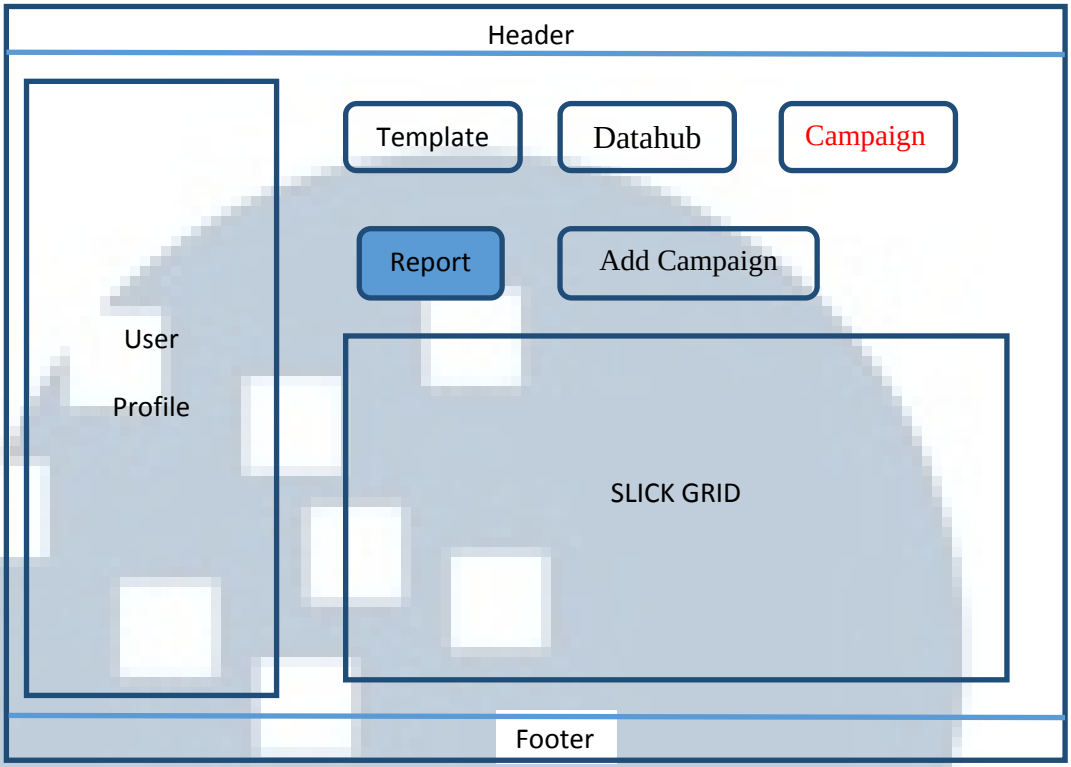


Gambar 3.9 *Sequence Diagram* Mengubah Informasi

Gambar 3.9 menjelaskan *sequence* diagram pada saat pengguna ingin mengubah tampilan tentang *user profile* yang lama dengan yang baru, dimana dalam halaman *setting* terdapat validasi untuk input yang dimasukkan pengguna.

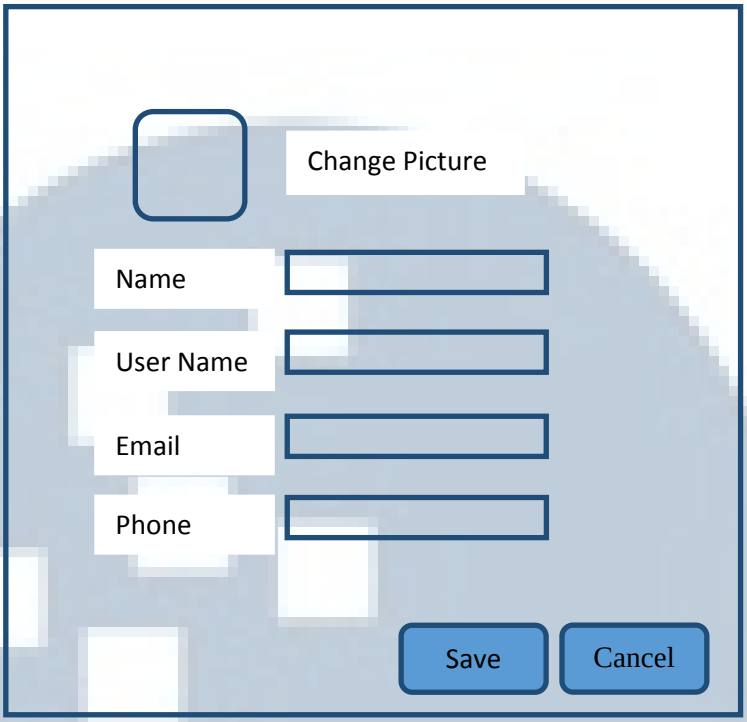
e. Rancangan Tampilan Antar Muka

Setelah dirancang dalam waktu yang cukup lama, akhirnya rancangan untuk situs *broadcast email* pun telah rampung. Dimulai dari beberapa tab yang harus diklik untuk mencapai tampilan yang diinginkan bagi pengguna yang ingin menggunakan *broadcast email*.



Gambar 3.10 Rancangan *interface* situs *broadcast email*

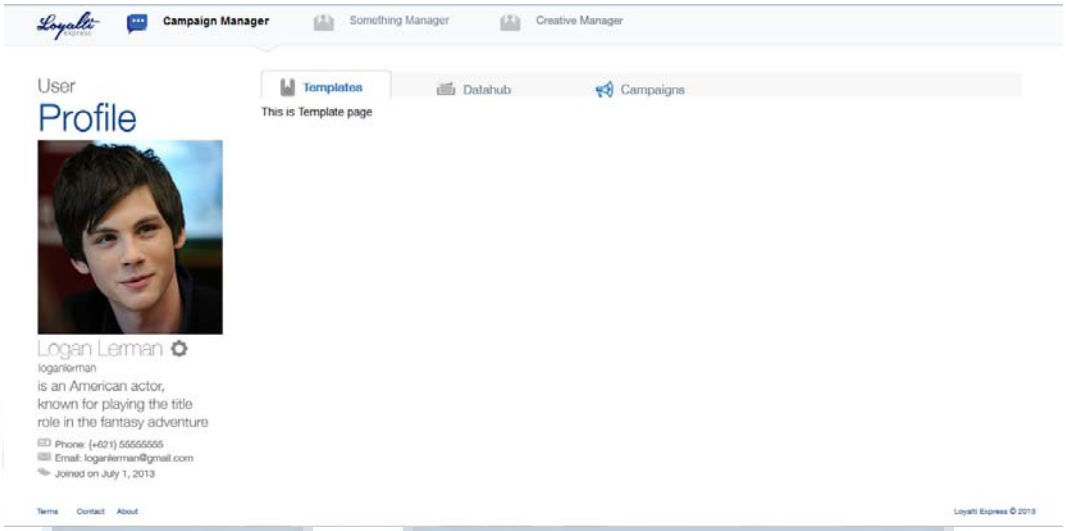
Untuk mengubah informasi yang terdapat di dalam user profile, terdapat rancangan halaman *setting* yang bertujuan memudahkan pengguna mengubah data dari informasi yang telah ada sebelumnya. Input informasi baru dilakukan pengguna dalam *pop up*. Halaman *Pop up* untuk input informasi baru dapat dilihat pada gambar 3.11.



Gambar 3.11 Desain halaman *setting*

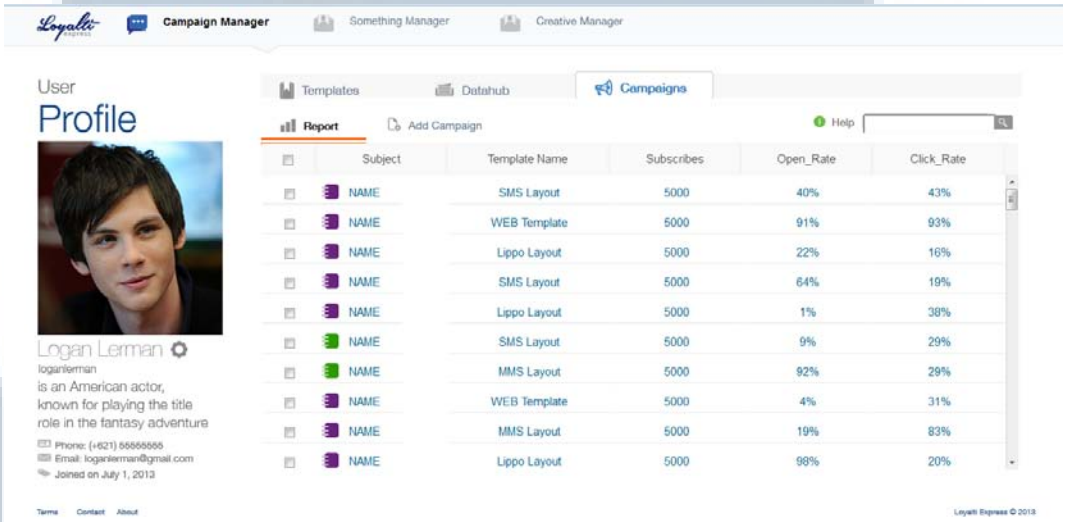
3.5.3 Implementasi

Pada tahap implementasi pengembangan situs *broadcast email* ini, penulis menggunakan metode MVC 4, CSS, Javascript, dan web browser seperti Mozilla Firefox dan Google Chrome. Dalam situs *broadcast email* ini terdapat tab-tab (menggunakan *javascript easytabs*) yang memiliki halaman masing-masing. Tampilan halaman tab pertama Campaign Manager yang berisi *user profile* dapat dilihat pada gambar 3.12



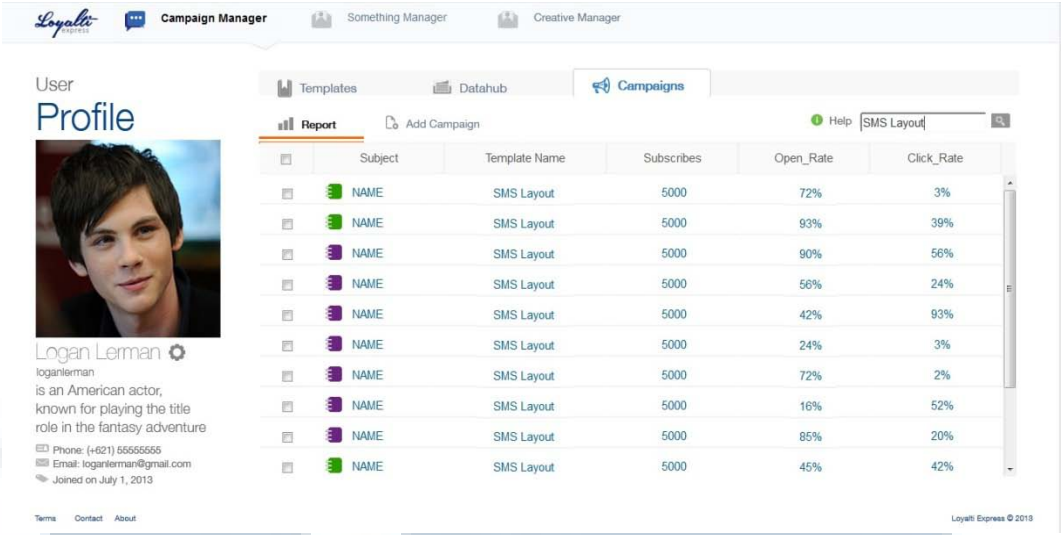
Gambar 3.12 Halaman tab Campaign Manager

Halaman selanjutnya adalah halaman untuk tab tahap kedua dan ketiga dimana terdapat tab yang bernama tab Campaigns yang isinya terdapat dua tab lainnya yang bernama tab Report dan tab Add Campaign (tab tahap ketiga). Peserta magang hanya akan membahas tentang tab Report yang didalamnya membahas tampilan untuk *broadcast email* yang berbentuk dalam tampilan slickgrid. Halaman tab kedua dapat dilihat pada gambar 3.13



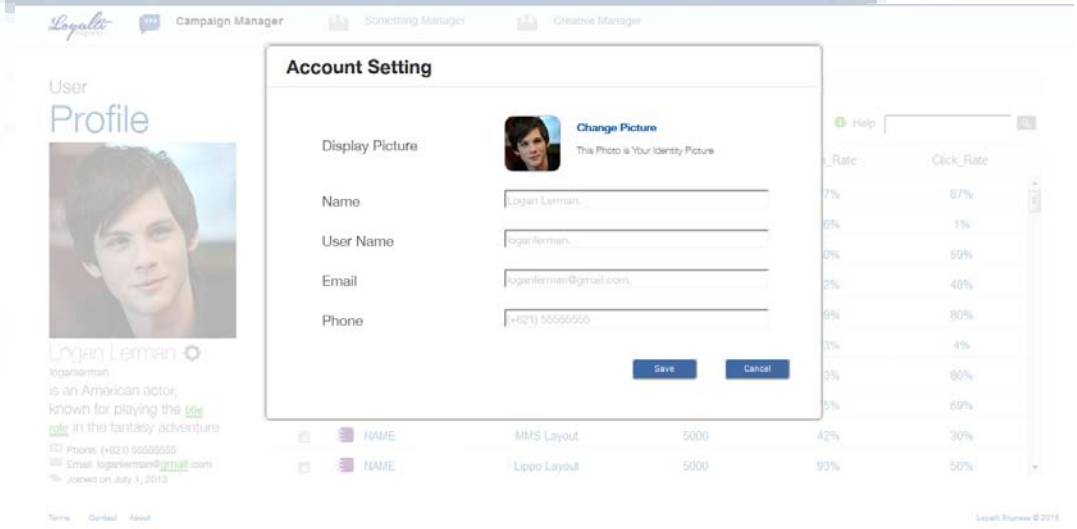
Gambar 3.13 Halaman tab Campaign dan tab Report

Untuk mencari nama *template* tertentu agar pengguna lebih mudah mengirim *email* ke beberapa pengguna dalam kurun waktu bersamaan, maka pengguna dapat memasukkan nama *template* yang dicari dalam *textbox* yang terdapat di halaman Campaign seperti dapat dilihat pada gambar 3.14



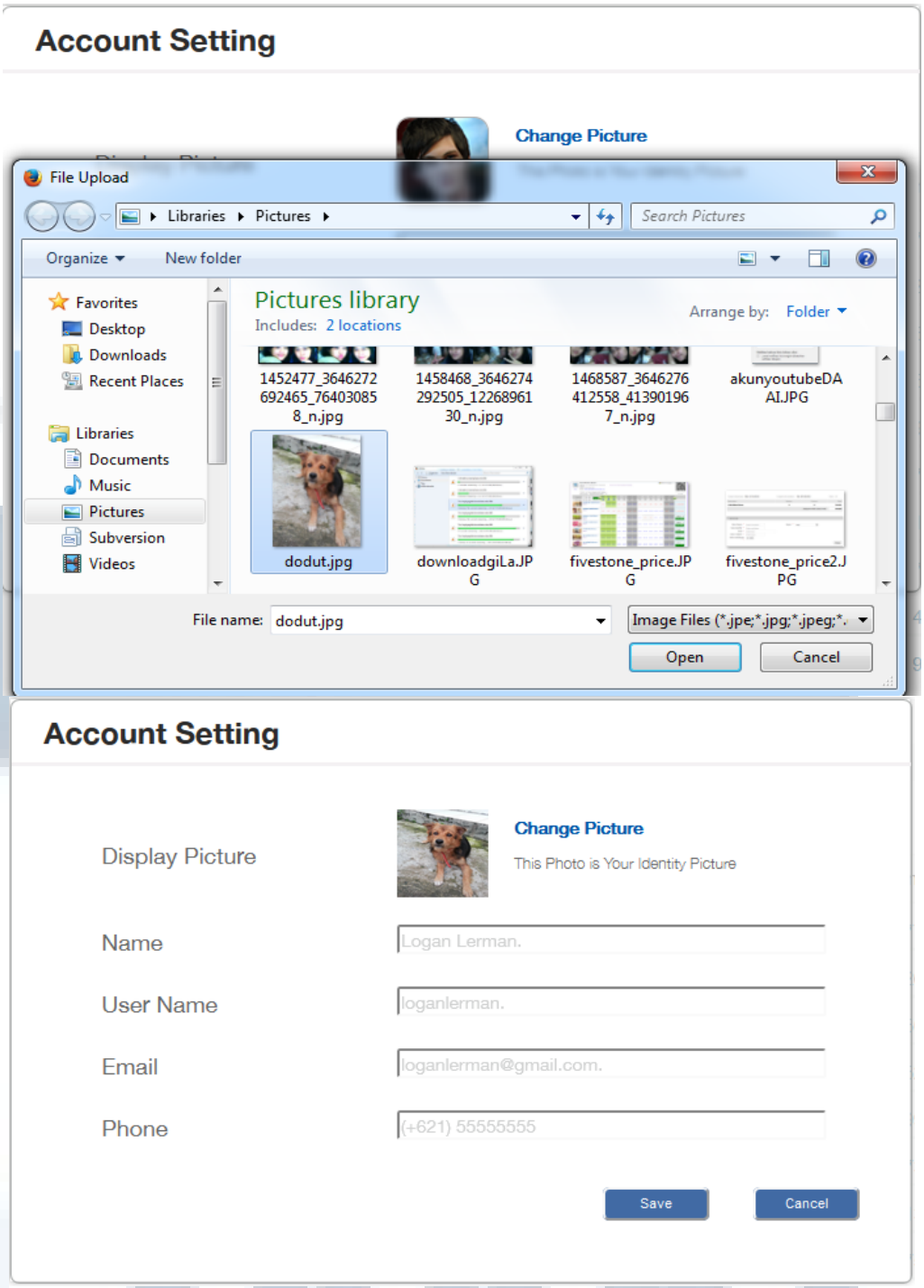
Gambar 3.14 Halaman pencarian *template name*

Dalam halaman di atas terdapat halaman lain yang bernama halaman *setting*. Halaman tersebut digunakan untuk mengubah informasi yang ada di dalam *user profile* sebelumnya menjadi informasi yang baru. Halaman *setting* akan terlihat dalam bentuk *pop up* setelah gambar setting diklik oleh pengguna. Halaman *setting* dapat dilihat pada gambar 3.15



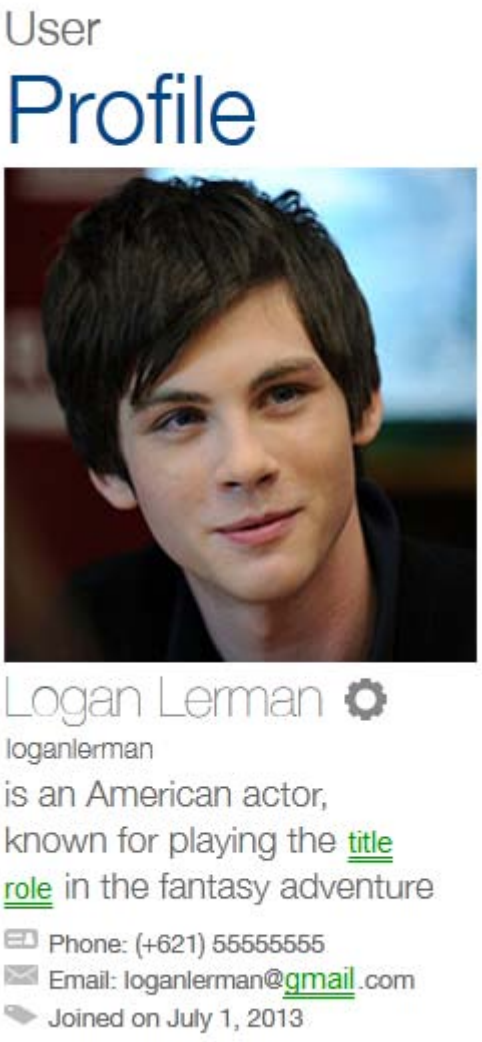
Gambar 3.15 Halaman *setting*

Untuk mengubah foto *profile* dengan foto yang diinginkan, pengguna harus melakukan klik tulisan *change picture* terlebih dahulu. Tampilan saat pengguna memilih foto yang diinginkan dalam folder yang dimiliki dapat dilihat pada gambar 3.16

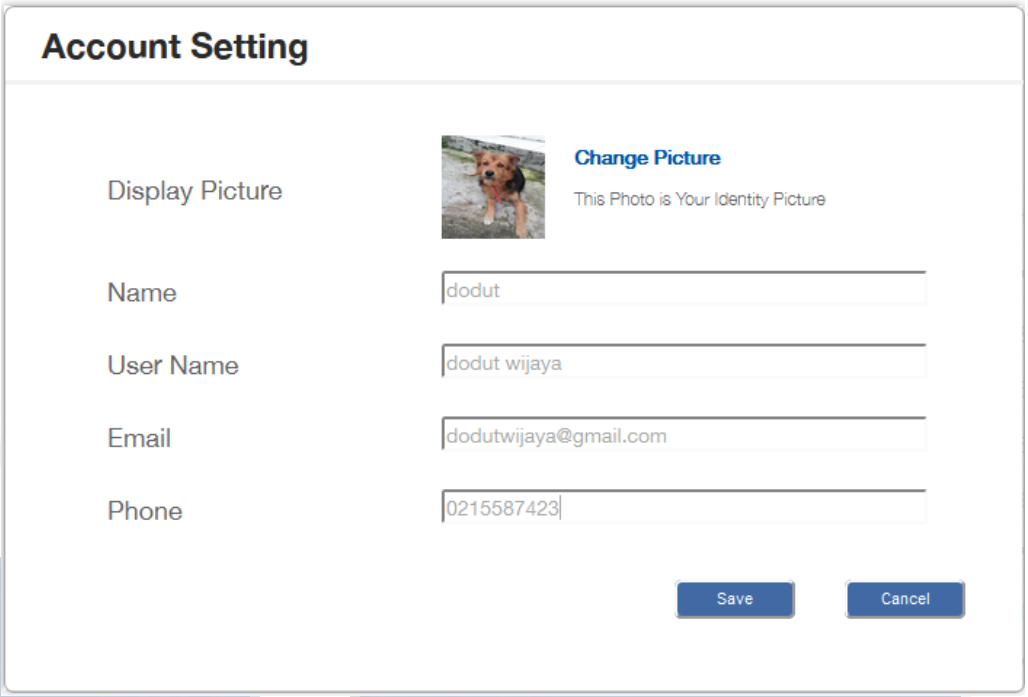


Gambar 3.16 Tampilan *Change Picture*

Untuk mengubah informasi lainnya seperti *name*, *user name*, *email* dan *phone*, pengguna harus mengisi terlebih dahulu *textbox* yang ada dalam halaman *setting*. Setelah selesai pengguna harus melakukan klik *button Save* untuk menyimpan informasi baru tersebut dan secara otomatis informasi yang lama akan tergantikan dengan informasi yang baru.



Gambar 3.17 Informasi *user profile* sebelum diubah



Gambar 3.18 Input *user* mengenai informasi baru *user profile*

Gambar 3.19 Informasi *user profile* setelah diubah

3.5.4 Kendala yang Ditemukan

Kendala yang ditemukan peserta magang dalam membangun web, yaitu sebagai berikut.

a. *User requirement* yang terus bertambah.
b. Kurangnya pengetahuan peserta magang dalam memenuhi *requirement* yang diinginkan pihak perusahaan.
c. Kesalahan (*error*) yang terjadi antara satu jQuery dengan jQuery lainnya disebabkan penggunaan versi jQuery yang berbeda-beda dimana di dalamnya terdapat *function* yang sama antar versi jQuery.
d. Tampilan yang tidak sama antara 2 *web browser* yang digunakan seperti Mozilla Firefox dan Google Chrome.

3.5.5 Solusi atas Kendala yang Ditemukan

Dari kendala yang ditemukan selama pengerjaan, peserta magang memberi solusi sebagai berikut.

a. Melakukan *review* tiap *requirement* yang telah diselesaikan.
b. Perbanyak menganalisa dan melakukan pencarian terhadap hal-hal yang berkaitan dengan *requirement* dari pihak perusahaan.
c. *Error* yang terjadi antara jQuery dapat diatasi dengan memilih satu jQuery yang memiliki segala *function* yang dibutuhkan.
d. Hindari penggunaan *position absolute* dan cari titik tengah dari ukuran margin antara *web browser* Mozilla Firefox dan Google Chrome.