

RANCANG BANGUN APLIKASI PENJADWALAN KEGIATAN AKADEMIK DISERTAI SISTEM REMINDER BERBASIS RESPONSIVE WEB DESIGN

Nopi Ramsari¹, Achmad Rifaldi²

^{1,2}Program Studi Teknik Informatika Universitas Nurtanio Bandung
Jl. Pajajaran 219 Husein Sastranegara Bandung
email: ¹nopiramsarihata@gmail.com, ²achmadrifaldi64@gmail.com

ABSTRAK

Fakultas Ilmu Komputer dan Informatika (FIKI) merupakan salah satu fakultas yang ada di Universitas Nurtanio Bandung. Fakultas memiliki kalender akademik untuk mengatur jadwal kegiatan akademik baik dalam jangka pendek maupun jangka panjang. Tingginya mobilitas dan tidak adanya *reminder* terkadang membuat dosen dan pejabat fakultas melupakan kegiatan akademik yang telah dijadwalkan. Berdasarkan hal tersebut, penulis mencoba membangun suatu aplikasi penjadwalan kegiatan akademik dengan memanfaatkan teknik Email Blast, SMS Blast dan Notification sebagai bentuk model pengingat agar dapat membantu meningkatkan keefektifitasan kegiatan. Aplikasi ini dibuat dengan gabungan framework Ruby on Rails, Postfix dan *Application Program Interface* (API) yang disediakan oleh layanan Twillio dan Onesignal. Twillio menyediakan API yang dapat digunakan untuk mengirimkan SMS reminder. Onesignal menyediakan API yang dapat digunakan untuk mengirimkan *reminder* dalam bentuk notifikasi. Sedangkan postfix digunakan mengirimkan reminder dalam bentuk email. Dalam pembangunan aplikasi ini, penulis menggunakan metode *Rapid Application Development* (RAD) dan metode *Responsive Web Design* (RWD) dalam pembuatan tampilan antarmuka sehingga memudahkan pengguna untuk mengakses aplikasi di berbagai perangkat. Secara garis besar aplikasi ini berbentuk kalender akademik disertai dengan sistem reminder. Adanya manajemen pengguna dan hak akses pengguna, membuat aplikasi ini sangat cocok digunakan untuk suatu organisasi atau instansi. Berdasarkan hasil implementasi, aplikasi penjadwalan kegiatan akademik dengan sistem reminder ini dapat mempermudah proses pencatatan jadwal dan pengingat kegiatan akademik sehingga dapat meningkatkan keefektifitasan kegiatan.

Kata kunci: penjadwalan, kegiatan akademik, *reminder*, Ruby on Rails, email, *Short Message Service* (SMS)

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Fakultas Ilmu Komputer dan Informatika (FIKI) merupakan salah satu fakultas yang ada di Universitas Nurtanio Bandung. Fakultas memiliki kalender akademik untuk mengatur jadwal kegiatan akademik baik dalam jangka pendek maupun jangka panjang. Kalender akademik ini memiliki penjadwalan administrasi akademik, administrasi keuangan, rapat kerja, ujian tengah semester, ujian akhir semester, skripsi, magang dan penjadwalan kegiatan lain. Penempatan Teknologi *reminder* pada kalender akademik akan sangat baik dilakukan untuk meningkatkan keefektifitasan kegiatan.

Dalam hal mengingat jadwal kegiatan akademik, pejabat fakultas dan dosen FIKI

Universitas Nurtanio Bandung masih menggunakan cara-cara konvensional untuk mencatat dan mengingat kegiatan akademik yang harus dilakukan sehari-hari, seperti menulis di kertas dan di tempelkan di meja, menggunakan alarm dan menggunakan aplikasi kalender di ponsel. Tingginya mobilitas membuat cara-cara seperti ini menjadi kurang efisien bagi pejabat fakultas dan dosen, karena harus membuat pengingat setiap kegiatan yang harus dilakukan. Semakin banyak kegiatan yang ingin diingatkan semakin banyak pula yang harus dicatat atau dimasukkan oleh masing-masing orang. Bahkan tidak jarang pejabat fakultas dan dosen melupakan kegiatan rapat karena lupa membuat pengingat, dan kadangkala dosen terlambat memberikan hasil evaluasi belajar mahasiswa karena tidak adanya *reminder* akan hal tersebut. Oleh karena itu,

diperlukan aplikasi penjadwalan kegiatan akademik yang dapat mengingatkan seluruh kegiatan tanpa memerlukan keterlibatan aktif pejabat fakultas dan dosen setiap kali memasukan jadwal kegiatan yang ingin diingatkan.

Dari permasalahan diatas maka dibutuhkan suatu aplikasi penjadwalan disertai *reminder* yang mempermudah proses pencatatan dan pengingat kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan di atas, masalah dapat diidentifikasi, yaitu pejabat fakultas dan dosen FIKI Universitas Nurtanio Bandung masih menggunakan cara-cara konvensional untuk mengingat kegiatan akademik, hal ini dapat dilihat dari cara mereka mengingat dan mencatat kegiatan akademik, seperti menulis di kertas, ditempelkan di meja, menggunakan alarm dan menggunakan aplikasi kalender di ponsel. Tingginya mobilitas membuat cara-cara konvensional untuk mengingat kegiatan akademik menjadi kurang efektif. Tidak jarang pejabat fakultas dan dosen FIKI lupa untuk membuat pengingat karena semakin banyak kegiatan yang ingin diingatkan semakin banyak pula yang harus dicatat oleh masing-masing orang.

1.3 Rumusan Masalah

Berdasarkan identifikasi permasalahan diatas, maka dapat diambil rumusan “Bagaimana membuat aplikasi penjadwalan disertai sistem *reminder* untuk mempermudah proses pencatatan dan pengingat kegiatan akademik FIKI Universitas Nurtanio Bandung?”

1.4 Batasan Masalah

Berdasarkan masalah-masalah yang telah dikemukakan di atas, pokok-pokok permasalahan yang menjadi batasan pembahasan penelitian ini supaya pembahasan masalah yang dilakukan dapat terarah dengan baik dan tidak menyimpang dari pokok permasalahan, maka penulis membatasi permasalahan yang akan dibahas, yakni:

1. Fokus pada perancangan aplikasi penjadwalan kegiatan disertai sistem *reminder*.

2. Kegiatan akademik yang dibahas adalah kegiatan akademik di Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung.
3. *Reminder* menggunakan teknik *EmailBlast*, *SMSBlast* dan *Notification*.
4. Sistem *Reminder* menggunakan aplikasi *third-party* seperti Twilio, NusaSMS dan Onesignal untuk mengirimkan notifikasi.

1.5 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini, yaitu membangun aplikasi penjadwalan dengan sistem *reminder* untuk kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung berbasis *Responsive Web Design* menggunakan *framework* Ruby on Rails.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data Observasi

Teknik pengumpulan data dengan melakukan pengamatan langsung ke lapangan untuk mendapat-kan data yang diperlukan untuk membangun apli-kasi web yang menarik dan sesuai dengan kebutuhan dan keadaan yang sebenarnya.

Interview

Teknik pengumpulan data dengan mengadakan tanya jawab dengan pihak fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung guna mendapatkan data-data serta keterangan yang dibutuhkan.

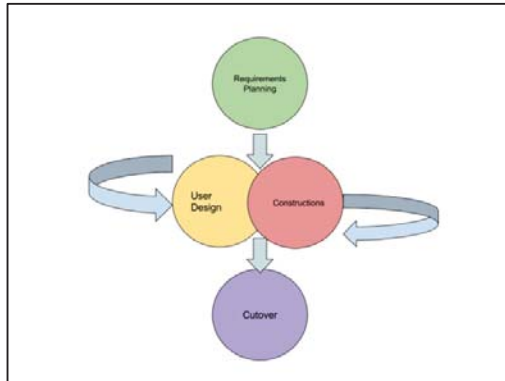
Studi Literatur

Teknik pengumpulan data dengan menggunakan atau mengambil informasi dari berbagai jurnal, buku panduan dan bacaan-bacaan lainnya, serta website yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis yang menunjang penelitian.

1.6.2 Pengembangan Sistem Aplikasi

Metode pengembangan perangkat lunak yang digunakan untuk membuat aplikasi penjadwalan ini yaitu dengan menggunakan *Rapid Application Development* (RAD). RAD merupakan metode yang memfokuskan pada kecepatan dalam pengembangan sistem untuk memenuhi kebutuhan pengguna atau pemilik

sistem seperti *prototyping* namun mempunyai cakupan yang lebih luas.



Gambar 1.1 Empat fase dari model RAD

2. LANDASAN TEORI

2.1 *Reminder*

Reminder adalah sebuah pesan yang menolong se-seorang untuk mengingatkan sesuatu. *Reminder* dapat lebih bermanfaat ketika informasi konteks-tual digunakan untuk menyajikan informasi pada waktu yang tepat dan tempat yang tepat. *Reminder* dapat digunakan dalam manajemen waktu yang berfungsi untuk memberi alarm peringatan berupa pemberitahuan berbasis lokasi, waktu maupun catatan yang bersifat kontekstual. (Indri, dkk, 2015:15)

2.2 SMS

Short Message Service (SMS) adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi telepon selular, dalam hal ini perangkat nirkabel yang digunakan adalah telepon selular. Salah satu kelebihan dari SMS adalah biaya yang murah. Selain itu SMS merupakan metode *store* dan *forward* sehingga keuntungan yang didapat adalah pada saat telepon selular penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar *service* area, penerima tetap dapat menerima SMS-nya apabila telepon selular tersebut sudah aktif kembali. SMS menyediakan mekanisme untuk mengirimkan pesan singkat dari dan menuju media-media wireless dengan menggunakan sebuah *Short Messaging Service Center* (SMSC), yang bertindak sebagai sistem yang berfungsi menyimpan dan mengirimkan kembali pesan-pesan singkat. Jaringan wireless menyediakan

mekanisme untuk menemukan station yang dituju dan mengirimkan pesan singkat.

Antara SMSC dengan *wireless station*. SMS mendukung banyak mekanisme input sehingga memungkinkan adanya interkoneksi dengan berbagai sumber dan tujuan pengiriman pesan yang berbeda (Pakpahan dalam Pradana, 2013:10).

2.3 *Email*

Electronic mail atau biasa disingkat sebagai *email*, merupakan sebuah metode untuk mengirimkan pesan dalam bentuk digital. Pesan ini biasanya dikirimkan melalui medium internet. Sebuah pesan elektronis terdiri dari isi, alamat pengirim, dan alamat-alamat yang dituju.

Sistem email beroperasi di atas jaringan berbasis pada model *store* dan *forward*. Sistem ini mengaplikasikan sebuah sistem server email yang menerima, meneruskan, mengirimkan, serta menyimpan pesan-pesan user. User hanya perlu mengkoneksikan PC ke dalam jaringan.

Email dapat dianalogikan dengan kotak surat yang ada di kantor pos, sedangkan server email dapat diibaratkan sebagai kantor pos. Dengan analogi ini, sebuah mail server dapat memiliki banyak akun email yang ada di dalamnya. (Haryanto, 2012:127)

2.4 *Application Program Interface* (API)

Application Programming Interface (API) adalah sebuah teknologi untuk memfasilitasi pertukaran informasi atau data antara dua atau lebih aplikasi perangkat lunak. API adalah antarmuka virtual antara dua fungsi perangkat lunak yang saling bekerja sama, seperti antara sebuah *word processor* dan *spreadsheet*. Sebuah API mendefinisikan bagaimana cara programmer memanfaatkan suatu fitur tertentu dari sebuah komputer. API tersedia untuk sistem windowing, sistem file, sistem basis data dan sistem jaringan.

Pada proses pengembangan sebuah perangkat lunak baik API pada khususnya maupun perangkat lunak lain memerlukan kaidah-kaidah yang perlu dipahami dan diterapkan. API yang baik adalah API yang memiliki sifat-sifat sebagai berikut:

1. Mudah untuk dipelajari.

2. Mudah untuk digunakan, bahkan bila tanpa dokumentasi yang menyertainya.
3. Susah untuk disalahgunakan.

Berkinerja tinggi dalam menyelesaikan tugasnya.

Mudah untuk dikembangkan lebih lanjut. (Yusaindera, 2017)

2.5 Responsive Web Design

Responsive Web Design (RWD) adalah pendekatan pengembangan web yang menciptakan perubahan dinamis pada tampilan situs web, bergantung pada ukuran dan orientasi layar perangkat yang digunakan untuk melihatnya. RWD adalah salah satu pendekatan terhadap masalah perancangan untuk banyak perangkat yang tersedia bagi pelanggan, mulai dari telepon kecil sampai monitor desktop besar.

RWD menggunakan apa yang disebut breakpoint untuk menentukan bagaimana tata letak sebuah situs akan muncul: satu desain digunakan di atas breakpoint dan desain lain diterapkan di bawah breakpoint tersebut. Breakpoint biasanya didasarkan pada lebar browser.

HTML yang sama disajikan ke semua perangkat, menggunakan CSS (yang menentukan tata letak halaman web) untuk mengubah tampilan halaman. Alih-alih membuat situs terpisah dan basis kode yang sesuai untuk monitor layar lebar, desktop, laptop, tablet, dan ponsel dengan segala ukuran, basis kode tunggal dapat mendukung pengguna dengan viewports dengan ukuran berbeda.

Dalam desain responsif, elemen halaman dirombak saat area pandang tumbuh atau menyusut. Desain desktop tiga kolom dapat dirombak menjadi dua kolom untuk tablet dan satu kolom untuk sebuah smartphone. Desain responsif bergantung pada grid berbasis proporsi untuk mengatur ulang elemen konten dan desain. (Amy Schade, 2014)

2.6 Ruby on Rails

Ruby adalah bahasa pemrograman open source dinamis berbasis skrip yang berorientasi obyek. Tujuan dari Ruby adalah menggabungkan kelebihan dari semua bahasa-bahasa pemrograman skrip yang ada didunia. Di dalam bahasa pemrograman Ruby, semua adalah

objek. Setiap informasi dan kode bisa diberi property dan action. Pemrograman berorientasi objek memanggil property dengan nama variabel instan dan *action*, yang disebut sebagai metode.

Ruby on Rails adalah sebuah *framework* pengembangan web yang ditulis dalam bahasa pemrograman Ruby. Ruby on Rails 100% open source, tersedia dibawah Lisensi MIT, dan akibatnya tidak ada biaya untuk mengunduh dan menggunakan.

Rails juga memberikan banyak keberhasilan pada desainnya yang elegan dan praktis; Dengan memanfaatkan kemudahan bahasa Ruby yang mendasarinya, Rails secara efektif membuat domain bahasa khusus untuk menulis aplikasi web. Sebagai konsekuensinya, banyak tugas pemrograman web yang umum seperti menghasilkan HTML, membuat model data, dan URL routing mudah dilakukan dengan Rails, dan kode aplikasi yang dihasilkanpun ringkas dan mudah dibaca. (Michael Hartl, 2015:23)

2.7 Postgresql

PostgreSQL adalah sistem basis data relasional open source yang powerfull. PostgreSQL memiliki lebih dari 15 tahun pengembangan aktif dan arsitektur yang telah terbukti yang telah mendapatkan reputasi yang kuat untuk keandalan, integritas data, dan kebenaran. PostgreSQL berjalan hampir pada semua sistem operasi, termasuk Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, macOS, Solaris, Tru64), dan Windows. PostgreSQL sepenuhnya memiliki dukungan penuh untuk *foreign keys*, *joins*, *views*, *triggers*, dan *stored procedures* (dalam banyak bahasa).

PostgreSQL mencakup sebagian besar tipe data SQL 2008, termasuk INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, dan TIMESTAMP. PostgreSQL juga mendukung penyimpanan binary large objects, termasuk gambar, suara, atau video. PostgreSQL memiliki antarmuka pemrograman asli untuk C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, dan dokumentasi yang luar biasa. (Postgresql, 2017)

2.8 Postfix

Postfix adalah sebuah program yang digunakan untuk mengirimkan email (MTA). Postfix menyediakan sebuah alternatif untuk program sendmail yang lebih sederhana dalam desain, lebih modular, lebih mudah dikonfigurasi, sedikit usaha untuk mengelola. Sama pentingnya, Postfix dirancang dengan skalabilitas, keandalan dan keamanan sebagai persyaratan mendasar. (Mick, dkk, 2000)

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Hal-hal yang akan dianalisis pada tahap analisis sistem ini adalah analisis prosedur sistem yang sedang berjalan, analisis masalah, analisis prosedur, dan analisis kebutuhan sistem.

3.1.1 Analisis Sistem yang sedang berjalan

Alur kerja dari sistem yang sedang berjalan dilakukan dengan cara sebagai berikut:

1. Bagian Tata Usaha membuat agenda untuk kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung.
2. Setelah agenda dibuat, agenda akan diinformasikan kepada pejabat fakultas dan dosen FIKI Universitas Nurtanio Bandung.
3. Pejabat fakultas dan dosen FIKI Universitas Nurtanio Bandung menerima informasi agenda dari bagian Tata Usaha.

3.1.2 Analisis Sistem yang diusulkan

Alur kerja dari sistem yang diusulkan dilakukan dengan cara sebagai berikut:

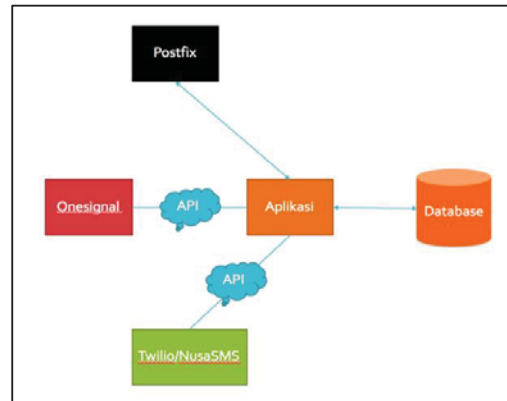
1. Bagian staf TI dapat mengelola data *role* dan *privilege user*.
2. Bagian Tata Usaha dapat mengelola data *user*.
3. Bagian Tata Usaha dapat mengelola data *event* kegiatan akademik.
4. Ketika event dibuat maka event tersebut akan muncul sebagai agenda kegiatan

akademik untuk pejabat fakultas dan dosen FIKI Universitas Nurtanio.

5. Aplikasi akan mengirimkan *reminder* jika waktu mulai *event* kurang dari atau sama dengan dua jam.

3.1.3 Gambaran Umum Aplikasi

Berikut ini adalah gambaran umum aplikasi penjadwalan disertai sistem *reminder* yang akan dibuat.



Gambar 3.1 Gambaran umum aplikasi

3.1.4 Kebutuhan fungsional

Analisis kebutuhan fungsional pada aplikasi penjadwalan dengan sistem reminder ini adalah sebagai berikut:

Tabel III.1 Tabel Kebutuhan Fungsional

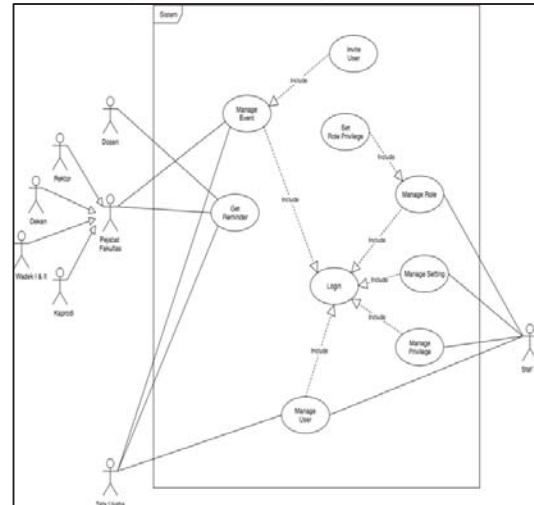
No	Kebutuhan Fungsional	Deskripsi
1.	<i>Login</i>	Jika <i>login valid</i> aplikasi menyediakan layanan sesuai otorisasi pekerjaannya.
2.	<i>Role Management</i>	<i>Role Management</i> merupakan fungsi untuk mengatur peran pengguna pada aplikasi.
	1. <i>Create Role</i>	Contoh peran pada aplikasi yaitu: Staf TI, Tata Usaha, Pejabat Fakultas, dan Dosen.
	2. <i>View Role</i>	
	3. <i>Update Role</i>	
	4. <i>Delete Role</i>	
	5. <i>Set Role Privilege</i>	

3.	<i>Privilege Management</i>	<i>Privilege Management</i> merupakan fungsi untuk mengatur akses pengguna kedalam aplikasi.
	1. <i>Create Privilege</i> 2. <i>View Privilege</i> 3. <i>Update Privilege</i> 4. <i>Delete Privilege</i>	Setiap <i>role</i> /peran memiliki hak akses yang berbeda, dan hal ini dapat diatur didalam fungsi <i>Role Management</i> . Aplikasi menyediakan fasilitas untuk membuat, melihat, memperbaharui, dan menghapus data <i>privilege</i>
4.	<i>User Management</i>	Aplikasi menyediakan fasilitas untuk membuat, melihat, memperbaharui, dan menghapus data <i>user</i>
	1. <i>Create User</i> 2. <i>View User</i> 3. <i>Update User</i> 4. <i>Delete User</i>	
5.	<i>Event Management</i>	<i>Event</i> merupakan kegiatan akademik yang telah dijadwalkan pada tanggal tertentu.
	1. <i>View Event</i> 2. <i>Create Event</i> 3. <i>Update Event</i> 4. <i>Delete Event</i> 5. <i>Invite User</i>	Aplikasi menyediakan fasilitas untuk membuat, melihat, memperbaharui, dan menghapus data <i>event</i> . Aplikasi menyediakan fasilitas untuk mengundang <i>user</i> kedalam <i>event</i> tertentu
6.	<i>Reminder</i>	Aplikasi menyediakan fasilitas untuk mengirimkan pengingat kepada <i>user</i> yang bersangkutan.
7.	<i>Setting Management</i>	Aplikasi menyediakan fitur untuk merubah pengaturan dasar aplikasi
	1. <i>View Setting</i> 2. <i>Update Setting</i>	

3.1.5 Diagram Use Case

Diagram *use case* memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor

dengan *use case* dalam sistem. Salah satu manfaat dari diagram *use case* adalah untuk komunikasi. Calon pengguna sistem dapat mengamati diagram *use case* untuk mendapatkan pengalaman yang utuh tentang sistem yang akan dikembangkan.

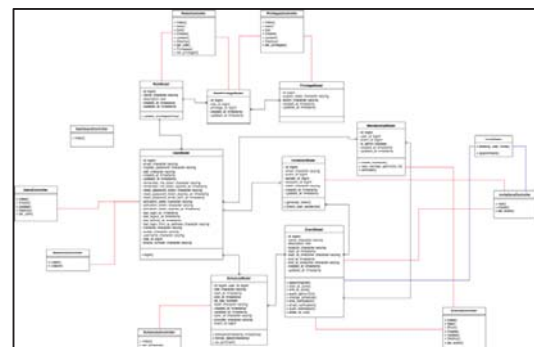


Gambar 3.2 Diagram use case

Pada bagian ini akan dijelaskan masing-masing deskripsi dari *use case* diagram yang dilakukan oleh Staf TI, Tata Usaha, Pejabat Fakultas, dan Dosen.

3.1.6 Class Diagram

Sebuah sistem biasanya mempunyai beberapa *Class Diagram*. *Diagram Class* menggambarkan struktur statis kelas di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem.



Gambar 3.3 Diagram class

Gambar 3.3 merupakan sekumpulan kelas di dalam aplikasi penjadwalan kegiatan akademik yang saling berhubungan.

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Bagian ini akan menjelaskan mengenai batasan implementasi, lingkungan perangkat keras dan perangkat lunak, serta implementasi yang dilakukan berdasarkan hasil pada tahap perancangan. Penjelasan pada subbab ini meliputi lingkungan pengembangan yang digunakan, batasan implementasi dan proses hasil implementasi.

4.1.1 Implementasi Kelas

Setiap kelas pada aplikasi diimplementasikan dalam bahasa ruby, dimana setiap kelas utama akan diimplementasikan dalam sebuah file. Kelas yang diimplementasikan mengacu pada perancangan yang telah dilakukan.

Kelas model

Tabel IV.1 Daftar Implementasi Kelas Model

Nama File Fisik	Nama Kelas
event.rb	Event
invitation.rb	Invitation
membership.rb	Membership
privilege.rb	Privilege
role.rb	Role
role_privilege.rb	RolePrivilege
schedule.rb	Schedule
user.rb	User
setting.rb	Setting
device.rb	Device

Kelas controller

Tabel IV.2 Daftar Implementasi Kelas Controller

Nama File Fisik	Nama Kelas
events_controller.rb	EventsController
invitations_controller.rb	InvitationsController
password_resets_controller.rb	PasswordResetsController
privileges_controller.rb	PrivilegesController
roles_controller.rb	RolesController
sessions_controller.rb	SessionsController
schedules_controller.rb	SchedulesController
users_controller.rb	UsersController
dashboard_controller.rb	DashboardController
settings_controller.rb	SettingController

Kelas mailer

Tabel IV.3 Daftar Implementasi Kelas Mailer

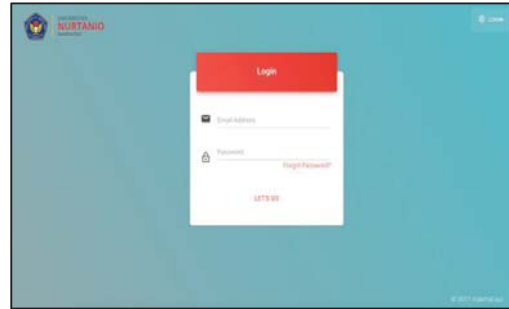
Nama File Fisik	Nama Kelas
invite_mailer.rb	InviteMailer
user_mailer.rb	UserMailer

4.1.2 Implementasi antar muka

Setiap kelas pada aplikasi diimplementasikan dalam bahasa ruby, dimana setiap kelas utama

akan diimplementasikan dalam sebuah file. Kelas yang diimplementasikan mengacu pada perancangan yang telah dilakukan.

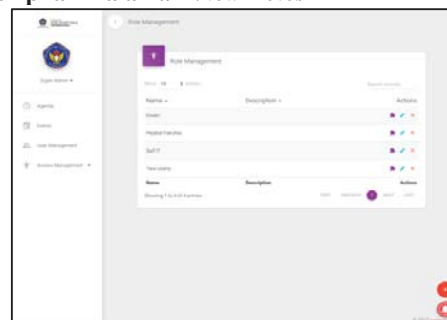
Tampilan Halaman Login



Gambar 4.1 Halaman Login

Pada gambar 4.1 merupakan implementasi halaman login dan menunjukkan bahwa untuk melakukan login dan masuk kedalam aplikasi, pengguna harus menginputkan email dan password yang valid terlebih dahulu.

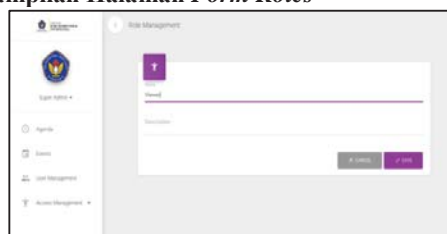
Tampilan Halaman View Roles



Gambar 4.2 Halaman view roles

Pada gambar 4.2 merupakan implementasi halaman roles management yang menampilkan seluruh daftar role yang ada didalam aplikasi penjadwalan. Dalam halaman ini terdapat fasilitas tambah role untuk menambahkan data role.

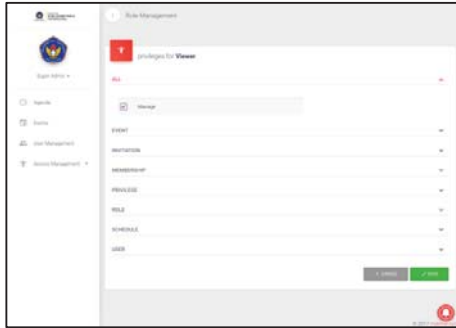
Tampilan Halaman Form Roles



Gambar 4.3 Halaman form role management

Pada gambar 4.3 merupakan implementasi halaman *formroles management* yang digunakan untuk menambahkan data *role*. Pengguna dapat menyimpan data *role* setelah menginputkan nama dan deskripsi *role*.

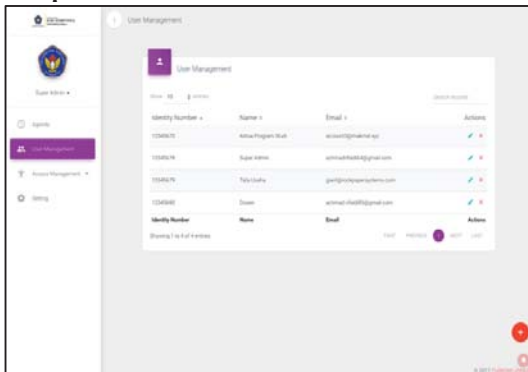
Tampilan set role privileges



Gambar 4.4 Halaman set role privilege

Pada gambar 4.4 merupakan implementasi halaman *set role privilege* yang digunakan untuk menambahkan mengatur hak akses *role* terhadap fitur-fitur yang ada didalam aplikasi.

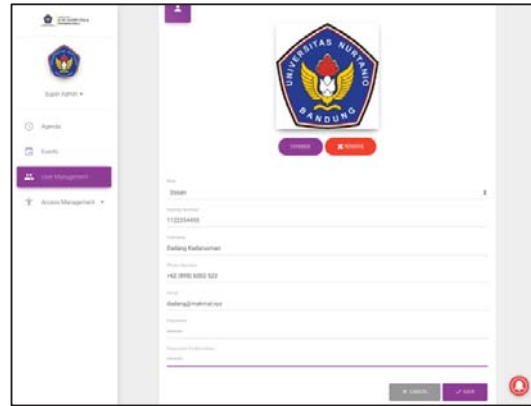
Tampilan halaman view users



Gambar 4.5 Halaman view users

Pada gambar 4.5 merupakan implementasi halaman roles management yang menampilkan seluruh data pengguna yang ada didalam aplikasi penjadwalan. Dalam halaman ini terdapat fasilitas tambah user untuk menambahkan data user

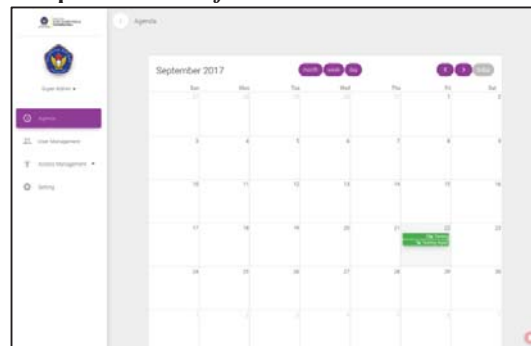
Tampilan halaman form users



Gambar 4.6 Halaman form user

Pada gambar 4.6 merupakan implementasi halaman *formuser management* yang digunakan untuk menambahkan data *user*. Hanya user dengan *role* Staf TI atau Tata Usaha saja yang dapat mengakses halaman *form user*.

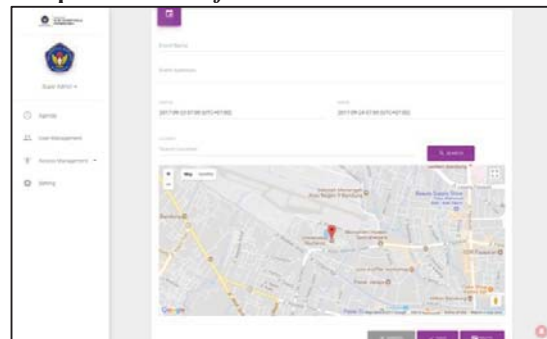
Tampilan halaman form users



Gambar 4.7 Halaman view agenda

Pada gambar 4.7 merupakan implementasi halaman *view agenda* yang menampilkan seluruh kegiatan yang sudah diagendakan.

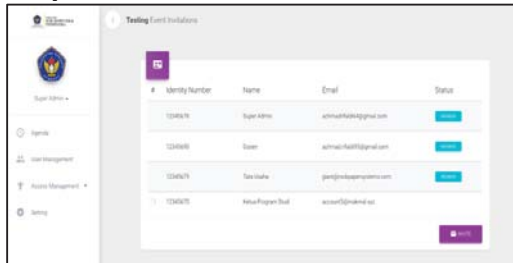
Tampilan halaman form event



Gambar 4.8 Halaman form event

Pada gambar 4.8 merupakan implementasi halaman *form event* yang digunakan untuk menambahkan data *event*. Hanya user dengan *role* Staf TI, Tata Usaha atau Pejabat Fakultas saja yang dapat mengakses halaman *form event*

Tampilan halaman *invite user*



Gambar 4.9 Halaman *invite user*

Pada gambar 4.9 merupakan implementasi halaman *invite user* yang menampilkan seluruh data pengguna yang ada didalam aplikasi penjadwalan. *User* pemilik *event* dapat mengundang *user* lain kedalam *event*-nya.

Tampilan *SMS Reminder*

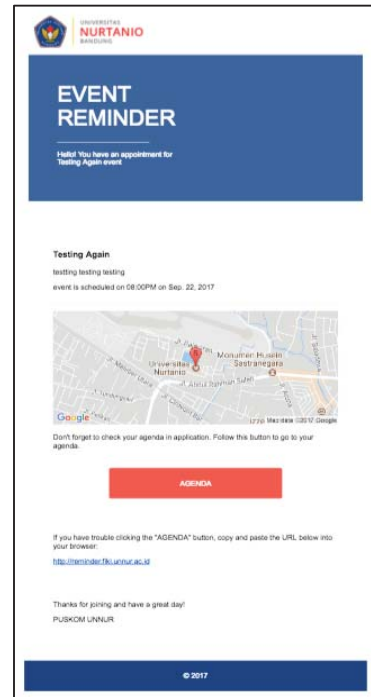
Tampilan ini merupakan tampilan pesan yang diterima oleh user yang dikirim oleh layanan NusaSMS dua jam sebelum *event* dimulai.



Gambar 4.10 Tampilan *reminder via SMS*

Tampilan *Email Reminder*

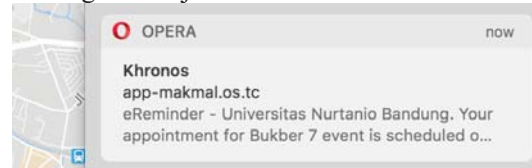
Tampilan ini merupakan tampilan email yang diterima oleh user yang dikirim dua jam sebelum *event* dimulai.



Gambar 4.11 Tampilan *reminder via email*

Tampilan *Popup Notification*

Tampilan ini merupakan tampilan pesan yang diterima oleh user dalam bentuk popup notification yang dikirim oleh layanan OneSignal dua jam sebelum *event* dimulai.



Gambar 4.12 Tampilan *reminder via popup notification*

4.2 Pengujian

Bagian ini akan memuat penjelasan mengenai pengujian terhadap perangkat lunak yang telah diimplementasikan. Metode pengujian yang dipakai adalah metode *black-box* yaitu pengujian yang memperhatikan masukan dan keluaran akhir dari *use case* yang dijalankan.

Tabel 4.4 Tabel Pengujian Aplikasi

Fungsi	Prosedur Pengujian	Hasil
Login	<ol style="list-style-type: none"> Mengakses halaman <i>login</i> Memasukan <i>email address</i> dan <i>password valid</i> Klik tombol <i>LET'S GO</i> 	OK

View Role	Klikmenu <i>Role Management</i> untuk mengakses halaman <i>role management</i>	OK
Create Role	<ol style="list-style-type: none"> 1. Klik tombol tambah pada halaman <i>Role Management</i> 2. Masukkan data <i>role</i> kedalam <i>form</i> yang tersedia 3. Klik tombol <i>save</i> untuk menyimpan data <i>user</i> 	OK
Update Role	<ol style="list-style-type: none"> 1. Klik tombol <i>edit</i> pada halaman <i>Role Management</i> 2. Perbarui data <i>role</i> 3. Klik tombol <i>save</i> untuk menyimpan perubahan 	OK
Delete Role	Klik tombol <i>delete</i> pada halaman <i>Role Management</i>	OK
Set Role Privilege	<ol style="list-style-type: none"> 1. Klik tombol <i>privilege</i> pada halaman <i>Role Management</i> 2. Pilih <i>privilege</i> untuk <i>role</i> 3. Klik <i>save</i> untuk menyimpan perubahan 	OK
View Privilege	Klikmenu <i>Privilege Management</i> untuk mengakses halaman <i>Privilege Management</i>	OK
Create Privilege	<ol style="list-style-type: none"> 1. Klik tombol tambah pada halaman <i>privilege management</i> 2. Masukkan data <i>privilege</i> kedalam <i>form</i> yang tersedia 3. Klik tombol <i>save</i> untuk menyimpan data <i>privilege</i> 	OK
Update Privilege	<ol style="list-style-type: none"> 1. Klik tombol edit pada halaman <i>privilege management</i> 2. Perbarui data <i>privilege</i> 3. Klik tombol <i>save</i> untuk menyimpan perubahan 	OK
Delete Privilege	Klik tombol <i>delete</i> pada halaman <i>Privilege Management</i>	OK

View User	Klikmenu <i>User Management</i> untuk mengakses halaman <i>users management</i>	OK
Create User	<ol style="list-style-type: none"> 1. Klik tombol tambah pada halaman <i>Users Management</i> 2. Masukkan data user kedalam <i>form</i> yang tersedia 3. Klik tombol <i>save</i> untuk menyimpan data <i>user</i> 	OK
Update User	<ol style="list-style-type: none"> 1. Klik tombol <i>edit</i> pada halaman <i>Users Management</i> 2. Perbarui data <i>user</i> 3. Klik tombol <i>save</i> untuk menyimpan perubahan 	OK
Delete User	Klik tombol <i>delete</i> pada halaman <i>Users Management</i>	OK
View Event	Klikmenu <i>Agenda</i> untuk mengakses halaman <i>Agenda</i>	OK
Create Event	<ol style="list-style-type: none"> 1. Klik tanggal pada halaman <i>Agenda</i> 2. Masukkan data <i>event</i> kedalam <i>form</i> yang tersedia 3. Klik tombol <i>save</i> untuk menyimpan data <i>event</i> atau, Klik tombol <i>invite</i> untuk menyimpan dan mengundang <i>user</i> 	OK
Update Event	<ol style="list-style-type: none"> 1. Klik tombol <i>view</i> pada halaman <i>events</i> atau, 2. Klik <i>event</i> pada halaman <i>agenda</i> 3. Perbarui data <i>event</i> 4. Klik tombol <i>save</i> untuk menyimpan data <i>event</i> atau, 5. Klik tombol <i>invite</i> untuk menyimpan dan mengundang <i>user</i> 	OK
Delete Event	<ol style="list-style-type: none"> 1. Klik tombol <i>delete</i> pada halaman <i>events</i> atau, 2. Klik tombol <i>delete</i> pada halaman <i>view event</i> 	OK
Invite User	1. Pada halaman <i>event invitations</i> pilih <i>user</i> untuk di	OK

	undang ke dalam <i>event</i> 2. Klik tombol <i>invite</i> untuk mengundang <i>user</i>	
Reminder	Aplikasi mengirimkan <i>reminder</i> jika waktu mulai <i>event</i> kurang dari atau sama dengan dua jam	OK
View Setting	Aplikasi mengirimkan <i>reminder</i> jika waktu mulai <i>event</i> kurang dari atau sama dengan dua jam	OK
Update Setting	1. Klik tombol <i>edit</i> pada halaman <i>setting</i> 2. Perbarui pengaturan aplikasi 3. Klik tombol <i>save</i> untuk menyimpan pengaturan	OK

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan pada bab sebelumnya, maka dapat ditarik kesimpulan bahwa aplikasi penjadwalan kegiatan akademik dengan sistem *reminder* bisa menjadi alat untuk mempermudah proses pencatatan dan pengingat kegiatan akademik FIKI Universitas Nurtanio Bandung sehingga dapat meningkatkan keefektifitasan kegiatan akademik Fakultas Ilmu Komputer dan Informatika Universitas Nurtanio Bandung.

5.2 Saran

Untuk penelitian selanjutnya penulis menyarankan agar aplikasi ini dapat terintegrasi dengan sistem aplikasi akademik yang ada sehingga kegiatan akademik yang sudah ada dapat diintegrasikan dengan aplikasi penjadwalan ini.

DAFTAR PUSTAKA

1. Bauer, Mick dan Brenno de Winter. 2000. *Using Postfix for Secure SMTP Gateways*. <http://www.linuxjournal.com/article/4241> (diakses tanggal 17 September 2017).
2. Concept Draw. *Diagramming Software for designing UML Sequence Diagrams*. [\[uml-sequence-diagram\]\(#\) \(diakses tanggal 30 September 2017\).](http://www.conceptdraw.com/How-To-Guide/diagrams-software-tools-for-design-

</div>
<div data-bbox=)

3. Handayani, Indri, Qurotul Aini dan Yessy Oktavyanti. 2015. *Penggunaan Rinfocal Sebagai Aplikasi Pengingat (Reminder) Kegiatan Akademik Pada Perguruan Tinggi*. Vol. 9, No, 1 dalam http://raharja.ac.id/raharja_file/file_jurnal/file/9010215.pdf (diakses tanggal 11 Agustus 2017).
4. Hartl, Michael. 2015. *Ruby on Rails Tutorial: Learn Web Development with Rails (3rd edition)*. United States : Addison-Wesley
5. Haryanto, Edy Victor. 2012. *Jaringan Komputer*. Yogyakarta : ANDI
6. Inderapermana, Yusa. *Pengertian Application Programming Interface (API)*. <http://www.yusaindera.com/2017/03/pengertian-application-programming.html> (diakses tanggal 11 Agustus 2017).
7. Pradana, Ryan Oki. 2013. *Rancang Bangun Sistem Informasi Pengingat Jadwal Pembayaran Angsuran Pada Kospin Jasa Cabang Pemalang Berbasis SMS Gateway*. Semarang : Tidak diterbitkan.
8. PostgreSQL. *About Postgresql*. <https://www.postgresql.org/about/> (diakses tanggal 2 Agustus 2017)
9. Schade, Amy “*Responsive Web Design (RWD) and User Experinece*”. <https://www.nngroup.com/articles/responsive-web-design-definition/> (diakses tanggal 8 Agustus 2017)