

Analisis dan Perancangan Aplikasi Penyusunan Jadwal Mengajar Sesuai Data Ketersediaan Mengajar Dosen (Studi Kasus: Jurusan Teknik Informatika)

Meliana Christianti J.^{#1}, Robby Tan^{*2}, Oscar Karnalim^{#3}, Egie Imandha^{*4}, Tedy Cahyadi^{#5}

^{#*}Jurusan Teknik Informatika, Universitas Kristen Maranatha
Jl. Surya Sumantri No. 65 Bandung

¹meliana.christianti@itmaranatha.org

²robbi.tan@itmaranatha.org

³oscar.karnalim@itmaranatha.org

⁴egieimandha@gmail.com

⁵timotiustedy@ymail.com

Abstract — The purpose of this research is to analyze and make a good courses scheduling in Department of Informatics Engineering at Maranatha Christian University. Several process include in scheduling process are determining number of classes based on registered students and lecturer allocation process based on lecturer availability time. The new system is expected to accommodate all data that required in courses scheduling and replace some processes that were previously done manually. The new system is developed using Java (desktop version) and PHP (web version) using MySQL database as data storage. In the last test, this system is able to accommodate and handle all required data and make courses scheduling in short amount of time.

Keywords—availability, class, courses, lecturer, scheduling.

I. PENDAHULUAN

Jurusan Teknik Informatika merupakan Jurusan Strata Satu yang sudah berdiri sejak pertengahan tahun 2003. Saat ini, proses penyusunan jadwal mengajar dosen di Jurusan Teknik Informatika masih dilakukan secara manual. Hal ini mengakibatkan proses penyusunan jadwal memakan waktu cukup lama, karena sekretaris Jurusan Teknik Informatika perlu menanyakan ketersediaan mengajar dosen dan waktu yang disediakan, membandingkan jadwal mengajar setiap dosen sehingga tidak terjadi bentrokan dengan jadwal mengajar dosen lain dan juga kebentrokan penggunaan ruangan.

Sebagai bagian dari Universitas Kristen Maranatha, Fakultas Teknologi Informasi maka Jurusan Teknik Informatika ingin mempersingkat waktu penyusunan jadwal mengajar dosen dan menyesuaikan dengan jadwal yang disediakan oleh seorang dosen untuk mengajar pada suatu

semester. Untuk menjamin penyusunan jadwal yang lebih ringkas, maka Jurusan Teknik Informatika melakukan analisis dan perancangan aplikasi yang dapat digunakan untuk menyusun jadwal mengajar dosen sesuai dengan informasi jadwal yang disediakan oleh dosen tersebut.

Kemudian untuk menjamin penyusunan jadwal di Jurusan Teknik Informatika, maka Jurusan Teknik Informatika membutuhkan aplikasi untuk penyusunan jadwal sesuai dengan jadwal ketersediaan dosen. Dalam penelitian ini, fokus utama disampaikan dalam bentuk pertanyaan penelitian yang disajikan dalam bentuk rumusan masalah mengenai pendataan jadwal ketersediaan dosen, proses penyusunan berdasarkan data yang diperoleh dan pembuatan pengumuman.

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana membuat aplikasi untuk mencatat seluruh jadwal ketersediaan mengajar yang disediakan oleh dosen untuk mengajar pada suatu semester?
2. Bagaimana membuat aplikasi untuk menyusun jadwal mengajar dosen pada suatu semester sesuai data ketersediaan mengajar yang diberikan dosen tersebut?
3. Bagaimana membuat aplikasi untuk membantu Jurusan Teknik Informatika mengumumkan jadwal kuliah yang telah disusun kepada mahasiswa dan dosen?

Tujuan dilakukannya penelitian ini adalah sebagai berikut:

1. Membuat aplikasi untuk mencatat jadwal yang disediakan oleh seorang dosen untuk mengajar di Jurusan Teknik Informatika pada suatu semester.
2. Membuat aplikasi untuk menyusun jadwal mengajar dosen sesuai dengan jadwal kesediaan mengajar yang diberikan oleh dosen.
3. Membuat aplikasi untuk mengumumkan jadwal kuliah yang telah disusun kepada dosen dan mahasiswa.

II. TINJAUAN PUSTAKA

A. Unified Modelling Language (UML)

“UML adalah sekumpulan simbol dan diagram untuk memodelkan software berorientasi objek” [1]. Dengan menggunakan UML, desain perangkat lunak dapat diwujudkan dalam bentuk simbol dan diagram, untuk diterjemahkan menjadi kode program.

Implementasi kode program dari diagram UML dapat menggunakan berbagai bahasa pemrograman dengan syarat bahasa pemrograman tersebut harus mendukung pemrograman berorientasi objek.

Pemrograman berorientasi objek adalah sebuah pendekatan yang menyediakan cara untuk memodularisasikan program dengan cara membuat area memory terpartisi untuk data dan fungsi yang bisa digunakan sebagai cetakan untuk membuat salinan seperti modul [2].

Salah satu bahasa yang menggunakan OOP adalah Java. Java pertama kali diperkenalkan pada tahun 1995 sebagai pemrograman berorientasi objek yang sederhana dan aman [3].

Ada beberapa kemungkinan yang menyebabkan orang-orang tertarik pada bahasa Java. Salah satunya adalah kemampuan *multiplatform* bahasa Java yang memungkinkan program Java bisa dijalankan di komputer manapun selama mesin *virtual* Java diimplementasi pada komputer tersebut.

Dalam penelitian ini juga digunakan *use case diagram* untuk menggambarkan fungsionalitas dari sebuah sistem. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case diagram* membantu dalam pembuatan *requirement* sebuah sistem [4].

B. Diagram Relasional Entitas

“Diagram relasional entitas adalah model, dalam bentuk diagram atau gambar, dari struktur data logis dari sebuah sistem. Struktur data logis dari sebuah sistem informasi adalah hubungan antar data dalam suatu sistem” [5].

Berikut ini merupakan definisi dari simbol yang digunakan dalam diagram relasional entitas.

1. Entitas

Entitas digambarkan dalam bentuk persegi panjang. Entitas adalah individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Entitas dapat berupa orang,

tempat, objek, atau kejadian. Contoh pekerja, supplier, bahan baku, dan lain-lain.

2. Atribut

Atribut digambarkan dalam bentuk elips. Atribut adalah karakteristik dari suatu entitas contoh pekerja memiliki atribut berupa ID Pekerja, nama, alamat, telepon, *handphone*, dan status. Berikut ini merupakan beberapa jenis atribut:

a. Key Attribute (Atribut Kunci)

Atribut yang digunakan untuk menentukan suatu entitas secara unik (atribut yang membedakan dalam satu entitas). Atribut ini ditandai dengan adanya garis bawah di nama atribut bersangkutan.

b. Simple Attribute (Atribut Sederhana)

Atribut yang tidak dapat dipilah lagi.

c. Composite Attribute (Atribut Komposit)

Atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.

d. Single-valued Attribute (Atribut Bernilai Tunggal)

Atribut yang bernilai tunggal.

e. Multivalued Attribute (Atribut Bernilai Banyak)

Atribut yang memiliki sekelompok nilai untuk setiap entitas. Atribut ini digambarkan dengan elips ganda.

f. Derived Attribute (Atribut Turunan)

Atribut yang dihasilkan dari atribut lain. Atribut ini digambarkan dengan elips titik-titik.

3. Relasi

Relasi digambarkan dalam bentuk belah ketupat. Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

4. Kardinalitas

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Terdapat 4 jenis kardinalitas yaitu sebagai berikut:

a. Relasi Satu Ke Satu

Relasi ini berarti setiap entitas pada himpunan entitas A hanya berhubungan dengan satu entitas pada himpunan entitas B, begitu juga sebaliknya.

b. Relasi Satu Ke Banyak

Relasi ini berarti setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

c. Relasi Banyak Ke Satu

Relasi ini berarti banyak entitas pada himpunan entitas A dapat berhubungan dengan satu entitas pada himpunan entitas B.

d. Relasi Banyak Ke Banyak

Relasi ini berarti semua entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B [6].

C. PHP Hypertext Preprocessor

Menurut dokumen resmi PHP, PHP adalah singkatan dari *PHP Hypertext Preprocessor*. PHP merupakan bahasa berbentuk skrip yang ditempatkan dalam *server* dan diproses dalam *server*. Hasilnya dikirim ke klien, tempat pemakai menggunakan *browser*.

Secara khusus, PHP dirancang untuk membentuk *web* dinamis artinya PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, menampilkan isi *database* ke halaman *web*. Pada prinsipnya, PHP mempunyai fungsi yang sama seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl.

Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat daftar riwayat hidupnya, yakni pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi *tools* yang disebut "*Personal home page*". Paket inilah yang menjadi cikal bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan yang kompleks.

Saat ini, PHP cukup populer sebagai peranti pemrograman *web*, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada *server-server* yang berbasis UNIX, Windows NT, dan Macintosh. Bahkan versi untuk Windows 95/98 pun tersedia.

Salah satu kelebihan dari PHP adalah mampu berkomunikasi dengan berbagai *database* yang terkenal. Dengan demikian, menampilkan data yang bersifat dinamis yang diambil dari *database*, merupakan hal mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa PHP cocok untuk membangun halaman *web* yang dinamis.

Saat ini PHP sudah dapat berkomunikasi dengan berbagai *database* meskipun dengan kelengkapan yang berbeda-beda; beberapa di antaranya adalah:

1. FilePro
2. Informix
3. Microsoft Access
4. Interbase
5. MSQL
6. MySQL
7. Oracle
8. Postgre SQL
9. dll.

Database yang digunakan untuk membangun aplikasi penyusunan jadwal mengajar sesuai data kesediaan mengajar dosen ini adalah MySQL, salah satu *database* yang mampu berkomunikasi dengan skrip PHP.

D. CodeIgniter

CodeIgniter adalah sebuah *framework* untuk *website* yang dibuat dalam format PHP [7]. CodeIgniter adalah *framework* yang berbasis *Model-View-Controller*(M-V-C). MVC adalah sistem yang memisahkan antara pengambilan data dari *database* dengan presentasi pada halaman *website*.

Keuntungan dari penggunaan *framework* CodeIgniter sebagai berikut:

1. CodeIgniter berbasis *open source*.
2. CodeIgniter berbasis *Model-View-Controller* (MVC).
3. CodeIgniter sangat ringan untuk dijalankan pada berbagai *platform*.
4. CodeIgniter kompatibel dengan PHP 4.
5. CodeIgniter memiliki keamanan yang cukup baik dan memiliki XSS filter.

E. JQuery

JQuery adalah *Javascript Library* merupakan kumpulan kode/ fungsi *javascript* siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode *javascript* [8].

Keuntungan dari jQuery sebagai berikut:

1. jQuery telah banyak digunakan oleh *website-website* terkemuka di dunia
2. jQuery cocok dengan semua *browser* yang populer
3. Kompatibel dengan semua versi CSS (CSS 1 sampai dengan CSS 3)
4. Ukuran dari file jQuery sangatlah kecil
5. jQuery *open source* dengan lisensi dari GNU (*General Public License*) dan *MT License*.

F. Cascading Style Sheet

CSS (*Cascading Style Sheet*) adalah sebuah pemrograman yang memberikan kontrol terhadap semua tampilan dan desain dari halaman *website*. Banyak tampilan yang dapat dilakukan di dalam CSS dari tampilan rollover, gambar, membuat kolom, dan efek dinamis dari *rollover* [9]. Beberapa keuntungan CSS adalah sebagai berikut:

1. *Designer* tidak dibatasi pada pembuatan tampilan dan permodelan dari HTML.
2. Memberikan banyak pilihan *formatting* daripada HTML
3. Dapat memasukkan sebuah gambar pada bagian belakang dari layar *website* tersebut dan dapat menentukan apakah tampilan gambar akan diulang-ulang atau tidak.
4. CSS dapat mengurangi banyak ukuran *file* dari halaman *website*.
5. Membantu dalam melakukan *update* tampilan *website* [10].

G. PHP Data Object (PDO)

PDO merupakan layer abstraksi dari PHP yang menyediakan fungsi-fungsi dan antarmuka untuk koneksi dan manipulasi basis data. PDO juga menyediakan *data-access abstraction layer* sehingga memudahkan pengguna untuk mengganti basis data yang digunakan tanpa perlu

menulis ulang *query*. Fungsi-fungsi PDO baru dapat digunakan dan diakses jika instalasi server PHP (Apache) minimal 5.1 [11].

H. PHP dan Pemrograman Berorientasi Objek

PHP baru mendukung konsep pemrograman berorientasi objek sejak PHP 3. Namun konsep OOP baru benar-benar diimplementasikan pada PHP 4 dan dikembangkan lebih lanjut pada PHP 5. PHP 5 mengimplementasi hampir seluruh fitur OOP yang belum dimasukkan dalam PHP 4. Perbedaan antara kode program PHP secara prosedural dan *object-oriented* dapat dilihat pada Gambar 1 [11].

```

1 <?php
2 $user_input = $_POST['field'];
3 $filtered_content = filter($user_input);
4 mysql_connect("dbhost","dbuser",
5             "dbpassword");
6 mysql_select_db("dbname");
7 $sql = "some query";
8 while ($data = mysql_fetch_assoc()) {
9     process($data);
10 }
11 process_user_input($filtered_content);
12
1 <?php
2 $input_filter = new filter();
3 $input_filter->filter_user_input();
4 $db = new dal("mysql");
5 $db->connect($dbconfig);
6 $result = $db->execute($sql);
7 ReportGenerator::makeReport($result);
8 $model = new PostModel($filter->
9             get_filtered_content());
10 $model->insert();
11
12
    
```

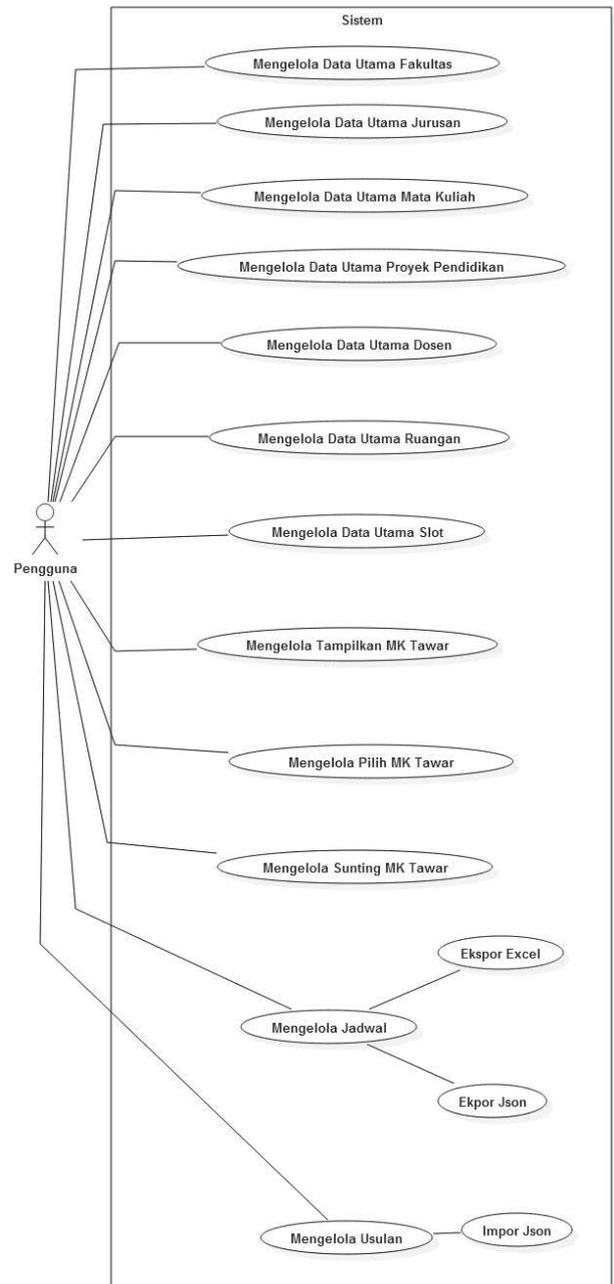
Gambar 1. Perbedaan Sintaks PHP Prosedural dan Object Oriented.

Tujuan dari penggunaan konsep pemrograman berorientasi objek adalah memudahkan dalam pembuatan aplikasi dan *maintenance*. Dengan menggunakan konsep ini, maka sebuah aplikasi dapat dibagi menjadi beberapa modul yang lebih kecil yang dapat dikembangkan secara terpisah tanpa mengganggu fungsi *web*.

III. ANALISIS DAN RANCANGAN SISTEM

Pemodelan perangkat lunak dibuat dengan menggunakan *Unified Modelling Language* (UML). Berikut ini adalah rancangan dari *use case diagram* yang akan digunakan dalam pembuatan aplikasi. Adapun aktor dalam aplikasi ini hanya satu yaitu pengguna. *Use case diagram* dapat dilihat pada Gambar 2.

Di dalam sistem ini, pengguna dapat mengelola semua data yang ada. Dalam mengelola data pengguna dapat menambahkan, memperbaharui, menghapus dan mencari data yang ada. Selain itu, pengguna dapat mengimpor dan mengekspor data dari format *json* dan juga pengguna dapat mengekspor dalam format *excel*.



Gambar 2 Rancangan use case diagram.

Tabel I – XI merupakan penjelasan dari use case diagram yang dibuat untuk aplikasi ini.

TABEL I
DESKRIPSI *USE CASE DIAGRAM* UNTUK FITUR DATA UTAMA FAKULTAS

Nama	Data Utama Fakultas
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna

Nama	Data Utama Fakultas
Pre-condition	Pengguna Memilih Menu Item Fakultas pada Menu Data Utama
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL II

DESKRIPSI USE CASE DIAGRAM UNTUK FITUR DATA UTAMA JURUSAN

Nama	Data Utama Jurusan
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Jurusan pada Menu Data Utama
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL III

DESKRIPSI USE CASE DIAGRAM UNTUK FITUR DATA UTAMA MATA KULIAH

Nama	Data Utama Mata Kuliah
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Mata Kuliah pada Menu Data Utama
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL IV

DESKRIPSI USE CASE DIAGRAM UNTUK FITUR DATA UTAMA PROYEK PENDIDIKAN

Nama	Data Utama Proyek Pendidikan
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Proyek Pendidikan pada Menu Data Utama
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk

Nama	Data Utama Proyek Pendidikan
	disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL V

DESKRIPSI USE CASE DIAGRAM UNTUK FITUR DATA UTAMA DOSEN

Nama	Data Utama Dosen
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Dosen pada Menu Data Utama
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL VI

DESKRIPSI USE CASE DIAGRAM UNTUK FITUR DATA UTAMA RUANGAN

Nama	Data Utama Ruangan
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Ruangan pada Menu Data Utama
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL VII

DESKRIPSI USE CASE DIAGRAM UNTUK FITUR TAMPILKAN MK TAWAR

Nama	Tampilkan MK Tawar
Deskripsi	Pengguna dapat menambah, memperbaharui, menghapus dan mencari data yang terdapat di dalam <i>database</i> .
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Tampilkan MK Tawar pada Menu MK Tawar
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data untuk disimpan di dalam <i>database</i> , memperbaharui, menghapus, dan mencari data yang ada di dalam <i>database</i> . Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui,

Nama	Tampilkan MK Tawar
	menghapus dan mencari data.

TABEL VIII
DESKRIPSI USE CASE DIAGRAM UNTUK FITUR PILIH MK TAWAR

Nama	Pilih MK Tawar
Deskripsi	Pengguna dapat menambah data Mata Kuliah.
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Pilih MK Tawar pada Menu MK Tawar
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memasukkan data mata kuliah dengan cara memilih mata kuliah yang ada pada <i>list</i> mata kuliah.

TABEL IX
DESKRIPSI USE CASE DIAGRAM UNTUK FITUR SUNTING MK TAWAR

Nama	Sunting MK Tawar
Deskripsi	Pengguna dapat memperbaharui dan mencari data MK Tawar.
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Sunting MK Tawar pada Menu MK Tawar
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memperbaharui dan mencari data MK Tawar. Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam memperbaharui dan mencari data.

TABEL X
DESKRIPSI USE CASE DIAGRAM UNTUK FITUR JADWAL

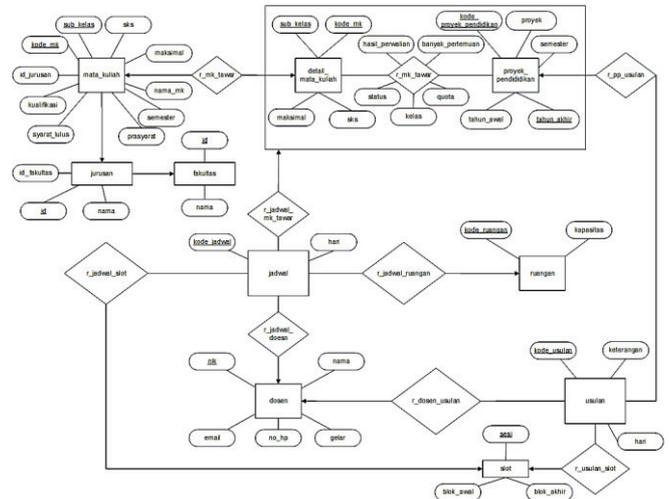
Nama	Jadwal
Deskripsi	Pengguna dapat memperbaharui, mencari dan ekspor data yang terdapat di dalam database baik berformat Json dan Ms. Excel.
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Jadwal pada Menu Jadwal
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat memperbaharui, mencari, ekspor dan impor data yang terdapat di dalam database baik berformat Json dan Ms. Excel. Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

TABEL XI
DESKRIPSI USE CASE DIAGRAM UNTUK FITUR JADWAL

Nama	Usulan
Deskripsi	Pengguna dapat mencari data yang terdapat di dalam database dan mengimpor data yang berformat Json.
Aktor	Pengguna
Pre-condition	Pengguna Memilih Menu Item Usulan pada Menu Usulan
Post-condition	Sistem akan menampilkan sebuah <i>JFrame</i> , lalu pengguna dapat mencari data yang terdapat di dalam database dan mengimpor data yang berformat Json.

Nama	Usulan
	Dan menampilkan pesan-pesan kesalahan jika terjadi suatu kesalahan dalam menyimpan, memperbaharui, menghapus dan mencari data.

Diagram relasional entitas untuk aplikasi ini dapat dilihat pada Gambar 2.



Gambar 2 Diagram Relasional Entitas

Berdasarkan hasil analisis dan perancangan aplikasi penyusunan jadwal mengajar sesuai data kesediaan mengajar dosen maka pada aplikasi ini terdapat 4 menu utama pada aplikasi penyusunan jadwal mengajar yaitu:

- Data utama
- MK Tawar
- Jadwal
- Usulan

Data utama yang ada pada aplikasi ini adalah sebagai berikut:

- Fakultas
- Jurusan
- Mata Kuliah
- Proyek Pendidikan
- Dosen
- Ruang
- Slot

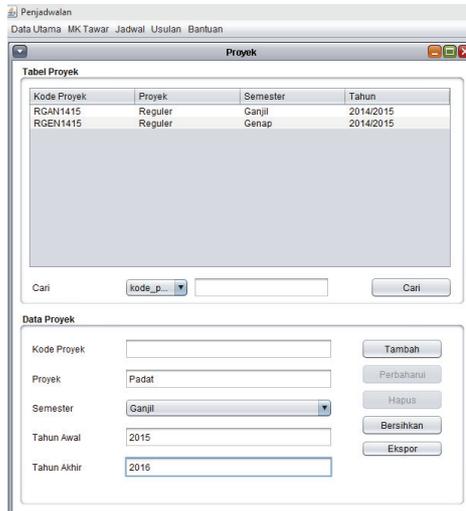
Pada menu MK Tawar, admin dapat melakukan proses berikut ini:

- Pilih MK Tawar
- Sunting MK Tawar
- Tampilkan MK Tawar

Pada menu Jadwal, admin dapat melakukan proses berikut ini:

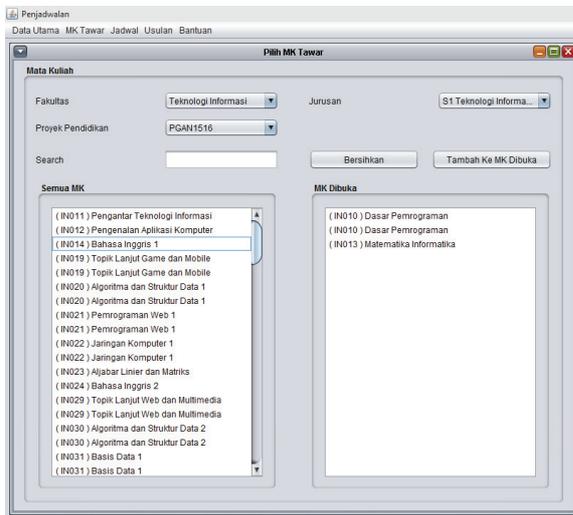
- Pembuatan jadwal
- Pilih Dosen per MK

Pada menu Usulan, admin dapat menampilkan data usulan jadwal dari dosen yang diperoleh dari aplikasi *web*. Langkah pertama pengguna akan menambahkan proyek pendidikan baru contoh tampilan aplikasi *desktop* dapat dilihat pada Gambar 4.



Gambar 3 Tambah Proyek Pendidikan

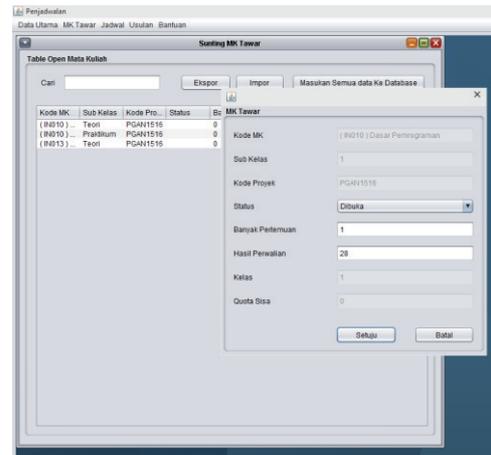
Langkah selanjutnya, pengguna memilih mata kuliah yang akan dibuka pada proyek pendidikan tersebut dan kemudian pengguna memasukkan data hasil perwalian dan aplikasi akan membuat saran kelas untuk setiap mata kuliah. Contoh tampilan pemilihan mata kuliah yang dibuka pada suatu semester dapat dilihat pada Gambar 5.



Gambar 4 Pilih mata kuliah yang akan dibuka

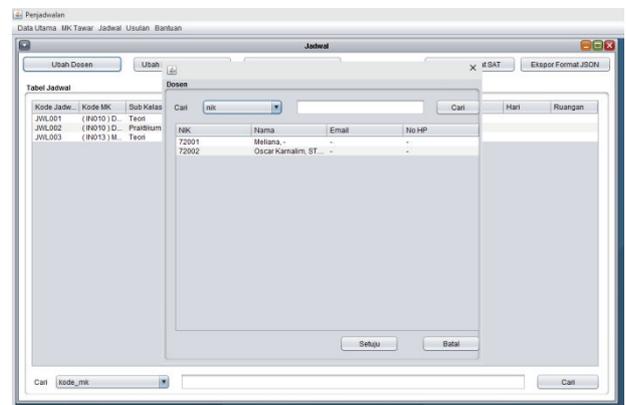
Setelah proses perwalian berlangsung admin perlu memasukkan jumlah kuota hasil perwalian seperti yang

ditampilkan pada Gambar 6 sehingga dapat diperhitungkan jumlah kelas yang perlu dibuka untuk mata kuliah tersebut.



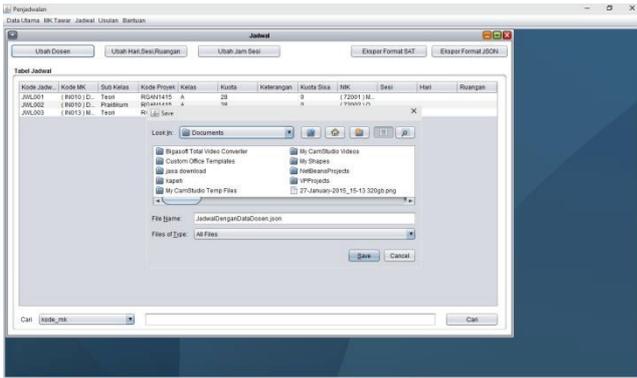
Gambar 6 Isi Hasil Perwalian

Selanjutnya, proses akan dilanjutkan dengan proses menentukan dosen yang akan mengajar untuk tiap mata kuliah dan akan memilih hari, ruangan dan sesi yang tersedia berdasarkan usulan dari setiap dosen yang ada atau berdasarkan ruangan yang tersedia. Tampilan dapat dilihat pada Gambar 7. Setelah semua mata kuliah telah memiliki jadwal pengguna akan mengeksport jadwal yang telah dibuat dalam format SAT. SAT merupakan nama aplikasi sistem akademik yang digunakan di Universitas Kristen Maranatha.



Gambar 7 Pilih dosen

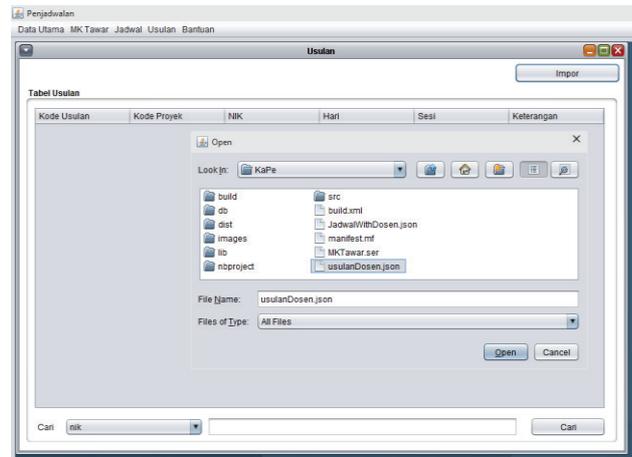
Setelah lengkap, data di-*export* ke Json untuk digunakan di aplikasi *web* usulan kesediaan mengajar dosen yang dapat dilihat pada Gambar 8 untuk meng-*eksport* data dan Gambar 9 untuk mengimpor data usulan. Apabila data berhasil diimpor contoh tampilan dapat dilihat pada Gambar 10.



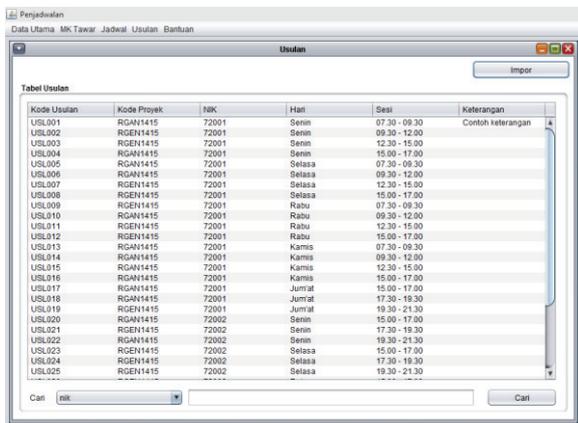
Gambar 8 Pilih source ekspor json



Gambar 11 Mengubah Tanggal Penerimaan Usulan



Gambar 9 Pilih source file impor json usulan

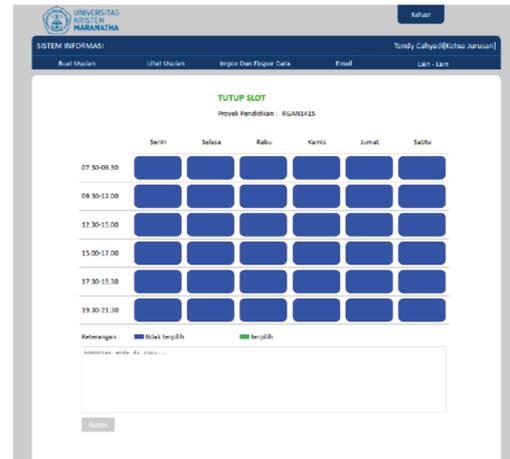


Gambar 10. Usulan berhasil di impor

Di Universitas Kristen Maranatha terdapat beberapa slot (sesi) jam mengajar yang telah ditetapkan oleh universitas.

- Sesi 1: 07.00 – 09.30
- Sesi 2: 09.30 – 12.00
- Sesi Istirahat: 12.00 – 12.30
- Sesi 3: 12.30 – 15.00
- Sesi 4: 15.00 – 17.30
- Sesi 5: 17.30 – Selesai

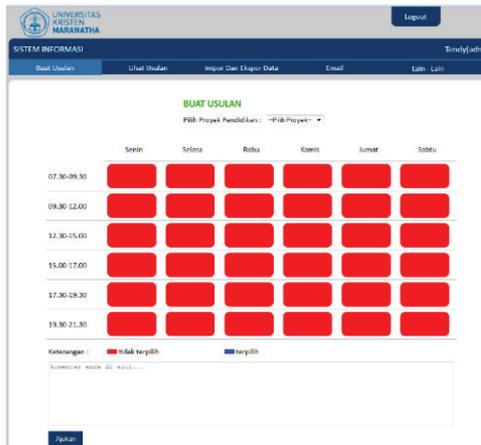
Admin dapat mengatur dan menentukan slot yang tidak dapat diusulkan oleh dosen. Tampilan pengaturan slot yang dibuka dan ditutup dapat dilihat pada Gambar 12.



Gambar 12 Tampilan Fitur Tutup Slot

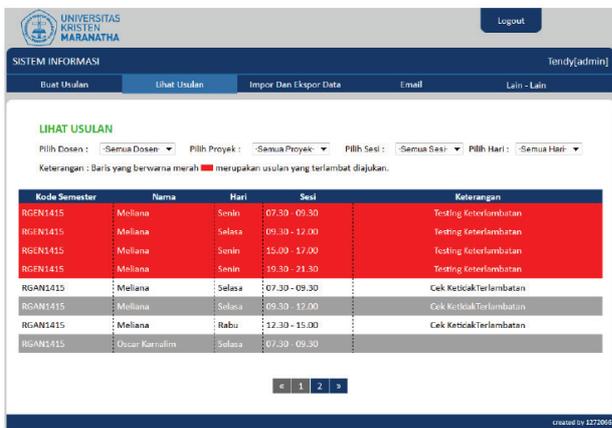
Dosen dapat login ke aplikasi web dan mengisi slot yang diusulkan sebagai jadwal kesediaan dapat dilihat pada Gambar 13.

Pada aplikasi web, admin atau dosen dapat login dengan menggunakan akun SAT. Pada aplikasi web usulan jadwal mengajar, dosen dapat mengusulkan jadwal kesediaan mengajar melalui aplikasi ini setelah admin membuka tanggal usulan. Tampilan untuk mengelola tanggal usulan dapat dilihat pada Gambar 11.



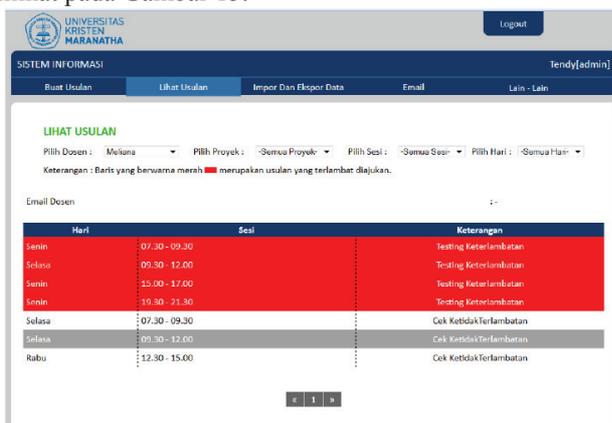
Gambar 13 Halaman Buat Usulan

Dosen dapat melihat usulan jadwal yang diajukan seperti yang ditampilkan pada Gambar 14.



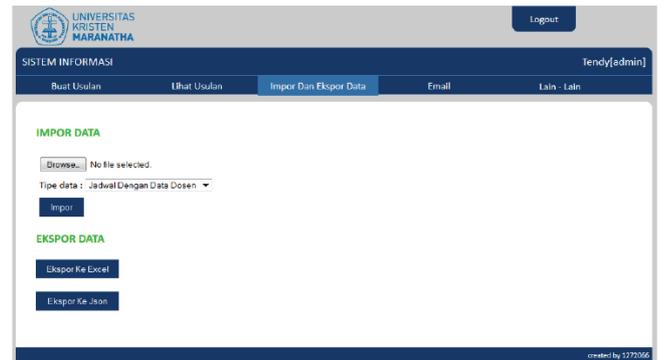
Gambar 14 Semua Usulan Yang Pernah Diajukan

Fitur pencarian data berdasarkan dosen tertentu dapat dilihat pada Gambar 15.



Gambar 15 Halaman Lihat Usulan Berdasarkan Dosen

Setelah data usulan terkumpul data tersebut akan diekspor sehingga dapat digunakan sebagai data usulan pada aplikasi *desktop*. Tampilan dapat dilihat pada Gambar 16.



Gambar 16 Halaman Ekspor Dan Impor Data Usulan

IV. SIMPULAN

Berdasarkan hasil pengamatan dalam melakukan analisis dan perancangan dapat ditarik simpulan sebagai berikut:

1. Aplikasi yang dihasilkan dapat digunakan untuk mencatat jadwal yang disediakan oleh seorang dosen untuk mengajar di Jurusan Teknik Informatika merupakan aplikasi berbasis *web* sehingga dosen dapat mengakses aplikasi ini untuk memberikan usulan kapanpun dan di manapun.
2. Aplikasi yang dihasilkan dapat digunakan untuk menyusun jadwal mengajar dosen sesuai dengan jadwal kesediaan mengajar yang diusulkan oleh dosen tersebut.
3. Aplikasi yang dihasilkan dapat menghasilkan file pengumuman jadwal kuliah untuk dicetak dan diberikan kepada dosen dan mahasiswa.

Saat ini, penelitian ini menghasilkan *prototype* aplikasi yang masih memerlukan pengembangan lebih lanjut sehingga dapat menjadi aplikasi yang diimplementasikan dan dipelihara dengan baik oleh Jurusan Teknik Informatika. Berikut ini merupakan saran pengembangan untuk aplikasi Penyusunan Jadwal Mengajar Sesuai Data Kesediaan Mengajar Dosen yang dihasilkan di Jurusan Teknik Informatika:

1. Untuk pengembangan aplikasi dapat dibuat sistem pengolahan jadwal secara otomatis menggunakan algoritma penjadwalan sehingga jadwal mengajar dapat diciptakan secara otomatis setelah pejabat di Jurusan Teknik Informatika menetapkan beban mengajar dosen pada suatu semester dan dosen mengisi usulan jadwal mengajar pada suatu semester.
2. Penambahan fitur untuk admin sistem untuk mengelola data usulan jadwal dosen tidak tetap sehingga tetap dapat memberikan usulan jadwal mengajar yang dimasukkan ke dalam sistem.

3. Aplikasi dapat diimplementasikan di semua jurusan yang ada di Fakultas Teknologi Informasi sehingga aplikasi ini dapat dimanfaatkan untuk mengatur penjadwalan kuliah dengan lebih baik. Selain itu, perlu dilakukan survey pengguna aplikasi sehingga dapat menjadi evaluasi penerapan sistem agar lebih bermanfaat.

DAFTAR PUSTAKA

- [1] I. F. Azis, *Object Oriented Prgramming Php 5*. Jakarta: PT Elex Media Komputindo, 2005.
- [2] E. Balagurusamy, *Programming With Java:A Primer 3E*. New Delhi: Tata McGraw-Hill Publishing Company Limited, 2007.
- [3] D. Poo, D. Kiong, &M. Ashok, *Object-Oriented Programming and Java*. London: Springer-Verlag London Limited, 2008.
- [4] M. Huda, *Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans*. Jakarta: PT Elex Media Komputindo, 2010.
- [5] R. V. Imbar, &B. R. Suteja, *Penrograman Web Commerce dengan Oracle & ASP*. Bandung: Informatika, 2006.
- [6] Fathansyah, *Basis Data*. Bandung: Informatika, 2007.
- [7] K. Wiswakarma, *9 Langkah Menjadi Master Framework Codeigniter*. Yogyakarta: Lokomedia, 2010.
- [8] L. Hakim, *Trik Dahsyat Menguasai Ajax dengan jQuery*. Yogyakarta: Lokomedia, 2011.
- [9] D. S. Mc Farland, *CSS: The Missing Manual*. Sebastopol: O'Reilly Media, 2006.
- [10] E. Freeman, &E. Freeman, *Head First HTML With CSS & XHTML*. Sebastopol: O'Reilly, 2006.
- [11] P. Lavin, *Object-Oriented PHP*. San Francisco: No Starch Press, 2006.