

# Pengembangan Fitur Notifikasi Pada Website Maranatha-Keimyung Korea Center dengan JavaServer Faces Framework

Nucky Vilano<sup>#1</sup>, Mewati Ayub<sup>\*2</sup>, Maresha Caroline Wijanto<sup>#3</sup>

<sup>1,2,3</sup> Program Studi Sistem Informasi, Universitas Kristen Maranatha  
Jl. Surya Sumantri no. 65 Bandung 40174  
Telp. (022) 2012186

<sup>1</sup>it1573001@student.it.maranatha.edu

<sup>2</sup>mewati.ayub@it.maranatha.org

<sup>3</sup>maresha.cw@it.maranatha.edu

**Abstract** — Maranatha-Keimyung Korea Center is an organization that provide services to learn Korean language which known by Korea Center. Korea Center is still manually record all of operational stuff. Therefore Korea Center in need of an application for managing data, and helping Korea Center to spread information. Thats why there are integration of email gateway for send emails. Website application with email gateway implemented using Java programming language, Javascript, CSS, MySQL also using JavaServer Faces framework within Neatbeans 8.1 also hosted online. This website provide user to access registration, downloading resources of class and also managed data via online website and also giving notification via website and emails to the student and admin for some purposes.

**Keywords**— Email Gateway, Registration, Managing Data, Website.

## I. PENDAHULUAN

### A. Latar Belakang

Maranatha-Keimyung Korea Center merupakan sebuah organisasi hasil kerjasama antara Universitas Kristen Maranatha dan Universitas Keimyung. Maranatha-Keimyung Korea Center juga biasa disebut sebagai Korea Center. Korea Center berperan sebagai wadah bagi mahasiswa atau pihak lainnya untuk dapat belajar tentang Bahasa Korea. Korea Center memiliki kegiatan manajemen, teknis, dan non-teknis yang berkaitan langsung dengan sumber daya manusia, jadwal pembelajaran, serta laporan-laporan lainnya.

Maranatha-Keimyung Korea Center sendiri memiliki manajemen pengelolaan data peserta, kepegawaian, dan informasi kegiatan yang dicatat pada agenda Korean Center. Semua ini masih tercatat secara manual. Seiring dengan semakin banyaknya peminat Korea Center ditambah dengan sistem yang masih manual, pihak Korea Center seringkali kewalahan dan kesulitan dalam mengelola administrasinya. Dikarenakan sistem yang masih manual ini, mengakibatkan laporan juga masih banyak yang berbentuk *hardcopy* serta seringkali memakan waktu yang cukup lama dan banyak terjadi kesalahan dalam pengelolaan datanya. Hal ini membuat data di Korea Center mudah hilang, rusak serta tidak jelas. Dalam penyebaran informasi kepada peserta pun masih menggunakan cara lama, yaitu pihak Korea Center akan mengontak peserta satu per satu. Hal ini juga mengakibatkan ada peserta atau informasi yang terlewat atau bahkan terlupakan.

Penelitian ini menawarkan pengembangan sistem informasi berbasis website agar pendataan data peserta, pegawai, pengajar, dan materi di Korea Center menjadi lebih terpusat. Selain itu, untuk dapat membantu menyebarkan informasi terkait pembelajaran kepada peserta, sistem ini juga akan dilengkapi dengan fitur *email gateway*.

## II. KAJIAN TEORI

### A. Penelitian Terkait

Pembuatan *website* yang menerapkan *email gateway*, pada penelitian oleh C. B. Susila dan R. A. Triyono mengembangkan aplikasi dengan *email gateway* yang dapat memanfaatkannya untuk keperluan mengirim *email* [6]. Sedangkan A. Wijayanto dan C. P memanfaatkan *email gateway* dengan pengiriman *email* pada pengguna *website* jika diperlukan pada saat pengiriman informasi [7]. J. C. W dan M. R. F memanfaatkan *website* untuk meningkatkan interaksi

dengan pengguna seperti meminta kuisioner pada pengguna yang telah melakukan kegiatan dalam *website* [8]. Serta pada penelitian J. A. Sihite dan M. C., *website* juga dapat digunakan untuk pencatatan data pengguna dalam menggunakan jasa yang tersedia seperti registrasi, jadwal acara yang tersedia. [9].

Untuk pengembangannya, penelitian ini memanfaatkan *email gateway* dalam berbagai bidang seperti pengiriman notifikasi pendaftaran, adanya aktivitas dalam akun pengguna, interaksi dengan pengguna, juga notifikasi pada admin pada adanya aktivitas yang terjadi dalam *website* serta pengiriman informasi melalui *email* pada peserta. Penelitian ini juga mengembangkan *website* agar dapat digunakan untuk pendaftaran jasa yang disediakan seperti pendaftaran peserta baru, *download* materi kelas yang ada, serta melakukan pengiriman *email* pada admin jika ada kesulitan.

## B. JavaServer Faces

*JavaServer Faces* (JSF) adalah sebuah *framework* antarmuka pengguna untuk *website* berbasis Java. JSF dapat memungkinkan pengguna untuk melakukan beberapa hal, yaitu[1]:

1. Membuat sebuah tampilan *website* dengan komponen *user interfaces* yang dapat digunakan kembali menggunakan *Model-ViewController*.
2. Mengikat komponen ke dalam server, agar memudahkan proses migrasi aplikasi antar data dan *user interfaces*.
3. Menangani navigasi halaman *website*.
4. Menyediakan model sederhana untuk menghubungkan *events* kepada *server-side application*.
5. Memudahkan pembuatan dan penggunaan ulang komponen *user interfaces* yang di sesuaikan.

*JavaServer Faces* (JSF) terdiri dari:

1. Satu set halaman *website* di mana komponen *user interfaces* ditata.
2. Satu set *managed beans*, yang digunakan untuk mengikat komponen kedalam *server-side* dan antar *acts* (aksi) sebagai *controller* aplikasi.
3. Sebuah deskripsi *deployment* yang opsional berupa *web.xml*.
4. Sebuah deskripsi konfigurasi data berupa *face-config.xml*.
5. Sebuah set kustom *objects* seperti *converters* dan *listeners*.

- 1) *Facelets*: bahasa deklarasi tampilan (*view handler*) untuk JSF. *Facelets* digunakan sebagai salah satu pengganti *JavaServer Pages* (JSP). *Facelets* dapat melakukan *composite components* serta *Ajax* dan mudah untuk *error reporting* [1].
- 2) *Composite Components*: JSF mendefinisikan *composite components* sebagai sebuah komponen yang memiliki satu atau beberapa komponen JSF yang didefinisikan dalam *Facelets Markup File* [1].
- 3) *Ajax*: JSF menyediakan *native support* untuk menambahkan *ajax capabilities* ke dalam halaman *website*. *Ajax* menyediakan *partial view processing*, dimana hanya beberapa komponen dalam *view* digunakan untuk melakukan proses [1]. Ada dua cara penggunaan *Ajax* yaitu secara *programmatic* menggunakan *JavaScript* dan secara deklaratif menggunakan *f:ajax*. Integrasi pemrograman *Ajax* dilakukan melalui *resource handling mechanism*. *Jsf.js* adalah sebuah sumber yang didefinisikan sebelumnya dalam *javax.faces*. Kode Program 1 adalah contoh penggunaan *Ajax* [1].

```
<h:body>
<!-- . . . -->
<h:outputScript
  name="jsf.js"
  library="javax.faces"
  target="body"/>
<!-- . . . -->
</h:body>
```

Kode Program 1. Penggunaan *Ajax*

- 4) *Enterprise JavaBeans* (EJB): sebuah komponen untuk melakukan *development* dan *deployment* dari komponen yang ada pada aplikasi. EJB biasa berisikan logika bisnis yang mengoperasikan *event* pada aplikasi. Sebuah *bean instance* di kelola pada *container*. *Bean* diakses pada sisi klien dan dijalankan oleh *container* yang di *deployed* [2]. *Session Scoped Beans* menyimpan *conversational state* untuk klien secara spesifik, kondisi ini disimpan dalam *Session Bean Instance*. Contoh sederhana *stateful session beans* menggunakan *@Stateful* [1] seperti pada Kode Program 2.

```
1. @Named(value = "bean001")
2. @SessionScoped
3. public class pesertaController implements Serializable {
```

Kode Program 2. Implementasi Penggunaan *SessionScoped*

- 5) *Java Persistence*: alat yang mendefinisikan sebuah standar *mapping* antar *database* dan tabel. *Java Persistence* mendefinisikan *syntax* untuk mendapatkan *primary* dan *foreign key* pada setiap *query*.

### C. JavaMail API

JavaMail API adalah representasi dari komponen sistem *email*, komponen yang direpresentasikan oleh kelas abstrak di dalam `javax.mail.package`. `javax.mail.Message` merepresentasikan sebuah *email*. JavaMail API mendeklarasikan *abstract class* untuk mendapatkan berbagai jenis *envelope* informasi untuk sebuah *email*, seperti pengirim dan alamatnya, tanggal kirim, dan subjek *email* [3].

```
Properties props = System.getProperties();
props.put("mail.smtp.host", "smtp.gmail.com");
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.port", "587");
props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
props.put("mail.smtp.socketFactory.port", "587");
props.put("mail.smtp.socketFactory.fallback", "false");
props.put("mail.smtp.starttls.enable", "true");

Session mailSess = Session.getDefaultInstance(props, null);
mailSess.setDebug(true);

Message mailMess = new MimeMessage(mailSess);

mailMess.setFrom(new InternetAddress("<from>"));
mailMess.setRecipient(Message.RecipientType.TO, new InternetAddress("<to>"));
mailMess.setContent("<Konten>", "text/html");
mailMess.setSubject("<Subjek>");

Transport trns = mailSess.getTransport("smtp");
trns.connect("smtp.gmail.com", "<username>", "<password>");

trns.sendMessage(mailMess, mailMess.getAllRecipients());
```

Kode Program 3. Sending Mail JavaMail

Pada Kode Program 3 merupakan ilustrasi penggunaan *email gateway* untuk melakukan pengiriman *email*.

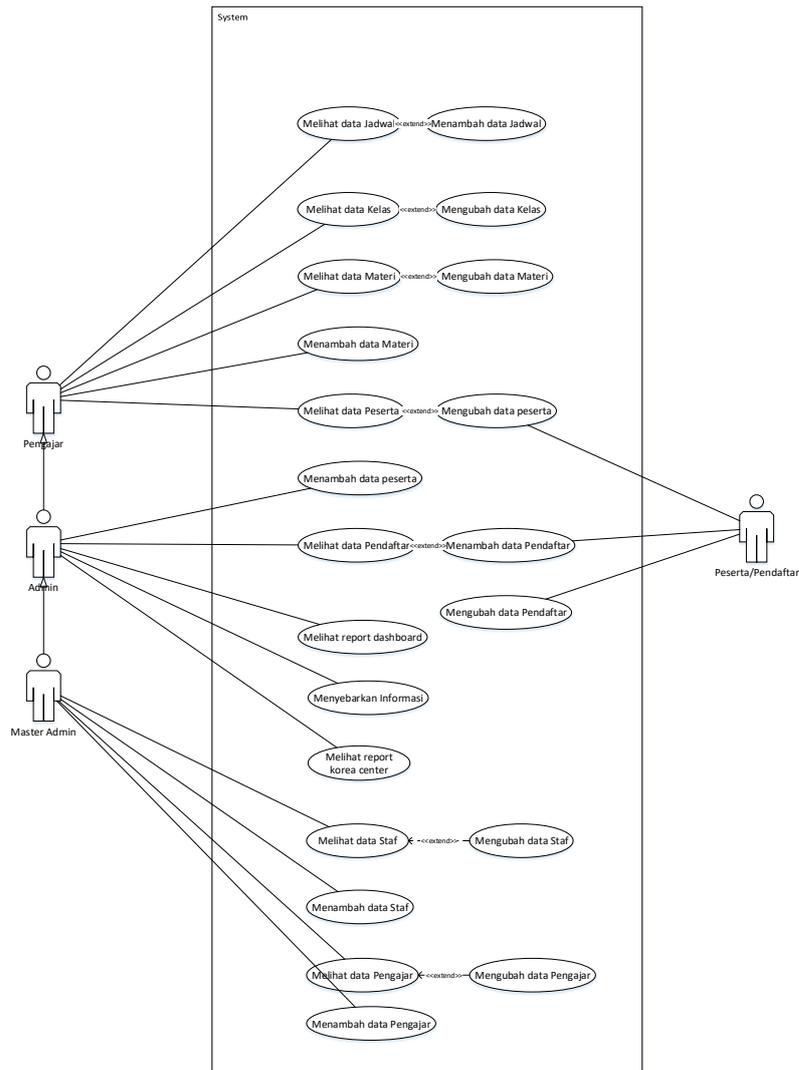
### D. Libraries Lainnya

- 1) *Primefaces*: sebuah *plugin* pemrograman XHTML yang merupakan bentuk reformulasi dari HTML menggunakan paradigma XML. *Primefaces libraries* digunakan untuk *user interfaces* didalam XHTML [4]. Dalam sistem ini digunakan untuk pembuatan berbagai UI seperti *text area*, *input secret*, dan *table* seperti pada Gambar 5.
- 2) *Rich Text*: sebuah *text editor* yang menyediakan perubahan format teks standar, seperti *bold*, *italic*, dan lainnya.
- 3) *Google Calendar*: layanan kalender yang dikembangkan oleh Google dan terintegrasi langsung dengan *email* Google.

## III. ANALISA DAN IMPLEMENTASI

### A. Use Case

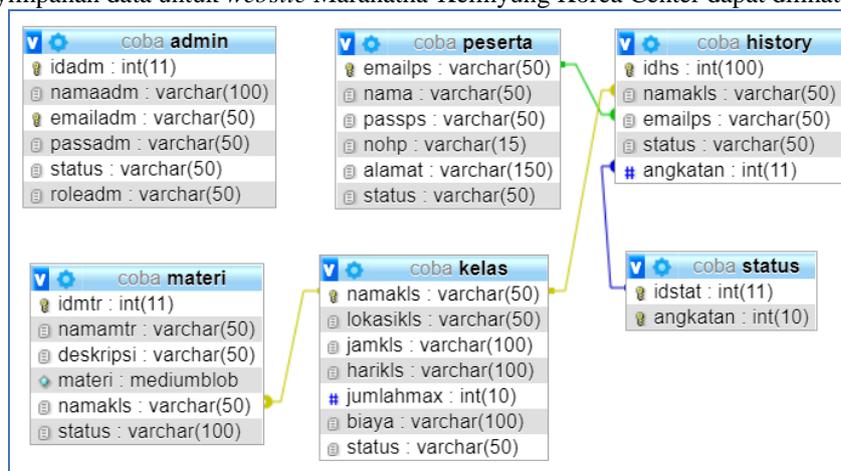
Berikut ini adalah *use case diagram* [10] terdapat 4 aktor, yaitu Master Admin, Admin, Pengajar, dan Peserta dengan hak akses yang berbeda beda, yang dapat dilihat pada Gambar 1.



Gambar 1. Use Case

**B. Implementasi Penyimpanan Data**

Implementasi penyimpanan data untuk *website* Maranatha-Keimyung Korea Center dapat dilihat pada Gambar 2.



Gambar 2. Desain Penyimpanan Data

Implementasi *Java Persistence* pada sistem untuk mengakses *database* dapat dilihat pada Kode Program 4.

```

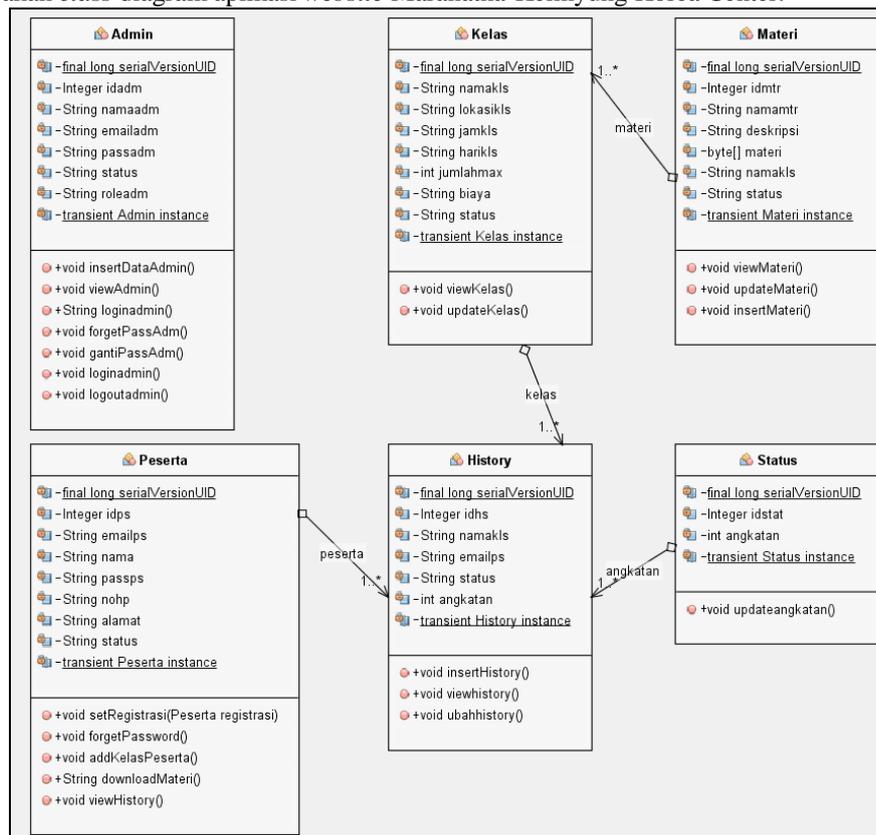
1. <?xml version="1.0" encoding="UTF-8"?>
2. <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001
   /XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
   http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
3.   <persistence-unit name="NKoreaCenter2_Db4freePU" transaction-type="JTA">
4.     <jta-data-source>java:app/d88</jta-data-source>
5.     <class>entity.Admin</class>
6.     <class>entity.History</class>
7.     <class>entity.Kelas</class>
8.     <class>entity.Materi</class>
9.     <class>entity.Peserta</class>
10.    <class>entity.Status</class>
11.    <exclude-unlisted-classes>>false</exclude-unlisted-classes>
12.    <properties>
13.      <property name="javax.persistence.schema-generation.database.action" value="create"/>
14.    </properties>
15.  </persistence-unit>
16. </persistence>

```

Kode Program 4. Kode Program Java Persistence

C. Implementasi Class/Modul

Gambar 3 merupakan *class diagram* aplikasi *website* Maranatha-Keimyung Korea Center.



Gambar 3. Class Diagram

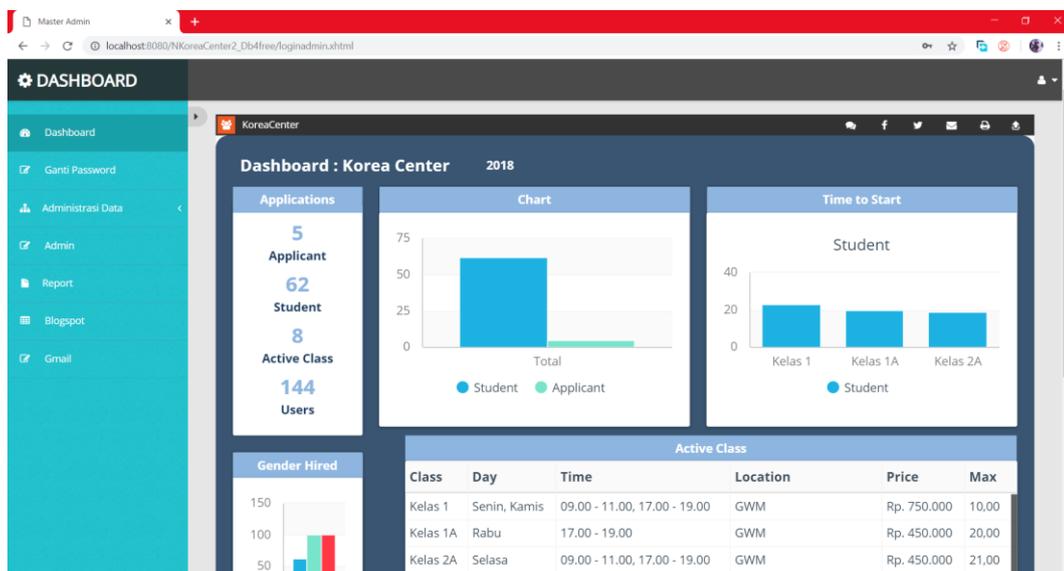
#### D. Implementasi Antar Muka

Halaman *home* merupakan halaman pertama yang ditampilkan untuk *user* saat pertama aplikasi dibuka. Pada halaman *home* terdapat menu-menu seperti *Home*, *News*, *Sign up*, *Login*, *Admin*, *Register Now*. Gambar 4 merupakan implementasi desain antarmuka untuk tampilan *home*.



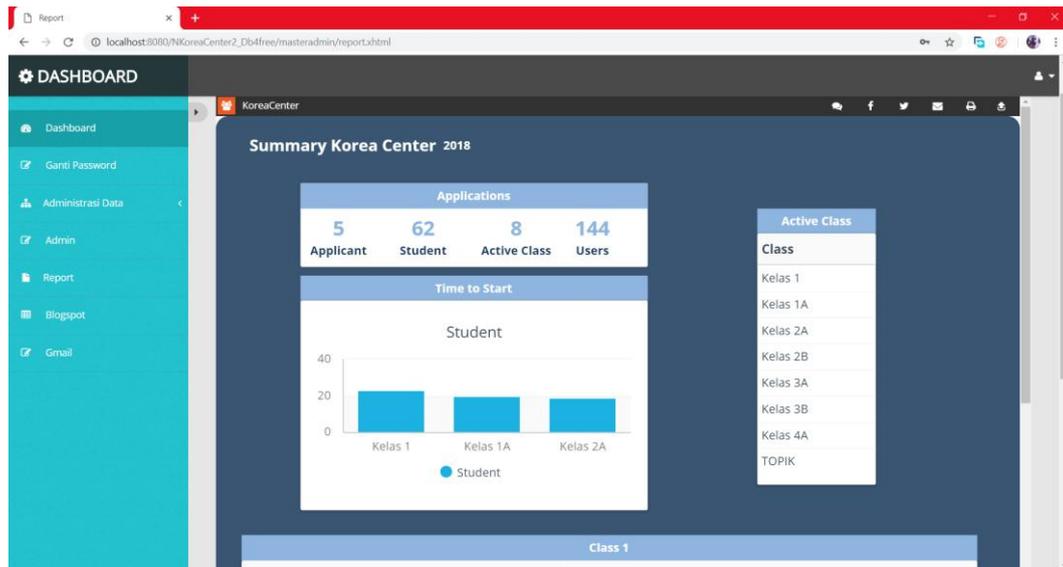
Gambar 4. Implementasi Tampilan Home

Halaman *Dashboard Admin* merupakan halaman pertama yang ditampilkan untuk *user* saat telah melakukan login. Pada halaman tersebut terdapat menu-menu seperti *dashboard*, *administrasi data*, *report*, *blogspot*, *gmail*, *logout*. Gambar 5 merupakan implementasi desain antarmuka untuk tampilan *Dashboard Admin*.



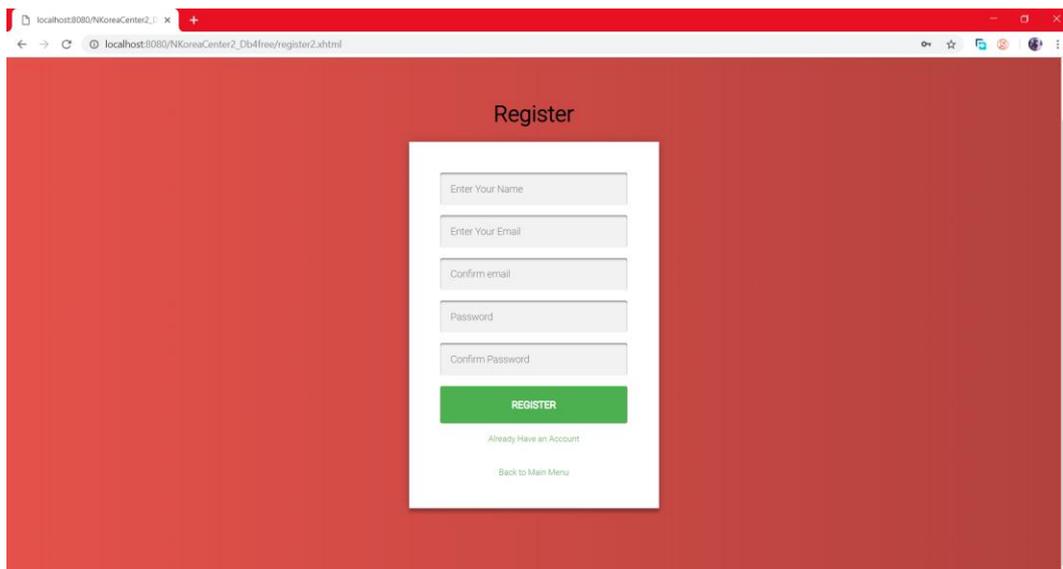
Gambar 5. Implementasi Tampilan Dashboard Admin

Sistem ini menyediakan halaman *Report* untuk membantu Korea Center melihat laporan keseluruhan. Halaman *Report* dapat dilihat pada Gambar 6. Laporan yang disediakan menunjukkan rangkuman keseluruhan dari kelas, peserta, dan pendaftar Korea Center.



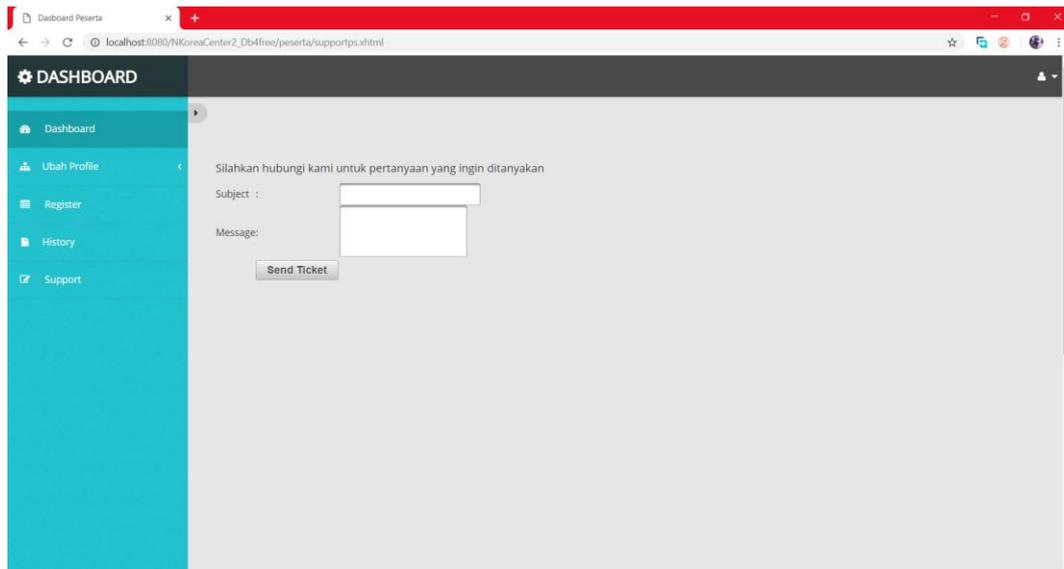
Gambar 6 Implementasi Halaman Report Admin

Halaman *Sign up* merupakan halaman yang ditampilkan untuk *user* saat menekan tombol *Sign up*. Pada halaman tersebut terdapat menu-menu seperti *register*, *already have an account*, *back to main menu*. Gambar 7 merupakan implementasi desain antarmuka untuk tampilan *Sign up*.



Gambar 7. Implementasi Tampilan Sign Up

Halaman *Support* Peserta merupakan halaman yang ditampilkan untuk *user* saat menekan tombol *support*. Pada halaman tersebut terdapat menu-menu seperti *download*, *dashboard*, *ubah profile*, *register*, *history*, *support*, *logout*. Gambar 8 merupakan implementasi desain antarmuka untuk tampilan *Support* Peserta. Setelah melakukan registrasi, email notifikasi akan dikirimkan pada peserta bahwa telah berhasil melakukan registrasi.



Gambar 8. Implementasi Tampilan Support Peserta

Halaman *Support* Peserta merupakan halaman yang ditampilkan untuk *user* saat menekan tombol *support*. Pada halaman tersebut terdapat menu-menu seperti *download*, *dashboard*, ubah profile, *register*, *history*, *support*, *logout*. Gambar 8 merupakan implementasi desain antarmuka untuk tampilan *Support* Peserta. Setelah melakukan pengiriman *ticket* akan ada notifikasi pada admin melalui *email*.

#### IV. PENGUJIAN

Dalam penelitian ini juga terdapat dua cara pengujian, yaitu *black box* dan dengan kuesioner. Dalam pengujian *black box*, hasilnya semua sudah sesuai dengan skenario yang ada. Sedangkan pengujian dengan kuesioner telah dilakukan untuk 10 orang sebagai pengguna dan 3 orang sebagai admin. Secara umum hasil kuesioner menyatakan sudah lebih dari 50% yang menyetujui bahwa sistem ini sudah baik, mudah digunakan, dan membantu proses administrasi.

Gambar 9 menunjukkan hasil kuesioner terkait fitur-fitur yang disediakan untuk *role* Pengguna. Secara umum, fitur yang ada mendapatkan respon yang baik dari pengisi kuesioner.

Bagaimana pendapat anda dalam penggunaan Fitur



Gambar 9. Hasil Kuesioner terkait Fitur Pengguna

Gambar 10 menunjukkan hasil kuesioner terkait fitur-fitur yang disediakan untuk *role* Admin. Secara umum juga pengisi kuesioner menilai fitur yang disediakan untuk Admin telah baik. Fitur yang disediakan untuk Admin lebih detail karena ada pengolahan data lebih banyak.



Gambar 10. Hasil Kuesioner terkait Fitur Admin

## V. KESIMPULAN

Pada penelitian ini telah dibuat sebuah *website* Maranatha-Keimyung Korea Center dengan memanfaatkan teknologi JavaServerFaces dan *Email Gateway*. Sistem ini dapat mengelola data peserta, data pegawai, data pengajar, dan data materi. *Website* ini juga dapat membantu mempermudah calon peserta melakukan pendaftaran sehingga tidak perlu datang ke kantor Korea Center untuk mengambil formulir berdasarkan kuisisioner peserta pada nomor 3. Dalam *website* ini juga dapat memberikan notifikasi kelas penuh, penerimaan peserta, pendaftaran peserta, serta pengiriman informasi melalui *email*.

## UCAPAN TERIMA KASIH

Dalam pelaksanaan penelitian ini, penulis mengucapkan terima kasih kepada Maranatha-Keimyung Korea Center yang telah memberikan bantuan serta kerja sama dalam pembuatan penelitian berikut ini. Serta ucapan terima kasih diucapkan sebesar-besarnya kepada para pengajar dan staf Fakultas Teknologi Informasi, Prodi Sistem Informasi Universitas Kristen Maranatha Bandung atas dukungan dan bantuannya dalam penyusunan laporan penelitian ini.

## DAFTAR PUSTAKA

- [1] A. Gupta, Java EE 7 Essential, O'REILLY, 2013.
- [2] A. Saputra, R. T. Subagio and S. , Membangun Aplikasi E-Library untuk Panduan Skripsi, Jakarta: PT. Elex Media Komputindo, 2012.
- [3] E. R. Harold, JavaMail API, O'Reilly Media, Inc., 2013.
- [4] R. V. Imbar and B. R. Suteja, Pemrograman Web Commerce dengan Oracle dan ASP, Bandung: Informatika Bandung, 2006.
- [5] W. Wisbisono and F. Baskoro, "Pengujian Perangkat Lunak Dengan Menggunakan Model Behavior UML," vol. 1, p. 8, 2002.
- [6] C. B. Susila and R. A. Triyono, "Sistem Informasi Nilai Mahasiswa Berbasis SMS Gateway Pada Sekolah Tinggi Ilmu Tarbiyah (STIT) Muhammadiyah Pacitan," *jutisi.maranatha.edu*, p. 8, 2015.
- [7] A. Wijayanto and C. P. , "PEMANFAATAN SMS GATEWAY DAN EMAIL GATEWAY UNTUK PENGIRIMAN INFORMASI DI PT. SEMESTA CITRA DANA," *jutisi.maranatha.edu*, p. 7, 2008.
- [8] J. C. W and M. R. F, "Pengembangan Sistem Informasi Penjadwalan dan Manajemen Keuangan Kegiatan Seminar dan Sidang Skripsi/Tugas Akhir (Studi Kasus Program Studi Sistem Informasi UNIKOM)," *jutisi.maranatha.edu*, p. 9, 2017.
- [9] J. A. Sihite and M. C. , "Sistem Informasi Pengelolaan Daftar Kehadiran dan Jadwal Pembicara Berbasis Website Dengan Mail Gateway Pada Komunitas JC Bandung," *jutisi.maranatha.edu*, p. 20, 2017.
- [10] M. Flower, UML Distilled Edisi 3, Yogyakarta: Andi Publisher, 2005.