

**SKRIPSI**

**KONVERSI FORMAT XLSX KE FORMAT ICS  
STUDI KASUS: JADWAL MENGAWAS UJIAN**



**ARIQ RAHMAERI**

**NPM: 2011730066**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2016**



**UNDERGRADUATE THESIS**

**CONVERSION XLSX FORMAT TO ICS FORMAT  
CASE STUDY: OVERSEEN EXAM SCHEDULE**



**ARIQ RAHMAERI**

**NPM: 2011730066**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND  
SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2016**



# **LEMBAR PENGESAHAN**

## **KONVERSI FORMAT XLSX KE FORMAT ICS STUDI KASUS: JADWAL MENGAWAS UJIAN**

**ARIQ RAHMAERI**

**NPM: 2011730066**

**Bandung, 14 Desember 2016**

**Menyetujui,**

**Pembimbing Utama**

**Pembimbing Pendamping**

**Dr. Veronica Sri Moertini**

**-**

**Ketua Tim Penguji**

**Anggota Tim Penguji**

**Gede Karya, M.T., CISA**

**Rosa De Lima, M.Kom.**

**Mengetahui,**

**Ketua Program Studi**

**Mariskha Tri Adithia, P.D.Eng**



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **KONVERSI FORMAT XLSX KE FORMAT ICS STUDI KASUS: JADWAL MENGAWAS UJIAN**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal 14 Desember 2016

Meterai Rp. 6000
---------------------

Ariq Rahmaeri  
NPM: 2011730066





## ABSTRAK

Setiap tahun dosen FTIS(Fakultas Teknologi Infomasi dan Sains) UNPAR menerima daftar jadwal mengawas ujian yang diterbitkan oleh TU(Tata Usaha) FTIS dalam bentuk tampilan file excel. Walaupun datanya bersifat digital, namun lebih baik jika data tersebut dibuat terstruktur sehingga dapat dibaca oleh mesin. Dari permasalahan tersebut maka dalam tugas akhir ini akan membahas pengembangan suatu program yang dapat membaca data excel tersebut dan merubahnya menjadi calendar digital atau biasa disebut .ics , dengan format .ics jadwal dapat di integrasikan dengan aplikasi iCalendar. Program ini akan menggunakan tiga *library* yaitu Apache POI, iCal4j, dan JavaFX. Apache POI bertugas membaca struktur data excel sehingga dapat dibaca oleh program, JavaFX berfungsi sebagai *interface* program, dan Ical4j bertugas mengkonversi data yang telah dibaca program menjadi iCalendar atau .ics .

**Kata-kata kunci:** Konverter, Kalender



## **ABSTRACT**

Every year lecture of FTIS(Fakultas Teknologi Infomasi dan Sains) UNPAR get list of exam schedule which published by FTIS Administration based in excel format. Although, the excel data is digital data, but it's better if the data is structured so can be read by machine. Based of the problem above, this thesis will be discussing about developing program that can read exam schedule in excel format and converting the schedule to digital calendar format or commonly called .ics, with .ics format the schedule can be integrated with iCalendar apps. This software will be build with three library Apache POI, iCal4j, and JavaFX. Apache POI will be handle reading input data so the data can be imported to software, JavaFX is functionate as interface of the software, and iCal4j handles of converting previously read data to iCalendar format or .ics .

**Keywords:** Converter, Calendar



*Keluarga dan kerabat*



## KATA PENGANTAR

Terimakasih yang sebesar-besarnya saya ucapkan bagi Allah SWT, keluarga, sahabat, dosen yang telah membantu saya menyelesaikan skripsi ini. Dengan dukungan dan semangat dari orang-orang terdekat saya dapat menyelesaikan skripsi ini. Mudah-mudahan apa yang tertuang dalam skripsi ini bermanfaat bagi saya pribadi dan dapat menjadi acuan untuk pengembangan selanjutnya.

Bandung, Desember 2016

Penulis





# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xix</b>
<b>DAFTAR TABEL</b>	<b>xxii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>3</b>
2.1 Microsoft Excel . . . . .	3
2.1.1 Fitur Dasar . . . . .	3
2.1.2 Format File . . . . .	4
2.2 iCalendar . . . . .	4
2.2.1 Komponen Utama . . . . .	4
2.2.2 Event(VEVENT) . . . . .	4
2.2.3 Tipe Komponen Lain . . . . .	5
2.3 Apache POI . . . . .	5
2.3.1 Komponen Apache POI . . . . .	5
2.3.2 Kelas Inti Apache POI . . . . .	6
2.4 iCal4j . . . . .	14
2.4.1 Komponen iCal4j . . . . .	14
2.4.2 Kelas Inti dari iCal4j . . . . .	14
2.4.3 net.fortuna.ical4j.data . . . . .	14
2.4.4 net.fortuna.ical4j.filter . . . . .	15
2.4.5 net.fortuna.ical4j.model . . . . .	16
2.4.6 net.fortuna.ical4j.model.component . . . . .	17
2.4.7 net.fortuna.ical4j.model.parameter . . . . .	18
2.4.8 net.fortuna.ical4j.model.property . . . . .	19
2.4.9 net.fortuna.ical4j.model.transform . . . . .	20
2.4.10 net.fortuna.ical4j.model.util . . . . .	21
2.5 JavaFX . . . . .	21
2.5.1 Scene Graph . . . . .	22
2.5.2 Java Public API untuk Fitur JavaFX . . . . .	22
2.5.3 Graphic System . . . . .	23
2.5.4 Glass Windowing Toolkit . . . . .	23

2.5.5	Media dan Gambar . . . . .	23
2.5.6	Komponen Web . . . . .	23
2.5.7	CSS . . . . .	24
2.5.8	UI Control . . . . .	24
2.5.9	Layout . . . . .	25
2.5.10	Transformasi 2-D dan 3-D . . . . .	25
2.5.11	Komponen JavaFX . . . . .	25
2.5.12	Property . . . . .	26
2.5.13	javafx.collections . . . . .	26
2.5.14	javafx.scene.control . . . . .	27
<b>3</b>	<b>ANALISIS</b>	<b>29</b>
3.1	Analisis Input . . . . .	29
3.1.1	Analisis File Excel Jadwal Mengawas Ujian . . . . .	29
3.1.2	Flowchart Konversi ArrayList Jadwal ke iCalendar . . . . .	33
3.1.3	Analisis Kebutuhan Perangkat Lunak . . . . .	33
3.2	Use Case Perangkat Lunak . . . . .	34
3.3	Diagram Aktifitas . . . . .	35
3.3.1	Memasukan Excel Jadwal Mengawas Ujian . . . . .	35
3.3.2	Sorting Nama Pengawas . . . . .	35
3.3.3	Unduh File iCal . . . . .	36
3.4	Pemodelan Kelas . . . . .	36
<b>4</b>	<b>PERANCANGAN</b>	<b>41</b>
4.1	Perancangan Diagram Kelas . . . . .	41
4.2	Rancangan <i>Method-Method</i> Utama . . . . .	44
4.3	Perancangan Antarmuka . . . . .	46
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>49</b>
5.1	Implementasi . . . . .	49
5.1.1	Lingkungan Implementasi . . . . .	49
5.1.2	Implementasi Program . . . . .	50
5.2	Implementasi Antarmuka . . . . .	59
5.3	Pengujian . . . . .	60
5.3.1	Pengujian Fungsional . . . . .	60
5.3.2	Pengujian Eksperimental . . . . .	70
5.3.3	Kesimpulan Pengujian . . . . .	87
<b>6</b>	<b>KESIMPULAN</b>	<b>89</b>
6.0.4	Kesimpulan . . . . .	89
6.0.5	Saran . . . . .	89
	<b>DAFTAR REFERENSI</b>	<b>91</b>
	<b>A KODE PROGRAM</b>	<b>93</b>
	<b>B FILE EXCEL</b>	<b>103</b>

## DAFTAR GAMBAR

2.1	<i>Spreadsheet</i> Microsoft Excel . . . . .	3
2.2	Contoh iCalendar . . . . .	4
2.3	Contoh VEVENT pada iCalendar . . . . .	4
2.4	Kelas diagram hssf . . . . .	6
2.5	Arsitektur JavaFX . . . . .	21
2.6	Contoh JavaFX UI Control . . . . .	24
3.1	Jadwal mengawas ujian FTIS . . . . .	29
3.2	<i>Flowchart</i> membaca excel . . . . .	31
3.3	<i>Flowchart</i> alur konversi excel ke iCalendar . . . . .	33
3.4	Diagram use case perangkat lunak konversi jadwal mengawas ujian . . . . .	34
3.5	Prosedur memasukan file excel jadwal mengawas ujian . . . . .	35
3.6	Prodsedur sorting nama pengawas . . . . .	36
3.7	Prosedur Mengunduh File iCal . . . . .	36
3.8	Gambar Pemodelan ExcelConverter . . . . .	37
3.9	Gambar Pemodelan ScheduleClass . . . . .	37
3.10	Gambar Pemodelan FXMLDocumentController . . . . .	37
3.11	Gambar Pemodelan CalendarConverter . . . . .	38
3.12	Gambar Pemodelan Kelas . . . . .	38
4.1	Gambar Kelas Diagram . . . . .	41
4.2	Tampilan awal Program . . . . .	46
4.3	Tampilan <i>Browse</i> file excel . . . . .	47
4.4	Tampilan setelah excel dibaca . . . . .	47
4.5	Tampilan untuk menyimpan iCal . . . . .	48
5.1	Implementasi kode program untuk mencari kolom No dan matakuliah . . . . .	50
5.2	Implementasi kode program ExcelConverter . . . . .	50
5.3	Implementasi kode program ExcelConverter bagian 2 . . . . .	51
5.4	Implementasi kode program ExcelConverter bagian 3 . . . . .	51
5.5	Implementasi kode program ExcelConverter bagian 4 . . . . .	52
5.6	Implementasi kode program ExcelConverter bagian 4.1 . . . . .	52
5.7	Implementasi kode program ExcelConverter bagian 5 . . . . .	52
5.8	Implementasi kode program ExcelConverter bagian 6 . . . . .	53
5.9	Implementasi kode program ExcelConverter bagian 7 . . . . .	53
5.10	Implementasi kode program ExcelConverter bagian 8 . . . . .	54
5.11	Implementasi kode program ExcelConverter bagian 9 . . . . .	54
5.12	Implementasi kode program CalendarConverter bagian 1 . . . . .	55
5.13	Implementasi kode program CalendarConverter bagian 2 . . . . .	55
5.14	Implementasi kode program CalendarConverter bagian 3 . . . . .	56
5.15	Implementasi kode program FXMLDocumentController bagian 1 . . . . .	56
5.16	Implementasi kode program FXMLDocumentController bagian 2 . . . . .	57
5.17	Implementasi kode program FXMLDocumentController bagian 3 . . . . .	57

5.18 Implementasi kode program FXMLDocumentController bagian 4 . . . . .	58
5.19 Implementasi kode program FXMLDocumentController bagian 5 . . . . .	58
5.20 Tampilan antarmuka perangkat lunak . . . . .	59
5.21 Tampilan antarmuka setelah file mengawas dimasukkan . . . . .	60
5.22 Tampilan browse file excel mengawas ujian . . . . .	62
5.23 Tampilan path file excel mengawas ujian . . . . .	62
5.24 PL menampilkan jadwal ke layar . . . . .	62
5.25 PL mengkonversi jadwal ke format iCal . . . . .	63
5.26 File iCal . . . . .	63
5.27 Hasil pengujian filter nama dosen . . . . .	64
5.28 Hasil pengujian convert hasil filter kedalam iCal . . . . .	64
5.29 File iCal Filter . . . . .	65
5.30 Hasil pengujian import kedalam Google Calendar . . . . .	65
5.31 Hasil import ke Google Calendar . . . . .	65
5.32 Hasil import ke Google Calendar bagian 2 . . . . .	66
5.33 <i>Import</i> file iCal kedalam MS Outlook . . . . .	66
5.34 File hasil Konversi berada pada kalender MS Outlook . . . . .	67
5.35 File hasil Konversi dapat dibuka di MS Outlook . . . . .	67
5.36 Hasil pengujian import file yang di filter kedalam Google Calendar . . . . .	67
5.37 Hasil import file yang di filter ke Google Calendar . . . . .	68
5.38 Hasil import file yang di filter ke Google Calendar bagian 2 . . . . .	68
5.39 <i>Import</i> file iCal yang telah di filter kedalam MS Outlook . . . . .	69
5.40 File iCal yang telah di filter berada pada kalender MS Outlook . . . . .	69
5.41 File hasil filter dapat dibuka di MS Outlook . . . . .	70
5.42 <i>Output</i> PL pada pengujian eksperimental . . . . .	71
5.43 <i>Output</i> PL pada pengujian eksperimental dari file <i>input</i> yang bukan excel . . . . .	71
5.44 PL memfilter ekstensi file excel yang akan dimasukkan . . . . .	73
5.45 PL menampilkan jadwal ke layar . . . . .	73
5.46 PL mengkonversi jadwal ke format iCal . . . . .	74
5.47 File iCal . . . . .	74
5.48 Hasil pengujian convert hasil filter kedalam iCal . . . . .	75
5.49 File iCal Filter . . . . .	75
5.50 Hasil pengujian import kedalam Google Calendar . . . . .	75
5.51 Hasil import ke Google Calendar . . . . .	76
5.52 Hasil import ke Google Calendar bagian 2 . . . . .	76
5.53 <i>Import</i> file iCal kedalam MS Outlook . . . . .	76
5.54 File hasil Konversi dapat dibuka di MS Outlook . . . . .	77
5.55 File hasil Konversi berada pada kalender MS Outlook . . . . .	77
5.56 Hasil pengujian import file yang di filter kedalam Google Calendar . . . . .	77
5.57 Hasil import file yang di filter ke Google Calendar . . . . .	78
5.58 Hasil import file yang di filter ke Google Calendar bagian 2 . . . . .	78
5.59 <i>Import</i> file iCal yang telah di filter kedalam MS Outlook . . . . .	79
5.60 File hasil filter dapat dibuka di MS Outlook . . . . .	79
5.61 File iCal yang telah di filter berada pada kalender MS Outlook . . . . .	79
5.62 Hasil pengujian <i>browse</i> pada OS X dengan <i>file input</i> data lama . . . . .	80
5.63 Hasil pengujian <i>browse</i> pada OS X dengan <i>file input</i> data baru . . . . .	81
5.64 Hasil pengujian menampilkan ke layar pada OS X dengan <i>file input</i> data lama . . . . .	81
5.65 Hasil pengujian menampilkan ke layar pada OS X dengan <i>file input</i> data baru . . . . .	82
5.66 Hasil pengujian menyimpan file iCal pada OS X dengan <i>file input</i> data lama . . . . .	82
5.67 Hasil pengujian menyimpan file iCal pada OS X dengan <i>file input</i> data baru . . . . .	83

5.68 Hasil pengujian menyimpan file iCal yang telah difilter pada OS X dengan <i>file input</i> data lama . . . . .	83
5.69 Hasil pengujian menyimpan file iCal yang telah difilter pada OS X dengan <i>file input</i> data baru . . . . .	84
5.70 Hasil pengujian <i>import</i> iCalendar pada OS X dengan <i>file input</i> data lama . . . . .	84
5.71 Notifikasi destinasi kalender pada OS X dengan <i>file input</i> data lama . . . . .	84
5.72 Tampilan iCalendar pada OS X dengan <i>file input</i> data lama . . . . .	85
5.73 Tampilan iCalendar bagian 2 pada OS X dengan <i>file input</i> data lama . . . . .	85
5.74 Tampilan iCalendar pada OS X dengan <i>file input</i> data baru . . . . .	86
5.75 Tampilan hasil filter pada OS X dengan <i>file input</i> data lama . . . . .	86
5.76 Tampilan hasil filter bagian 2 pada OS X dengan <i>file input</i> data lama . . . . .	86
5.77 Tampilan hasil filter pada OS X dengan <i>file input</i> data baru . . . . .	87
 B.1 File Excel Jadwal Mengawas Format Lama . . . . .	 103
B.2 File Excel Jadwal Mengawas Format Lama . . . . .	104

## DAFTAR TABEL

2.1	Tabel kelas Konstruktor XSSFWorkbook . . . . .	7
2.2	Tabel kelas method XSSFWorkbook . . . . .	7
2.3	Tabel kelas Konstruktor XSSFSheet . . . . .	8
2.4	Tabel kelas method XSSFSheet . . . . .	8
2.5	Tabel kelas method XSSFRow . . . . .	8
2.6	Tabel ringkasan tipe cell . . . . .	9
2.7	Tabel kelas method XSSFCell . . . . .	9
2.8	Tabel ringkasan cell style . . . . .	10
2.9	Tabel kelas konstruktor XSSFCellStyle . . . . .	10
2.10	Tabel kelas method XSSFCellStyle . . . . .	11
2.11	Tabel kelas konstruktor XSSFColor . . . . .	11
2.12	Tabel kelas method XSSFColor . . . . .	12
2.13	Tabel kelas konstruktor XSSFFont . . . . .	12
2.14	Tabel kelas method XSSFFont . . . . .	12
2.15	Tabel field XSSFHyperlink . . . . .	13
2.16	Tabel kelas method XSSFHyperlink . . . . .	13
2.17	Tabel kelas method XSSFCreationHelper . . . . .	13
2.18	Tabel kelas method XSSFPrintSetup . . . . .	13
2.19	Tabel komponen iCal4j . . . . .	14
2.20	Tabel ringkasan interface net.fortuna.ical4j.data . . . . .	15
2.21	Tabel ringkasan kelas net.fortuna.ical4j.data . . . . .	15
2.22	Tabel ringkasan interface net.fortuna.ical4j.filter . . . . .	15
2.23	Tabel ringkasan kelas net.fortuna.ical4j.filter . . . . .	16
2.24	Tabel ringkasan interface net.fortuna.ical4j.model . . . . .	16
2.25	Tabel ringkasan kelas net.fortuna.ical4j.model . . . . .	17
2.26	Tabel ringkasan kelas net.fortuna.ical4j.model.component . . . . .	18
2.27	Tabel ringkasan kelas net.fortuna.ical4j.model.parameter . . . . .	19
2.28	Tabel ringkasan kelas net.fortuna.ical4j.model.property . . . . .	20
2.30	Tabel ringkasan kelas net.fortuna.ical4j.model.transform . . . . .	20
2.31	Tabel ringkasan interface net.fortuna.ical4j.model.util . . . . .	21
2.32	Tabel ringkasan kelas net.fortuna.ical4j.model.util . . . . .	21
2.33	Tabel komponen JavaFX . . . . .	26
2.34	Tabel method ObservableList . . . . .	27
2.35	Tabel method ListChangeListener . . . . .	27
2.36	Tabel method TextField . . . . .	27
2.37	Tabel method TableView . . . . .	27
2.38	Tabel method TableColumn . . . . .	28
3.1	Tabel penjelasan kolom pada excel mengawas ujian . . . . .	30
3.2	Tabel analisa kolom pada excel mengawas ujian . . . . .	32
4.1	Tabel Kelas <i>ScheduleClass</i> . . . . .	42
4.2	Tabel Kelas <i>ExcelConverter</i> . . . . .	42

4.3	Tabel Kelas <i>CalendarConverter</i> . . . . .	43
4.4	Tabel Kelas <i>FXMLDocumentController</i> . . . . .	43
4.5	Tabel Kelas <i>iCalendarMainConverter</i> . . . . .	44
5.1	Tabel hasil pengujian fungsional . . . . .	61
5.2	Tabel hasil pengujian eksperimental . . . . .	70
5.3	Tabel hasil pengujian eksperimental setelah revisi . . . . .	72
5.4	Tabel hasil pengujian pada OS X . . . . .	80





# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Jadwal mengawas ujian di FTIS(Fakultas Teknologi Infomasi dan Sains) merupakan hal yang rutin dibagikan filenya kepada dosen setiap semester. Jadwal mengawas tersebut dipublikasikan oleh TU(Tata Usaha). Format jadwal mengawas ujian bersifat umum, dalam arti jadwal tersebut menyimpan nama semua dosen yang mengawas, nama mata kuliah, dan tempat pelaksanaan ujian. Dosen diharuskan melihat satu persatu baris untuk mendapatkan informasi mengenai waktu, nama matakuliah, dan tanggal dosen tersebut mengawas. Walaupun jadwal mengawas tersebut telah disusun dalam file excel, namun tetap dirasa kurang efisien karena tidak tersusun berdasarkan dosen yang mengawas dan memungkinkan terjadi kesalahan dalam membaca jadwal oleh dosen. iCalendar merupakan format file *calendar* pada komputer yang memudahkan penggunaanya untuk mengirimkan undangan *meeting* dan melakukan pekerjaan bersama pengguna lainnya, via email, atau file *sharing* menggunakan ekstensi .ics . Format iCalendar sendiri telah didukung dan kompatibel dengan produk lainnya, seperti Google Calendar, Microsoft Outlook, Apple Calendar. Tugas akhir ini dimaksudkan untuk memudahkan dosen untuk melihat jadwal mengawas ujian. Pengembangan perangkat lunak ini menggunakan tiga *library* yaitu Apache POI, JavaFX, dan iCal4j.

### 1.2 Rumusan Masalah

Berdasarkan penjelasan di latar belakang, maka dapat dipaparkan rumusan masalah sebagai berikut :

1. Bagaimana perangkat lunak dapat membaca file excel jadwal mengawas ujian yang dibuat oleh TU ?
2. Bagaimana perangkat lunak mengkonversi jadwal mengawas menjadi iCalendar ?
3. Bagaimana kompatibilitas dengan aplikasi lain ?

### 1.3 Tujuan

Tujuan dari skripsi ini dapat dipaparkan sebagai berikut:

1. Merancang PL(Perangkat Lunak) yang mampu membaca file excel yang dipublikasikan oleh TU.
2. Merancang PL(Perangkat Lunak) dapat mengkonversi jadwal mengawas menjadi iCalendar.
3. Merancang PL(Perangkat Lunak) yang kompatibel dengan aplikasi lain.

## 1.4 Batasan Masalah

Batasan masalah dalam penelitian ini agar dapat fokus pada pengembangan perangkat lunak konversi jadwal mengawas ujian :

- Diasumsikan format excel jadwal mengawas yang menjadi *input* sudah fix.
- File excel jadwal mengawas berekstensi .xlsx.
- *Sheet* yang dibaca dan dikonversi oleh PL adalah *Sheet* pertama.

## 1.5 Metodologi

Untuk menunjang penelitian maka diperlukan data untuk pengujian maupun pengetahuan teori yang akan diterapkan. Berikut adalah kegiatan yang akan dilakukan:

1. Melakukan studi pustaka mengenai
  - (a) Apache POI
  - (b) JavaFX
  - (c) iCal4j
2. Melakukan analisis pada file excel jadwal mengawas ujian yang dikeluarkan oleh TU.
3. Melakukan perancangan yang terdiri dari use case, diagram aktifitas, dan *user interface*.
4. Melakukan pengujian perangkat lunak dengan berbagai kemungkinan kasus.
5. Menyimpulkan atas serangkaian pengembangan yang dilakukan
6. Menulis dokumen skripsi

## 1.6 Sistematika Pembahasan

1. Bab 1 Pendahuluan  
Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.
2. Bab 2 Dasar Teori  
Bab ini berisi tentang teori dasar tentang File Excel, iCalendar, Apache POI, iCal4j, Java FX.
3. Bab 3 Analisis  
Bab ini berisi tentang analisis kebutuhan dan fitur PL, diagram aktifitas PL, use case, diagram kelas.
4. Bab 4 Perancangan  
Bab ini berisi tentang perancangan kelas dalam PL dan gambaran *user interface*.
5. Bab 5 Implementasi dan Pengujian  
Bab ini berisi tentang penerapan hasil rancangan pada bab sebelumnya serta pengujian perangkat lunak.
6. Bab 6 Kesimpulan dan Saran  
Bab ini berisi tentang kesimpulan yang didapatkan dari hasil pengujian serta saran apabila ingin melanjutkan pengembangan ini.

## BAB 2

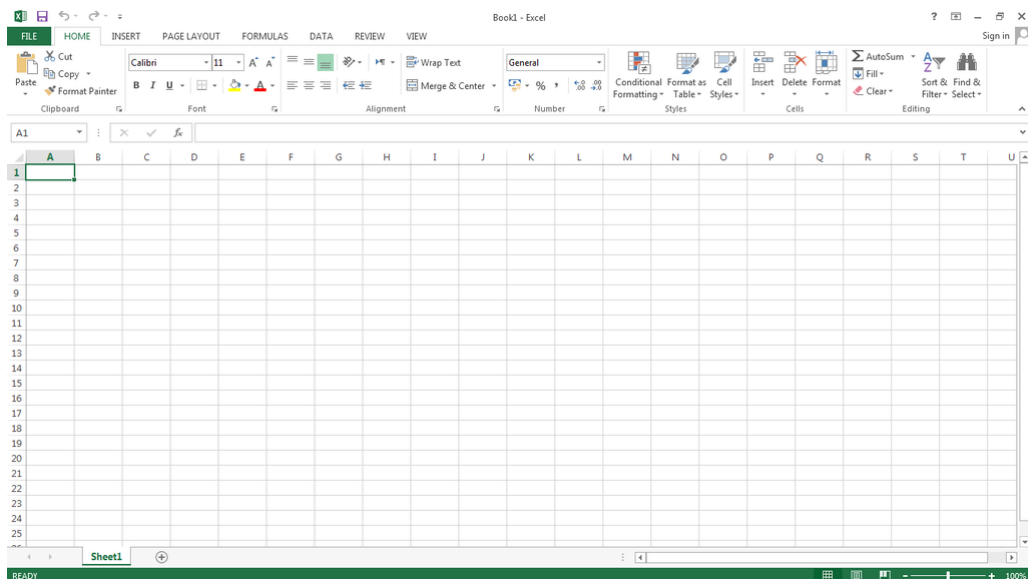
### LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai konsep-konsep dasar pendukung PL, yaitu File Excel, iCalendar, JavaFX, Apache POI, iCal4j.

#### 2.1 Microsoft Excel

Microsoft Excel atau Microsoft Office Excel adalah sebuah program aplikasi lembar kerja *spreadsheet* yang dibuat dan didistribusikan oleh Microsoft Corporation yang dapat dijalankan pada Microsoft Windows dan Mac OS. Aplikasi ini memiliki fitur kalkulasi, pembuatan grafik.<sup>[1]</sup>

##### 2.1.1 Fitur Dasar



Gambar 2.1: *Spreadsheet* Microsoft Excel

Microsoft Excel mempunyai semua fitur dasar dari *spreadsheet*. MS(Microsoft) Excel memiliki sebuah grid dari sel yang diatur dalam beberapa kolom. Setiap kolom berisi huruf kapital yang berfungsi untuk mengatur atau memanipulasi data, contohnya operasi aritmatika. Terdapat beberapa fungsi atau rumus yang disediakan untuk membantu menjawab persoalan statistik, teknik, dan kebutuhan keuangan. Sebagai tambahan, MS Excel dapat menampilkan data dalam *line graph*, histogram dan chart.<sup>[1]</sup>

### 2.1.2 Format File

Sebelum 2007 MS Excel memiliki paten dari format file biner yang dinamakan *Excel Binary File Format*(.XLS). Pada tahun 2007 dengan penambahan XML maka format file dari excel berubah menjadi .xlsx . [1]

## 2.2 iCalendar

iCalendar merupakan format file komputer yang memungkinkan pengguna internet untuk mengirimkan undangan undangan pertemuan dengan pengguna internet lainnya dengan berbagai atau mengirimkan file dalam format yang memiliki ekstensi .ics .[2]

### 2.2.1 Komponen Utama

Pada elemen bagian atas iCalendar terdiri dari rangkaian kalender, dan informasi penjadwalan. Sebuah informasi jadwal dapat berisi objek iCalendar tunggal maupun beberapa objek iCalendar yang buat group.

Baris pertama adalah BEGIN:VCALENDAR, dan baris terakhir berisi END:VCALENDAR, konten jadwal berada diantara kedua baris ini. Baris konten biasa disebut "*icalbody*". Baris kedua merupakan versi format yang digunakan. Ada dua versi dari vCalendar yaitu VERSION:1.0 dan VERSION:2.0. Bodi dari objek iCalendar terdiri dari list properti kalender dan satu atau lebih komponen kalender. Properti dari kalender dapat di aplikasikan ke seluruh kalender. Komponen Kalender terdiri dari beberapa properti kalender untuk membuat sebuah skema kalender(desain). Misalnya, komponen kalender dapat menentukan sebuah event, daftar to-do, sebuah entri jurnal, informasi zona waktu, informasi waktu sibuk/lenggang, atau sebuah alarm.

Berikut ini contoh dari sebuah objek iCalendar. Obek iCalendar ini berisi hari ulang tahun bastille yang dilaksanakan pada 14 juli 1997, pukul 17:00(UTC) sampai 15 juli 1997, pukul 03:59(UTC).[2]

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
UID:uid1@example.com
DTSTAMP:19970714T170000Z
ORGANIZER;CN=John Doe:MAILTO:john.doe@example.com
DTSTART:19970714T170000Z
DTEND:19970715T035959Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

Gambar 2.2: Contoh iCalendar

### 2.2.2 Event(VEVENT)

VEVENT menggambarkan sebuah event yang dijadwalkan dengan rentang waktu tertentu. Sebuah VEVENT memiliki DTSTART untuk menetapkan waktu awal, dan DTEND menetapkan waktu akhir dari sebuah event.[2]

```
BEGIN:VALARM
TRIGGER:-PT1440M
ACTION:DISPLAY
DESCRIPTION:Reminder
END:VALARM
```

Gambar 2.3: Contoh VEVENT pada iCalendar

### 2.2.3 Tipe Komponen Lain

Terdapat komponen lain pada iCalendar, yaitu VTIMEZONE (zona waktu) and VALARM (alarm). VTIMEZONE menerangkan zona waktu pada kalender. VALARM untuk memasang alarm pengingat pada kalender. UID merupakan tanda pengenal yang membuat kalender. [2]

## 2.3 Apache POI

Apache POI(*Poor Obfuscation Implementation*) pada hakikatnya merupakan *library* untuk memanipulasi dan menciptakan sesuatu melalui Java API( *application programming interface*) dengan memanipulasi berbagai format file berdasarkan *Office Open XML standards*(OOXML) dan dokumen *Microsoft OLE 2 Document Compound Format*(OLE2), Singkatnya dengan library ini memungkinkan untuk membaca dan menulis pada MS Excel menggunakan Java.[3]

### 2.3.1 Komponen Apache POI

Untuk membaca aplikasi MS Office Apache POI mempunyai modul berisi komponen java api untuk membaca dokumen dengan format OLE2 dan OOXML. Berikut ini komponen-komponen dalam Apache POI.[3]

1. Excel *workbooks* (HSSF dan XSSF)
2. Word *document* (HWPF dan XWPF)
3. PowerPoint *presentation* (HSLF dan XSLF)
4. Outlook (HSMF)
5. Visio (HDGF dan XDGF)
6. Publisher (HPBF)

HSSF merupakan singkatan dari *Horrible Spreadsheet Format*, sedangkan XSSF merupakan singkatan dari *XML Spreadsheet Format*. HSSF dan XSSF memberikan cara untuk membuat, membaca, dan memodifikasi XLS spreadsheet. Pada sub bab ini akan difokuskan untuk membahas XSSF sesuai kebutuhan untuk menganalisa file excel jadwal mengawas ujian yang dikeluarkan oleh TU FTIS.[3]

- **HSSFWorkbook** : Kelas ini mempunyai method yang dapat membaca dan menulis file Microsoft Excel dengan format .xls. Kelas ini kompatibel dengan MS-Office versi 97-2003.
- **XSSFWorkbook** : Kelas ini mempunyai method untuk menulis dan membaca Microsoft Excel dan OpenOffice xml dengan format .xlsx. Kelas ini kompatibel dengan MS-Office versi 2007 atau versi barunya.

## XSSFWorkbook

Kelas ini merepresentasikan baik *high* dan *low* level format file excel. XSSFWorkbook merupakan kelas yang berada dalam *package* **org.apache.xssf.usermodel** dan mengimplementasikan antarmuka workbook. Berikut ini list method dan constructor dalam kelas ini.[4]

### Class Constructor

Tabel 2.1: Tabel kelas Konstruktur XSSFWorkbook

No	Constructor dan Deskripsi
1	<b>XSSFWorkbook()</b> Membuat objek XSSFWorkbook.
2	<b>XSSFWorkbook(java.io.File file)</b> Membangun sebuah objek XSSFWorkbook dari file yang diberikan.
3	<b>XSSFWorkbook(java.io.InputStream is)</b> Membangun sebuah object XSSFWorkbook dengan <i>buffering</i> semua input stream ke dalam memory, dilanjutkan dengan membuka objek OPCPackage.
4	<b>XSSFWorkbook(java.lang.String path)</b> Membangun sebuah objek XSSFWorkbook dengan diberikan <i>full path</i> dari sebuah file.

### Class Methods

Tabel 2.2: Tabel kelas method XSSFWorkbook

No	Method dan Deskripsi
1	<b>CreateSheet()</b> Menciptakan atau menambahkan sebuah XSSFSheet pada workbook.
2	<b>createSheet(java.lang.String sheetname)</b> Membuat sheet baru untuk workbook .
3	<b>createFont()</b> Membuat font baru dan menambahkannya pada tabel font pada workbook.
4	<b>createCellStyle()</b> Membuat XSSFCellStyle Baru dan menambahkannya pada tabel style workbook.
5	<b>setPrintArea(int sheetIndex, int startColumn, int endColumn, int startRow, int endRow)</b> Menentukan area print dari kertas dengan parameter yang spesifik.

## Sheet

*Sheet* merupakan sebuah interface dibawah package org.apache.ss.usermodel. *Sheet* merupakan representasi sebuah *grid* dari cell.[4]

## XSSFSheet

Kelas ini merupakan representasi dari *high level* excel spreadsheet. Kelas ini berada di bawah package org.apache.poi.hssf.usermodel.[4]

### Class Constructor

Tabel 2.3: Tabel kelas Konstruktor XSSFSheet

No	Constructor dan Deskripsi
1	<b>XSSFSheet()</b> Membuat baru XSSFSheet yang disebut XSSFWorkbook dalam pembuatan sheet baru .

### Class Methods

Tabel 2.4: Tabel kelas method XSSFSheet

No	Constructor dan Deskripsi
1	<b>addMergedRegion(CellRangeAddress region)</b> Menambahkan gabungan wilayah dari cell.(beberapa cell menjadi satu) .
2	<b>autoSizeColumn(int column)</b> Menyesuaikan lebar kolom agar sesuai dengan isinya.
3	<b>iterator()</b> Method ini alias rowIterator() untuk memungkinkan foreach loop .
4	<b>addHyperlink(XSSFHyperlink hyperlink)</b> Mendaftarkan sebuah hyperlink kedalam koleksi hyperlink yang ada di sheet.

### Row

Row merupakan interface berada di bawah *package* **org.apache.poi.ss.usermodel**. Row ini digunakan untuk *high-level representation* dari sebuah row pada sebuah spreadsheet. Row juga merupakan super-interface dari semua kelas yang mewakili row dalam POI *Library*.<sup>[4]</sup>

### XSSFRow

XSSFRow merupakan sebuah kelas di bawah *package* **org.apache.poi.xssf.usermodel** dan mengimplementasi Row interface. Selain itu, kelas ini dapat membuat row dalam sebuah spreadsheet. List dibawah ini merupakan *method* dan constructors pada kelas ini.<sup>[4]</sup>

### Class Methods

Tabel 2.5: Tabel kelas method XSSFRow

No	Deskripsi
1	<b>createCell(int columnIndex)</b> Membuat cell baru dalam baris.
2	<b>setHeight(short height)</b> Mengatur tinggi dalam satuan short.

### Cell

Cell merupakan interface yang berada di bawah *package* **org.apache.poi.ss.usermodel**. Cell merupakan sebuah super-interface dari semua kelas yang mewakili cell dalam baris sebuah spreadsheet.

Cell dapat berupa berbagai atribut seperti *blank*, *numeric*, *date*, *error*. Sebelum ditambahkan ke baris cell memiliki nomer tersendiri(dari mulai 0).<sup>[4]</sup>



## XSSFCell

Kelas ini berada di bawah *package* **org.apache.poi.xssf.usermodel**. Kelas ini mewakili cell interface. XSSFCell adalah *high-level representation* cell dalam row dari sebuah spreadsheet.[4]

### Ringkasan Tipe Cell

List di bawah ini adalah sebagian *field* dari kelas XSSFCell beserta deskripsinya.

Tabel 2.6: Tabel ringkasan tipe cell

Tipe Cell	Deskripsi
CELL_TYPE_BLANK	Representasi cell kosong
CELL_TYPE_BOOLEAN	Representasi cell Boolean (True atau False)
CELL_TYPE_ERROR	Representasi nilai error dari cell
CELL_TYPE_FORMULA	Representasi dari hasil sebuah formula dalam cell
CELL_TYPE_NUMERIC	Representasi dari data numerik dalam cell
CELL_TYPE_STRING	Representasi dari String(teks) dalam cell

### Class Methods

Tabel 2.7: Tabel kelas method XSSFCell

No	Deskripsi
1	<b>setCellStyle(CellStyle style)</b> Mengatur style untuk cell.
2	<b>setCellType(int cellType)</b> Mengatur tipe cell (numeric, formula, atau String).
3	<b>setCellValue(boolean value)</b> Mengatur nilai boolean dalam sebuah cell.
4	<b>setCellValue(java.util.Calendar value)</b> Mengatur nilai tanggal dari cell .
5	<b>setCellValue(double value)</b> Mengatur nilai numerik dari cell.
6	<b>setCellValue(java.lang.String str)</b> Mengatur nilai String dari cell.
7	<b>setHyperlink(Hyperlink hyperlink)</b> Menambahkan sebuah hyperlink kedalam cell

## XSSFCellStyle

XSSFCellStyle merupakan sebuah kelas yang berada di bawah *package* **org.apache.poi.usermodel**. kelas ini memberikan informasi yang mungkin mengenai format konten pada suatu cell dari spreadsheet. Kelas ini juga memberikan opsi untuk merubah format tersebut. Kelas ini mewakili *CellStyle interface*. [4]

### Ringkasan Cell Style

List di bawah ini adalah sebagian *field* yang diwariskan dari CellStyle interface.[4]

Tabel 2.8: Tabel ringkasan cell style

Nama Field	Deskripsi Field
ALIGN_CENTER	Rata tengah <i>content cell</i>
ALIGN_CENTER_SELECTION	Posisi seleksi tengah horizontal
ALIGN_FILL	Mencocokkan ukuran <i>cell content</i>
ALIGN_JUSTIFY	Mencocokkan ukuran <i>cell content</i> terhadap lebarnya
ALIGN_LEFT	Rata kiri <i>cell content</i>
ALIGN_RIGHT	Rata kanan <i>cell content</i>
BORDER_DASH_DOT	<i>Cell style</i> dengan garis dan titik
BORDER_DOTTED	<i>Cell style</i> dengan border titik
BORDER_DASHED	<i>Cell style</i> dengan border garis
BORDER_THICK	<i>Cell style</i> dengan border tebal
BORDER_THIN	<i>Cell style</i> dengan border tipis
VERTICAL_BOTTOM	Posisi <i>cell content</i> vertikal kebawah
VERTICAL_CENTER	<i>cell content</i> vertikal ketengah
VERTICAL_JUSTIFY	Posisi <i>cell content</i> sejajar secara vertikal
VERTICAL_TOP	Posisi selaras keatas secara vertikal

### Class Constructor

Tabel 2.9: Tabel kelas konstruktor XSSFCellStyle

No	Constructor dan Deskripsi
1	<b>XSSFCellStyle(int cellXfId, int cellStyleXfId, StylesTable stylesSource, ThemesTable theme)</b> Menciptakan cell style dengan bagian yang sudah disediakan.
2	<b>XSSFCellStyle(StylesTable stylesSource)</b> Membuat cell Style kosong.

### Class Methods

Tabel 2.10: Tabel kelas method XSSFCellStyle

No	Method dan Deskripsi
1	<b>setAlignment(short align)</b> Mengatur style secara horizontal untuk cell.
2	<b>setBorderBottom(short border)</b>
3	<b>setBorderColor(XSSFCellStyle.BorderSide side, XSSFColor color)</b> Mengatur warna untuk border yang dipilih.
4	<b>setBorderLeft(Short border)</b> Mengatur tipe border untuk border kiri dari cell .
5	<b>setBorderRight(short border)</b> Mengatur tipe border untuk border kanan dari cell .
6	<b>setBorderTop(short border)</b> Mengatur tipe border untuk border atas dari cell
7	<b>setFillBackgroundColor(XSSFColor color)</b> Mengatur latar belakang warna yang diwakili oleh nilai XSSFColor
8	<b>setFillForegroundColor(XSSFColor color)</b> Mengatur latar depan warna yang diwakili oleh nilai XSSFColor
9	<b>setFillPattern(short fp)</b> Menentukan isi informasi cell dengan pola dan warna solid
10	<b>setFont(Font font)</b> Mengatur font
11	<b>setRotation(short rotation)</b> Mengatur derajat rotasi pada teks dalam cell.
12	<b>setVerticalAlignment(short align)</b> Menetapkan tipe posisi vertical pada cell

### XSSFColor

XSSFColor merupakan sebuah kelas dibawah *package* **org.apache.poi.xssf.usermodel**. Kelas ini mewakili warnma pada spreadsheet. Kelas ini mengimplementasika interface warna. List dibawah ini merupakan beberapa method XSSFColor dan constructornya.<sup>[4]</sup>

#### Class Constructor

Tabel 2.11: Tabel kelas konstruktor XSSFColor

No	Constructor dan Deskripsi
1	<b>XSSFColor()</b> Menciptakan <i>instance</i> baru dari XSSFColor.
2	<b>XSSFColor(byte[] rgb)</b> Membuat <i>instance</i> baru dari XSSFColor menggunakan RGB.
3	<b>XSSFColor(java.awt.Color clr)</b> Membuat <i>instance</i> baru dari XSSFColor menggunakan kelas warna dari <i>awt package</i> .

#### Class Methods

Tabel 2.12: Tabel kelas method XSSFCOLOR

No	Method dan Deskripsi
1	<b>setAuto(boolean auto)</b> Mengatur sebuah nilai boolean untuk mengindikasikan bahwa ctColor bersifat otomatis dan bergantung pada sistem ctColor.
2	<b>setIndexed(int indexed)</b> Mengatur nilai indeks ctColor sebagai sistem ctColor.

### XSSFFont

XSSFFont merupakan kelas dibawah *package* **org.apache.poi.xssf.usermodel**. Kelas ini mengimplementasikan *Font interface* dan oleh sebab itu kelas ini dapat menangani font berbeda pada sebuah workbook.[4]

#### Class Constructor

Tabel 2.13: Tabel kelas konstruktor XSSFFont

No	Constructor dan Deskripsi
1	<b>XSSFFont()</b> Menciptakan <i>instance</i> baru dari XSSFFont.

#### Class Methods

Tabel 2.14: Tabel kelas method XSSFFont

No	Method dan Deskripsi
1	<b>setBold(boolean bold)</b> Mengatur sebuah nilai boolean untuk atribut 'bold'.
2	<b>setColor(short color)</b> Mengatur nilai indeks warna untuk font.
3	<b>setColor(XSSFCOLOR color)</b> Mengatur warna untuk font dalam standar nilai warna Alpha RGB.
4	<b>setFontHeight(short height)</b> Mengatur tinggi font dalam poin.
5	<b>setFontName(java.lang.String name)</b> Mengatur nama dari font.
6	<b>setItalic(boolean italic)</b> Mengatur nilai boolean pada properti 'italic'.

### XSSFHyperlink

XSSFHyperlink merupakan kelas dibawah *package* **org.apache.poi.xssf.usermodel**. Kelas ini mengimplementasikan *Hyperlink interface*. Kelas ini digunakan untuk mengatur sebuah hyperlink pada *cell content* dalam sebuah spreadsheet.[4]

#### Field

Field dalam kelas ini akan didefinisikan sebagai berikut. Field disini dalam arti tipe dari hyperlink yang dipakai.

Tabel 2.15: Tabel field XSSFHyperlink

Field	Deskripsi
LINK_DOCUMENT	Dipakai untuk menghubungkan dengan dokumen lainnya
LINK_EMAIL	Digunakan untuk menghubungkan dengan email
LINK_FILE	Digunakan untuk menghubungkan dengan file lain dalam berbagai format
LINK_URL	Digunakan untuk menghubungkan dengan URL <i>website</i>

### Class Methods

Tabel 2.16: Tabel kelas method XSSFHyperlink

No	Method dan Deskripsi
1	<b>setAddress(java.lang.String address)</b> Alamat Hyperlink.

### XSSFCreationHelper

XSSFCreationHelper merupakan kelas dibawah *package* **org.apache.poi.xssf.usermodel**. Kelas ini mengimplementasikan *CreationHelper interface*. Kelas ini digunakan sebagai bentuk kelas pendukung untuk *formula evaluation* dan penyusun hyperlink.<sup>[4]</sup>

### Class Methods

Tabel 2.17: Tabel kelas method XSSFCreationHelper

No	Method dan Deskripsi
1	<b>createFormulaEvaluator()</b> Membuat sebuah <i>instance</i> XSSFFormulaEvaluator, objek yang dapat mengevaluasi formula dalam cell.
2	<b>createHyperlink(int type)</b> Membuat sebuah XSSFHyperlink baru.

### XSSFPrintSetup

XSSFPrintSetup merupakan kelas dibawah *package* **org.apache.poi.xssf.usermodel**. Kelas ini mengimplementasikan *PrintSetup interface*. Kelas ini digunakan untuk mengatur ukuran cetak pada halaman, wilayah cetak, opsi, dan pengaturan.<sup>[4]</sup>

### Class Methods

Tabel 2.18: Tabel kelas method XSSFPrintSetup

No	Method dan Deskripsi
1	<b>setLandscape(boolean ls)</b> Mengatur sebuah nilai boolean yang dapat mengijinkan atau menolak <i>landscape printing</i> .
2	<b>setLeftToRight(boolean ltor)</b> Mengatur perintah ke kiri, kanan, atas, atau bawah ketika proses cetak.
3	<b>setPaperSize(short size)</b> Mengatur ukuran kertas.

## 2.4 iCal4j

iCal4j merupakan *Java library* yang digunakan untuk membaca dan menulis data iCalendar yang didefinisikan dalam RFC2445. iCalendar standar menyediakan sebuah format data yang umumnya digunakan untuk menyimpan informasi tentang spesifikasi kalender seperti acara, pertemuan, *to-do list*, dll. Semua *tool* kalender yang populer, seperti Lotus Notes, Outlook, Google Calendar, Apple iCal mensupport standar iCalendar.[5]

Sebagai pengurai kalender dan *object model*, iCal4j memudahkan untuk memodifikasi data kalender yang sudah ada atau membuat model data baru. Validasi juga diperlukan untuk memastikan data terjaga baik dan konsisten dengan spesifikasi yang diperlukan.[5]

### 2.4.1 Komponen iCal4j

Berikut ini merupakan kumpulan *package* yang ada dalam iCal4j.[6]

Tabel 2.19: Tabel komponen iCal4j

No	Package dan Deskripsi
1	<b>net.fortuna.ical4j.data</b> Menyediakan berbagai tipe RFC2445 input, output, serta fungsi parsing.
2	<b>net.fortuna.ical4j.filter</b> Aturan untuk menyaring list komponen yang digunakan, <i>properties</i> , maupun parameter yang digunakan.
3	<b>net.fortuna.ical4j.model</b> Berisikan komponen utama yang digunakan untuk mendefinisikan model iCalendar.
4	<b>net.fortuna.ical4j.model.component</b> Berisikan representasi tipe yang digunakan dalam komponen model iCalendar.
5	<b>net.fortuna.ical4j.model.parameter</b> Berisikan representasi tipe yang digunakan dalam parameter model iCalendar.
6	<b>net.fortuna.ical4j.model.property</b> Berisikan representasi tipe yang digunakan dalam properti model iCalendar.
7	<b>net.fortuna.ical4j.transform</b> Berisikan perubahan tipe yang digunakan komponen model iCalendar sesuai RFC2446 .
8	<b>net.fortuna.ical4j.util</b> Berisikan tipe utilitas yang mendukung fungsi dari iCal4j.

### 2.4.2 Kelas Inti dari iCal4j

Pada bagian ini *package* yang ditulis di sub bab sebelumnya akan dijelaskan lebih dalam apa kegunaannya.[6]

### 2.4.3 net.fortuna.ical4j.data

#### Ringkasan Interface

Tabel 2.20: Tabel ringkasan interface net.fortuna.ical4j.data

No	Method dan Deskripsi
1	<b>CalendarParser</b> Pelaksana yang menyediakan fungsi parsing pada iCalendar.
2	<b>ContentHandler</b> Pelaksana yang menyediakan fungsi yang berlaku selama parsing aliran data dari iCalendar (misalnya membangun model objek).

### Ringkasan Kelas

Tabel 2.21: Tabel ringkasan kelas net.fortuna.ical4j.data

No	Method dan Deskripsi
1	<b>AbstractOutputter</b> kelas dasar untuk model <i>output</i> .
2	<b>CalendarBuilder</b> Parsing dan menangani sebuah model iCalendar dari input stream.
3	<b>CalendarOutputter</b> Menuliskan sebuah model iCalendar pada output stream.
4	<b>CalendarParserFactory</b> Menyediakan akses pada CalendarParser yang telah dikonfigurasi.
5	<b>CalendarParserImpl</b> Implementasi <i>default</i> dari CalendarParser.
6	<b>DefaultCalendarParserFactory</b> Implementasi <i>default</i> dari CalendarParser.
7	<b>FoldingWriter</b> Fungsi penulisan yang mendukung penulisan iCalendar berlipat.
8	<b>HCalendarParser</b> Menguraikan dokumen XHTML yang meliputi data kalender, ditandai dengan mikroformat hCalendar.
9	<b>HCalendarParserFactory</b> kumpulan parser untuk mikroformat hCal
10	<b>UnfoldingReader</b> Fungsi membaca bagian iCalendar yang wajib dibaca.

#### 2.4.4 net.fortuna.ical4j.filter

##### Ringkasan Interface[\[6\]](#)

Tabel 2.22: Tabel ringkasan interface net.fortuna.ical4j.filter

No	Method dan Deskripsi
1	<b>Rule</b> Pelaksana yang menentukan apakah suatu objek tertentu diklasifikasikan sebagai pasangannya dapat dijadikan sebagai filter lampiran.

##### Ringkasan Kelas[\[6\]](#)

Tabel 2.23: Tabel ringkasan kelas net.fortuna.ical4j.filter

No	Method dan Deskripsi
1	<b>DateInRangeRule</b> Mengimplementasikan Rule.
2	<b>Filter</b> Melakukan filtering dari seperangkat aturan. Sebuah filter dapat menentukan apakah setidaknya satu aturan tersebut cocok atau tidak.
3	<b>HasPropertyRule</b> Sebuah aturan yang mencocokkan komponen memuat properti yang spesifik.
4	<b>PeriodRule&lt;T extends Component&gt;</b> Sebuah aturan yang mencocokkan komponen terjadi atau tidak dalam jangka waktu yang ditentukan.

### 2.4.5 net.fortuna.ical4j.model

#### Ringkasan Interface[\[6\]](#)

Tabel 2.24: Tabel ringkasan interface net.fortuna.ical4j.model

No	Method dan Deskripsi
2	<b>Escapable</b> Pelaksana yang mengkonversi ke/dari nilai string kedalam bentuk iCalendar.
4	<b>ParameterFactory&lt;T extends Parameter&gt;</b> Pelaksana yang menyediakan pembuatan <i>service</i> parameter.
5	<b>PropertyFactory&lt;T extends Property&gt;</b> Membuat properti iCalendar.
6	<b>TimeZoneRegistry</b> Menyediakan daftar definisi wilayah yang berlaku untuk digunakan objek iCalendar.

#### Ringkasan Kelas[\[6\]](#)



Tabel 2.25: Tabel ringkasan kelas net.fortuna.ical4j.model

No	Method dan Deskripsi
1	<b>AddressList</b> Mendefinisikan list dari alamat pada iCalendar.
2	<b>Calendar</b> Mendefinisikan kalender pada iCalendar.
3	<b>CalendarDateFormatFactory</b> Membuat objek dateFormat untuk optimisasi pola tanggal pada iCalendar.
4	<b>Date</b> Representasi dari objek DATE sesuai RFC5445.
5	<b>DateList</b> Representasi list tanggal dari iCalendar.
6	<b>DateTime</b> Representasi dari objek DATE-TIME sesuai RFC5445.
7	<b>LocationTypeList</b> Menetapkan sebuah list tipe lokasi dari iCalendar.
8	<b>NumberList</b> Menetapkan list dari nomer.
9	<b>Parameter</b> Mendefinisikan parameter.
10	<b>Period</b> Mendefinisikan periode waktu.
11	<b>Property</b> Mendefinisikan properti dari iCalendar.
12	<b>Time</b> Sebuah tipe yang merepresentasikan nilai waktu pada iCalendar.
13	<b>TimeZone</b> Implementasi zona waktu java.
14	<b>WeekDay</b> Mendefinisikan hari dalam seminggu dengan diimbangi terkait dengan kejadian bulanan atau tahunan.

#### 2.4.6 net.fortuna.ical4j.model.component

Ringkasan Kelas[\[6\]](#)

Tabel 2.26: Tabel ringkasan kelas net.fortuna.ical4j.model.component

No	Method dan Deskripsi
1	<b>Available</b> Mendefinisikan komponen tersedia di iCalendar.
2	<b>Daylight</b> Mendefinisikan waktu siang dalam zona waktu.
3	<b>Standard</b> Mendefinisikan komponen zona waktu standar.
4	<b>Standard.Factory VAlarm</b> Mendefinisikan komponen VALARM pada iCalendar.
5	<b>Standard.Factory VAvailability</b> Mendefinisikan komponen VAvailability pada iCalendar.
6	<b>VAvailability.Factory VEvent</b> Mendefinisikan komponen VEvent pada iCalendar.
7	<b>VEvent.Factory VFreeBusy</b> Mendefinisikan komponen VFreeBusy pada iCalendar.
8	<b>VFreeBusy.Factory VJournal</b> Mendefinisikan komponen VJournal pada iCalendar.
9	<b>VJournal.Factory VTimeZone</b> Mendefinisikan komponen VTimeZone pada iCalendar.
10	<b>VTimeZone.Factory VToDo</b> Mendefinisikan komponen VToDo pada iCalendar.
11	<b>VToDo.Factory VVenue</b> Mendefinisikan komponen VVenue pada iCalendar.

#### 2.4.7 net.fortuna.ical4j.model.parameter

Ringkasan Kelas[\[6\]](#)

Tabel 2.27: Tabel ringkasan kelas net.fortuna.ical4j.model.parameter

No	Method dan Deskripsi
1	<b>Abbrev</b> Mendefinisikan parameter singkatan.
2	<b>AltRep</b> Mendefinisikan alternatif representasi parameter teks.
3	<b>Rsvp</b> Mendefinisikan parameter RSVP.
4	<b>ScheduleAgent</b> Mendefinisikan penjadwalan.
5	<b>ScheduleStatus</b> Mendefinisikan status penjadwalan.
6	<b>SentBy</b> Mendefinisikan parameter pengirim.
7	<b>Type</b> Mendefinisikan parameter tipe.
8	<b>TzId</b> Mendefinisikan parameter zona waktu.
9	<b>Value</b> Mendefinisikan parameter nilai tipe data.
10	<b>Vvenue</b> Mendefinisikan parameter Vvenue.
11	<b>XParameter</b> Mendefinisikan parameter pemicu alarm.
12	<b>Related</b> Mendefinisikan penambahan parameter.

#### 2.4.8 net.fortuna.ical4j.model.property

Ringkasan Kelas[\[6\]](#)

Tabel 2.28: Tabel ringkasan kelas net.fortuna.ical4j.model.property

No	Method dan Deskripsi
1	<b>Action</b> Mendefinisikan aksi dari komponen properti iCalendar .
2	<b>Attach</b> Mendefinisikan lampiran dari komponen properti iCalendar.
3	<b>Attendee</b> Mendefinisikan kedatangan dari komponen properti iCalendar.
4	<b>Contact</b> Mendefinisikan kontak pada komponen properti.
5	<b>Country</b> Mendefinisikan negara pada komponen properti.
6	<b>Created</b> Mendefinisikan pembuatan pada komponen properti.
7	<b>Description</b> Mendefinisikan deskripsi pada komponen properti.
8	<b>DtEnd</b> Mendefinisikan DtEnd pada komponen properti.
9	<b>DtStamp</b> Mendefinisikan DtStamp pada komponen properti.
10	<b>DtStart</b> Mendefinisikan DtStart pada komponen properti.
11	<b>Due</b> Mendefinisikan Due pada komponen properti.
12	<b>Duration</b> Mendefinisikan Durasi pada komponen properti.
13	<b>Location</b> Mendefinisikan lokasi pada komponen properti.
14	<b>Name</b> Mendefinisikan nama pada komponen properti
15	<b>StreetAddress</b> Mendefinisikan alamat pada komponen properti.
16	<b>Summary</b> Mendefinisikan ringkasan pada komponen properti.
17	<b>Url</b> Mendefinisikan url pada komponen properti.

#### 2.4.9 net.fortuna.ical4j.model.transform

##### Ringkasan Kelas[\[6\]](#)

Tabel 2.30: Tabel ringkasan kelas net.fortuna.ical4j.model.transform

No	Method dan Deskripsi
1	<b>PublishTransformer</b> Merubah kalender untuk dipublikasikan.
2	<b>Transformer</b> <i>Base Class</i> untuk transforasi kalender.

### 2.4.10 net.fortuna.ical4j.model.util

#### Ringkasan Interface[6]

Tabel 2.31: Tabel ringkasan interface net.fortuna.ical4j.model.util

No	Method dan Deskripsi
1	<b>HostInfo</b> Menyediakan informasi host berupa <i>paltform</i> yang independen.

#### Ringkasan Kelas[6]

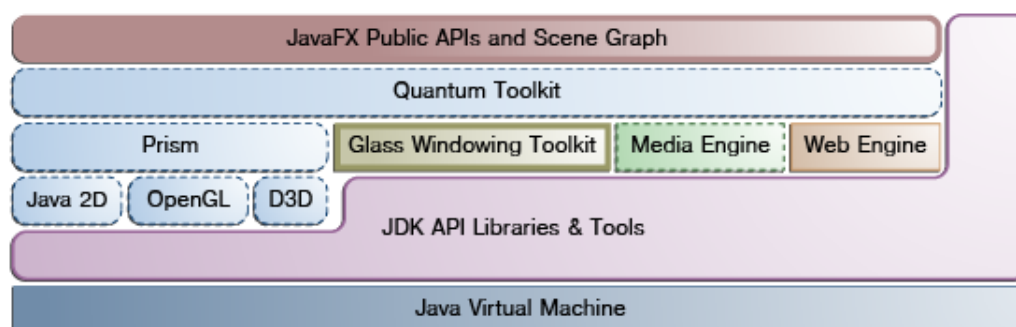
Tabel 2.32: Tabel ringkasan kelas net.fortuna.ical4j.model.util

No	Method dan Deskripsi
1	<b>Calendars</b> Method utility untuk bekerja dengan kalender.
2	<b>Dates</b> Mengimplementasikan koleksi dari method utility yang relevan untuk memproses tanggal.
3	<b>Numbers</b> kelas utility untuk memproses nomer.
4	<b>Strings</b> method utility yang bekerja dengan parameter.
5	<b>TimeZones</b> method utility yang relevan dengan zona waktu Java.

## 2.5 JavaFX

JavaFX merupakan seperangkat grafis dan paket media yang memungkinkan pengembang untuk merancang, membuat, menguji, *debug* , dan beroperasi secara konsisten di seluruh platform yang beragam.[7]

Berikut ini ilustrasi arsitektur dari gambar JavaFX.



Gambar 2.5: Arsitektur JavaFX

Ilustrasi dari gambar 2.1 mendeskripsikan setiap komponen saling berhubungan. Dibawah JavaFX Public API terdapat mesin yang menjalankan code JavaFX. Mesin tersebut terdiri dari sub komponen termasuk mesin grafis berperforma tinggi yang dinamakan Prism. Selain itu, terdapat sistem *windowing* kecil dan efisien yang dinamakan Glass. Terakhir dalam mesin dibawah JavaFX Public API terdapat sebuah *media engine* dan *web engine*. Berikut ini elemen-elemen yang terdapat pada arsitektur JavaFX : [8]

1. *Scene Graph*
2. Java Public API untuk Fitur JavaFX
3. *Graphics System*
4. *Glass Windowing Toolkit*
5. Gambar dan Media
6. Komponen Web
7. CSS
8. *UI Control*
9. Layout
10. Transformasi 2-D dan 3-D
11. *Visual Effects*

### 2.5.1 Scene Graph

*Scene Graph* merupakan sebuah pohon hirarki dari sekumpulan node yang merepresentasikan elemen visual dari antarmuka suatu aplikasi. Sebuah elemen dari *scene graph* dinamakan node. Setiap node mempunyai ID, *style class* dan *bounding volume*. Node dalam *scene graph* juga memiliki :[8]

1. *Effect*, seperti blur dan shadow
2. *Opacity*
3. *Transform*
4. *Event handler* (Mouse, keyboard, dan input method lainnya)
5. Perintah spesifik dari sebuah aplikasi

Penggunaan **javafx.scene** API memungkinkan *developer* untuk menggunakan beberapa jenis konten di dalamnya, seperti : [8]

1. **Node** : Bentuk(2-D dan 3-D), gambar, media, *embedded web browser*, teks, *UI control*, grafik, grup, dan *container*.
2. **State** : Transformasi (posisi dan orientasi dari node), efek visual, dan konten visual lainnya.
3. **Effect** : objek sederhana yang dapat merubah penampilan dari node *scene graph*, seperti blur, shadow, dan *color adjustment*

### 2.5.2 Java Public API untuk Fitur JavaFX

Pada lapisan atas arsitektur JavaFX pada gambar 2.1 API Java memberikan kebebasan dan fleksibilitas untuk membangun berbagai *client* dari sebuah aplikasi. Platform JavaFX menggabungkan kemampuan terbaik yang dimiliki platform Java secara menyeluruh dan mendalam serta intuitif dengan memasukan fungsi media kedalamnya, sehingga tercipta lingkup konsep *one-stop development*. Berikut contoh kegunaan Java API untuk fitur JavaFX :[8]

1. Memungkinkan penggunaan fitur Java yang powerful seperti *generics*, *annotations*, *multithreading*.

2. Lebih mudah mengembangkan web menggunakan JavaFX dibanding *JVM-base dynamic languages* lainnya seperti Groovy, dan JavaScript.
3. Memungkinkan Java developer untuk menggunakan bahasa sistem seperti Groovy untuk menulis file besar atau kompleks pada aplikasi JavaFX.
4. Memungkinkan penggunaan binding.
5. Menambahkan koleksi library Java dengan memasukan urutan dan memetakan perubahan sehingga memungkinkan aplikasi untuk menghubungkan antarmuka kedalam data model, mengamati perubahan pada data model, dan memperbarui kontrol UI yang sesuai dengan perubahan tersebut.

### 2.5.3 Graphic System

*JavaFX Graphic System* pada gambar **2.1** merupakan implementasi dari JavaFX *scene graph layer*. Sistem grafis pada JavaFX mendukung tampilan 2-D dan 3-D, selain itu sistem grafis ini menyediakan *software rendering* untuk mendukung akselerasi *rendering* dari *hardware*. Berikut ini merupakan dua *graphic accelerated pipeline* yang ada pada JavaFX platform :[8]

1. **Prism** yang bekerja pada proses render. Prism dapat bekerja pada kedua sisi baik *hardware* maupun *software* rendering termasuk 3-D rendering. Prism juga bertanggung jawab untuk proses *rasterization*(mengubah vektor menjadi pixel atau dot) dan rendering pada JavaFX.
2. **Quantum Toolkit** merupakan perpaduan Prism dan Windowing Toolkit yang bekerja di lapisan teratas pada JavaFX untuk mengatur *threading rule* yang berhubungan dengan rendering dan *event handling*.

### 2.5.4 Glass Windowing Toolkit

Tugas pada lapisan ini adalah membantu *service* pada sistem operasi, seperti mengatur windows, waktu , dan *surface*. Glass Toolkit juga bertanggung jawab atas pengaturan *event queue*. [8]

### 2.5.5 Media dan Gambar

Fungsi -fungsi media pada JavaFX tersedia pada **javafx.scene.media** API. JavaFX mensupport baik visual maupun audio. Beberapa format yang disupport seperti MP3, AIFF, WAV pada file audio dan format FLV pada video. Ada tiga komponen yang berperan pada JavaFX media, yaitu :[8]

1. *Media object* merepresentasikan sebuah file media.
2. *Media Player* memutar sebuah file media.
3. *Media View* merupakan sebuah node yang menampilkan media tersebut.

### 2.5.6 Komponen Web

Mesin Web pada JavaFX merupakan bagian dari JavaFX UI control yang berbasis Webkit, dimana mesin web ini dapat menampilkan sebuah website dan melakukan browsing melalui APInya. Berikut ini fitur JavaFX yang dapat di implementasikan pada program java :[8]

1. Render konten HTML dari local atau remote URL.
2. Mendukung history dan menyediakan navigasi Back dan Forward.
3. *Reload Content*.

4. Edit konten HTML.
5. Mengeksekusi perintah JavaScript.
6. *Handle event*.

Komponen dari browser tersebut terbagi kedalam ke kelas-kelas berikut :[8]

1. **WebEngine** : menyediakan kemampuan dasar dari halaman web.
2. **WebView** : merangkum sebuah *WebEngine object*, Menggabungkan konten HTML kedalam layar aplikasi, dan mendukung *field* dan *method* untuk menerapkan efek dan transformasi berupa ekstensi maupun sebuah kelas Node.

### 2.5.7 CSS

JavaFX Cascading Style Sheet (CSS) mendukung kemampuan untuk mengkustom styling pada antarmuka sebuah aplikasi JavaFX tanpa merubah *source code* aplikasi tersebut.[8]

### 2.5.8 UI Control

JavaFX UI Control dalam JavaFX API dibangun menggunakan node pada scene graph. JavaFX UI Control dapat mengambil keuntungan dari fitur yang diberikan platform JavaFX dan bersifat *portable* pada platform yang berbeda.[8]



Gambar 2.6: Contoh JavaFX UI Control

pada gambar 2.2 menunjukan UI Control yang sementara didukung oleh JavaFX. Java UI control baru seperti TitlePane atau Accordion sebelumnya telah diperkenalkan pada JavaFX SDK. UI control tersebut terdapat pada **javafx.scene.control** package.[8]



### 2.5.9 Layout

*Layout container* atau panel digunakan untuk pengaturan UI control secara dinamis dan fleksibel dalam scene graph pada aplikasi JavaFX. JavaFX Layout API mempunyai kelas-kelas yang dapat mengotomatiskan tata letak model sebagai berikut:[8]

1. **BorderPane** merupakan kelas yang mengatur bagian atas, bawah, kiri, kanan layout.
2. **Hbox** merupakan kelas yang mengatur konten node secara horizontal dalam satu baris.
3. **Vbox** merupakan kelas yang mengatur konten node secara vertikal dalam satu baris.
4. **StackPane** adalah kelas yang menempatkan *back-to-front* konten node pada suatu *stack*.
5. **GridPane** adalah kelas yang memungkinkan developer untuk membuat sebuah grid baris dan kolom secara flexible untuk memetakan konten node.
6. **FlowPane** adalah kelas yang mengatur alur konten node baik horizontal maupun vertical, *wrapping* pada batas lebar konten (untuk horizontal) atau tinggi konten (untuk vertical).
7. **AnchorPane** adalah kelas yang memungkinkan developer untuk membuat *anchor* node pada layout atas, bawah, sisi kiri atau ditengah layout.

### 2.5.10 Transformasi 2-D dan 3-D

Setiap node pada JavaFX scene graph dapat ditransformasikan dalam koordinat x-y melalui kelas-kelas *javafx.scene.transform* berikut ini:[8]

1. **translate** - Memindahkan sebuah node dari satu posisi ke posisi lain bersama koordinat x,y,z yang relatif terhadap posisi awalnya.
2. **scale** - Meresize sebuah node untuk membesar atau mengecil sesuai koordinat x,y,z tergantung skala faktornya.
3. **rotate** - Merotasi sebuah node sesuai titik porosnya.
4. **affine** - Melakukan pemetaan linear dari koordinat 2-D / 3-D ke koordinat 2-D / 3-D lainnya dengan menjaga lurus dan paralel sifat garis tersebut. Kelas ini digunakan bersamaan dengan kelas lainnya dibanding penggunaan langsung.

### 2.5.11 Komponen JavaFX

Berikut ini merupakan kumpulan *package* yang ada dalam JavaFX.[9]

Tabel 2.33: Tabel komponen JavaFX

No	Package dan Deskripsi
1	<b>javafx.application</b> Menyediakan kelas-kelas dalam siklus aplikasi.
2	<b>javafx.event</b> Memberikan kerangka dasar untuk FX event, dari mulai pengiriman hingga handling.
3	<b>javafx.fxml</b> Berisi kelas untuk membuat hirarki objek dari markup.
4	<b>javafx.scene</b> Memberikan set basis kelas - kelas untuk JavaFX Scene Graph API .
5	<b>javafx.scene.control</b> JavaFX <i>User Interface Control</i> (kontrol UI atau kontrol saja) dimana node khusus dalam JavaFX Scenegraph yang dapat digunakan untuk banuak konteks aplikasi yang berbeda.
6	<b>javafx.scene.input</b> Menyediakan set kelas - kelas untuk mouse dan keyboard <i>input event handling</i> .
7	<b>javafx.scene.layout</b> Menyediakan kelas - kelas untuk mendukung UI layout.
8	<b>javafx.scene.text</b> Menyediakan set kelas - kelas untuk font dan teks node yang dapat di render.
9	<b>javafx.util</b> Berisi berbagai utilitas dan kelas pembantu.
10	<b>javafx.util.converter</b> <i>Package</i> ini untuk konversi String pada JavaFX.
11	<b>javafx.beans</b> <i>Package</i> ini berisi interface yang mendefinisikan bentuk umum dari <i>observability</i> .
12	<b>javafx.beans.binding</b> <i>Package</i> ini untuk menjelaskan karakter dari <i>Binding</i> .
13	<b>javafx.beans.value</b> <i>Package</i> ini berisi fundamental interface dari <i>observableValue</i> dan <i>WritableValue</i> dan semua sub interface di dalamnya.
14	<b>javafx.collections</b> <i>Package</i> ini berisi koleksi penting dari javaFX dan koleksi utilitas lainnya.

### 2.5.12 Property

Properti merupakan *interface* generik yang mendefinisikan metode umum untuk sifat yang dapat ditulis secara independen sesuai tipenya.[9]

### 2.5.13 javafx.collections

#### ObservableList

Dalam *package* javafx.collections terdapat sebuah *interface* yang bernama ObservableList. *Interface* ini merupakan sebuah *list* yang memungkinkan *listeners* melacak ketika terjadi perubahan pada *list* tersebut. . Berikut ini *method* yang ada pada *interface* ObservableList.[9]

Tabel 2.34: Tabel method ObservableList

No	Method dan Deskripsi
1	<b>addAll(E... elements)</b> Sebuah metode untuk menambah semua variabel dari sebuah elemen.
2	<b>addListener(ListChangeListener&lt;? super E&gt; listener)</b> Menambahkan <i>listeners</i> pada sebuah observableList.

### ListChangeListener

*Interface* ini menerima informasi perubahan dari sebuah ObservableList. Berikut ini *method* yang ada pada *interface* ListChangeListener.[9]

Tabel 2.35: Tabel method ListChangeListener

No	Method dan Deskripsi
1	<b>onChanged(ListChangeListener.Change&lt;? extends E&gt; c)</b> Metode ini dipanggil ketika perubahan telah dilakukan terhadap sebuah ObservableList.

### 2.5.14 javafx.scene.control

JavaFX UI *control* (kontrol UI atau Kontrol) merupakan Node khusus pada JavaFX *scenegraph* yang dapat digunakan berulang kali menyesuaikan kebutuhan rancangan aplikasi. Kontrol UI dapat disesuaikan visualnya oleh desainer atau pengembang aplikasi. *Interface* ini didesain untuk berkerja dalam *layout system*. Contoh *interface* ini adalah Button, Label, ListView, TableView, dan TextField.[9]

### TextField

TextField merupakan komponen *input* teks yang memungkinkan pengguna untuk memasukan satu baris teks terformat. Berikut ini *method* pada kelas TextField.[9]

Tabel 2.36: Tabel method TextField

No	Method dan Deskripsi
1	<b>getOnAction()</b> Metode ini menangani aksi terkait <i>text field</i> ini. Method ini mengembalikan nilai null bila tidak ada aksi yang diberikan.

### TableView

TableView dirancang untuk memvisualisasikan jumlah baris data. Jumlah baris data tersebut akan dipecah menjadi kolom-kolom dalam TabelView. Berikut *method* dalam kelas TabelView.[9]

Tabel 2.37: Tabel method TabelView

No	Method dan Deskripsi
1	<b>setItems(ObservableList&lt;S&gt; value)</b> Menetapkan nilai dari properti item pada tabel.
2	<b>getOnSort()</b> <i>Method</i> ini dipanggil ketika ada permintaan untuk <i>sorting</i> .

## TableColumn

Sebuah TableView terdiri dari sejumlah TableColumn. Setiap TableColumn pada tabel bertanggung jawab menampilkan dan mengedit isi kolom. Berikut *method* pada TableColumn.[9]

Tabel 2.38: Tabel method TableColumn

No	Method dan Deskripsi
1	<b>setCellValueFactory(Callback&lt;TableColumn.CellDataFeatures&lt;S,T&gt;,ObservableValue&lt;T&gt;» value)</b> Menetapkan nilai properti <i>CellFactory</i> yang dibutuhkan untuk menentukan bagaimana sel terisi pada satu TableColumn .

## BAB 3

### ANALISIS

Pada bab ini, akan dijelaskan mengenai analisis *input* dan fitur perangkat lunak, deskripsi pengguna perangkat lunak, pengembangan perangkat lunak, *use case* dari perangkat lunak serta diagram aktifitas dari perangkat lunak.

### 3.1 Analisis Input

#### 3.1.1 Analisis File Excel Jadwal Mengawas Ujian

Sub bab ini akan membahas analisis file excel yang dikeluarkan oleh TU(Tata Usaha). TU FTIS mengeluarkan jadwal setiap semester yang dibagikan kepada dosen FTIS. berikut ini contoh file excel yang dikeluarkan oleh TU.

	A	B	C	D	E	F	G	H	I	J	K	L
	No.	Hari, Tgl.	Jam	Sem.	PS	Nama Mata kuliah	9120	9121	9122	10316	10317	10323
1												
2												
3	1	Senin, 14 Mrt. 2016	08.00-10.00	2	MA	Kalkulus 2	Mariskha	Pascal		Ferry	Maria	
4	2	Senin, 14 Mrt. 2016	08.00-10.00	6p	IT	Keamanan Informasi, Reologi						
5	3	Senin, 14 Mrt. 2016	11.00-13.00	4	FI	Elektronika 2, Analisis Real	Vero	Gede	Owen	Elok	Iwan	Philips
6	4	Senin, 14 Mrt. 2016	11.00-13.00	4	IT	Manajemen Informasi dan Basis Data						
7	5	Senin, 14 Mrt. 2016	14.00-16.00	2	IT	Matematika Informatika	Maria	Haryanto	Janto			
8	6	Senin, 14 Mrt. 2016	14.00-16.00	2	IT	Matematika Informatika						
9	7	Selasa, 15 Mrt. 2016	08.00-10.00	2	FI	Fisika Dasar 2, Pemr. Apl. Bergerak				Reinard	Gede	Liem
10	8	Selasa, 15 Mrt. 2016	08.00-10.00	p	MA	Komputasi Keuangan						
11	9	Selasa, 15 Mrt. 2016	11.00-13.00	6	FI	Mekanika Kuantum, Pers. Dif. Biasa				Bagoes	Iwan	Sylvia
12	10	Selasa, 15 Mrt. 2016	11.00-13.00	6p	IT	Jaringan Syaraf Tiruan						
13	11	Selasa, 15 Mrt. 2016	14.00-16.00	8p	IT	Majemen Pengetahuan	Taufik	Janto				
14	12	Selasa, 15 Mrt. 2016	14.00-16.00	8p	IT	Majemen Pengetahuan						
15	13	Rabu, 16 Mrt. 2016	08.00-10.00	4	FI	Fisika Matematika 4, Proy. Inform., Logika Inf.	Risti	Pascal	Mariskha	Philips	Vania	Benny
16	14	Rabu, 16 Mrt. 2016	08.00-10.00	p	MA	Pengantar Matematika Asuransi						
17	15	Rabu, 16 Mrt. 2016	10.00-12.00	4	IT	Desain dan Analisis Algoritma	Shift 1: Luciana, Flaviana					
18	16	Rabu, 16 Mrt. 2016	12.00-14.00	4	IT	Desain dan Analisis Algoritma	Shift 2: Pascal, Luciana					
19	17	Rabu, 16 Mrt. 2016	11.00-13.00	2	MA	Aljabar Matriks	Taufik	Sylvia		Henri	Rusli	Haryanto
20	18	Rabu, 16 Mrt. 2016	11.00-13.00	8/8	FI/IT	Etika Profesi						
	19	Rabu, 16 Mrt. 2016	14.00-16.00	6	IT	Penulisan Ilmiah	Lab.: Anunz, Vania					

Gambar 3.1: Jadwal mengawas ujian FTIS

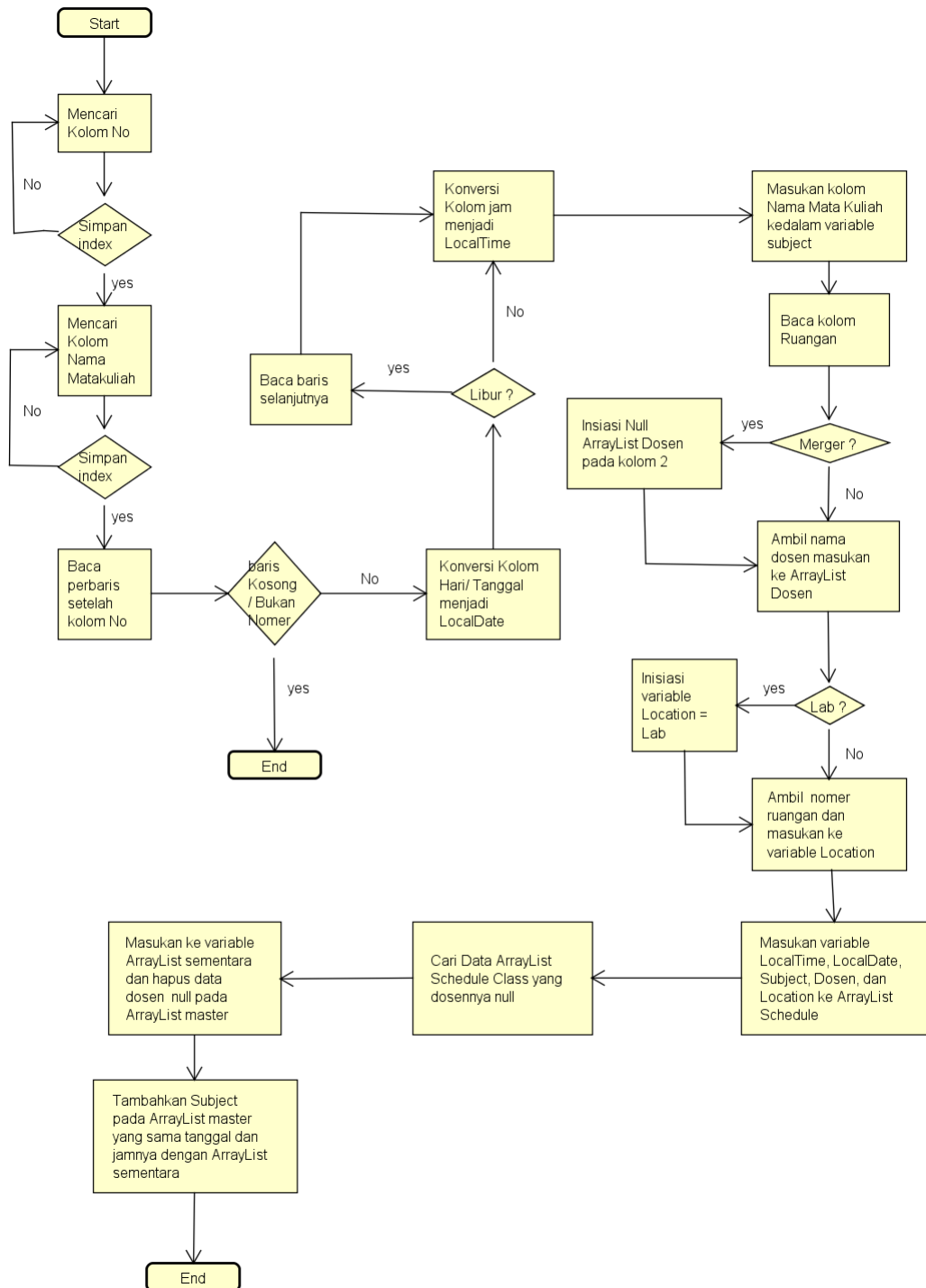
Berikut penjelasan kolom-kolom yang ada di gambar 3.1.

File excel jadwal mengawas ujian yang dipublikasikan Tata Usaha FTIS yang terdiri dari 12 kolom A sampai dengan L. TU meentukan jumlah ruangan yang dipakai berdasarkan jumlah peserta ujian dibagi dengan jumlah dosen pengawas ujian. Setiap dosen dapat mengawas lebih dari satu matakuliah pada satu ruangan. Tabel 3.1 menjelaskan rincian dari masing-masing kolom pada excel tersebut.

Tabel 3.1: Tabel penjelasan kolom pada excel mengawas ujian

No	Kolom dan Deskripsi
1	<b>No</b> Menyatakan nomer urut jadwal mengawas ujian.
2	<b>Hari, Tanggal</b> Kolom dalam bentuk String berisi hari dan tanggal. Terdapat singkatan yang diberikan TU dalam contoh ini Mrt. menunjukan bulan Maret
3	<b>Jam</b> Kolom ini bertipe String dan menerangkan pukul dilaksakannya mengawas ujian.
4	<b>Semester</b> Kolom ini bertipe String dan mengerangkan semester dari mata kuliah yang diujikan . Terdapat simbol p yang menerangkan matakuliah pilihan
5	<b>PS</b> Bertipe string berisi program studi yang terkait dengan ujian mata kuliah tersebut.
6	<b>Nama Mata Kuliah</b> Kolom bertipe String dan berisi tentang mata kuliah yang diujikan.
7	<b>Ruangan</b> Kolom dengan merge 6 kolom dan pada baris kedua terdapat 6 kolom ruangan ujian yaitu 9120, 9121, 9122, 10316, 10317, 10323. Masing kolom ruangan berisi nama pengawas ujian bertipe String. Ruangan tidak terpakai ditandai dengan garis-garis miring. Jika ada isi kolom ruangan yang dimerge sebanyak 6 kolom menandakan ruangan tersebut adalah laboratorium komputer FTIS.

Dari rincian tabel [3.1](#) berikut analisis *flowchart* cara perangkat lunak membaca file excel :



Gambar 3.2: Flowchart membaca excel

1. *Sheet* yang dibaca merupakan *Sheet* pertama.
2. Kolom **No.** dapat dijadikan acuan dalam membaca baris jadwal Excel pada perangkat lunak. Jika PL menemukan kolom No pada excel maka simpan baris dan kolomnya pada variabel tertentu untuk menandakan bahwa baris selanjutnya merupakan data yang dibutuhkan oleh perangkat lunak. Selanjutnya nomer pada kolom No. juga dapat dijadikan penanda dalam perangkat lunak menentukan banyak data yang dibaca, jika baris selanjutnya dari kolom No. merupakan angka maka dipastikan baris tersebut memuat data jadwal mengawas.
3. Kolom **Hari, Tgl.** memuat tanggal dan hari ujian menggunakan koma (,) sebagai pemisah

hari dan tanggal dan titik (.) sebagai penanda singkatan bulan. Untuk mendapatkan tanggal yang sesuai dengan format *LocalDate* maka perangkat lunak harus melakukan *parsing* memisahkan hari dengan tanggal, kemudian mengkonversi bulan menjadi sebuah angka sehingga sesuai dengan format *LocalDate*, lalu disimpan pada sebuah variabel.

4. Kolom **Jam** menggunakan *hyphen(-)* sebagai pemisah antara jam dimulainya ujian dan waktu ujian berakhir. perangkat lunak dapat melakukan *parsing* untuk memisahkan waktu tersebut menjadi dua variabel, lalu dikonversi sesuai dengan ketentuan *LocalTime*.
5. Kolom **Nama Mata kuliah** memuat nama mata kuliah yang diujikan, karena satu dosen dapat mengawas salah satu dari 2 matakuliah sehingga pada kolom ruangan yang memuat dua matakuliah yang ditandai dengan *merger* dua baris yang berisi nama pengawas ujian.
6. kolom **Ruangan** pada kolom ini terdapat 6 ruangan yang masing-masing kolom dan baris akan disimpan pada variable untuk dicocokkan nanti pada saat membaca file excel satu per satu untuk menentukan lokasi ujian tersebut berlangsung.
7. Jika perangkat lunak menemukan kata *LIBUR* atau menemukan mata kuliah yang tidak ada dosen pengawasnya maka baris tersebut akan dilewati menuju baris selanjutnya.
8. Jika perangkat lunak menemukan kata *Shift* atau *Lab* maka otomatis perangkat lunak akan menginisiasi tempat berlangsungnya ujian adalah Laboratorium Komputer.
9. Semua variabel *LocalDate*, *LocalTime*, *subject*, dosen, dan *location* yang telah terisi kemudian dimasukan ke dalam sebuah *ArrayList schedule*.
10. *ArrayList schedule* kemudian diperiksa kembali sehingga jika pada data excel dosen memiliki 2 jadwal mengawas, maka pada *ArrayList schedule* satu dosen tersebut memiliki 2 *subject*.
11. Poses ini menghasilkan *output* berupa sebuah *ArrayList schedule* yang memuat semua data file excel jadwal mengawas ujian yang telah dikonversi dan distandarisasi.

Selanjutnya adalah analisis berupa wawancara dengan dosen mengenai jadwal mengawas ujian. Hasilnya dosen FTIS tidak memerlukan kolom semseter dan program studi pada file excel jadwal mengawas. Hal ini disebabkan karena dosen FTIS dapat saling mengawas ujian antar program studi selama program studi tersebut masih dalam lingkup FTIS. Dari analisis tersebut maka kolom yang butuhkan untuk ditampilkan pada PL, yaitu :

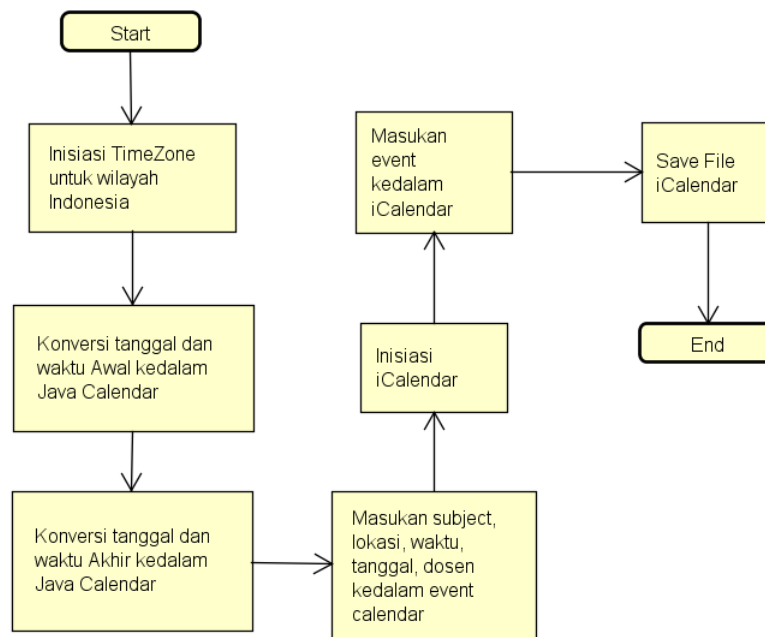
Tabel 3.2: Tabel analisa kolom pada excel mengawas ujian

No	Kolom dan Deskripsi
1	<b>Hari, Tanggal</b> Kolom ini bertipe String dan terdapat singkatan seperti Mrt, maka akan dibuatkan fungsi pada saat implementasi agar seragam dan sesuai dengan format tgl dan waktu pada Java.
2	<b>Jam</b> Kolom ini bertipe String, maka dibutuhkan konversi String kedalam fungsi jam pada saat implementasi.
3	<b>Nama Mata Kuliah</b> Kolom ini dapat menerangkan deskripsi mata kuliah pada PL.
4	<b>Dosen</b> Kolom ini merupakan isi dari tabel kelas pada excel, pada PL akan ditampilkan sebagai kolom tersendiri menerangkan nama pengawas ujian.
7	<b>Ruangan</b> Kolom ini akan berisi ruangan ujian, mata kuliah, waktu dan tanggal.



### 3.1.2 Flowchart Konversi ArrayList Jadwal ke iCalendar

Setelah dibaca file excel akan berubah menjadi ArrayList *schedule*, kemudian data list tersebut yang menjadi *input* untuk dikonversi menjadi iCalendar. Berikut alur dari konversi tersebut :



Gambar 3.3: *Flowchart* alur konversi excel ke iCalendar

1. Menginisiasi waktu indonesia.
2. Mengkonversi waktu awal atau waktu masuk ujian kedalam format Java Calendar.
3. Mengkonversi waktu akhir atau waktu selesai ujian kedalam format Java Calendar.
4. Masukan mata kuliah/*subject*, lokasi, jam, tanggal dari ArrayList jadwal kedalam variabel event Kalender.
5. Inisiasi iCalendar.
6. Masukan semua komponen event pada iCalendar.
7. *Output* icalendar dapat disimpan pada alamat folder yang diinginkan.

### 3.1.3 Analisis Kebutuhan Perangkat Lunak

Perangkat Lunak ini akan memiliki fitur sebagai berikut :

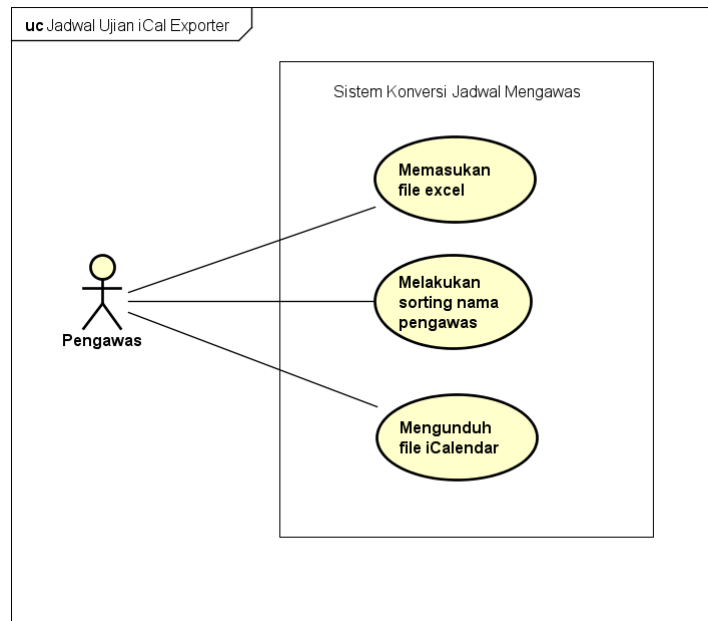
1. Perangkat lunak ini dapat menerima dan membaca *input* file excel jadwal mengawas ujian yang dikeluarkan oleh TU FTIS.
2. Perangkat lunak ini dapat mengubah file excel menjadi iCalendar.
3. File iCalendar dapat diunduh oleh pengguna.
4. Pengguna dapat melakukan *sort* terhadap nama pengawas ujian.

#### Deskripsi Pengguna

Pengguna perangkat lunak ini merupakan dosen yang aktif mengajar di FTIS.

## 3.2 Use Case Perangkat Lunak

Berikut diagram use case berserta skenario yang tertera pada gambar 3.4



Gambar 3.4: Diagram use case perangkat lunak konversi jadwal mengawas ujian

### 1. Skenario Memasukan input file excel

Deskripsi : Kegiatan memasukan input file excel.

Aktor : Pengawas

Prakondisi : -

Skenario :

- Pengawas memasukan file excel mengawas ujian.
- PL membaca file excel lalu menkonversi ke ArrayList jadwal.
- PL menampilkan jadwal dalam bentuk TableView.

### 2. Skenario Melakukan Sorting nama

Deskripsi : Kegiatan mensorting jadwal mengawas.

Aktor : pengawas

Prakondisi : -

Skenario :

- Pengawas menuliskan nama pengawas pada kolom filter di PL.
- PL mengambil data nama pengawas dan mencocokkannya dengan ArrayList jadwal.
- PL mengabil data nama pengawas yang sesuai dengan masukan Pengawas, lalu memisahkannya ke variable ArrayList sementara.
- PL mengatur kembali urutan tabel jadwal sesuai nama pengawas yang dimasukan Pengawas.

### 3. Skenario Mengunduh File iCal

Deskripsi : Kegiatan Mengunduh file iCal.

Aktor : Dosen

Prakondisi : -

Skenario :

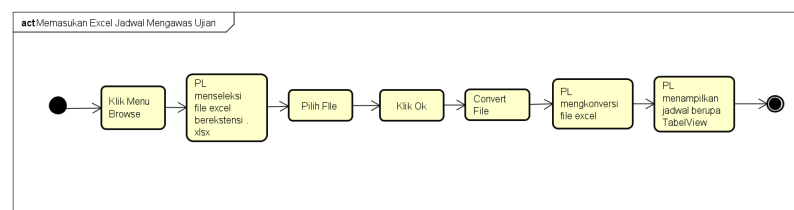
- Pengawas memilih jadwal yang akan dikonversi.
- Pengawas mengklik tombol Convert to iCal pada PL.
- PL mengambil item yang diseleksi oleh Pengawas.
- PL mengkonversi jadwal yang dipilih Pengawas.
- Pengawas menentukan alamat destinasi file iCalendar dan memasukan nama file iCalendar.
- PL akan menyimpan file iCalendar ditempat yang dipilih oleh Pengawas.

### 3.3 Diagram Aktifitas

Pada subbab ini akan dibahas mengenai prosedur setiap aktifitas dari perangkat lunak.

#### 3.3.1 Memasukan Excel Jadwal Mengawas Ujian

Tahap ini merupakan tahap awal proses file input dimasukkan kedalam perangkat lunak dimana file input yang dimaksud merupakan excel jadwal yang dikeluarkan TU FTIS. Berikut step-step untuk memasukan excel kedalam perangkat lunak.

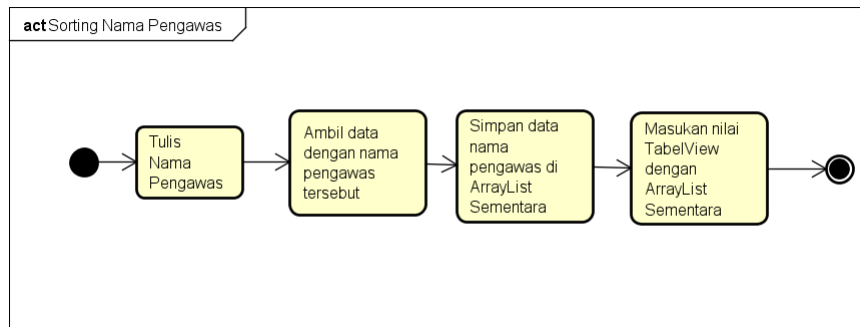


Gambar 3.5: Prosedur memasukan file excel jadwal mengawas ujian

1. Pengguna mengklik menu Browse yang ada di perangkat lunak.
2. Perangkat lunak menseleksi file yang berekstensi .xlsx yang dapat dipilih oleh pengguna.
3. Pengguna memilih file excel yang akan dimasukkan.
4. Pengguna mengklik tombol oke pada window.
5. Pengguna mengklik tombol Convert untuk mengkonversi file excel.
6. Perangkat lunak mengkonversi file excel menjadi ArrayList jadwal.
7. Perangkat Lunak menampilkan TableView jadwal yang telah dikonversi.

#### 3.3.2 Sorting Nama Pengawas

Tahap ini menjelaskan bagaimana pengguna dapat memanfaatkan fitur dari perangkat lunak dengan mensorting nama dosen, dengan begitu jadwal pengawas yang dicari dapat diunduh dengan mudah. Berikut step-step untuk mensorting nama pengawas.

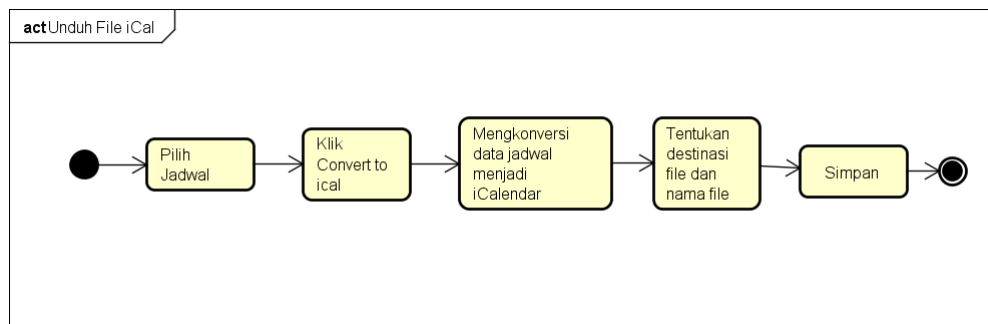


Gambar 3.6: Prodsedur sorting nama pengawas

1. Pengguna menuliskan nama pengawas yang dicari.
2. Perangkat lunak mencocokkan nama pengawas dengan ArrayList jadwal.
3. Data dengan nama pengawas yang sama akan disimpan pada ArrayList sementara.
4. Atur kembali urutan tabel menggunakan ArrayList sementara lalu tampilkan ke pengguna.

### 3.3.3 Unduh File iCal

Tahap ini merupakan tahap terakhir, file excel yang telah dimasukkan lalu dikonversi menjadi iCal selanjutnya pengguna tinggal memilih jadwal mana yang akan diunduh. File iCal yang telah dikonversi tersebut dapat diintegrasikan dengan aplikasi iCalendar seperti Google Calendar, Microsoft Outlook, dan Apple Calendar. Berikut step-step untuk mengunduh file iCal.



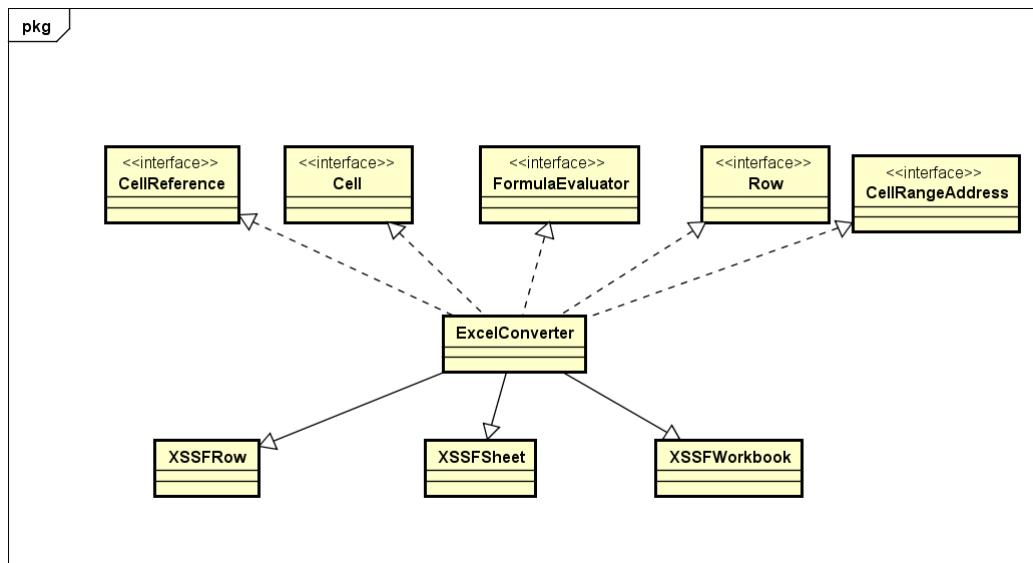
Gambar 3.7: Prosedur Mengunduh File iCal

1. Pengguna memilih jadwal yang akan dikonversi pada tabel.
2. Pengguna mengklik tombol Convert to iCal pada perangkat lunak.
3. Perangkat lunak mengkonversi jadwal yang dipilih oleh pengguna menjadi iCalendar.
4. Pengguna menentukan destinasi folder penyimpanan dan menuliskan nama file iCalendar.
5. Pengguna mengklik tombol simpan pada perangkat lunak.

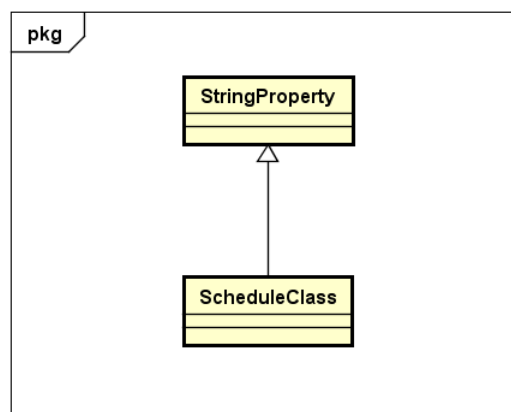
## 3.4 Pemodelan Kelas

Setelah file excel mengawas ujian tersebut dijabarkan dan dianalisis, pada subbab ini akan di jelaskan mengenai pembagian fungsi kelas dalam rancangan perangkat lunak.

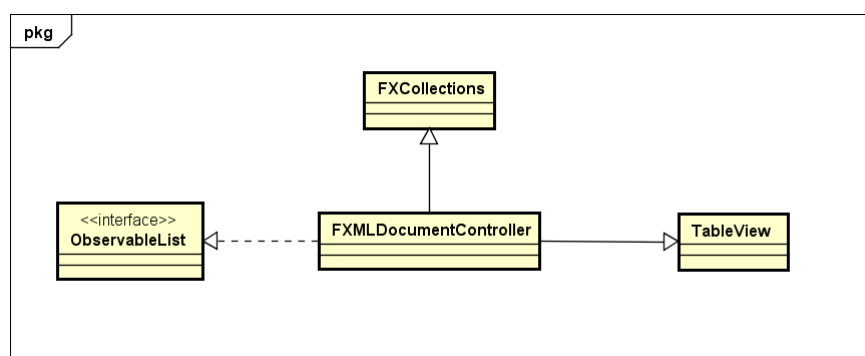
## Pemodelan Rancangan Kelas



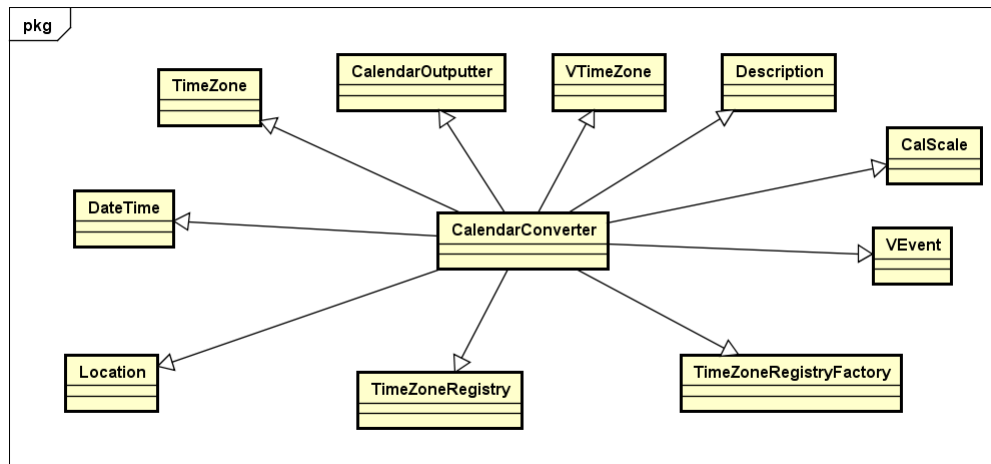
Gambar 3.8: Gambar Pemodelan ExcelConverter



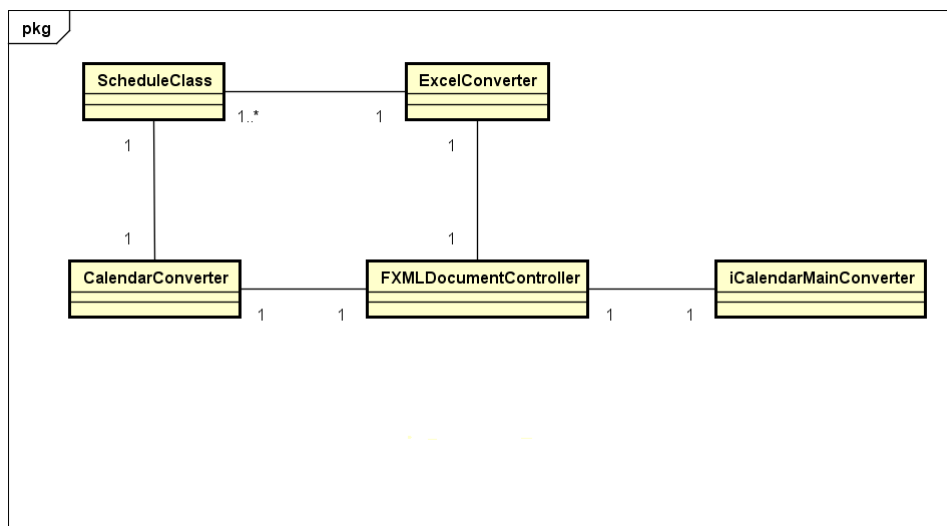
Gambar 3.9: Gambar Pemodelan ScheduleClass



Gambar 3.10: Gambar Pemodelan FXMLElementController



Gambar 3.11: Gambar Pemodelan CalendarConverter



Gambar 3.12: Gambar Pemodelan Kelas

Berikut penjelasan fungsi dari kelas dari gambar 3.12 :

1. **ScheduleClass**

Kelas ini berfungsi menampung jadwal pengawas yang telah dikonversi oleh kelas ExcelConverter.

2. **ExcelConverter**

Kelas ini bertugas membaca file excel jadwal mengawas ujian sehingga dapat ditampilkan oleh perangkat lunak.

3. **FXMLDocumentController**

Kelas ini mempunyai peran untuk mendapatkan file *input* yang dimasukkan oleh pengguna, memberi perintah kepada kelas ExcelConverter untuk membaca *input*, menampilkannya kembali ke perangkat lunak dan memberikan perintah kepada CalendarConverter untuk mengkonversikannya dalam iCal.

4. **CalendarConverter**

Kelas ini berfungsi mengkonversi file yang telah dibaca kedalam format .ics atau iCalendar.

---

### 5. **iCalendarMainConverter**

kelas ini berfungsi sebagai *main* pada perangkat lunak dimana kelas ini mengeksekusi dan menghubungkan seluruh elemen kelas pada perangkat lunak ini.





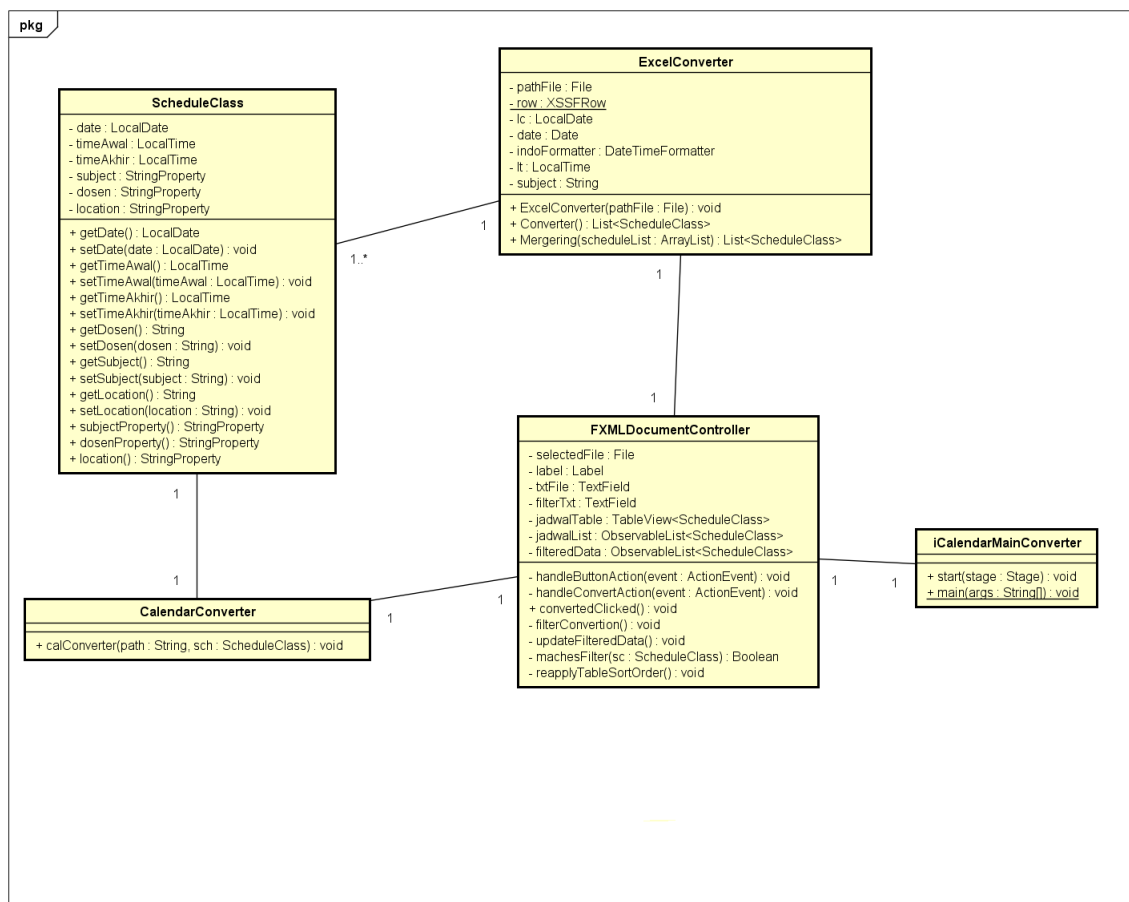
## BAB 4

### PERANCANGAN

Berdasarkan analisa dari bab 3, pada bab ini akan dibahas mengenai perancangan diagram kelas, dan perancangan antarmuka dari program.

#### 4.1 Perancangan Diagram Kelas

Berdasarkan hasil analisis dari bab 3, telah dijelaskan pemodelan struktur kelas yang akan digunakan pada program, selanjutnya pada subbab ini merupakan terjemahan dari pemodelan kelas dalam bentuk rancangan diagram kelas pada gambar 4.1.



Gambar 4.1: Gambar Kelas Diagram

Berikut ini rincian kelas pada diagram kelas yang tercantum dalam tabel-tabel dibawah ini :

Tabel 4.1: Tabel Kelas *ScheduleClass*

Atribut		
Nama atribut	Tipe Data	Fungsi
<b>date</b>	LocalDate	Atribut tanggal
<b>timeAwal</b>	LocalTime	Atribut jam ujian dimulai
<b>timeAkhir</b>	LocalTime	Atribut jam ujian berakhir
<b>subject</b>	StringProperty	Atribut mata kuliah
<b>dosen</b>	StringProperty	Atribut nama dosen
<b>location</b>	StringProperty	Atribut lokasi ujian
Method		
Nama Method		Fungsi
<b>getDate()</b>		Mendapatkan tanggal
<b>setDate(date: LocalDate)</b>		Set tanggal
<b>getTimeAwal()</b>		Mendapatkan jam awal ujian
<b>setTimeAwal(timeAwal: LocalTime)</b>		Set jam awal ujian
<b>getTimeAkhir()</b>		Mendapatkan jam akhir ujian
<b>setTimeAkhir(timeAkhir: LocalTime)</b>		Set jam akhir ujian
<b>getDosen()</b>		Mendapatkan nama dosen
<b>setDosen(dosen: String)</b>		Set nama dosen
<b>getSubject()</b>		Mendapatkan nama mata kuliah
<b>setSubject(subject: String)</b>		Set mata kuliah
<b>getLocation()</b>		Mendapatkan lokasi ujian
<b>setLocation(location: String)</b>		Set lokasi ujian
<b>subjectProperty()</b>		Mendapatkan properti mata kuliah
<b>dosenProperty()</b>		Mendapatkan properti dosen
<b>location()</b>		Mendapatkan properti lokasi

Tabel 4.2: Tabel Kelas *ExcelConverter*

Atribut		
Nama atribut	Tipe Data	Fungsi
<b>pathFile</b>	File	Atribut path file excel mengawas
<b>row</b>	XSSFRow	Atribut baris dari Excel
<b>lc</b>	LocalDate	Atribut tanggal ujian
<b>indoFormater</b>	DateTimeFormatter	Atribut konversi ke zona waktu jakarta
<b>lt</b>	LocalTime	Atribut jam ujian
<b>subject</b>	String	Atribut matakuliah
Method		
Nama Method		Fungsi
<b>ExcelConverter(path: File)</b>		Konstruktor untuk mendapatkan path file dari excel mengawas ujian
<b>Converter()</b>		Konversi excel menjadi list scheduleClass
<b>Mergering(scheduleList: ArrayList)</b>		Mengabungkan duplikat entri mengawas dosen

Tabel 4.3: Tabel Kelas *CalendarConverter*

Method	
Nama Method	Fungsi
<b>calConverter(path: String, sch: ScheduleClass)</b>	Mengkonversi schedule-Class yang dipilih kedalam iCal dan menyimpannya pada path yang ditentukan

Tabel 4.4: Tabel Kelas *FXMLDocumentController*

Atribut		
Nama atribut	Tipe Data	Fungsi
<b>selectedFile</b>	File	Atribut file yang dipilih pengguna
<b>label</b>	Label	Atribut label
<b>txtFile</b>	TextField	Atribut menampilkan path file yang dipilih
<b>filterTxt</b>	TextField	Atribut untuk menampilkan filter teks
<b>jadwalTable</b>	TableView<ScheduleClass>	Atribut menampilkan tabel jadwal
<b>jadwalList</b>	ObservableList<ScheduleClass>	Atribut untuk menyimpan jadwal
<b>filteredData</b>	ObservableList<ScheduleClass>	Atribut untuk menyimpan data yang telah di filter
Method		
Nama Method		Fungsi
<b>handleButtonAction(event: ActionEvent)</b>		Method untuk melakukan browse dan mendapatkan file excel
<b>handleConvertAction(event: ActionEvent)</b>		Method untuk membaca file excel
<b>convertedClicked()</b>		Method untuk konversi <i>selected item</i> menjadi iCal
<b>filterConversion()</b>		Method untuk menerima masukan filter dari pengguna
<b>updateFilteredData()</b>		Menginisiasi list <i>filteredData</i>
<b>matchesFilter(sc: ScheduleClass)</b>		Mencocokkan nama dosen sesuai yang diinginkan pengguna
<b>reapplyTableSortOrder()</b>		Mengatur urutan tabel setelah di filter

Tabel 4.5: Tabel Kelas *iCalendarMainConverter*

Method	
Nama Method	Fungsi
<b>start(stage: Stage)</b>	Menampilkan window
<b>main(args: String[])</b>	Mengeksekusi program

## 4.2 Rancangan *Method-Method* Utama

Berikut ini adalah rancangan *method* utama program jadwal mengawas ujian yang berperan penting dalam perangkat lunak:

### 1. readExcel() - ExcelConverter

- Input : -  
 Output : List<ScheduleClass>  
 Deskripsi : Method ini membaca excel yang di *input* oleh pengguna dan mengkonversikannya kedalam bentuk list.  
 Algoritma :
- Ambil path file yang telah di *input* oleh pengguna.
  - Cari kolom No. pada file excel dan jadikan acuan bahwa program akan membaca setelah dari index kolom tersebut.
  - Baca baris per baris namun cek terlebih dahulu apakah di kolom No. baris tersebut masih berupa nomer, apabila tidak maka berhenti membaca karena baris yang berisi jadwal sudah terbaca semua.
  - Cek apakah baris mengandung kata *LIBUR* bila ya maka lewati saja.
  - Pisahkan hari dan tanggal lalu konversi menjadi LocalDate.
  - Pisahkan jam menjadi jamAwal dan jamAkhir, lalu konversi menjadi LocalTime.
  - jika menemukan kata *Shift* atau *Lab* maka lokasi ujian adalah Lab.
  - Masukan semua kedalam sebuah ArrayList<ScheduleClass>.

### 2. mergingList() - ExcelConverter

Input : List<ScheduleClass>  
 Output : List<ScheduleClass>  
 Deskripsi : Method ini menghapus *entry* duplikat dari dosen yang mempunyai dua jadwal mengawas pada hari yang sama.  
 Algoritma :
 

- (a) Cari subject/mata kuliah yang tidak memiliki dosen pada ArrayList yang telah di proses oleh *method* Convert() karena bila program membaca kolom yang di-*merger* maka hanya kolom pertama saja yang dibaca sehingga kolom keduanya kosong.
- (b) Masukkan baris yang tidak memiliki dosen kedalam ArrayList baru.
- (c) Hapus baris yang tidak memiliki dosen pada ArrayList master.
- (d) Cocokkan waktu dan tanggal ujian ArrayList master dengan temp, bila sama maka tambahkan matkuliah/subject pada ArrayList master.

### 3. calConverter() - CalendarConverter

Input : sch: ScheduleClass , path: String  
 Output : void  
 Deskripsi : Method ini mengkonversi ScheduleClass menjadi iCal  
 Algoritma :
 

- (a) Inisiasi variable zona waktu Indonesia.
- (b) Konversi tanggal, bulan, dan tahun kedalam GregorianCalender.
- (c) Masukkan event berdasarkan subject/mata kuliah , lokasi, dan dosen yang mengawas.
- (d) Inisiasi kalender dan masukan variable tanggal dan event yang telah dibuat sebelumnya kedalam variable kalender tersebut.
- (e) Simpan pada path yang telah di pilih oleh pengguna.

### 4. filterConversion() - FXMLDocumentController

Input : void  
 Output : void  
 Deskripsi : Method ini menjalankan filter data dosen sesuai *input* pengguna  
 Algoritma :
 

- (a) Inisiasi tabel dengan list yang sudah di filter.
- (b) Masukkan nilai yang sama kedalam list filter jika dosen yang dicari sesuai dengan *input* pengguna.
- (c) Atur kembali urutan tabel pada perangkat lunak.

### 4.3 Perancangan Antarmuka

Setelah melalui serangkaian analisis dan perancangan diagram kelas pada sub bab ini akan dijelaskan mengenai gambaran bentuk program mengawas ujian tersebut.

1. Halaman awal program

Ini adalah tampilan awal pada saat pengguna pertama kali menjalankan program.

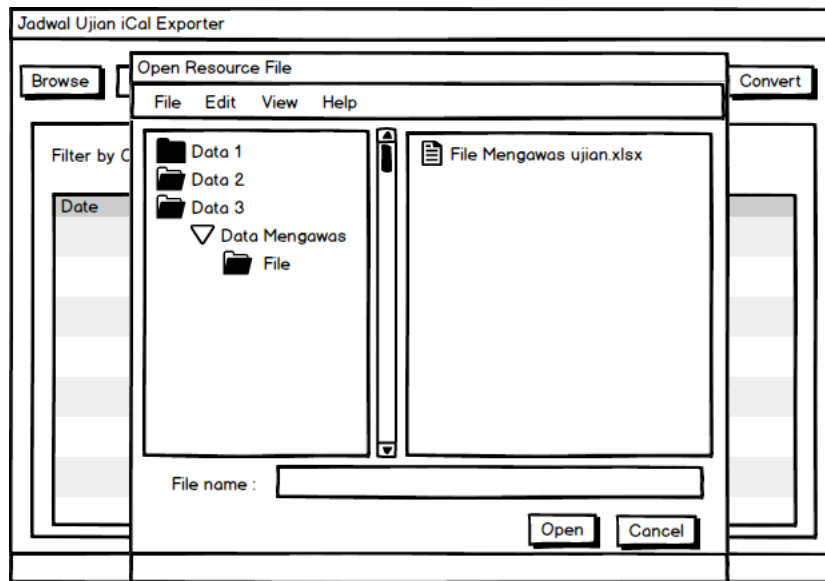
Gambar 4.2: Tampilan awal Program

Pada gambar 4.2 terdapat beberapa *button* dan *textbox* yang memiliki fungsi sebagai berikut.

- *Browse* : berfungsi untuk membuka *pop-up window* sebagai sarana pengguna memilih file excel yang akan dimasukkan.
- *Textbox path* : alamat file yang telah dipilih oleh pengguna akan dicatat pada *textbox* ini.
- *Convert* : tombol ini berfungsi mengkonversi kolom pada tabel excel jadwal mengawas ujian menjadi *ArrayList* jadwal.
- *Textbox filter* : merupakan fitur untuk memfilter jadwal mengawas berdasarkan nama dosen yang sesuai dengan *input* pengguna.
- *TableView* : jadwal yang telah dibaca pada excel selanjutnya akan ditampilkan pada tabel ini. tabel ini terdiri dari kolom tanggal, waktu, matakuliah, dosen, lokasi, dan *download* untuk mengunduh file iCal.

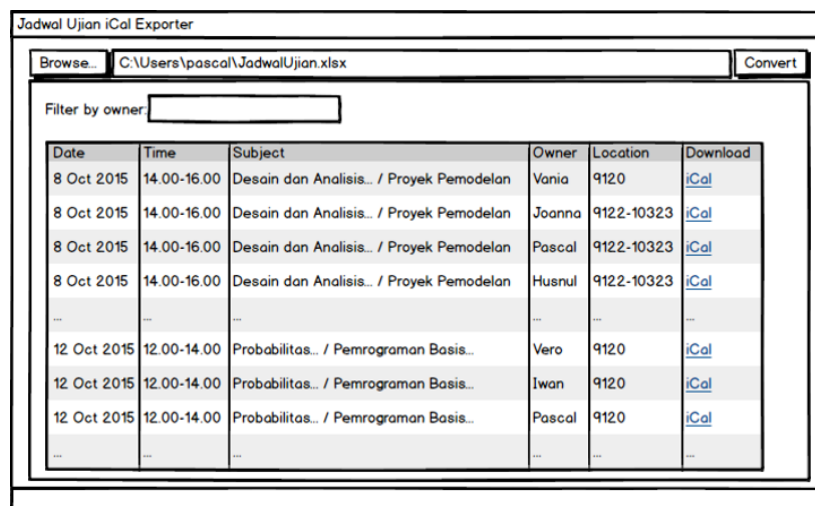
2. Halaman untuk melakukan *Browse* file excel

Halaman ini merupakan halaman dimana pengguna melakukan pemilihan *input* file excel jadwal mengawas. Halaman *browser* menyesuaikan tipe sistem operasi yang dipakai.

Gambar 4.3: Tampilan *Browse* file excel

### 3. Halaman setelah excel dibaca

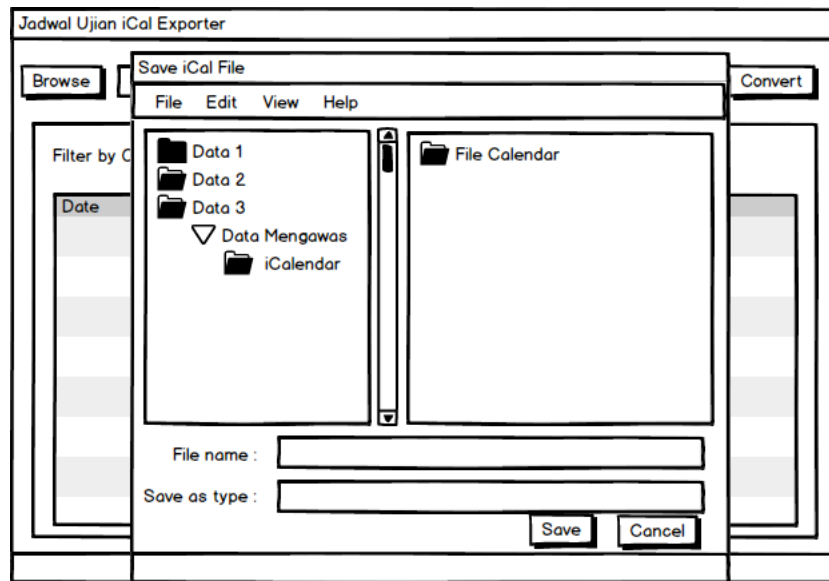
Halaman ini menunjukan ketika file excel telah sukses dibaca dan ditampilkan pada *tableview*.



Gambar 4.4: Tampilan setelah excel dibaca

### 4. Halaman untuk menyimpan iCal

Halaman ini dimana pengguna telah memilih salah satu jadwal dan akan menyimpannya dalam bentuk iCal. Halaman *save* menyesuaikan sistem operasi yang dipakai.



Gambar 4.5: Tampilan untuk menyimpan iCal



## BAB 5

### IMPLEMENTASI DAN PENGUJIAN

Pada bagian ini merupakan rincian atau penjelasan lanjut mengenai lingkungan implementasi perangkat keras maupun perangkat lunak, serta implementasi program iCalendar Converter dan tampilan antarmukanya. Terakhir akan dibahas mengenai pengujian pada perangkat lunak ini.

#### 5.1 Implementasi

Pada bab ini akan dijabarkan mengenai lingkungan pengembangan perangkat lunak disertai dengan pengujian.

##### 5.1.1 Lingkungan Implementasi

Dalam mengimplementasikan program terdapat dua lingkungan pendukung, yaitu lingkungan perangkat keras dan lingkungan perangkat lunak.

##### Lingkungan Perangkat Keras

Dalam mengembangkan perangkat ini, digunakan spesifikasi perangkat keras sebagai berikut:

- *Processor* : Intel Core i7 2.4 Ghz
- *Memory* : 8 GB
- *Hardisk* : 640 GB
- *VGA* : Nvidia GeForce 540M
- *keyboard* dan *mouse standard*

##### Lingkungan Perangkat Lunak

Untuk pengembangan perangkat lunak iCalendar Converter, digunakan spesifikasi sebagai berikut:

- IDE : Netbeans 8.1
- JDK : 1.8 [10]
- JRE : Java Runtime Enviroment 8 [10]
- Serta library pihak ketiga seperti JavaFX, Apache POI, dan iCal4j
- Editor antarmuka menggunakan SceneBuilder

### 5.1.2 Implementasi Program

Subbab ini menjelaskan tahap dimana program akan dibuat dan dikembangkan dari hasil analisis dan perancangan kelas-kelas maupun *method* yang digunakan. Kode program lengkap dapat dilihat pada Lampiran A. Berikut ini merupakan penjelasan kode program dari perangkat lunak iCalendarConverter :

1. Kode Program untuk menyimpan jadwal

ScheduleClass merupakan kelas model yang ditujukan untuk menyimpan informasi jadwal yang telah dibaca.

Baris (115-127) kelas ScheduleClass pada lampiran A.1 menjelaskan tentang penggunaan StringProperty, StringProperty memungkinkan untuk memberitahu jika ada perubahan pada *variable* tersebut. *Property* membantu untuk menjaga tampilan agar sinkron dengan data. Pada ScheduleClass *variable* yang menggunakan StringProperty adalah dosen, subject, dan location.

2. Kode program untuk membaca Excel

ExcelConverter merupakan kelas yang dikhususkan untuk membaca excel dan mengeluarkan *output* berupa ArrayList dari kelas model ScheduleClass.

Berikut ini merupakan urutan dari algoritma yang digunakan pada Excel Converter :

```

outerloop :
for (int j = 0; j < sheet.getLastRowNum(); j++) {
    row = sheet.getRow(j);
    for (int f = 0; f < row.getLastCellNum(); f++) {
        Cell cell = row.getCell(f);
        if (cell.getCellType() == Cell.CELL_TYPE_STRING
            && cell.getStringCellValue().equalsIgnoreCase("No. ")) {
            rowNoIdx = j;
            colNoIdx = cell.getColumnIndex();
        }
        else if (cell.getCellType() == Cell.CELL_TYPE_STRING
            && cell.getStringCellValue().equalsIgnoreCase("Nama Mata Kuliah"))
        {
            colMatkulIdx = cell.getColumnIndex();
            break outerloop;
        }
    }
}

```

Gambar 5.1: Implementasi kode program untuk mencari kolom No dan matakuliah

- (a) Gambar 5.1 menjelaskan bagaimana program mencari kolom No. dan Nama Mata Kuliah pada excel. Dua kolom tersebut merupakan acuan data jadwal yang akan dibaca oleh program. Setelah diketahui dimana kolom No. dan Nama Kuliah berada, maka nomer baris dan kolomnya akan dimasukkan kedalam *variable* sebagai acuan membaca program dimulai pada baris itu. Pemilihan kolom No. sebagai acuan dikarenakan isi kolom No. menandakan berapa banyak data jadwal yang ada, sehingga bila isi dari kolom No. bukan angka, maka program akan berhenti membaca. Selanjutnya, pemilihan Nama Mata Kuliah sebagai acuan selain karena data pada kolom itu akan dimasukkan ke kelas model, pun juga karena setelah kolom tersebut terdapat kolom ruang kuliah yang akan dimasukkan kedalam *variable* lokasi pada program. Selain itu, nomer kolom ruangan dapat menjadi acuan lokasi dosen mengawas.

```

for (int i = 0; i < sheet.getLastRowNum(); i++) {
    outerloop :
        for (int j = 0; j < row.getLastCellNum(); j++) {

```

Gambar 5.2: Implementasi kode program ExcelConverter

- (b) Gambar 5.1 menjelaskan bahwa *i* sebagai acuan program membaca baris, sedangkan *j* sebagai acuan program membaca kolom.

```

if (row == null)
{
    i = sheet.getLastRowNum();
    break outerloop2;
}
Cell cell = row.getCell(j);
FormulaEvaluator evaluator = wb.getCreationHelper().createFormulaEvaluator();
if (cell.getColumnIndex() == colNoIdx && i > rowNoIdx + 3
    && cell.getCellType() != Cell.CELL_TYPE_BLANK
    && evaluator.evaluate(cell).getCellType() != Cell.CELL_TYPE_NUMERIC) {
    i = sheet.getLastRowNum();
    break outerloop2;
}
if (cell.getColumnIndex() == colNoIdx && i > rowNoIdx+3
    && cell.getCellType() == Cell.CELL_TYPE_BLANK )
{
    i = i + 1;
    break outerloop;
}

if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 1)) {
    if (cell.getCellType() == Cell.CELL_TYPE_BLANK)
    {
        i = i+1;
        break outerloop;
    }
}

```

Gambar 5.3: Implementasi kode program ExcelConverter bagian 2

- (c) Gambar 5.3 menjelaskan bahwa bila baris ke *i* program membaca dan isinya kosong maka berhenti membaca.
- (d) Gambar 5.3 menjelaskan bila isi pada kolom No. bukanlah angka dan *blank* maka program berhenti membaca.

```

if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 1)) {
    if (cell.getCellType() == Cell.CELL_TYPE_BLANK)
    {
        i = i+1;
        break outerloop;
    }
}
String delims = "[, . ]";
String[] summary = cell.getStringCellValue().split(delims);
for (int l = 0; l < summary.length; l++) {
    if (summary[l].equalsIgnoreCase("Mrt")) {
        summary[l] = "3";
    }
    if (summary[l].equalsIgnoreCase("Okt")) {
        summary[l] = "10";
    }
    if (summary[l].equalsIgnoreCase("`16")) {
        summary[l] = "2016";
    }
}

lc = LocalDate.of(Integer.parseInt(summary[5]), Integer.parseInt(summary[3]), Integer.parseInt(summary[2]));

```

Gambar 5.4: Implementasi kode program ExcelConverter bagian 3

- (e) Gambar 5.4 menjelaskan bila isi kolom No. adalah kosong maka lewati barisnya dan baca baris selanjutnya.
- (f) Gambar 5.4 menjelaskan bahwa bila kolom tersebut adalah tanggal dan jika isinya kosong, maka lewati barisnya. Jika isinya tidak kosong, maka pisahkan isinya menurut

tanda "-" dan tanda "." . Lalu, jika ada singkatan Mrt ganti menjadi 3. Jika ada singkatan Okt ganti menjadi 10 dan jika ada singkatan 16 maka ganti menjadi 2016. Sehingga format tanggal menjadi 2016-03-01 sebagai contoh. Setelah itu masukan ke *variable* beritipe LocalDate.

```
if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 2)) {
    if (cell.getStringCellValue().equalsIgnoreCase("LIBUR"))
    {
        i = i+1;
        break outerloop;
    }
    else
    {
        if (cell.getStringCellValue().equalsIgnoreCase("Shift 1")
            || cell.getStringCellValue().equalsIgnoreCase("Shift 2"))
        {
            CellReference cr = new CellReference(cell.getRowIndex()+1, cell.getColumnIndex());
            Row row2 = sheet.getRow(cr.getRow());
            Cell c = row2.getCell(cr.getCol());
            String delimsJam = "[-]";
            String[] arrJam = c.getStringCellValue().split(delimsJam);
            for (int k = 0; k < arrJam.length; k++) {
                arrJam[k] = arrJam[k].replace('.', ':');
            }
            lt = LocalTime.parse(arrJam[0]);
        }
    }
}
```

Gambar 5.5: Implementasi kode program ExcelConverter bagian 4

```
else
{
    String delimsJam = "[-]";
    String[] arrJam = cell.getStringCellValue().split(delimsJam);
    for (int k = 0; k < arrJam.length; k++) {
        arrJam[k] = arrJam[k].replace('.', ':');
    }
    lt = LocalTime.parse(arrJam[0]);
}
```

Gambar 5.6: Implementasi kode program ExcelConverter bagian 4.1

- (g) Gambar 5.5 menjelaskan bahwa bila kolom tersebut adalah jam dan jika isinya LIBUR maka lewati baris tersebut. Jika isinya Shift maka pasti baris dibawahnya adalah jam. Sehingga, ambil *value* baris dibawahnya, lalu pisahkan menurut tanda "-" dan ganti tanda "." dengan tanda ":" . Kemudian, masukan ke *variable* bertipe LocalTime. Jika isi kolom berisi jam saja, maka pisahkan menurut tanda "-" dan ganti tanda "." dengan tanda ":" . Lalu, masukan ke *variable* bertipe LocalTime.

```
if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == colMatkulIdx ) {
    subject = cell.getStringCellValue();
}
```

Gambar 5.7: Implementasi kode program ExcelConverter bagian 5

- (h) Gambar 5.7 menjelaskan bila kolom tersebut adalah Nama Mata Kuliah, maka masukan ke *variable* String Subject.

```

if (cell.getRowIndex() > rowNoIdx && cell.getColumnIndex() >= colMatkulIdx+1
    && cell.getColumnIndex() < row.getLastCellNum()) {
    if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC) {
    }
    if (cell.getCellType() == Cell.CELL_TYPE_STRING) {
        if (cell.getStringCellValue().contains(":") ) {
            String[] spl1 = cell.getStringCellValue().split(":");
            String[] spl2 = spl1[1].split(",");
            for (int l = 0; l < spl2.length; l++) {
                dosen.add(spl2[l].trim());
                location.add("Lab");
            }
        } else {
            CellReference cr = new CellReference(1, cell.getColumnIndex());
            Row row2 = sheet.getRow(cr.getRow());
            Cell c = row2.getCell(cr.getCol());
            if (!cell.getStringCellValue().isEmpty())
            {
                dosen.add(cell.getStringCellValue().trim());
                location.add(String.valueOf((int) c.getNumericCellValue()).trim());
            }
        }
    }
}
}

```

Gambar 5.8: Implementasi kode program ExcelConverter bagian 6

- (i) Gambar 5.8 menjelaskan bila kolom tersebut adalah ruangan, yang bearti isinya adalah nama dosen yang mengawas. Jika isi kolom diawali dengan Lab, maka pisahkan menurut tanda ":" dan pisahkan kembali menurut tanda ",", sehingga menghasilkan nama dosen saja. Selanjutnya, masukan nama dosen ke ArrayList dosen dan isi ArrayList location dengan kata Lab. Jika isi kolom tidak diawali dengan kata lab, maka masukan nama dosen ke ArrayList dosen dan isi ArrayList location dengan mengambil nomor kolom dari ruangan tersebut dan mencocokkannya dengan posisi nama dosen tersebut berada dan isi nomer ruangan kedalam ArrayList Location.

```

if (cell.getCellType() == Cell.CELL_TYPE_BLANK && cell.getRowIndex() > 2) {
    CellReference cr = new CellReference(cell.getRowIndex() - 1, cell.getColumnIndex());
    Row row2 = sheet.getRow(cr.getRow());
    Cell c = row2.getCell(cr.getCol());
    CellReference cr2 = new CellReference(1, cell.getColumnIndex());
    Row row3 = sheet.getRow(cr2.getRow());
    Cell c2 = row3.getCell(cr2.getCol());
    if (c.getStringCellValue().contains(":") ) {
        String[] spl1 = c.getStringCellValue().split(":");
        String[] spl2 = spl1[1].split(",");
        for (int l = 0; l < spl2.length; l++) {
            dosen.add(spl2[l].trim());
            location.add("");
        }
    } else {
        if (!c.getStringCellValue().isEmpty())
        {
            dosen.add("");
            location.add("");
        }
    }
}
}

```

Gambar 5.9: Implementasi kode program ExcelConverter bagian 7

- (j) Gambar 5.9 menjelaskan bahwa karena dua mata kuliah berisikan dua baris kolom dosen yang di *merger* jadi satu dan Apache POI hanya dapat membaca baris pertama kolom yang digabungkan, maka pada baris yang kosong pada ArrayList dosen dan location diisi dengan String kosong.

```

for (int j = 0; j < dosen.size(); j++) {
    scheduleList.add(new ScheduleClass(lc, lt, lt.plusHours(2), subject, dosen.get(j), location.get(j)));
}
dosen.clear();
location.clear();
}
return mergingList(scheduleList);

```

Gambar 5.10: Implementasi kode program ExcelConverter bagian 8

- (k) Gambar 5.10 menjelaskan masukan semua *variable* yang diisikan sebelumnya kedalam sebuah ArrayList ScheduleClass sesuai jumlah ArrayList nama dosen.
- (l) Gambar 5.10 menjelaskan hapus semua isi ArrayList dosen agar tidak ada duplikasi.
- (m) Gambar 5.10 menjelaskan hasil ArrayList method readExcel() akan kembali dicek oleh *method* mergingList().

```

for (int i = 0; i < scheduleList.size(); i++) {
    if (scheduleList.get(i).getDosen().isEmpty() )
    {
        scheduleListSmt.add(scheduleList.get(i));
    }
}
for (int i = 0; i < scheduleListSmt.size() ; i++) {
    for (int j = 0; j < scheduleList.size(); j++) {
        if(scheduleList.get(j).equals(scheduleListSmt.get(i)))
        {
            scheduleList.remove(j);
        }
    }
}
for (int i = 0; i < scheduleList.size(); i++) {
    outerloop :
    for (int j = 0; j < scheduleListSmt.size(); j++) {
        if (scheduleList.get(i).getDate().equals(scheduleListSmt.get(j).getDate())
            && scheduleList.get(i).getTimeAwal().equals(scheduleListSmt.get(j).getTimeAwal()))
        {
            String ss = scheduleList.get(i).getSubject();
            scheduleList.get(i).setSubject(ss+" "+scheduleListSmt.get(j).getSubject());
            j = j + 1;
            break outerloop;
        }
    }
}
return scheduleList;

```

Gambar 5.11: Implementasi kode program ExcelConverter bagian 9

- (n) Gambar 5.11 pada lampiran A.2 menjelaskan jika ada dosen yang isinya kosong pada ArrayList ScheduleList, maka pindahkan isinya ke ArrayList baru yang bernama ScheduleListSmt.
  - (o) Gambar 5.11 menjelaskan cara menghapus isi ArrayList ScheduleList yang sama dengan ArrayList ScheduleListSmt.
  - (p) Gambar 5.11 menjelaskan jika tanggal dan jam pada ArrayList ScheduleList sama dengan ArrayList ScheduleListSmt, maka tambahkan subject dari ArrayList ScheduleList dengan subject yang ada di ArrayList ScheduleListSmt.
  - (q) Gambar 5.11 menjelaskan kembalian ArrayList ScheduleList.
3. Kode Program untuk Konversi Kalender
- Kelas CalendarConverter merupakan kelas yang dikhususkan untuk mengkonversi ScheduleClass menjadi file iCalendar.
- Berikut ini penjelasan dari implementasi kelas CalendarConverter :

```

44 public void calConverter (String path, ScheduleClass sch) throws SocketException
45     , FileNotFoundException, IOException, ValidationException
46 {
47     //creating timezone
48     TimeZoneRegistry registry = TimeZoneRegistryFactory.getInstance().createRegistry();
49     TimeZone timezone = registry.getTimeZone("Asia/Jakarta");
50     VTimeZone tz = timezone.getVTimeZone();
51
52     //Start Date
53     java.util.Calendar startDate = new GregorianCalendar();
54     startDate.setTimeZone(timezone);
55     startDate.set(java.util.Calendar.MONTH, sch.getDate().getMonthValue());
56     startDate.set(java.util.Calendar.DAY_OF_MONTH, sch.getDate().getDayOfMonth());
57     startDate.set(java.util.Calendar.YEAR, sch.getDate().getYear());
58     startDate.set(java.util.Calendar.HOUR_OF_DAY, sch.getTimeAwal().getHour());
59     startDate.set(java.util.Calendar.MINUTE, sch.getTimeAwal().getMinute());
60
61     //EndDate
62     java.util.Calendar endDate = new GregorianCalendar();
63     endDate.setTimeZone(timezone);
64     endDate.set(java.util.Calendar.MONTH, sch.getDate().getMonthValue());
65     endDate.set(java.util.Calendar.DAY_OF_MONTH, sch.getDate().getDayOfMonth());
66     endDate.set(java.util.Calendar.YEAR, sch.getDate().getYear());
67     endDate.set(java.util.Calendar.HOUR_OF_DAY, sch.getTimeAkhir().getHour());
68     endDate.set(java.util.Calendar.MINUTE, sch.getTimeAkhir().getMinute());

```

Gambar 5.12: Implementasi kode program CalendarConverter bagian 1

- Gambar 5.12 menjelaskan pembuatan *timeZone* untuk wilayah Indonesia.
- Gambar 5.12 menjelaskan pembuatan tanggal dan waktu dimulainya ujian dengan mengkonversi tahun, bulan, tanggal, jam, dan menit dari ScheduleClass.
- Gambar 5.12 menjelaskan pembuatan tanggal dan waktu ujian tersebut berakhir dengan mengkonversi tahun, bulan, tanggal, jam, dan menit dari ScheduleClass.

```

69 //creating an event
70 String eventName = sch.getSubject();
71 String location2 = sch.getLocation();
72 String desc = "Mengawas Ujian "+sch.getDosen();
73 DateTime start = new DateTime(startDate.getTime());
74 DateTime end = new DateTime(endDate.getTime());
75 VEvent mengawas = new VEvent(start,end,eventName);
76 mengawas.getProperties().add(new Location(location2));
77 mengawas.getProperties().add(new Description());
78
79 try {
80     mengawas.getProperties().getProperty(Property.DESCRPTION).setValue(desc);
81 } catch (Exception e) {
82 }
83
84 //add timezone info
85 mengawas.getProperties().add(tz.getTimeZoneId());
86
87 //generate unique identifier
88 UidGenerator uidgenerator = new UidGenerator("uidGen");
89 Uid uid = uidgenerator.generateUid();
90 mengawas.getProperties().add(uid);

```

Gambar 5.13: Implementasi kode program CalendarConverter bagian 2

- Gambar 5.13 menjelaskan pembuatan *event* pada kalendar.
- Gambar 5.13 memasukan *timeZone* pada kalendar.
- Gambar 5.13 menjelaskan identitas pembuat calendar.

```

//creating calendar
net.fortuna.ical4j.model.Calendar calendar = new net.fortuna.ical4j.model.Calendar();
calendar.getProperties().add(new ProdId("-//Ben Fortuna//iCal4j 1.0//EN"));
calendar.getProperties().add(Version.VERSION_2_0);
calendar.getProperties().add(CalScale.GREGORIAN);

// Add the event and print
calendar.getComponents().add(mengawas);
System.out.println(calendar);

//saving iCal
String calFile = sch.getSubject();

FileOutputStream fout = new FileOutputStream(path);

CalendarOutputter outputter = new CalendarOutputter();
outputter.setValidating(false);
outputter.output(calendar, fout);

```

Gambar 5.14: Implementasi kode program CalendarConverter bagian 3

- (g) Gambar 5.14 menjelaskan cara pembuatan calendar.
- (h) Gambar 5.14 memasukan *event* yang telah dibuat kedalam kalender dan *print* sesudahnya.
- (i) Gambar 5.14 menjelaskan cara menyimpan file iCalendar pada direktori tertentu.

#### 4. Kode Program *Controller*

Kelas FXMLDocumentController merupakan kelas yang bertugas menjadi penghubung kelas *view* dengan kelas-kelas lainnya. Di kelas ini hasil dari excel yang telah dibaca akan ditampilkan pada tabelview dan semua fungsi *button* dan *textbox* di atur dalam kelas ini.

```

@FXML
private void handleButtonAction(ActionEvent event) {

    FileChooser fileChooser = new FileChooser();
    fileChooser.setTitle("Open Resource File");

    //Set extension filter
    FileChooser.ExtensionFilter extFilter = new FileChooser.ExtensionFilter("Excel files (*.xlsx)", "*.xlsx");
    FileChooser.ExtensionFilter extFilter2 = new FileChooser.ExtensionFilter("Excel files (*.xls)", "*.xls");
    fileChooser.getExtensionFilters().add(extFilter);
    fileChooser.getExtensionFilters().add(extFilter2);
    selectedFile = fileChooser.showOpenDialog(null);

    if (selectedFile != null)
    {
        txtFile.setText(selectedFile.getAbsolutePath());
    }
}

```

Gambar 5.15: Implementasi kode program FXMLDocumentController bagian 1

Berikut ini penjelasan dari kode-kode dalam kelas FXMLDocumentController :

- (a) Gambar 5.15 menjelaskan file yang akan dipilih nanti harus berekstensi .xlsx.
- (b) Gambar 5.15 menjelaskan bagaimana memunculkan *pop-up window* untuk memilih file *input*.
- (c) Gambar 5.15 menjelaskan jika file tidak kosong maka ambil *path* file tersebut dan isikan *textbox* browse dengan *path* file tersebut.



```

@FXML
private void handleConvertAction(ActionEvent event) throws FileNotFoundException, IOException
{
    ExcelConverter con = new ExcelConverter(selectedFile);
    jadwalList = FXCollections.observableArrayList(con.readExcel());

    jadwalTable.setItems(jadwalList);
    jadwalTable.getColumns().get(0).setCellValueFactory(new PropertyValueFactory("Date"));
    jadwalTable.getColumns().get(1).setCellValueFactory(new PropertyValueFactory<>("timeAwal"));
    jadwalTable.getColumns().get(2).setCellValueFactory(new PropertyValueFactory<>("timeAkhir"));
    jadwalTable.getColumns().get(3).setCellValueFactory(new PropertyValueFactory("Subject"));
    jadwalTable.getColumns().get(4).setCellValueFactory(new PropertyValueFactory("Dosen"));
    jadwalTable.getColumns().get(5).setCellValueFactory(new PropertyValueFactory("Location"));

    filteredData.addAll(jadwalList);

    jadwalList.addListener( new ListChangeListener<ScheduleClass>()
    {
        @Override
        public void onChanged(ListChangeListener.Change<? extends ScheduleClass> change)
        {
            updateFilteredData();
        }
    });
}

```

Gambar 5.16: Implementasi kode program FXMLDocumentController bagian 2

- (d) Gambar 5.16 menjelaskan konversi file excel tersebut, kemudian ambil hasilnya dan masukan kedalam ObservableArrayList.
- (e) Gambar 5.16 menjelaskan cara menampilkan hasil konversi kedalam *tableview* dengan mengisikan sesuai urutan kolom pada *tableview*.
- (f) Gambar 5.16 menjelaskan masukan semua ObservableArrayList jadwalList kedalam ObservableArrayList filteredData untuk keperluan filter nanti dan jika ada perubahan pada ObservableArrayList jadwalList maka *update* pula ObservableArrayList filteredData.

```

101 public void convertClicked() throws FileNotFoundException,
102     IOException, SocketException, ValidationException
103 {
104     ScheduleClass selected = jadwalTable.getSelectionModel().getSelectedItem();
105     //System.out.println("selected = "+selected);
106     FileChooser fileChooser = new FileChooser();
107     fileChooser.setTitle("Save iCal File");
108
109     File save = fileChooser.showSaveDialog(null);
110     int idx = jadwalTable.getSelectionModel().getSelectedIndex();
111     //System.out.println("idx = "+idx);
112     String path;
113     if(save != null)
114     {
115         path = save.getAbsolutePath();
116         CalendarConverter cc = new CalendarConverter();
117         cc.calConverter(path, selected);
118     }
119     else
120     {
121         System.out.println("Canceled !");
122     }
123 }

```

Gambar 5.17: Implementasi kode program FXMLDocumentController bagian 3

- (g) Gambar 5.17 menjelaskan ambil kelas ScheduleClass yang dipilih oleh *user* dan masukan kedalam *variable* selected.

- (h) Gambar 5.17 menjelaskan bahwa setiap file yang akan disimpan diberikan ekstensi .ics .
- (i) Gambar 5.17 menjelaskan cara memunculkan *save dialog*.
- (j) Gambar 5.17 menjelaskan ambil path direktori dimana *user* akan menyimpan file, lalu konversi jadwal yang telah dipilih oleh *user* dan simpan di direktori yang sudah ditentukan.

```

121  @FXML
122  private void filterConversion()
123  {
124      jadwalTable.setItems(filteredData);
125      filterTxt.textProperty().addListener(new ChangeListener<String>()
126      {
127          @Override
128          public void changed(ObservableValue<? extends String> observable,
129                          String oldValue, String newValue)
130          {
131              updateFilteredData();
132          }
133      });
134  }
135
136  private void updateFilteredData()
137  {
138      filteredData.clear();
139      for (ScheduleClass sc : jadwalList)
140      {
141          if (matchesFilter(sc))
142          {
143              filteredData.add(sc);
144          }
145      }
146      reapplyTableSortOrder();
147  }

```

Gambar 5.18: Implementasi kode program FXMLDocumentController bagian 4

- (k) Gambar 5.18 menjelaskan jika *textbox filter* di isi oleh *user* maka jadwal di tabel pun berubah sesuai dengan nama dosen yang di tuliskan *user*.
- (l) Gambar 5.18 menjelaskan cara *update* filteredData sesuai nama dosen yang di input oleh *user*.

```

149  private boolean matchesFilter(ScheduleClass sc)
150  {
151      String filterString = filterTxt.getText();
152
153      if (filterString == null || filterString.isEmpty())
154      {
155          return true;
156      }
157      String lowerCaseFilterString = filterString.toLowerCase();
158
159      if (sc.getDosen().toLowerCase().indexOf(lowerCaseFilterString) != -1)
160      {
161          return true;
162      }
163      return false;
164  }
165
166  private void reapplyTableSortOrder()
167  {
168      ArrayList<TableColumn<ScheduleClass, ? >> sortOrder = new ArrayList<>(jadwalTable.getSortOrder());
169      jadwalTable.getSortOrder().clear();
170      jadwalTable.getSortOrder().addAll(sortOrder);
171  }
172  }

```

Gambar 5.19: Implementasi kode program FXMLDocumentController bagian 5

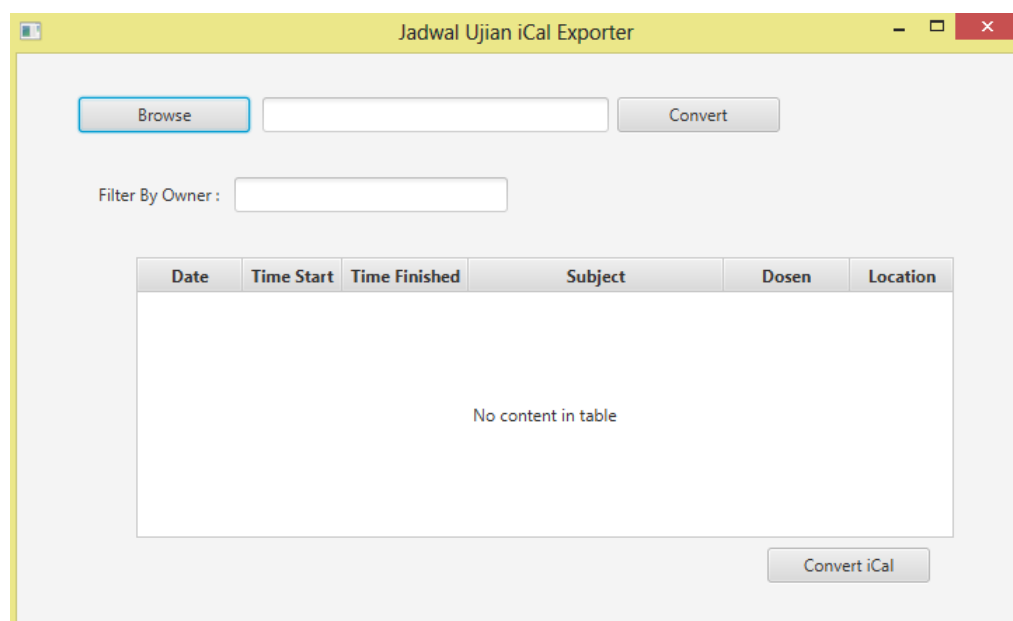
- (m) Gambar 5.19 menjelaskan bila nama dosen yang ditulis di *textbox* filter sama dengan nama dosen yang ada di *ScheduleClass*, maka kembalikan nilai *true*. Jika tidak, maka kembalikan nilai *false*.
- (n) Gambar 5.19 menjelaskan pengurutan kembali tabel, sehingga isi tabelnya sesuai dengan apa yang di masukan pada *textbox* oleh *user* sebelumnya.

## 5.2 Implementasi Antarmuka

Pada subbab ini akan dibahas implementasi antarmuka dari perangkat lunak. Terdapat beberapa perubahan dari perancangan antarmuka perangkat lunak pada bab 4.

Berikut ini implementasi antarmuka dari perangkat lunak iCalendarConverter :

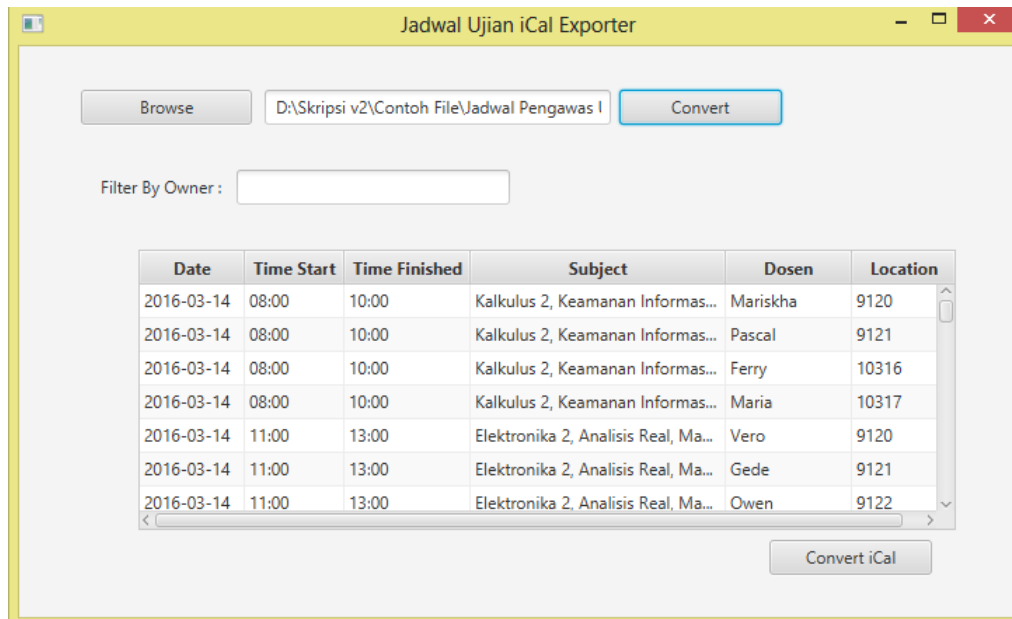
1. Tampilan perangkat lunak iCalendarConverter



Gambar 5.20: Tampilan antarmuka perangkat lunak

Pada tampilan 5.20 terlihat beberapa perubahan dengan perancangan antarmuka pada bab sebelumnya. Beberapa perubahan tersebut diantaranya, fungsi fitur *download* yang terletak pada tabel digantikan dengan *button* Convert to iCal pada antarmuka. Hal ini dikarenakan sangat sulit untuk membuat *button* pada setiap isi dari tabel yang berbeda-beda isinya. Selain itu, jika setiap isi tabel mempunyai file iCal akan sangat memakan memori apalagi jika jumlah datanya sangat banyak. Maka, diputuskan untuk membuat satu *button* Convert to iCal yang menangkap *selected item* dari kursor pengguna pada tabel. Selanjutnya, jika pengguna menekan tombol tersebut akan memunculkan *pop-up window* meminta pengguna menentukan tempat penyimpanan file iCal dari jadwal yang telah dipilih sebelumnya.

2. Tampilan Antarmuka ketika file excel jadwal mengawas telah dimasukkan



Gambar 5.21: Tampilan antarmuka setelah file mengawas dimasukkan

## 5.3 Pengujian

Pada subbab ini akan dilakukan pengujian pada perangkat lunak untuk mengetahui apakah program dapat berjalan sesuai dengan apa yang di inginkan. Terdapat dua pengujian yaitu :

1. Pengujian Fungsional.
2. Pengujian Eksperimental.

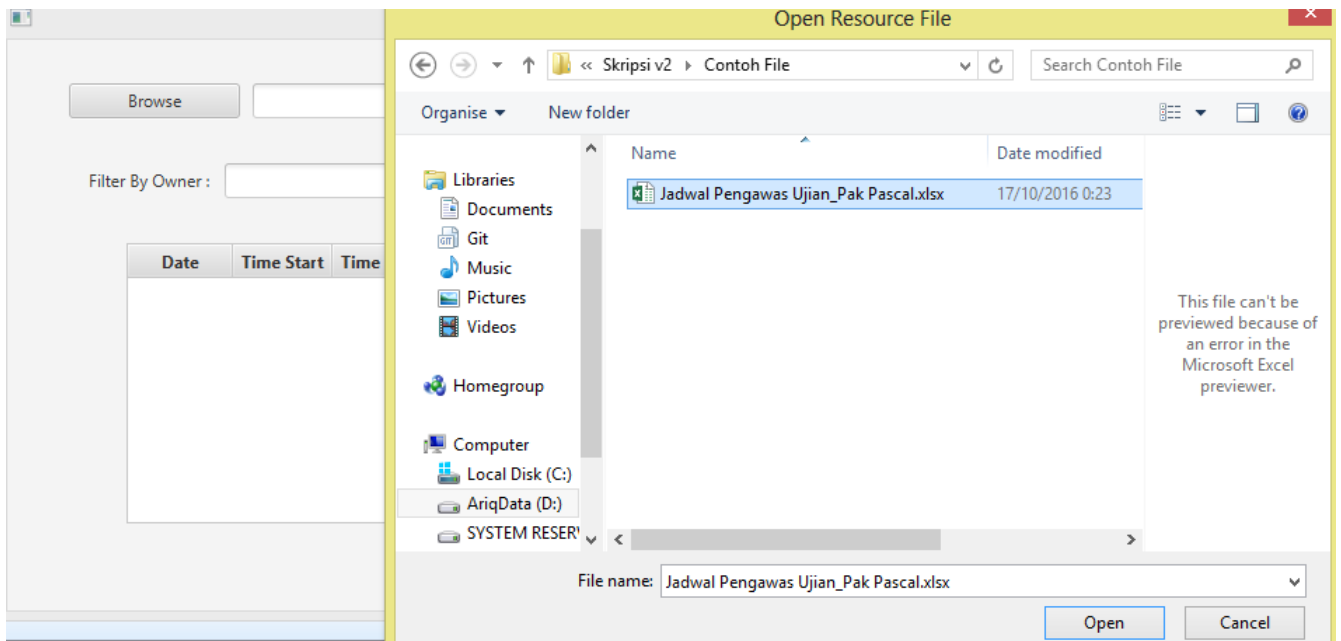
### 5.3.1 Pengujian Fungsional

Pada pengujian ini akan di uji mengenai fungsionalitas dari perangkat lunak. Selain itu, dalam pengujian fungsional menggunakan file uji [B.1](#) untuk mengetahui apakah program berjalan sesuai harapan dan beberapa fungsi berjalan dengan baik. Berikut hasil pengujiannya :

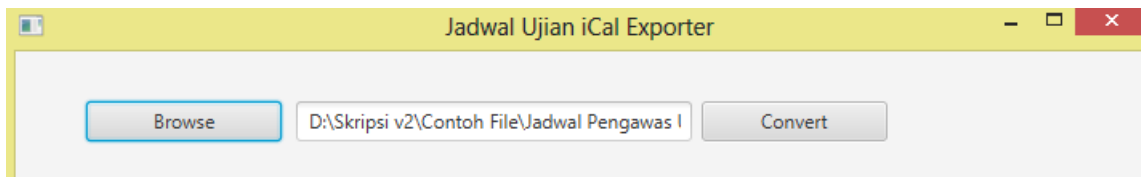
Tabel 5.1: Tabel hasil pengujian fungsional

Hal yang diuji	Hasil yang diharapkan	Hasil
Browse file excel	PL dapat melakukan browse file excel	Berhasil <a href="#">5.22</a>
Path file excel	PL dapat menangkap alamat file dari <i>input file</i> excel	Berhasil <a href="#">5.23</a>
Menampilkan Jadwal ke layar	PL menampilkan ke layar file excel yang telah dibaca	Berhasil <a href="#">5.24</a>
Konversi ke iCal	PL dapat mengkonversi jadwal yang diseleksi pengguna kedalam iCalendar	Berhasil <a href="#">5.25</a>
Filter nama dosen	PL dapat menampilkan nama dosen yang telah di filter, sesuai yang di yang dimasukkan oleh pengguna pada <i>textbox</i> filter	Berhasil <a href="#">5.27</a>
Hasil Filter dapat di-konversi ke iCal	Hasil Filter pada PL dan diseleksi oleh pengguna, dapat di konversikan kedalam iCal	Berhasil <a href="#">5.28</a>
<i>Import</i> Google Calendar	Hasil konversi PL dapat di masukan kedalam Google Calendar	Berhasil <a href="#">5.50</a>
Dapat dibuka di Outlook	Hasil konversi PL dapat di buka di Outlook	Berhasil <a href="#">5.35</a>
Hasil filter dapat di <i>import</i> Google Calendar	Hasil filter konversi PL dapat di masukan kedalam Google Calendar	Berhasil <a href="#">5.36</a>
Hasil filter Dapat dibuka di Outlook	Hasil filter konversi PL dapat di buka di Outlook	Berhasil <a href="#">5.61</a>

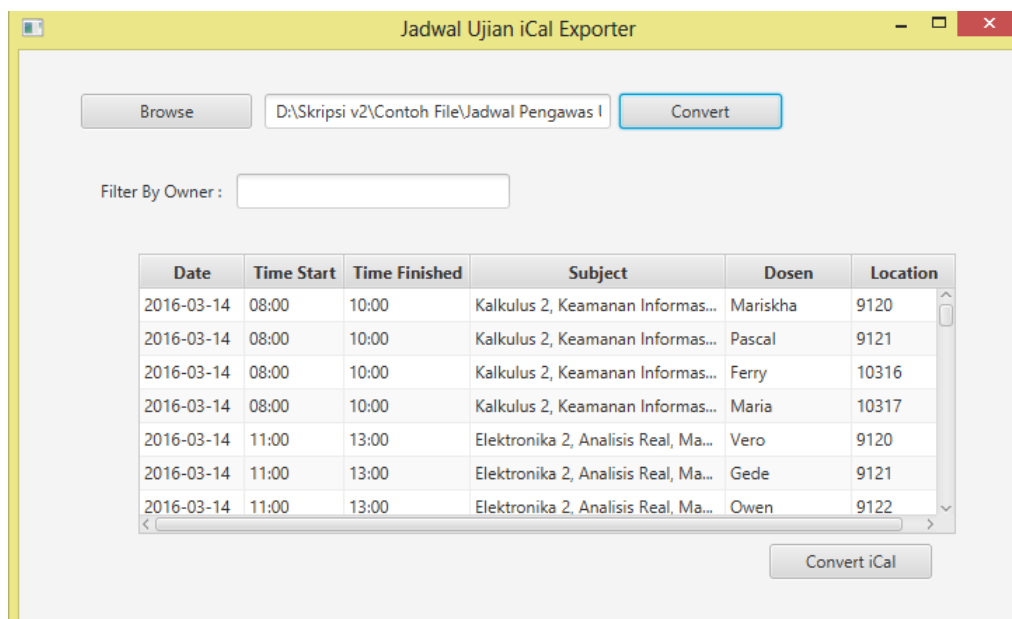
Berikut ini adalah tampilan dari hasil pengujian yang telah dilakukan pada tabel [5.1](#) :



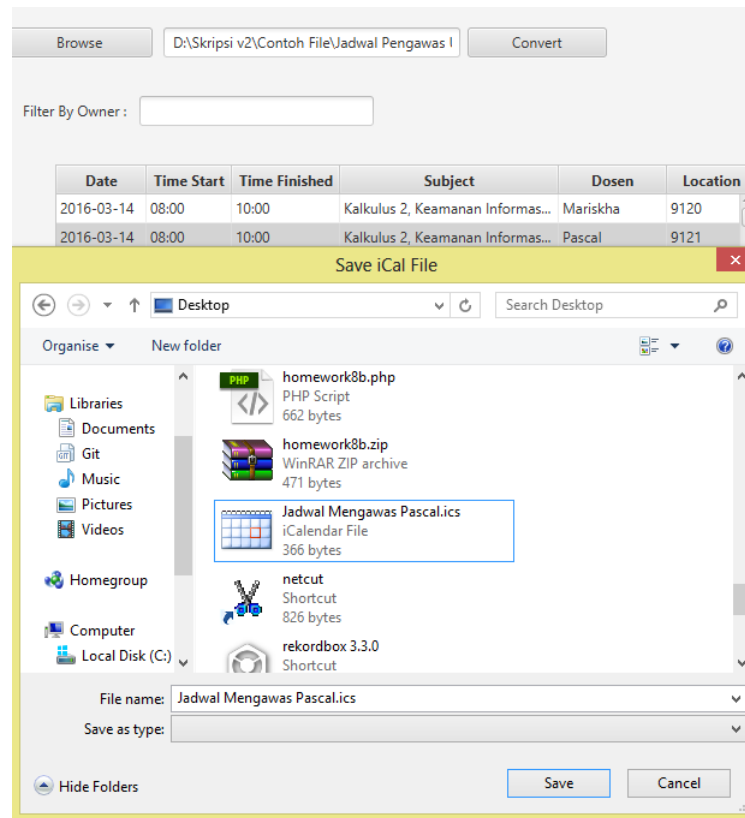
Gambar 5.22: Tampilan browse file excel mengawas ujian



Gambar 5.23: Tampilan path file excel mengawas ujian



Gambar 5.24: PL menampilkan jadwal ke layar



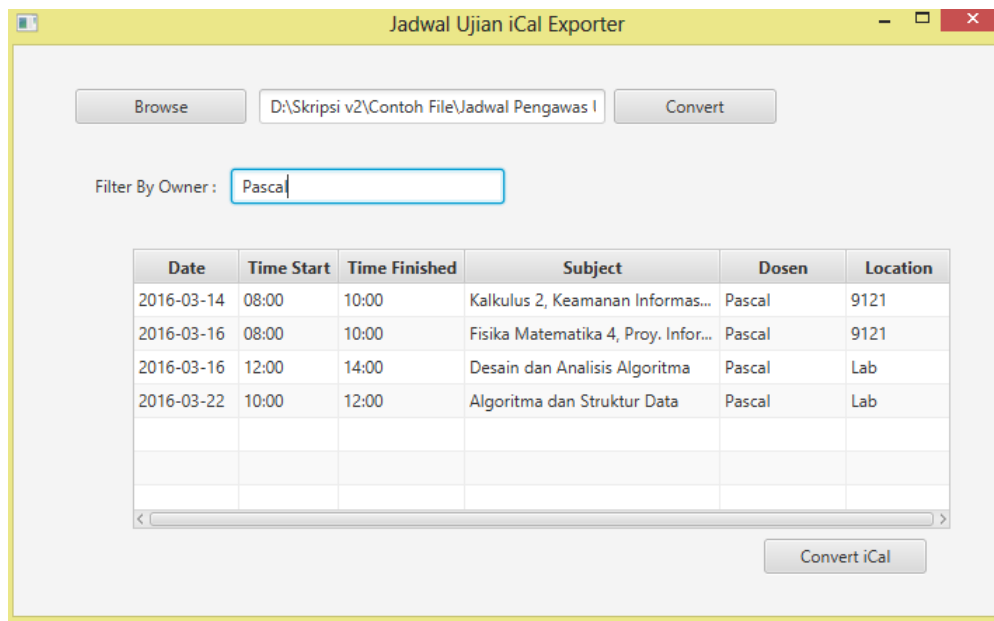
Gambar 5.25: PL mengkonversi jadwal ke format iCal

```

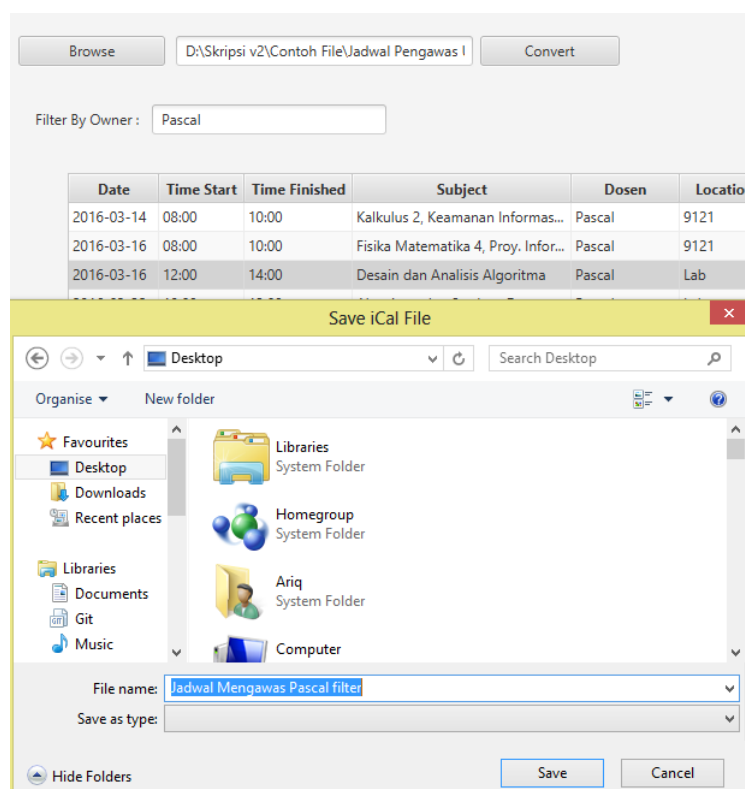
1 BEGIN:VCALENDAR
2 PRODID:-//Ben Fortuna//iCal4j 1.0//EN
3 VERSION:2.0
4 CALSCALE:GREGORIAN
5 BEGIN:VEVENT
6 DTSTAMP:20161116T135352Z
7 DTSTART:20160314T080052
8 DTEND:20160314T100052
9 SUMMARY:Kalkulus 2\, Keamanan Informasi\, Reologi
10 LOCATION:9121
11 DESCRIPTION:Mengawas Ujian Pascal
12 TZID:Asia/Jakarta
13 UID:20161116T135353Z-uidGen@AriqRahmaeri
14 END:VEVENT
15 END:VCALENDAR
16

```

Gambar 5.26: File iCal



Gambar 5.27: Hasil pengujian filter nama dosen



Gambar 5.28: Hasil pengujian convert hasil filter kedalam iCal

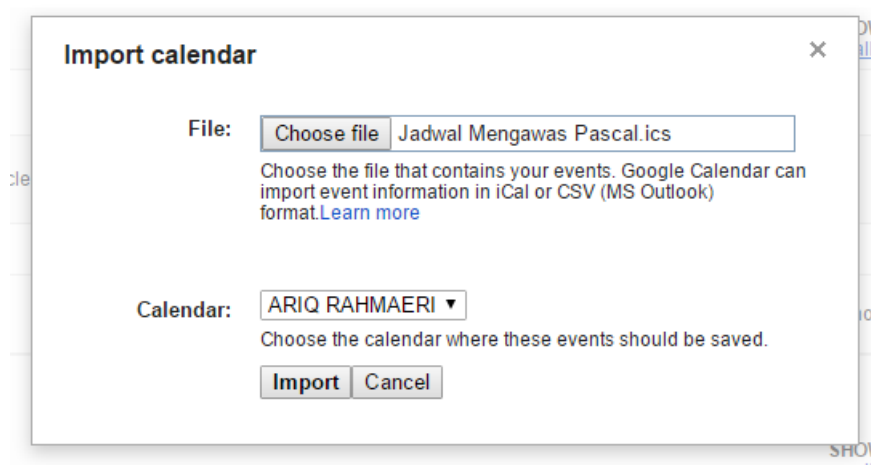


```

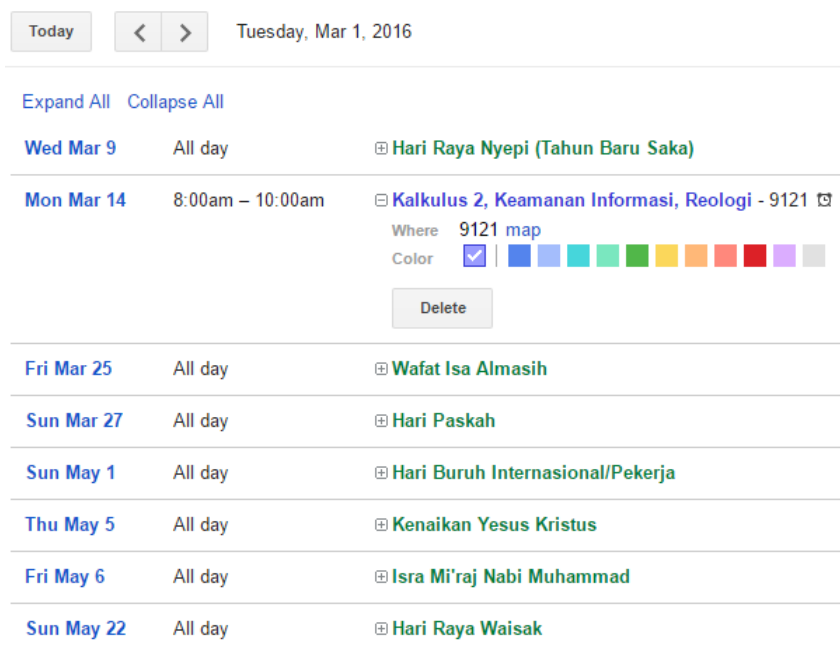
1 BEGIN:VCALENDAR
2 PRODID:-//Ben Fortuna//iCal4j 1.0//EN
3 VERSION:2.0
4 CALSCALE:GREGORIAN
5 BEGIN:VEVENT
6 DTSTAMP:20161116T135417Z
7 DTSTART:20160316T120017
8 DTEND:20160316T140017
9 SUMMARY:Desain dan Analisis Algoritma
10 LOCATION:Lab
11 DESCRIPTION:Mengawas Ujian Pascal
12 TZID:Asia/Jakarta
13 UID:20161116T135417Z-uidGen@AriqRahmaeri
14 END:VEVENT
15 END:VCALENDAR

```

Gambar 5.29: File iCal Filter



Gambar 5.30: Hasil pengujian import kedalam Google Calendar



Gambar 5.31: Hasil import ke Google Calendar

**Kalkulus 2, Keamanan Informasi, Reologi**

3/14/2016 8:00am to 10:00am 3/14/2016 [Time zone](#)

☐ All day ☐ Repeat...

Event details [Find a time](#)

Where  [map](#)

Add: [Guests](#) | [Rooms, etc.](#)

Video call [Add video call](#)

Calendar

Description

Guests can

- ☐ modify event
- ☒ invite others
- ☒ see guest list

Attachment [Add attachment](#)

Event color ☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Notifications

[Add a notification](#)

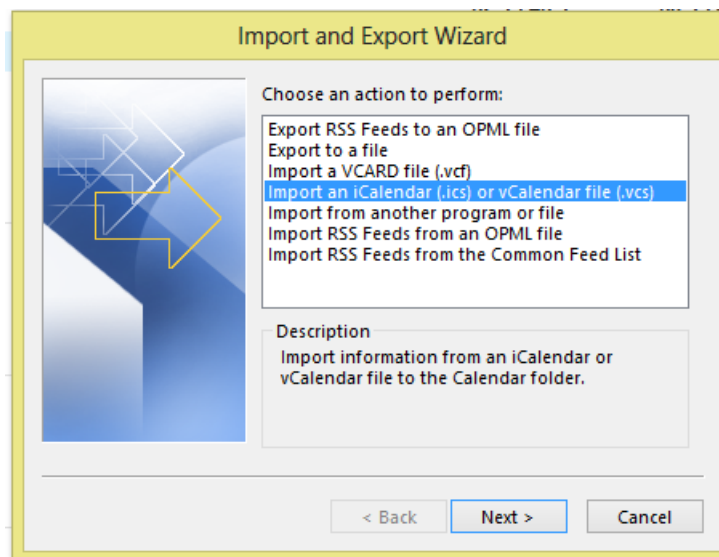
Show me as ☐ Available ☒ Busy

Visibility ☒ Calendar default ☐ Public ☐ Private

By default this event will follow the [sharing settings](#) of this calendar. event details will be visible to anyone who can see details of other events in this calendar. [Learn more](#)

[Publish event](#)

Gambar 5.32: Hasil import ke Google Calendar bagian 2

Gambar 5.33: *Import* file iCal kedalam MS Outlook

◀ ▶ Maret 2016

SENIN	SELASA
29 Feb	<b>1 Mar</b>
7	8
14 8:01 Kalkulus 2, Keamanan Informasi, Reologi; 9121	15
21	22

Gambar 5.34: File hasil Konversi berada pada kalender MS Outlook

Subject	Kalkulus 2, Keamanan Informasi, Reologi		
Location	9121		
Start time	Sen 14/03/2016	8:01	<input type="checkbox"/> All day event
End time	Sen 14/03/2016	10:01	

Mengawas Ujian Pascal

Gambar 5.35: File hasil Konversi dapat dibuka di MS Outlook

**Import calendar** ×

File: Choose file Jadwal Mengawas Pascal filter.ics

Choose the file that contains your events. Google Calendar can import event information in iCal or CSV (MS Outlook) format. [Learn more](#)

Calendar: ARIQ RAHMAERI ▼

Choose the calendar where these events should be saved.

Import Cancel

Gambar 5.36: Hasil pengujian import file yang di filter kedalam Google Calendar

Showing events after 2/14/2016. [Look for earlier events](#)

Wed Mar 9	All day	⊕ Hari Raya Nyepi (Tahun Baru Saka)
Mon Mar 14	8:00am – 10:00am	⊕ Kalkulus 2, Keamanan Informasi, Reologi - 9121
Wed Mar 16	12:00pm – 2:00pm	⊖ Desain dan Analisis Algoritma - Lab
Where <a href="#">Lab</a> <a href="#">map</a> Color <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>		
<a href="#">Delete</a>		
Fri Mar 25	All day	⊕ Wafat Isa Almasih
Sun Mar 27	All day	⊕ Hari Paskah
Sun May 1	All day	⊕ Hari Buruh Internasional/Pekerja
Thu May 5	All day	⊕ Kenaikan Yesus Kristus
Fri May 6	All day	⊕ Isra Mi'raj Nabi Muhammad

Gambar 5.37: Hasil import file yang di filter ke Google Calendar

Desain dan Analisis Algoritma

3/16/2016

12:00pm

to

2:00pm

3/16/2016

Time zone

☐ All day
 ☐ Repeat...

Event details

Find a time

Where

Lab

map

Video call

Add video call

Calendar

ARIQ RAHMAERI

Description

Mengawas Ujian Pascal

Attachment

Add attachment

Event color

☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Notifications

Notification

10

minutes

×

Add a notification

Show me as

☐ Available
 ☒ Busy

Visibility

☒ Calendar default
 ☐ Public
 ☐ Private

By default this event will follow the [sharing settings](#) of this calendar: event details will be visible to anyone who can see details of other events in this calendar. [Learn more](#)

[Publish event](#)

Add: Guests | [Rooms, etc.](#)

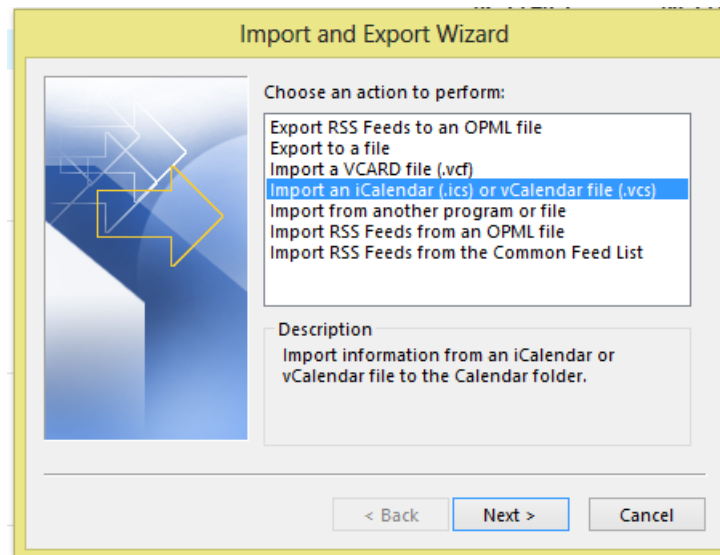
Enter guest email address

Add

Guests can

☐ modify event
 ☒ invite others
 ☒ see guest list

Gambar 5.38: Hasil import file yang di filter ke Google Calendar bagian 2



Gambar 5.39: *Import* file iCal yang telah di filter kedalam MS Outlook

◀ ▶ Maret 2016

SENIN	SELASA
29 Feb	<b>1 Mar</b>
7	8
14 8:01 Kalkulus 2, Keamanan Informasi, Reologi; 9121	15
21	22

Gambar 5.40: File iCal yang telah di filter berada pada kalender MS Outlook

Subject	Desain dan Analisis Algoritma		
Location	Lab		
Start time	Rab 16/03/2016	12:00	<input type="checkbox"/> All day event
End time	Rab 16/03/2016	14:00	

Mengawas Ujian Pascal

Gambar 5.41: File hasil filter dapat dibuka di MS Outlook

### 5.3.2 Pengujian Eksperimental

Pengujian eksperimental merupakan pengujian yang dilakukan dengan melibatkan skenario yang bersifat eksperimental. Pengujian ini ditujukan untuk melihat reaksi program menerima berbagai kejadian. Selain itu, file uji B.2 pada pengujian eksperimental ini merupakan format file baru yang dikeluarkan TU FTIS untuk 2016. Berikut hasil pengujiannya :

Tabel 5.2: Tabel hasil pengujian eksperimental

Hal yang diuji	Hasil yang diharapkan	Hasil
Browse file excel	PL dapat melakukan browse file excel	Berhasil 5.22
Path file excel	PL dapat menangkap alamat file dari <i>input file</i> excel	Berhasil 5.23
Memasukan file yang bukan excel	PL mengeluarkan notifikasi kesalahan file <i>input</i>	Berhasil 5.43
Menampilkan Jadwal ke layar	PL menampilkan ke layar file excel yang telah dibaca	Tidak berhasil Karena format tahun pada file excel jadwal baru menggunakan tanda '' untuk menandakan tahun ('16). Sehingga, <i>output</i> yang dikeluarkan PL berupa notifikasi kesalahan format 5.42

Berikut ini merupakan gambar hasil pengujian eksperimental :

```

at javafx.scene.Scene.impl_processMouseEvent(Scene.java:1762)
at javafx.scene.Scene$ScenePeerListener.mouseEvent(Scene.java:2494)
at com.sun.javafx.tk.quantum.GlassViewEventHandler$MouseEventNotification.run(GlassViewEventHandler.java:380)
at com.sun.javafx.tk.quantum.GlassViewEventHandler$MouseEventNotification.run(GlassViewEventHandler.java:294)
at java.security.AccessController.doPrivileged(Native Method)
at com.sun.javafx.tk.quantum.GlassViewEventHandler.lambda$handleMouseEvent$354(GlassViewEventHandler.java:416)
at com.sun.javafx.tk.quantum.QuantumToolkit.runWithoutRenderLock(QuantumToolkit.java:389)
at com.sun.javafx.tk.quantum.GlassViewEventHandler.handleMouseEvent(GlassViewEventHandler.java:415)
at com.sun.glass.ui.View.handleMouseEvent(View.java:555)
at com.sun.glass.ui.View.notifyMouse(View.java:937)
at com.sun.glass.ui.win.WinApplication._runLoop(Native Method)
at com.sun.glass.ui.win.WinApplication.lambda$null$148(WinApplication.java:191)
at java.lang.Thread.run(Thread.java:745)
Caused by: java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at sun.reflect.misc.Trampoline.invoke(MethodUtil.java:71)
at sun.reflect.GeneratedMethodAccessor1.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at sun.reflect.misc.MethodUtil.invoke(MethodUtil.java:275)
at javafx.fxml.FXMLLoader$MethodHandler.invoke(FXMLLoader.java:1769)
... 48 more
Caused by: java.lang.NumberFormatException: For input string: ""16"
at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
at java.lang.Integer.parseInt(Integer.java:580)
at java.lang.Integer.parseInt(Integer.java:615)
at icalendarconverter.ExcelConverter.Converter(ExcelConverter.java:108)
at icalendarconverter.FXMLDocumentController.handleConvertAction(FXMLDocumentController.java:74)
... 58 more

```

Gambar 5.42: *Output* PL pada pengujian eksperimental

```

at com.sun.javafx.tk.quantum.GlassViewEventHandler.lambda$handleMouseEvent$354(GlassViewEventHandler.java:416)
at com.sun.javafx.tk.quantum.QuantumToolkit.runWithoutRenderLock(QuantumToolkit.java:389)
at com.sun.javafx.tk.quantum.GlassViewEventHandler.handleMouseEvent(GlassViewEventHandler.java:415)
at com.sun.glass.ui.View.handleMouseEvent(View.java:555)
at com.sun.glass.ui.View.notifyMouse(View.java:937)
at com.sun.glass.ui.win.WinApplication._runLoop(Native Method)
at com.sun.glass.ui.win.WinApplication.lambda$null$148(WinApplication.java:191)
at java.lang.Thread.run(Thread.java:745)
Caused by: java.lang.reflect.InvocationTargetException
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at sun.reflect.misc.Trampoline.invoke(MethodUtil.java:71)
at sun.reflect.GeneratedMethodAccessor1.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at sun.reflect.misc.MethodUtil.invoke(MethodUtil.java:275)
at javafx.fxml.FXMLLoader$MethodHandler.invoke(FXMLLoader.java:1769)
... 111 more
Caused by: org.apache.poi.openxml4j.exceptions.InvalidFormatException: Package should contain a content type part [M1
at org.apache.poi.util.PackageHelper.open(PackageHelper.java:39)
at org.apache.poi.xssf.usermodel.XSSFWorkbook.<init>(XSSFWorkbook.java:274)
at icalendarconverter.ExcelConverter.Converter(ExcelConverter.java:61)
at icalendarconverter.FXMLDocumentController.handleConvertAction(FXMLDocumentController.java:74)
... 121 more
Caused by: org.apache.poi.openxml4j.exceptions.InvalidFormatException: Package should contain a content type part [M1.13]
at org.apache.poi.openxml4j.opc.ZipPackage.getPartsImpl(ZipPackage.java:197)
at org.apache.poi.openxml4j.opc.OPCPackage.getParts(OPCPackage.java:696)
at org.apache.poi.openxml4j.opc.OPCPackage.open(OPCPackage.java:280)
at org.apache.poi.util.PackageHelper.open(PackageHelper.java:37)
... 124 more

```

Gambar 5.43: *Output* PL pada pengujian eksperimental dari file *input* yang bukan excel

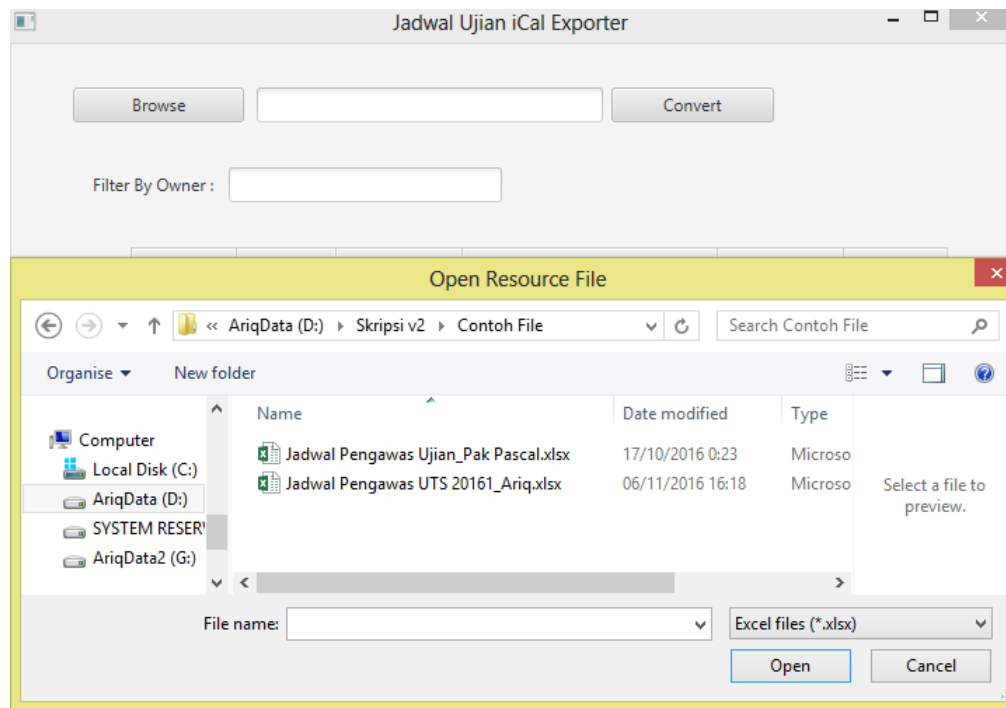
Berdasarkan hasil pengujian tersebut maka dilakukan perubahan pada cara membaca PL sehingga dapat membaca kedua file excel, baik format lama maupun format baru yang dikeluarkan TU FTIS. Kelas ExcelConverter sebelum revisi dapat dilihat pada A.5. Berikut hasil pengujian setelah revisi :

Tabel 5.3: Tabel hasil pengujian eksperimental setelah revisi

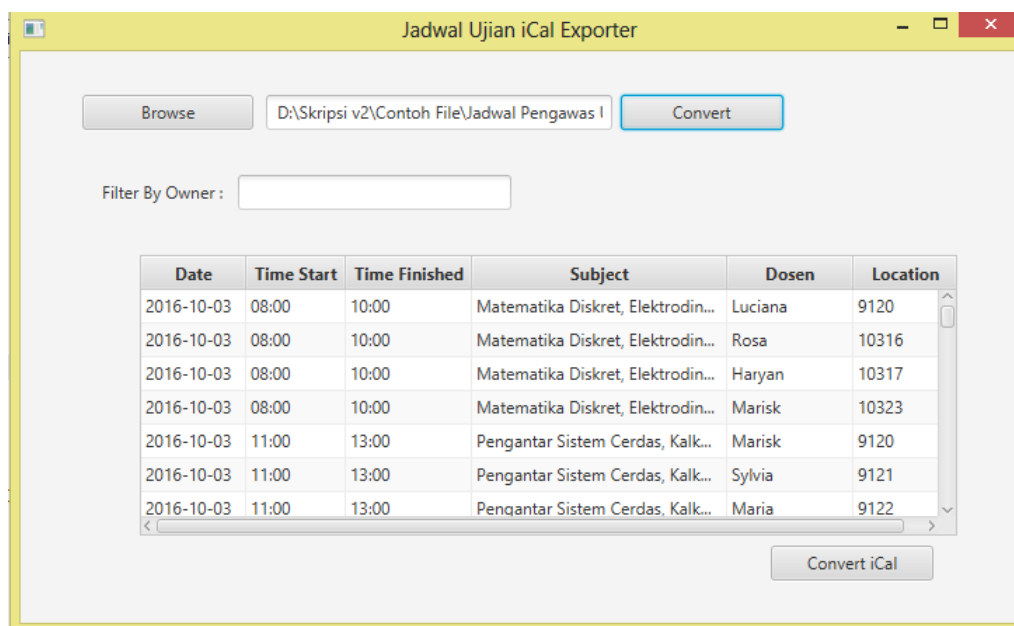
Hal yang diuji	Hasil yang diharapkan	Hasil
Browse file excel	PL dapat melakukan browse file excel	Berhasil <a href="#">5.22</a>
Path file excel	PL dapat menangkap alamat file dari <i>input file</i> excel	Berhasil <a href="#">5.23</a>
Memasukan file yang bukan excel	PL menggunakan <i>extension filter</i> sehingga file bukan excel tidak dapat menjadi file <i>input</i>	Berhasil <a href="#">5.44</a>
Menampilkan Jadwal ke layar	PL menampilkan ke layar file excel yang telah dibaca	Berhasil <a href="#">5.45</a>
Konversi ke iCal	PL dapat mengkonversi jadwal yang diseleksi pengguna kedalam iCalendar	Berhasil <a href="#">5.46</a>
Filter nama dosen	PL dapat menampilkan nama dosen yang telah di filter, sesuai yang di yang dimasukkan oleh pengguna pada <i>textbox</i> filter	Berhasil <a href="#">5.27</a>
Hasil Filter dapat di-konversi ke iCal	Hasil Filter pada PL dan diseleksi oleh pengguna, dapat di konversikan kedalam iCal	Berhasil <a href="#">5.48</a>
<i>Import</i> Google Calendar	Hasil konversi PL dapat di masukan kedalam Google Calendar	Berhasil <a href="#">5.51</a>
Dapat dibuka di Outlook	Hasil konversi PL dapat di buka di Outlook	Berhasil <a href="#">5.54</a>
Hasil filter dapat di <i>import</i> Google Calendar	Hasil filter konversi PL dapat di masukan kedalam Google Calendar	Berhasil <a href="#">5.57</a>
Hasil filter Dapat dibuka di Outlook	Hasil filter konversi PL dapat di buka di Outlook	Berhasil <a href="#">5.60</a>

Berikut merupakan tampilan dari hasil pengujian menggunakan file excel dengan format baru [B.2](#) :

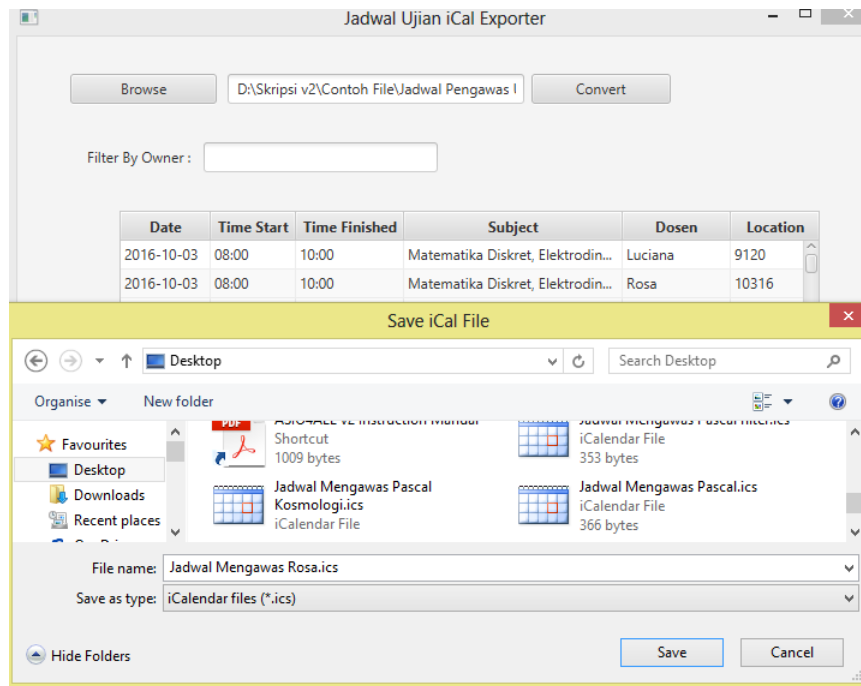




Gambar 5.44: PL memfilter ekstensi file excel yang akan dimasukkan



Gambar 5.45: PL menampilkan jadwal ke layar



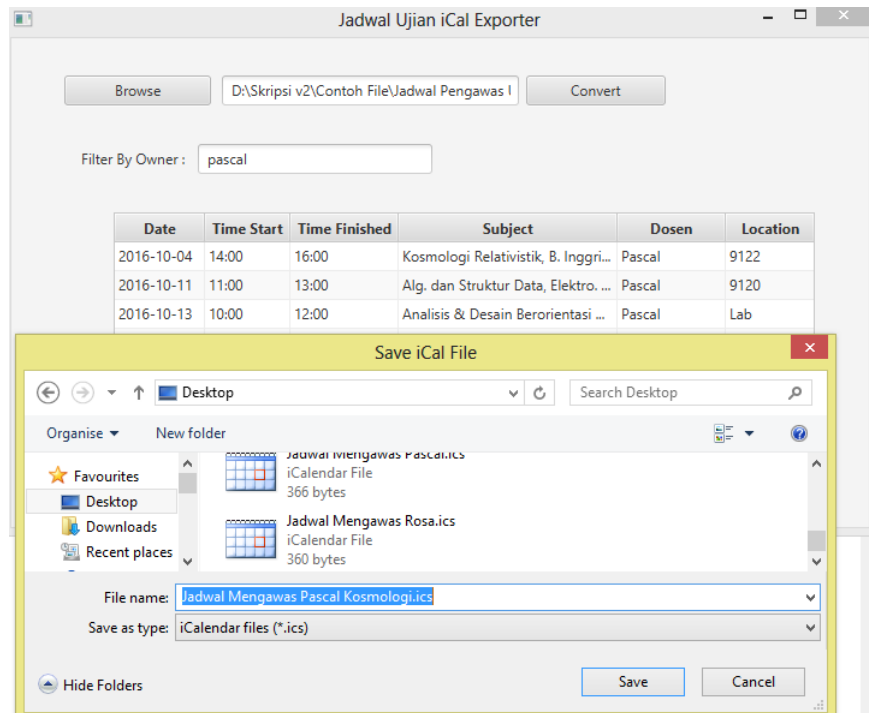
Gambar 5.46: PL mengkonversi jadwal ke format iCal

```

BEGIN:VCALENDAR
PRODID:-//Ben Fortuna//iCal4j 1.0//EN
VERSION:2.0
CALSCALE:GREGORIAN
BEGIN:VEVENT
DTSTAMP:20161116T205843Z
DTSTART:20161003T080043
DTEND:20161003T100043
SUMMARY:Matematika Diskret\, Elektrodinamika
LOCATION:10316
DESCRIPTION:Mengawas Ujian Rosa
TZID:Asia/Jakarta
UID:20161116T205843Z-uidGen@AriqRahmaeri
END:VEVENT

```

Gambar 5.47: File iCal



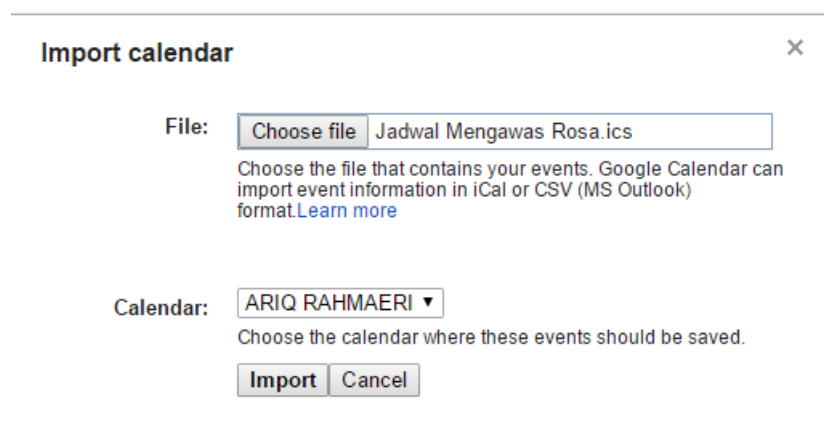
Gambar 5.48: Hasil pengujian convert hasil filter kedalam iCal

```

BEGIN:VCALENDAR
PRODID:-//Ben Fortuna//iCal4j 1.0//EN
VERSION:2.0
CALSCALE:GREGORIAN
BEGIN:VEVENT
DTSTAMP:20161116T211248Z
DTSTART:20161004T140048
DTEND:20161004T160048
SUMMARY:Kosmologi Relativistik\, B. Inggri...
LOCATION:9122
DESCRIPTION:Mengawas Ujian Pascal
TZID:Asia/Jakarta
UID:20161116T211248Z-uidGen@AriqRahmaeri
END:VEVENT

```

Gambar 5.49: File iCal Filter



Gambar 5.50: Hasil pengujian import kedalam Google Calendar



Subject	Matematika Diskret, Elektrodinamika		
Location	10316		
Start time	Sen 03/10/2016	8:01	<input type="checkbox"/> All day event
End time	Sen 03/10/2016	10:01	

Mengawas Ujian Rosa

Gambar 5.54: File hasil Konversi dapat dibuka di MS Outlook

◀ ▶ Oktober 2016

SENIN	SELASA
26 Sep	27
3 8:01 Matematika Diskret, Elektrodinamika; 10316	4
10	11
17	18

Gambar 5.55: File hasil Konversi berada pada kalender MS Outlook

Import calendar

×

File:  Jadwal Mengawas Pascal Kosmologi.ics

Choose the file that contains your events. Google Calendar can import event information in iCal or CSV (MS Outlook) format. [Learn more](#)

Calendar:

Choose the calendar where these events should be saved.

Gambar 5.56: Hasil pengujian import file yang di filter kedalam Google Calendar

<b>Tue Oct 4</b>	2:00pm – 4:00pm	<b>Kosmologi Relativistik, B. Inggris, Str. SI dan Ars. Perush. Berskala Bsr. - 9122</b>
		Where 9122 map
		Color
<b>Wed Oct 12</b>	10:00am – 11:00am	<b>Bimbingan Ariq - UNPAR</b>
<b>Wed Oct 19</b>	10:00am – 11:00am	<b>Bimbingan Ariq - UNPAR</b>

Gambar 5.57: Hasil import file yang di filter ke Google Calendar

**Kosmologi Relativistik, B. Inggris, Str. SI dan Ars. Perush. B**

10/4/2016 2:00pm to 4:00pm 10/4/2016 Time zone

☐ All day ☐ Repeat...

Event details Find a time

Where 9122 map

Video call Add video call

Calendar ARIQ RAHMAERI

Description Mengawas Ujian Pascal

Attachment Add attachment

Event color

Notifications Notification 10 minutes x Add a notification

Show me as ☐ Available ☒ Busy

Visibility ☒ Calendar default ☐ Public ☐ Private

By default this event will follow the sharing settings of this calendar: event details will be visible to anyone who can see details of other events in this calendar. [Learn more](#)

[Publish event](#)

Add: Guests | Rooms, etc.

Enter guest email address Add

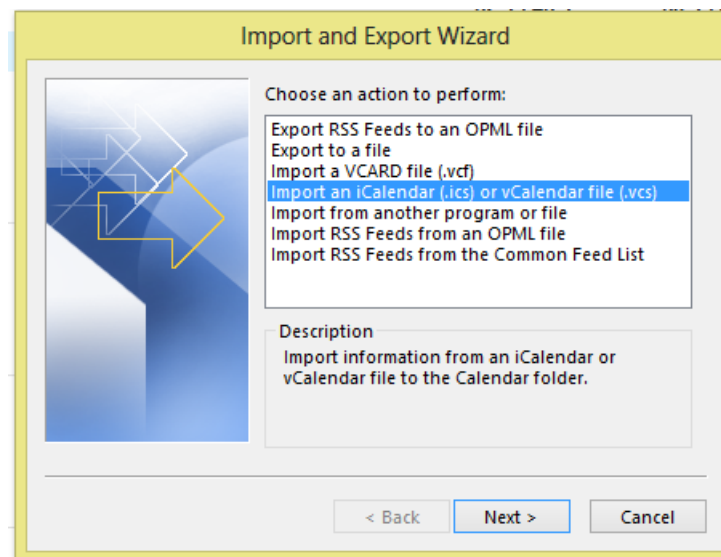
Guests can

☐ modify event

☒ invite others

☒ see guest list

Gambar 5.58: Hasil import file yang di filter ke Google Calendar bagian 2



Gambar 5.59: *Import* file iCal yang telah di filter kedalam MS Outlook

Subject	Matematika Diskret, Elektrodinamika		
Location	10316		
Start time	Sen 03/10/2016	8:01	<input type="checkbox"/> All day event
End time	Sen 03/10/2016	10:01	

Mengawas Ujian Rosa

Gambar 5.60: File hasil filter dapat dibuka di MS Outlook

◀ ▶ Oktober 2016

SENIN	SELASA
26 Sep	27
3	4
8:01 Matematika Diskret, Elektrodinamika; 10316	14:01 Kosmologi Relativistik, B. Inggris, S...
10	11

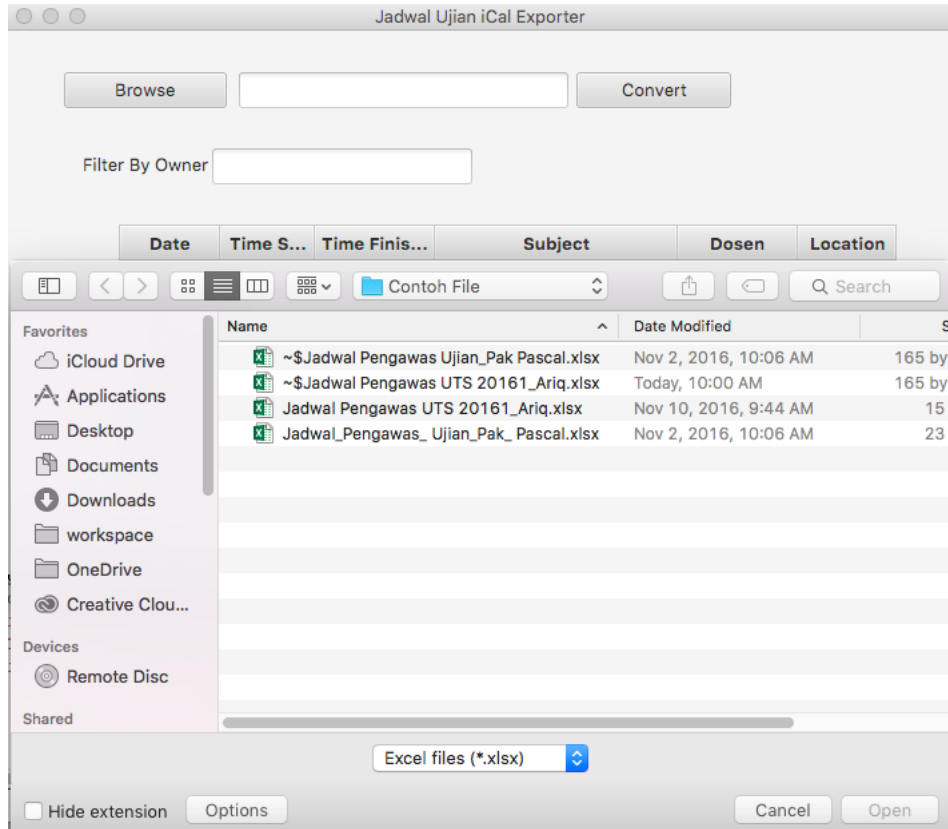
Gambar 5.61: File iCal yang telah di filter berada pada kalender MS Outlook

Selain menguji pada sistem operasi windows, PL juga diuji pada sistem operasi Mac OS. Pada kesempatan kali ini PL diuji kinerjanya dalam OS X. PL diuji pada Macbook Pa Pascal dengan skenario yang sama pada sistem operasi windows. Berikut hasil pengujiannya.

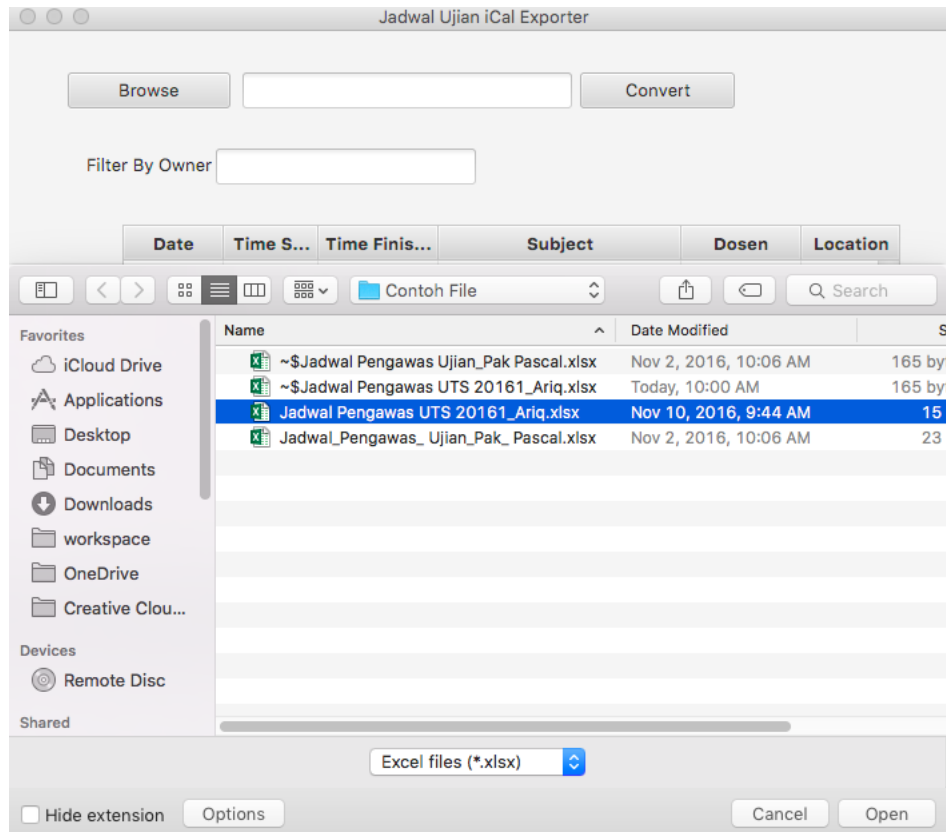
Tabel 5.4: Tabel hasil pengujian pada OS X

Hal yang diuji	Hasil yang diharapkan	Hasil
Browse file excel	PL dapat melakukan browse file excel	Berhasil 5.62 5.63
Menampilkan Jadwal ke layar	PL menampilkan ke layar file excel yang telah dibaca	Berhasil 5.64 5.65
Konversi ke iCal	PL dapat mengkonversi jadwal yang diseleksi pengguna kedalam iCalendar	Berhasil 5.66 5.67
Hasil Filter dapat di-konversi ke iCal	Hasil Filter pada PL dan diseleksi oleh pengguna, dapat di konversikan kedalam iCal	Berhasil 5.68 5.69
Import iCalendar Mac	Hasil konversi PL dapat di masukan kedalam iCalendar pada OS X	Berhasil 5.73 5.74
Hasil filter dapat di <i>import</i> Google Calendar	Hasil filter konversi PL dapat di masukan kedalam Google Calendar	Berhasil 5.76 5.77

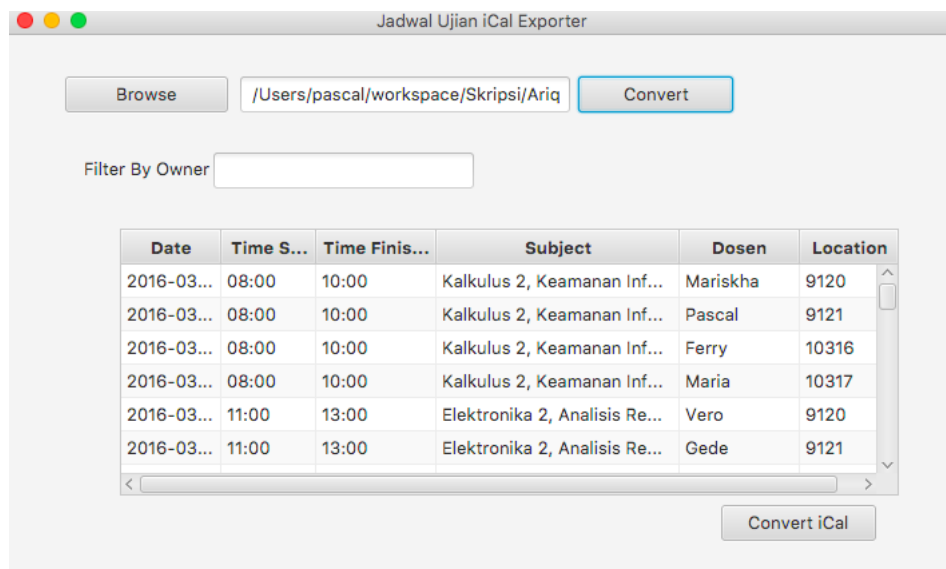
Berikut tampilan dari hasil pengujian PL pada OS X :

Gambar 5.62: Hasil pengujian *browse* pada OS X dengan *file input* data lama

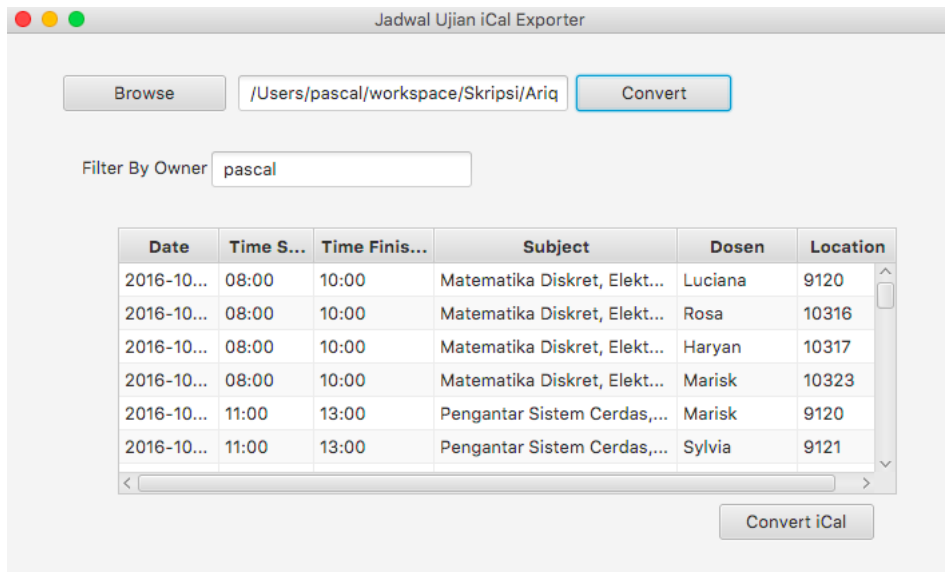




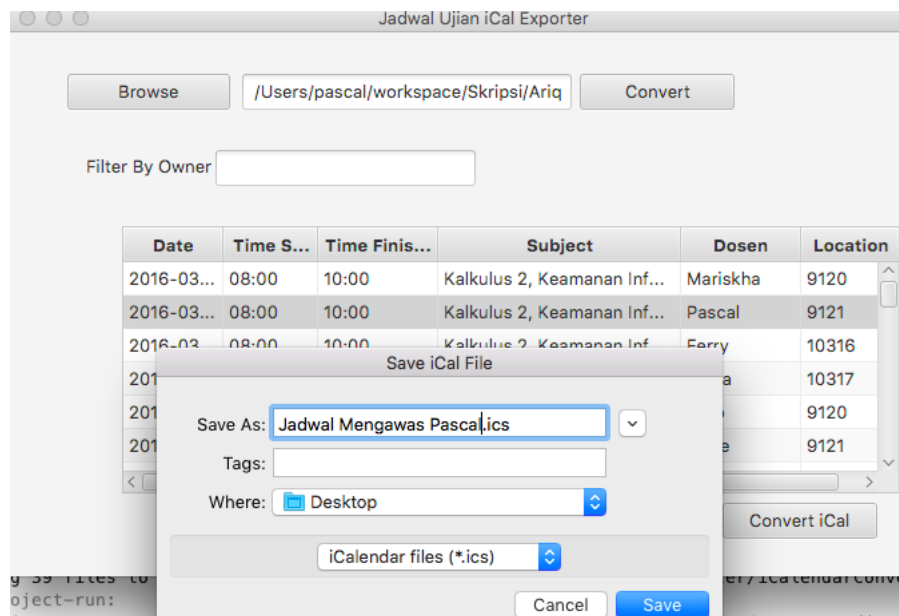
Gambar 5.63: Hasil pengujian *browse* pada OS X dengan *file input* data baru



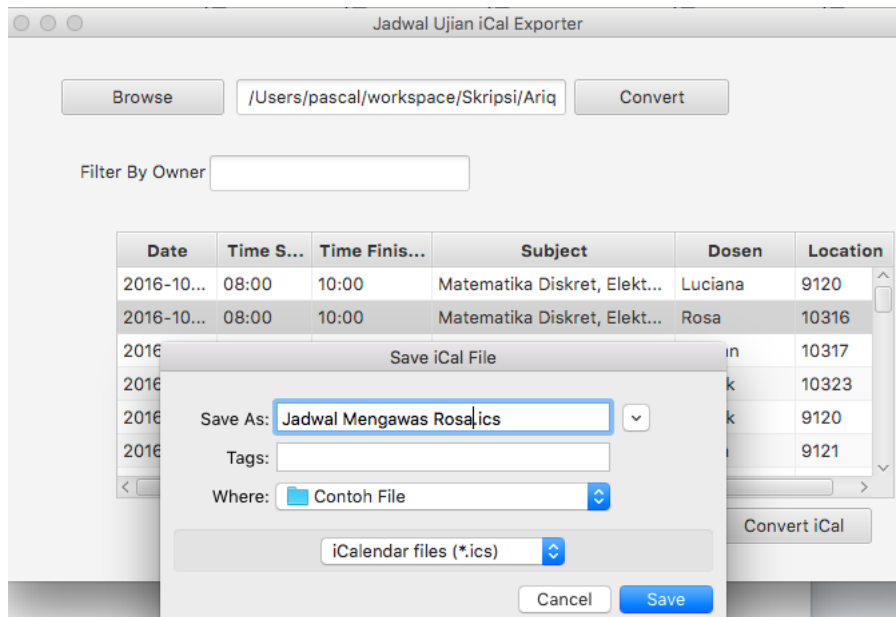
Gambar 5.64: Hasil pengujian menampilkan ke layar pada OS X dengan *file input* data lama



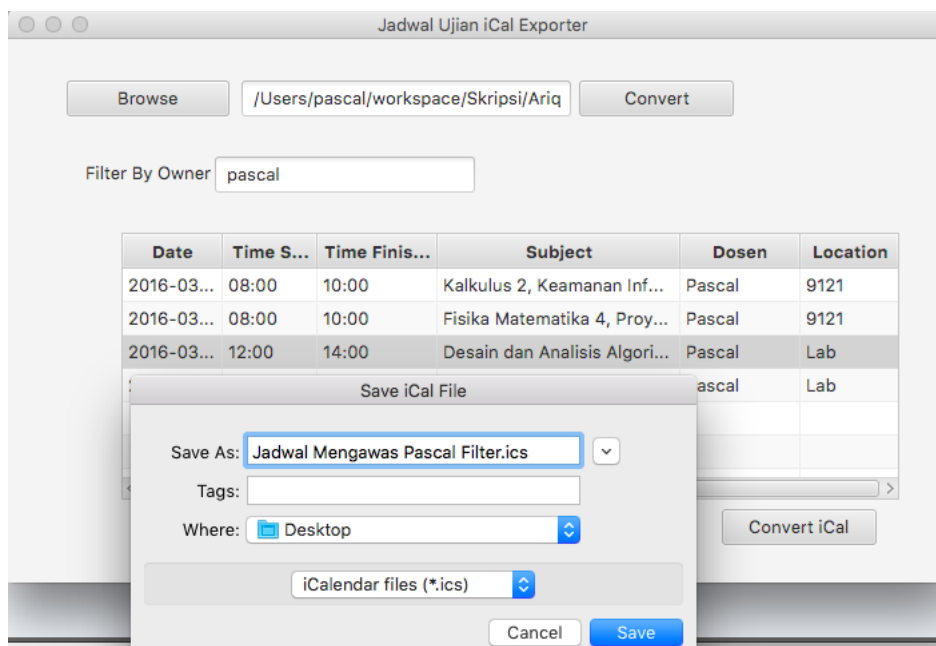
Gambar 5.65: Hasil pengujian menampilkan ke layar pada OS X dengan *file input* data baru



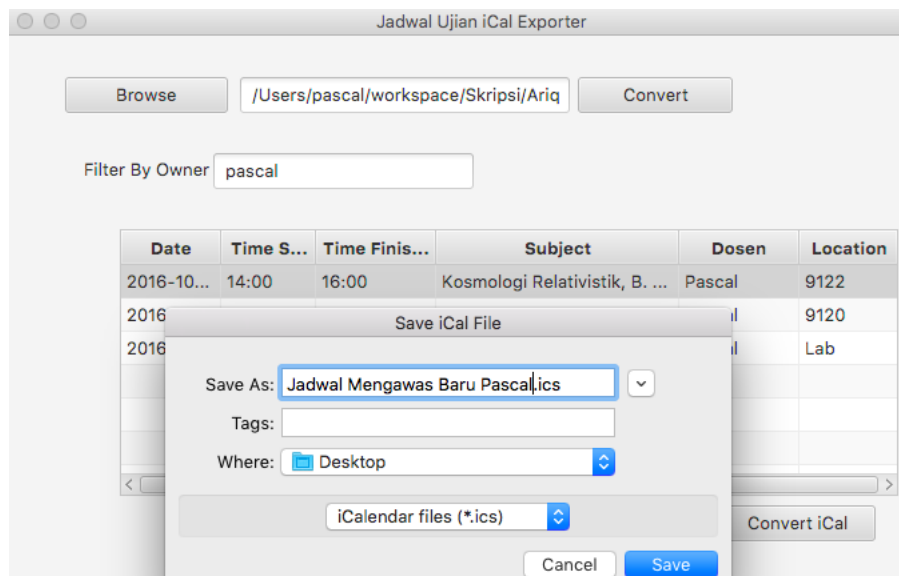
Gambar 5.66: Hasil pengujian menyimpan file iCal pada OS X dengan *file input* data lama



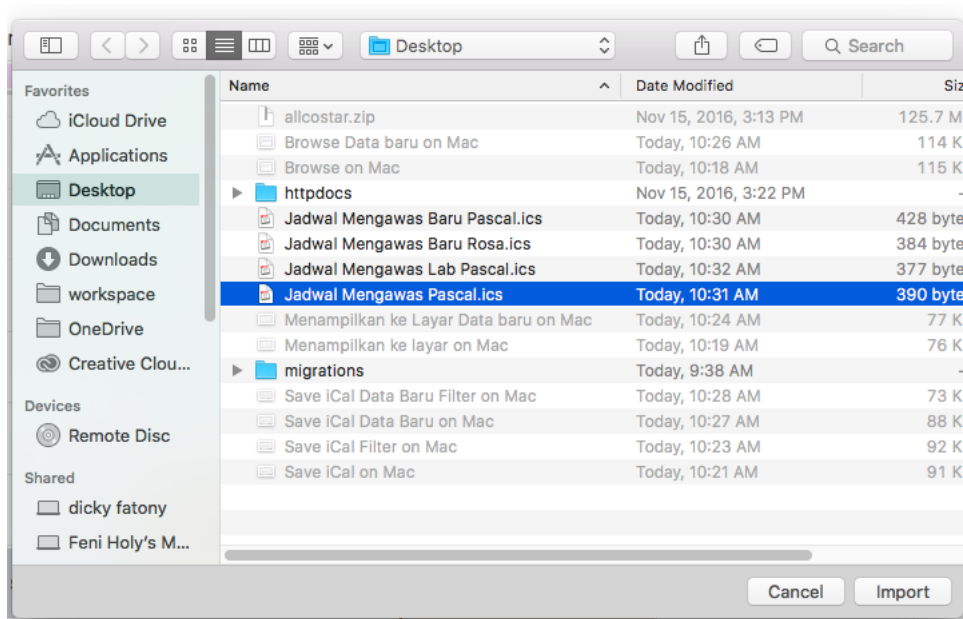
Gambar 5.67: Hasil pengujian menyimpan file iCal pada OS X dengan *file input* data baru



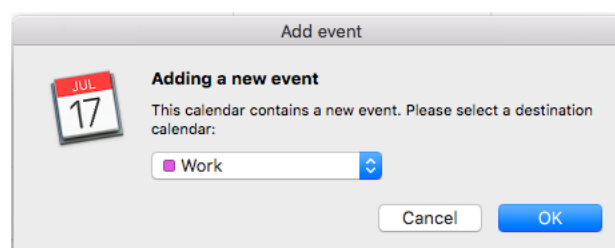
Gambar 5.68: Hasil pengujian menyimpan file iCal yang telah difilter pada OS X dengan *file input* data lama



Gambar 5.69: Hasil pengujian menyimpan file iCal yang telah difilter pada OS X dengan *file input* data baru



Gambar 5.70: Hasil pengujian *import* iCalendar pada OS X dengan *file input* data lama



Gambar 5.71: Notifikasi destinasi kalender pada OS X dengan *file input* data lama

March 2016

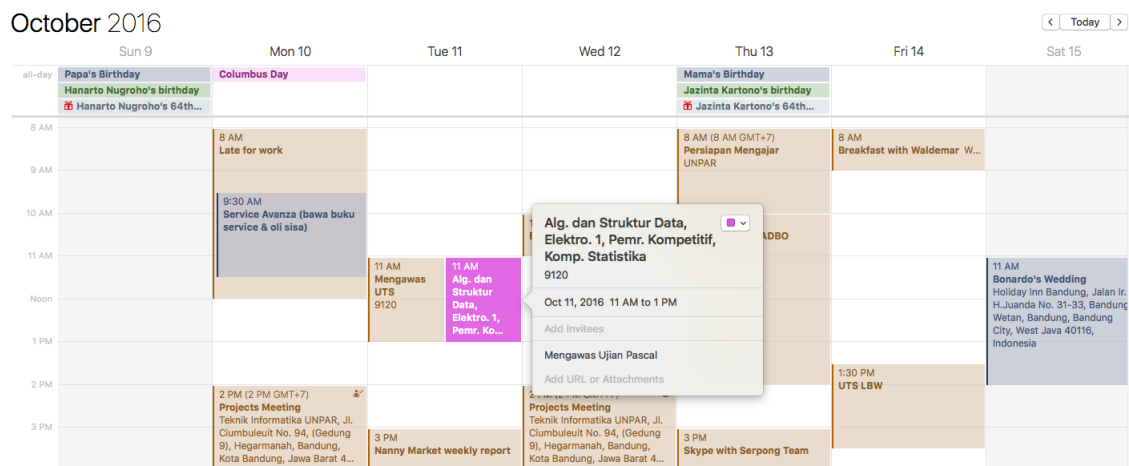
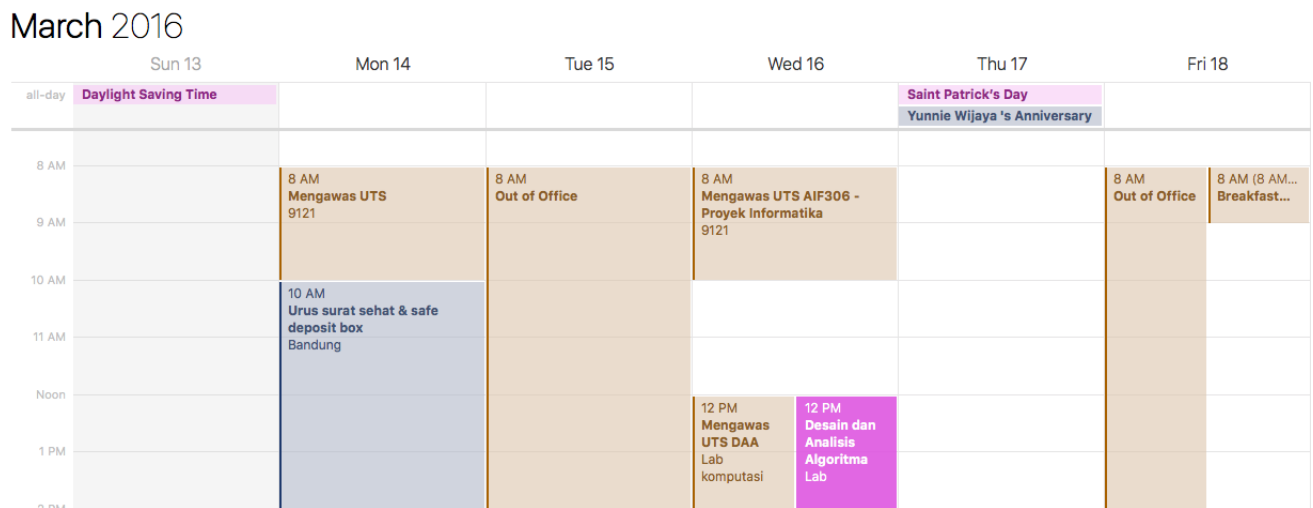
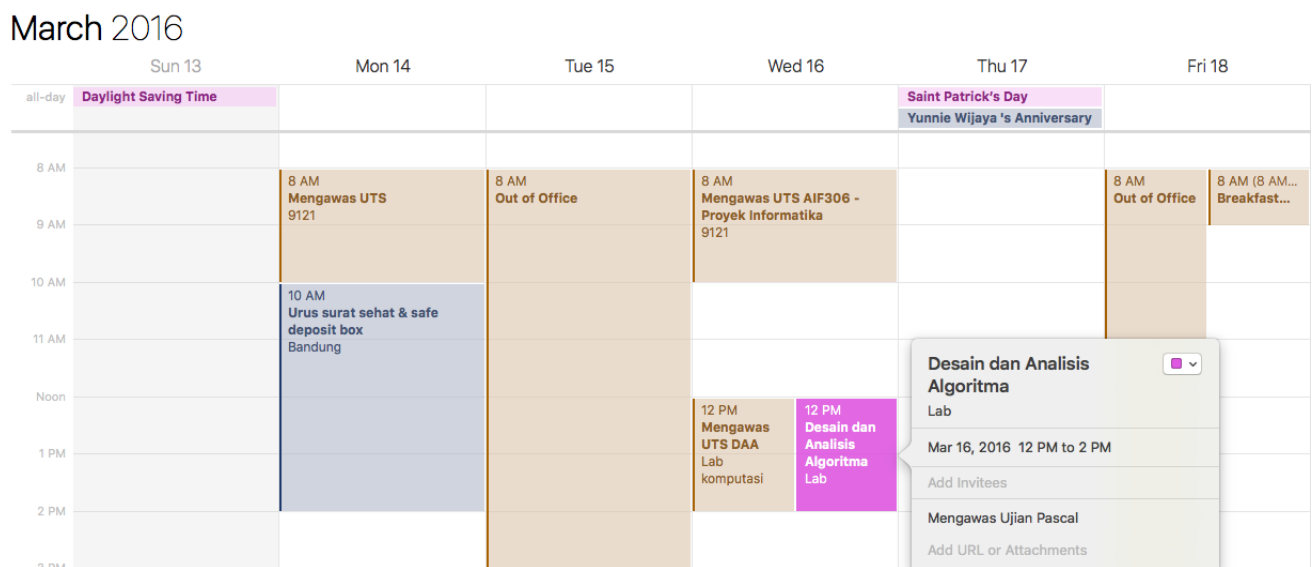
	Sun 13	Mon 14	Tue 15	Wed 16	Thu 17	Fri 18
all-day	Daylight Saving Time				Saint Patrick's Day Yunnie Wijaya 's Anniversary	
4 AM						
5 AM						
6 AM						
7 AM						
8 AM		8 AM Mengawas UTS 9121	8 AM Kalkulus 2, Keamanan Informasi, Reologi 9121	8 AM Out of Office	8 AM Mengawas UTS AIF306 - Proyek Informatika 9121	8 AM Out of Office
9 AM						8 AM (8 AM... Breakfast...
10 AM		10 AM Urus surat sehat & safe deposit box Bandung				
11 AM						
Noon						
1 PM				12 PM Mengawas UTS DAA Lab komputasi		

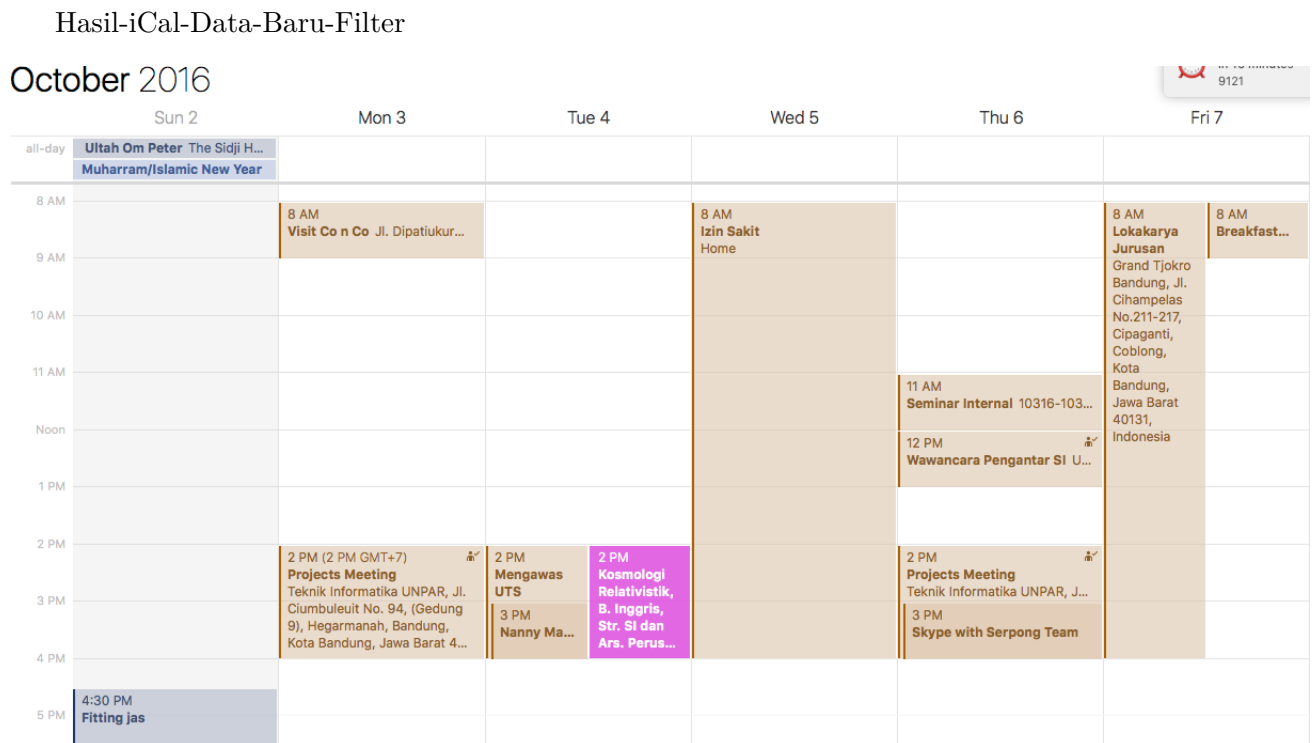
Gambar 5.72: Tampilan iCalendar pada OS X dengan *file input* data lama

March 2016

	Sun 13	Mon 14	Tue 15	Wed 16	Thu 17	Fri 18
all-day	Daylight Saving Time				Saint Patrick's Day Yunnie Wijaya 's Anniversary	
4 AM						
5 AM						
6 AM						
7 AM						
8 AM		8 AM Mengawas UTS 9121	8 AM Kalkulus 2, Keamanan Informasi, Reologi 9121	8 AM Out of Office	8 AM Out of Office	8 AM (8 AM... Breakfast...
9 AM						
10 AM		10 AM Urus surat sehat & safe deposit box Bandung				
11 AM						
Noon						
1 PM				12 PM Mengawas UTS DAA Lab komputasi		

Gambar 5.73: Tampilan iCalendar bagian 2 pada OS X dengan *file input* data lama

Gambar 5.74: Tampilan iCalendar pada OS X dengan *file input* data baruGambar 5.75: Tampilan hasil filter pada OS X dengan *file input* data lamaGambar 5.76: Tampilan hasil filter bagian 2 pada OS X dengan *file input* data lama



Gambar 5.77: Tampilan hasil filter pada OS X dengan *file input* data baru

### 5.3.3 Kesimpulan Pengujian

Berikut ini merupakan kesimpulan dari hasil pengujian fungsional maupun eksperimental.

#### 1. Pengujian Fungsional

- Pengguna dapat melakukan *browse* untuk mencari file excel.
- Alamat file excel muncul di kolom alamat pada PL.
- PL dapat membaca file excel dan menampilkannya ke layar.
- Pengguna dapat melakukan *filter* nama pengawas.
- Pengguna dapat memilih jadwal pada tabel dan mengkonversikannya kedalam file iCalendar.
- Pengguna dapat memasukan jadwal ke Goggle Calendar dan Microsoft Outlook.

#### 2. Pengujian Eksperimental

- File selain excel tidak dapat dibaca oleh PL.
- Sheet* yang dapat dibaca oleh PL adalah *Sheet* pertama.
- PL dapat mengkonversi file excel jadwal baru.
- Hasil konversi PL dapat dimasukan kedalam Apple Calendar.





## BAB 6

### KESIMPULAN

#### 6.0.4 Kesimpulan

Kesimpulan dari penelitian ini adalah PL dapat membaca file excel mengawas ujian yang diterbitkan oleh TU FTIS dengan menggunakan *library* Apache POI dan kelas internal ExcelConverter. Hasil dari kelas ExcelConverter adalah sebuah ArrayList jadwal yang diteruskan ke kelas FXMLDocumentController. Pada kelas FXMLDocumentController ArrayList tersebut dimasukan kedalam ObservableList dan ditampilkan ke layar menggunakan TableView. Pengguna dapat mengkonversi jadwal menjadi file iCal dengan cara memilih jadwal pada tabel, kemudian baris yang dipilih diteruskan ke kelas CalendarConverter. Pada kelas CalendarConverter, potongan baris yang dipilih pengguna yang merupakan kelas ScheduleClass, dikonversikan menjadi iCal menggunakan *library* iCal4j. Hasil dari kelas CalendarConverter merupakan file berekstensi .ics yang dapat dimasukan ke dalam Google Calendar, Microsoft Outlook, dan Apple Calendar.

#### 6.0.5 Saran

Pengembangan selanjutnya dari skripsi ini dapat di arahkan untuk membuat SI(Sistem Informasi) agar file jadwal mengawas konsisten. Selain itu, pengembangan SI dapat mengotomatisasi penjadwalan tugas mengajar dosen, sehingga kedepannya jadwal mengawas dosen akan dikirimkan melalui email dengan file iCalendar.



## DAFTAR REFERENSI

- [1] Wikipedia (2016) Microsoft excel. [https://en.wikipedia.org/wiki/Microsoft\\_Excel](https://en.wikipedia.org/wiki/Microsoft_Excel). 21 Desember 2016.
- [2] Wikipedia (2016) icalendar. <https://en.wikipedia.org/wiki/ICalendar>. 22 Desember 2016.
- [3] The Apache Software Foundation (2016) Apache POI - the Java API for Microsoft Documents. <https://poi.apache.org/>. [Online; diakses 23-09-2016].
- [4] The Apache Software Foundation (2016) POI API Documentation. <http://poi.apache.org/apidocs/index.html>. [Online; diakses 23-09-2016].
- [5] Ben Fortune (2007) ical4j. <http://ical4j.sourceforge.net/introduction.html>. [Online; diakses 14-09-2016].
- [6] Ben Fortune (2016) cal4j/ical4j: A Java library for parsing and building iCalendar data models. <http://ical4j.github.io/docs/ical4j/api/2.0-beta1>. [Online; diakses 14-09-2016].
- [7] Oracle and/or its affiliates (2013) What is javafx? <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>. [Online; diakses 28-09-2016].
- [8] Oracle and/or its affiliates (2013) JavaFX Architecture. <http://docs.oracle.com/javafx/2/architecture/jfxpub-architecture.htm>. [Online; diakses 28-09-2016].
- [9] Oracle and/or its affiliates (2015) Javafx 8. <http://docs.oracle.com/javase/8/javafx/api/toc.htm>. [Online; diakses 28-09-2016].
- [10] Oracle and/or its affiliates (2016) Java se at a glance. <http://www.oracle.com/technetwork/java/javase/overview/index.html>. 17 November 2016.



# LAMPIRAN A

## KODE PROGRAM

Listing A.1: ScheduleClass.java

```
1
2 import java.time.LocalDate;
3 import java.time.LocalTime;
4 import javafx.beans.property.SimpleStringProperty;
5 import javafx.beans.property.StringProperty;
6
7 /**
8  *
9  * @author Ariq
10 */
11 public class ScheduleClass {
12     private LocalDate date;
13     private LocalTime timeAwal;
14     private LocalTime timeAkhir;
15     private StringProperty subject;
16     private StringProperty dosen;
17     private StringProperty location;
18
19     public ScheduleClass (LocalDate date, LocalTime timeAwal
20         ,LocalTime timeAkhir, String subject, String dosen
21         , String location)
22     {
23         this.date = date;
24         this.timeAwal = timeAwal;
25         this.timeAkhir = timeAkhir;
26         this.subject = new SimpleStringProperty(subject);
27         this.dosen = new SimpleStringProperty(dosen);
28         this.location = new SimpleStringProperty(location);
29     }
30
31     /**
32     * @return the date
33     */
34     public LocalDate getDate() {
35         return date;
36     }
37
38     /**
39     * @param date the date to set
40     */
41     public void setDate(LocalDate date) {
42         this.date = date;
43     }
44
45     /**
46     * @return the time
47     */
48     public LocalTime getTimeAwal() {
49         return timeAwal;
50     }
51
52     /**
53     * @param time the time to set
54     */
55     public void setTimeAwal(LocalTime timeAwal) {
56         this.timeAwal = timeAwal;
57     }
58
59     /**
60     * @return the time
61     */
62     public LocalTime getTimeAkhir() {
63         return timeAkhir;
64     }
65
66     /**
67     * @param time the time to set
68     */
69     public void setTimeAkhir(LocalTime timeAkhir) {
70         this.timeAkhir = timeAkhir;
71     }
72
73     /**
```

```

74      * @return the subject
75      */
76      public String getSubject() {
77          return subject.get();
78      }
79
80      /**
81      * @param subject the subject to set
82      */
83      public void setSubject(String subject) {
84          this.subject.set(subject);
85      }
86
87      /**
88      * @return the dosen
89      */
90      public String getDosen() {
91          return dosen.get();
92      }
93
94      /**
95      * @param dosen the dosen to set
96      */
97      public void setDosen(String dosen) {
98          this.dosen.set(dosen);
99      }
100
101      /**
102      * @return the location
103      */
104      public String getLocation() {
105          return location.get();
106      }
107
108      /**
109      * @param location the location to set
110      */
111      public void setLocation(String location) {
112          this.location.set(location);
113      }
114
115      public StringProperty subjectProperty()
116      {
117          return subject;
118      }
119
120      public StringProperty dosenProperty()
121      {
122          return dosen;
123      }
124      public StringProperty location()
125      {
126          return location;
127      }
128
129  }

```

Listing A.2: ExcelConverter.java

```

1  package icalendarconverter;
2
3  import java.io.File;
4  import java.io.FileInputStream;
5  import java.io.FileNotFoundException;
6  import java.io.IOException;
7  import java.text.SimpleDateFormat;
8  import java.time.LocalDate;
9  import java.time.LocalDateTime;
10 import java.time.format.DateTimeFormatter;
11 import java.time.format.FormatStyle;
12 import java.util.ArrayList;
13 import java.util.Date;
14 import java.util.Iterator;
15 import java.util.List;
16 import java.util.Locale;
17 import org.apache.poi.ss.usermodel.Cell;
18 import org.apache.poi.ss.usermodel.FormulaEvaluator;
19 import org.apache.poi.ss.usermodel.Row;
20 import org.apache.poi.ss.util.CellRangeAddress;
21 import org.apache.poi.ss.util.CellReference;
22 import org.apache.poi.xssf.usermodel.XSSFRow;
23 import org.apache.poi.xssf.usermodel.XSSFSheet;
24 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
25
26 /**
27  *
28  * @author Ariq
29  */
30 public class ExcelConverter {
31
32     private File pathFile;
33     static XSSFRow row;
34     private int rowNoIdx;
35     private int colMatkulIdx;
36     private LocalDate lc;
37     private SimpleDateFormat sp;
38     private Date date;
39     private DateTimeFormatter indoFormatter;

```

```

40 | private LocalTime lt;
41 | private String subject;
42 |
43 | public ExcelConverter(File pathFile)
44 | {
45 |     this.pathFile = pathFile;
46 |     this.rowNoIdx = 0;
47 | }
48 |
49 | public List<ScheduleClass> Converter() throws FileNotFoundException, IOException
50 | {
51 |     ArrayList<ScheduleClass> scheduleList = new ArrayList<>();
52 |
53 |     FileInputStream fis = new FileInputStream(pathFile);
54 |
55 |     XSSFWorkbook wb = new XSSFWorkbook(fis);
56 |     XSSFSheet sheet = wb.getSheetAt(0);
57 |     Iterator< Row > rowIterator = sheet.iterator();
58 |
59 |     CellRangeAddress add;
60 |     int colNoIdx = 0;
61 |     ArrayList<String> dosen = new ArrayList<>();
62 |     ArrayList<Integer> idxDosen = new ArrayList<>();
63 |     ArrayList<Integer> colDosen = new ArrayList<>();
64 |     ArrayList<String> location = new ArrayList<>();
65 |     int idxNumber = 0;
66 |     ArrayList<Integer> locationIdx = new ArrayList<>();
67 |     outerloop :
68 |     for (int j = 0; j < sheet.getLastRowNum(); j++) {
69 |         row = sheet.getRow(j);
70 |         for (int f = 0; f < row.getLastCellNum(); f++) {
71 |             Cell cell = row.getCell(f);
72 |             if (cell.getCellType() == Cell.CELL_TYPE_STRING && cell.getStringCellValue().
73 |                 equalsIgnoreCase("No. ")) {
74 |                 rowNoIdx = j;
75 |                 colNoIdx = cell.getColumnIndex();
76 |             }
77 |             else if (cell.getCellType() == Cell.CELL_TYPE_STRING && cell.getStringCellValue().
78 |                 equalsIgnoreCase("Nama_Mata_Kuliah"))
79 |             {
80 |                 colMatkulIdx = cell.getColumnIndex();
81 |                 break outerloop;
82 |             }
83 |         }
84 |     }
85 |     outerloop2 :
86 |     for (int i = 0; i < sheet.getLastRowNum(); i++) {
87 |         outerloop :
88 |         for (int j = 0; j < row.getLastCellNum(); j++) {
89 |             row = sheet.getRow(i);
90 |             if (row == null)
91 |             {
92 |                 i = sheet.getLastRowNum();
93 |                 break outerloop2;
94 |             }
95 |             Cell cell = row.getCell(j);
96 |             FormulaEvaluator evaluator = wb.getCreationHelper().createFormulaEvaluator();
97 |             if (cell.getColumnIndex() == colNoIdx && i > rowNoIdx + 3 && cell.getCellType() != Cell.
98 |                 CELL_TYPE_BLANK
99 |                 && evaluator.evaluate(cell).getCellType() != Cell.CELL_TYPE_NUMERIC) {
100 |                 i = sheet.getLastRowNum();
101 |                 break outerloop2;
102 |             }
103 |             if (cell.getColumnIndex() == colNoIdx && i > rowNoIdx+3
104 |                 && cell.getCellType() == Cell.CELL_TYPE_BLANK )
105 |             {
106 |                 i = i + 1;
107 |                 break outerloop;
108 |             }
109 |             if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 1)) {
110 |                 if (cell.getCellType() == Cell.CELL_TYPE_BLANK)
111 |                 {
112 |                     i = i+1;
113 |                     break outerloop;
114 |                 }
115 |                 String delims = "[.,_]";
116 |                 String[] sumary = cell.getStringCellValue().split(delims);
117 |                 for (int l = 0; l < sumary.length; l++) {
118 |                     if (sumary[l].equalsIgnoreCase("Mrt")) {
119 |                         sumary[l] = "3";
120 |                     }
121 |                     if (sumary[l].equalsIgnoreCase("Okt")) {
122 |                         sumary[l] = "10";
123 |                     }
124 |                     if (sumary[l].equalsIgnoreCase("'16")) {
125 |                         sumary[l] = "2016";
126 |                     }
127 |                 }
128 |                 lc = LocalDate.of(Integer.parseInt(sumary[5]), Integer.parseInt(sumary[3]), Integer.
129 |                     parseInt(sumary[2]));
130 |             }
131 |             if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 2)) {
132 |                 if (cell.getStringCellValue().equalsIgnoreCase("LIBUR"))
133 |                 {
134 |                     i = i+1;
135 |                     break outerloop;

```

```

135     }
136     else
137     {
138         if (cell.getStringCellValue().equalsIgnoreCase("Shift_1")
139             || cell.getStringCellValue().equalsIgnoreCase("Shift_2"))
140         {
141             CellReference cr = new CellReference(cell.getRowIndex()+1, cell.getColumnIndex());
142             Row row2 = sheet.getRow(cr.getRow());
143             Cell c = row2.getCell(cr.getCol());
144             String delimsJam = "-";
145             String[] arrJam = c.getStringCellValue().split(delimsJam);
146             for (int k = 0; k < arrJam.length; k++) {
147                 arrJam[k] = arrJam[k].replace('.', ':');
148             }
149             lt = LocalTime.parse(arrJam[0]);
150         }
151         else
152         {
153             String delimsJam = "-";
154             String[] arrJam = cell.getStringCellValue().split(delimsJam);
155             for (int k = 0; k < arrJam.length; k++) {
156                 arrJam[k] = arrJam[k].replace('.', ':');
157             }
158             lt = LocalTime.parse(arrJam[0]);
159         }
160     }
161 }
162
163 }
164 if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == colMatkulIdx) {
165     subject = cell.getStringCellValue();
166 }
167
168 if (cell.getRowIndex() > rowNoIdx && cell.getColumnIndex() >= colMatkulIdx+1
169     && cell.getColumnIndex() < row.getLastCellNum()) {
170     if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC) {
171     }
172     if (cell.getCellType() == Cell.CELL_TYPE_STRING) {
173         if (cell.getStringCellValue().contains(":")) {
174             String[] spl1 = cell.getStringCellValue().split(":");
175             String[] spl2 = spl1[1].split(",");
176             for (int l = 0; l < spl2.length; l++) {
177                 dosen.add(spl2[l].trim());
178                 location.add("Lab");
179             }
180         }
181         else {
182             CellReference cr = new CellReference(1, cell.getColumnIndex());
183             Row row2 = sheet.getRow(cr.getRow());
184             Cell c = row2.getCell(cr.getCol());
185             if (!cell.getStringCellValue().isEmpty())
186             {
187                 dosen.add(cell.getStringCellValue().trim());
188                 location.add(String.valueOf((int) c.getNumericCellValue()).trim());
189             }
190         }
191     }
192 }
193 if (cell.getCellType() == Cell.CELL_TYPE_BLANK && cell.getRowIndex() > 2) {
194     CellReference cr = new CellReference(cell.getRowIndex() - 1, cell.getColumnIndex());
195     Row row2 = sheet.getRow(cr.getRow());
196     Cell c = row2.getCell(cr.getCol());
197     CellReference cr2 = new CellReference(1, cell.getColumnIndex());
198     Row row3 = sheet.getRow(cr2.getRow());
199     Cell c2 = row3.getCell(cr2.getCol());
200     if (c.getStringCellValue().contains(":")) {
201         String[] spl1 = c.getStringCellValue().split(":");
202         String[] spl2 = spl1[1].split(",");
203         for (int l = 0; l < spl2.length; l++) {
204             dosen.add(spl2[l].trim());
205             location.add("");
206         }
207     }
208     else {
209         if (!c.getStringCellValue().isEmpty())
210         {
211             dosen.add("");
212             location.add("");
213         }
214     }
215 }
216 }
217 }
218
219 for (int j = 0; j < dosen.size(); j++) {
220     scheduleList.add(new ScheduleClass(lc, lt, lt.plusHours(2), subject, dosen.get(j),
221         location.get(j)));
222     dosen.clear();
223     location.clear();
224 }
225
226 return Mergering(scheduleList);
227 }
228
229 public List<ScheduleClass> Mergering (ArrayList<ScheduleClass> scheduleList)
230

```



```

231 {
232     int count = 0;
233     ArrayList<ScheduleClass> scheduleListSmt = new ArrayList<>();
234
235     for (int i = 0; i < scheduleList.size(); i++) {
236
237         if (scheduleList.get(i).getDosen().isEmpty() )
238         {
239             scheduleListSmt.add(scheduleList.get(i));
240         }
241     }
242
243     for (int i = 0; i < scheduleListSmt.size(); i++) {
244         for (int j = 0; j < scheduleList.size(); j++) {
245             if (scheduleList.get(j).equals(scheduleListSmt.get(i)))
246             {
247                 scheduleList.remove(j);
248             }
249         }
250     }
251
252     for (int i = 0; i < scheduleList.size(); i++) {
253         outerloop :
254         for (int j = 0; j < scheduleListSmt.size(); j++) {
255             if (scheduleList.get(i).getDate().equals(scheduleListSmt.get(j).getDate())
256                 && scheduleList.get(i).getTimeAwal().equals(scheduleListSmt.get(j).getTimeAwal()))
257             {
258                 String ss = scheduleList.get(i).getSubject();
259                 scheduleList.get(i).setSubject(ss+"_"+scheduleListSmt.get(j).getSubject());
260                 j = j + 1;
261                 break outerloop;
262             }
263         }
264     }
265     return scheduleList;
266 }

```

Listing A.3: CalendarConverter.java

```

1 package icalendarconverter;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.net.SocketException;
8 import java.util.GregorianCalendar;
9 import java.util.List;
10 import net.fortuna.ical4j.data.CalendarOutputter;
11 import net.fortuna.ical4j.model.DateTime;
12 import net.fortuna.ical4j.model.Property;
13 import net.fortuna.ical4j.model.TimeZone;
14 import net.fortuna.ical4j.model.TimeZoneRegistry;
15 import net.fortuna.ical4j.model.TimeZoneRegistryFactory;
16 import net.fortuna.ical4j.model.ValidationException;
17 import net.fortuna.ical4j.model.component.VEvent;
18 import net.fortuna.ical4j.model.component.VTimeZone;
19 import net.fortuna.ical4j.model.property.CalScale;
20 import net.fortuna.ical4j.model.property.Description;
21 import net.fortuna.ical4j.model.property.Location;
22 import net.fortuna.ical4j.model.property.ProdId;
23 import net.fortuna.ical4j.model.property.Uid;
24 import net.fortuna.ical4j.model.property.Version;
25 import net.fortuna.ical4j.util.UidGenerator;
26
27 /**
28  *
29  * @author Ariq
30  */
31 public class CalendarConverter {
32
33     public CalendarConverter()
34     {
35
36     }
37
38     public void calConverter (String path, ScheduleClass sch) throws SocketException,
39         FileNotFoundException, IOException, ValidationException
40     {
41         //creating timezone
42         TimeZoneRegistry registry = TimeZoneRegistryFactory.getInstance().createRegistry();
43         TimeZone timezone = registry.getTimeZone("Asia/Jakarta");
44         VTimeZone tz = timezone.getVTimeZone();
45
46         //Start Date
47         java.util.Calendar startDate = new GregorianCalendar();
48         startDate.setTimeZone(timezone);
49         startDate.set(java.util.Calendar.MONTH, sch.getDate().getMonthValue()-1);
50         startDate.set(java.util.Calendar.DAY_OF_MONTH, sch.getDate().getDayOfMonth());
51         startDate.set(java.util.Calendar.YEAR, sch.getDate().getYear());
52         startDate.set(java.util.Calendar.HOUR_OF_DAY, sch.getTimeAwal().getHour());
53         startDate.set(java.util.Calendar.MINUTE, sch.getTimeAwal().getMinute());
54
55         //EndDate
56         java.util.Calendar endDate = new GregorianCalendar();
57         endDate.setTimeZone(timezone);
58         endDate.set(java.util.Calendar.MONTH, sch.getDate().getMonthValue()-1);
59         endDate.set(java.util.Calendar.DAY_OF_MONTH, sch.getDate().getDayOfMonth());

```

```

59     endDate.set(java.util.Calendar.YEAR, sch.getDate().getYear());
60     endDate.set(java.util.Calendar.HOUR_OF_DAY, sch.getTimeAkhir().getHour());
61     endDate.set(java.util.Calendar.MINUTE, sch.getTimeAkhir().getMinute());
62
63
64     //creating an event
65     String eventName = sch.getSubject();
66     String location2 = sch.getLocation();
67     String desc = "Mengawas Ujian "+sch.getDosen();
68     DateTime start = new DateTime(startDate.getTime());
69     DateTime end = new DateTime(endDate.getTime());
70     VEvent mengawas = new VEvent(start, end, eventName);
71     mengawas.getProperties().add(new Location(location2));
72     mengawas.getProperties().add(new Description());
73
74     try {
75         mengawas.getProperties().getProperty(Property.DESRIPTION).setValue(desc);
76     } catch (Exception e) {
77     }
78
79     //add timezone info
80     mengawas.getProperties().add(tz.getTimeZoneId());
81
82     //generate unique identifier
83     UidGenerator uidgenerator = new UidGenerator("uidGen");
84     Uid uid = uidgenerator.generateUid();
85     mengawas.getProperties().add(uid);
86
87
88     //creating calendar
89     net.fortuna.ical4j.model.Calendar calendar = new net.fortuna.ical4j.model.Calendar();
90     calendar.getProperties().add(new ProdId("-//Ben Fortuna//iCal4j 1.0//EN"));
91     calendar.getProperties().add(Version.VERSION_2_0);
92     calendar.getProperties().add(CalScale.GREGORIAN);
93
94     // Add the event and print
95     calendar.getComponents().add(mengawas);
96     System.out.println(calendar);
97
98     //saving iCal
99     String calFile = sch.getSubject();
100
101     FileOutputStream fout = new FileOutputStream(path);
102
103     CalendarOutputter outputter = new CalendarOutputter();
104     outputter.setValidating(false);
105     outputter.output(calendar, fout);
106 }
107 }

```

Listing A.4: FXMLDocumentController.java

```

1 package icalendarconverter;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.net.SocketException;
7 import java.net.URL;
8 import java.util.ArrayList;
9 import java.util.ResourceBundle;
10 import javafx.beans.value.ChangeListener;
11 import javafx.beans.value.ObservableValue;
12 import javafx.collections.FXCollections;
13 import javafx.collections.ListChangeListener;
14 import javafx.collections.ObservableList;
15 import javafx.event.ActionEvent;
16 import javafx.fxml.FXML;
17 import javafx.fxml.Initializable;
18 import javafx.scene.control.Alert;
19 import javafx.scene.control.Alert.AlertType;
20 import javafx.scene.control.Label;
21 import javafx.scene.control.TextField;
22 import javafx.stage.FileChooser;
23 import javafx.scene.control.TableColumn;
24 import javafx.scene.control.TableView;
25 import javafx.scene.control.cell.PropertyValueFactory;
26 import javafx.util.Callback;
27 import javax.swing.JFileChooser;
28 import net.fortuna.ical4j.model.ValidationException;
29 /**
30  *
31  * @author Ariq
32  */
33 public class FXMLDocumentController implements Initializable {
34     private File selectedFile;
35
36     @FXML
37     private Label label;
38     @FXML
39     private TextField txtFile;
40     @FXML
41     private TextField filterTxt;
42     @FXML
43     private TableView<ScheduleClass> jadwalTable;
44
45     ObservableList<ScheduleClass> jadwalList;
46     ObservableList<ScheduleClass> filteredData = FXCollections.observableArrayList();

```

```

47 |
48 | @FXML
49 | private void handleButtonAction(ActionEvent event) {
50 |
51 |     FileChooser fileChooser = new FileChooser();
52 |     fileChooser.setTitle("Open Resource File");
53 |
54 |     //Set extension filter
55 |     FileChooser.ExtensionFilter extFilter = new FileChooser.ExtensionFilter("Excel files (*.xlsx)",
56 |                                     "*.xlsx");
57 |     FileChooser.ExtensionFilter extFilter2 = new FileChooser.ExtensionFilter("Excel files (*.xls)",
58 |                                     "*.xls");
59 |     fileChooser.getExtensionFilters().add(extFilter);
60 |     fileChooser.getExtensionFilters().add(extFilter2);
61 |     selectedFile = fileChooser.showOpenDialog(null);
62 |
63 |     if (selectedFile != null)
64 |     {
65 |         txtFile.setText(selectedFile.getAbsolutePath());
66 |     }
67 |     else
68 |     {
69 |     }
70 | }
71 |
72 | @FXML
73 | private void handleConvertAction(ActionEvent event) throws FileNotFoundException, IOException
74 | {
75 |     ExcelConverter con = new ExcelConverter(selectedFile);
76 |     jadwalList = FXCollections.observableArrayList(con.Converter());
77 |
78 |     jadwalTable.setItems(jadwalList);
79 |     jadwalTable.getColumns().get(0).setCellValueFactory(new PropertyValueFactory("Date"));
80 |     jadwalTable.getColumns().get(1).setCellValueFactory(new PropertyValueFactory("<?>(\"timeAwal\")");
81 |     jadwalTable.getColumns().get(2).setCellValueFactory(new PropertyValueFactory("<?>(\"timeAkhir\")");
82 |     jadwalTable.getColumns().get(3).setCellValueFactory(new PropertyValueFactory("Subject"));
83 |     jadwalTable.getColumns().get(4).setCellValueFactory(new PropertyValueFactory("Dosen"));
84 |     jadwalTable.getColumns().get(5).setCellValueFactory(new PropertyValueFactory("Location"));
85 |
86 |     filteredData.addAll(jadwalList);
87 |
88 |     jadwalList.addListener( new ListChangeListener<ScheduleClass>()
89 |     {
90 |         @Override
91 |         public void onChanged(ListChangeListener.Change<? extends ScheduleClass> change)
92 |         {
93 |             updateFilteredData();
94 |         }
95 |     });
96 | }
97 |
98 |
99 | @Override
100 | public void initialize(URL url, ResourceBundle rb) {
101 |
102 | }
103 |
104 |
105 | public void convertClicked() throws FileNotFoundException,
106 |     IOException, SocketException, ValidationException
107 | {
108 |     ScheduleClass selected = jadwalTable.getSelectionModel().getSelectedItem();
109 |     FileChooser fileChooser = new FileChooser();
110 |     fileChooser.setTitle("Save iCal File");
111 |     FileChooser.ExtensionFilter extFilter = new FileChooser.ExtensionFilter("iCalendar files (*.ics)",
112 |                                     "*.ics");
113 |     fileChooser.getExtensionFilters().add(extFilter);
114 |     File save = fileChooser.showSaveDialog(null);
115 |
116 |     int idx = jadwalTable.getSelectionModel().getSelectedIndex();
117 |     String path;
118 |     if(save != null)
119 |     {
120 |         path = save.getAbsolutePath();
121 |         CalendarConverter cc = new CalendarConverter();
122 |         cc.calConverter(path, selected);
123 |     }
124 |     else
125 |     {
126 |         System.out.println("Canceled!");
127 |     }
128 | }
129 | @FXML
130 | private void filterConversion()
131 | {
132 |     jadwalTable.setItems(filteredData);
133 |     filterTxt.textProperty().addListener(new ChangeListener<String>()
134 |     {
135 |         @Override
136 |         public void changed(ObservableValue<? extends String> observable,
137 |                             String oldValue, String newValue)
138 |         {
139 |             updateFilteredData();
140 |         }
141 |     });
142 | }

```

```

143 private void updateFilteredData()
144 {
145     filteredData.clear();
146     for (ScheduleClass sc : jadwalList)
147     {
148         if (matchesFilter(sc))
149         {
150             filteredData.add(sc);
151         }
152     }
153     reapplyTableSortOrder();
154 }
155
156 private boolean matchesFilter(ScheduleClass sc)
157 {
158     String filterString = filterTxt.getText();
159
160     if (filterString == null || filterString.isEmpty())
161     {
162         return true;
163     }
164     String lowerCaseFilterString = filterString.toLowerCase();
165
166     if (sc.getDosen().toLowerCase().indexOf(lowerCaseFilterString) != -1)
167     {
168         return true;
169     }
170     return false;
171 }
172
173 private void reapplyTableSortOrder()
174 {
175     ArrayList<TableColumn<ScheduleClass, ? >> sortOrder = new ArrayList<>(jadwalTable.getSortOrder());
176     jadwalTable.getSortOrder().clear();
177     jadwalTable.getSortOrder().addAll(sortOrder);
178 }
179 }

```

Listing A.5: ExcelConverter.java sebelum revisi

```

1
2 import java.io.File;
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.text.SimpleDateFormat;
7 import java.time.LocalDate;
8 import java.time.LocalDateTime;
9 import java.time.format.DateTimeFormatter;
10 import java.util.ArrayList;
11 import java.util.Date;
12 import java.util.Iterator;
13 import java.util.List;
14 import org.apache.poi.ss.usermodel.Cell;
15 import org.apache.poi.ss.usermodel.FormulaEvaluator;
16 import org.apache.poi.ss.usermodel.Row;
17 import org.apache.poi.ss.util.CellRangeAddress;
18 import org.apache.poi.ss.util.CellReference;
19 import org.apache.poi.xssf.usermodel.XSSFRow;
20 import org.apache.poi.xssf.usermodel.XSSFSheet;
21 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
22
23 /**
24  *
25  * @author Ariq
26  */
27 public class ExcelConverter {
28
29     private File pathFile;
30     static XSSFRow row;
31     private int rowNoIdx;
32     private LocalDate lc;
33     private SimpleDateFormat sp;
34     private Date date;
35     private DateTimeFormatter indoFormatter;
36     private LocalDateTime lt;
37     private String subject;
38
39     public ExcelConverter(File pathFile)
40     {
41         this.pathFile = pathFile;
42         this.rowNoIdx = 0;
43     }
44
45     public List<ScheduleClass> Converter() throws FileNotFoundException
46         , IOException
47     {
48         ArrayList<ScheduleClass> scheduleList = new ArrayList<>();
49
50         FileInputStream fis = new FileInputStream(pathFile);
51
52         XSSFWorkbook wb = new XSSFWorkbook(fis);
53         XSSFSheet sheet = wb.getSheetAt(0);
54         Iterator<Row> rowIterator = sheet.iterator();
55
56         CellRangeAddress add;
57         int colNoIdx = 0;
58         ArrayList<String> dosen = new ArrayList<>();

```

---

```

59 | ArrayList<Integer> idxDosen = new ArrayList<>();
60 | ArrayList<Integer> colDosen = new ArrayList<>();
61 | ArrayList<String> location = new ArrayList<>();
62 | int idxNumber = 0;
63 | ArrayList<Integer> locationIdx = new ArrayList<>();
64 | outerloop :
65 | for (int j = 0; j < sheet.getLastRowNum(); j++) {
66 |     row = sheet.getRow(j);
67 |     for (int f = 0; f < row.getLastCellNum(); f++) {
68 |         Cell cell = row.getCell(j);
69 |         if (cell.getStringCellValue().contains("No. ")) {
70 |             rowNoIdx = j;
71 |             colNoIdx = cell.getColumnIndex();
72 |
73 |             break outerloop;
74 |         }
75 |     }
76 | }
77 | outerloop2 :
78 | for (int i = 0; i < sheet.getLastRowNum(); i++) {
79 |     row = sheet.getRow(i);
80 |     outerloop :
81 |     for (int j = 0; j < row.getLastCellNum(); j++) {
82 |         Cell cell = row.getCell(j);
83 |         FormulaEvaluator evaluator = wb.getCreationHelper().createFormulaEvaluator();
84 |         if (cell.getColumnIndex() == colNoIdx && i > rowNoIdx+3
85 |             && evaluator.evaluate(cell).getCellType() != Cell.CELL_TYPE_NUMERIC) {
86 |             i = sheet.getLastRowNum();
87 |             break outerloop2;
88 |         }
89 |
90 |         if (cell.getRowIndex() > rowNoIdx+1
91 |             && cell.getColumnIndex() == (colNoIdx + 1)) {
92 |             String delims = "[.,_]";
93 |             String[] summary = cell.getStringCellValue().split(delims);
94 |             for (int l = 0; l < summary.length; l++) {
95 |                 if (summary[l].equalsIgnoreCase("Mrt")) {
96 |                     summary[l] = "3";
97 |                 }
98 |             }
99 |
100 |             lc = LocalDate.of(Integer.parseInt(summary[5])
101 |                               , Integer.parseInt(summary[3]) , Integer.parseInt(summary[2]));
102 |         }
103 |         if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 2)) {
104 |             if (cell.getStringCellValue().equalsIgnoreCase("LIBUR"))
105 |             {
106 |                 i = i+1;
107 |                 break outerloop;
108 |             }
109 |             else
110 |             {
111 |                 String delimsJam = "[-]";
112 |                 String[] arrJam = cell.getStringCellValue().split(delimsJam);
113 |                 for (int k = 0; k < arrJam.length; k++) {
114 |                     arrJam[k] = arrJam[k].replace('.', ':');
115 |                 }
116 |                 lt = LocalTime.parse(arrJam[0]);
117 |             }
118 |
119 |         }
120 |         if (cell.getRowIndex() > rowNoIdx+1 && cell.getColumnIndex() == (colNoIdx + 5)) {
121 |             subject = cell.getStringCellValue();
122 |         }
123 |
124 |         if (cell.getRowIndex() > rowNoIdx
125 |             && cell.getColumnIndex() >= colNoIdx+6
126 |             && cell.getColumnIndex() < row.getLastCellNum()) {
127 |             if (cell.getCellType() == Cell.CELL_TYPE_NUMERIC) {
128 |             }
129 |             if (cell.getCellType() == Cell.CELL_TYPE_STRING) {
130 |                 if (cell.getStringCellValue().contains(":")) {
131 |                     String[] splt = cell.getStringCellValue().split(":");
132 |                     String[] splt2 = splt[1].split(",");
133 |                     for (int l = 0; l < splt2.length; l++) {
134 |                         dosen.add(splt2[l].trim());
135 |                         location.add("Lab");
136 |                     }
137 |                 } else {
138 |                     CellReference cr = new CellReference(1, cell.getColumnIndex());
139 |                     Row row2 = sheet.getRow(cr.getRow());
140 |                     Cell c = row2.getCell(cr.getCol());
141 |                     if (!cell.getStringCellValue().isEmpty())
142 |                     {
143 |                         dosen.add(cell.getStringCellValue().trim());
144 |                         location.add(String.valueOf((int) c.getNumericCellValue()).trim());
145 |                     }
146 |                 }
147 |             }
148 |
149 |             if (cell.getCellType() == Cell.CELL_TYPE_BLANK && cell.getRowIndex() > 2) {
150 |                 CellReference cr = new CellReference(cell.getRowIndex() - 1
151 |                                                         , cell.getColumnIndex());
152 |                 Row row2 = sheet.getRow(cr.getRow());
153 |                 Cell c = row2.getCell(cr.getCol());
154 |                 CellReference cr2 = new CellReference(1, cell.getColumnIndex());
155 |                 Row row3 = sheet.getRow(cr2.getRow());
156 |                 Cell c2 = row3.getCell(cr2.getCol());
157 |                 if (c.getStringCellValue().contains(":")) {

```

```

158         String[] spl1 = c.getStringCellValue().split(":");
159         String[] spl2 = spl1[1].split(",");
160         for (int l = 0; l < spl2.length; l++) {
161             dosen.add("".trim());
162             location.add("");
163         }
164     } else {
165         if (!c.getStringCellValue().isEmpty())
166         {
167             dosen.add("");
168             location.add("");
169         }
170     }
171 }
172 }
173 }
174
175     for (int j = 0; j < dosen.size(); j++) {
176         scheduleList.add(new ScheduleClass(lc, lt, lt.plusHours(2)
177             , subject, dosen.get(j), location.get(j)));
178     }
179     dosen.clear();
180     location.clear();
181 }
182
183
184     return Mergering(scheduleList);
185 }
186
187 public List<ScheduleClass> Mergering (ArrayList<ScheduleClass> scheduleList)
188 {
189     int count = 0;
190     ArrayList<ScheduleClass> scheduleListSmt = new ArrayList<>();
191
192     for (int i = 0; i < scheduleList.size(); i++) {
193
194         if (scheduleList.get(i).getDosen().isEmpty() )
195         {
196             scheduleListSmt.add(scheduleList.get(i));
197         }
198     }
199
200     for (int i = 0; i < scheduleListSmt.size(); i++) {
201         for (int j = 0; j < scheduleList.size(); j++) {
202             if (scheduleList.get(j).equals(scheduleListSmt.get(i)))
203             {
204                 scheduleList.remove(j);
205             }
206         }
207     }
208
209     for (int i = 0; i < scheduleList.size(); i++) {
210         outerloop :
211         for (int j = 0; j < scheduleListSmt.size(); j++) {
212             if (scheduleList.get(i).getDate().equals(scheduleListSmt.get(j).getDate())
213                 && scheduleList.get(i).getTimeAwal().equals(scheduleListSmt.get(j).getTimeAwal()))
214             {
215                 String ss = scheduleList.get(i).getSubject();
216                 scheduleList.get(i).setSubject(ss+" , "+scheduleListSmt.get(j).getSubject());
217                 j = j + 1;
218                 break outerloop;
219             }
220         }
221     }
222     return scheduleList;
223 }

```

# LAMPIRAN B

## FILE EXCEL

JADWAL MENGAWAS UTS SEM. GENAP 2015/2016

TIS-UNPAR					Ruang						
No.	Hari, Tgl.	Jam	Sem.	PS	Nama Mata kuliah	9120	9121	9122	10316	10317	10323
1	Senin, 14 Mrt. 2016	08.00-10.00	2	MA	Kalkulus 2						
2	Senin, 14 Mrt. 2016	08.00-10.00	6p	IT	Komaman Informasi, Reologi	Mariskha	Pascal		Ferry	Maria	
3	Senin, 14 Mrt. 2016	11.00-13.00	4	FI	Elektronika 2, Analisis Real	Vero	Gede	Owen	Elok	Iwan	Philips
4	Senin, 14 Mrt. 2016	11.00-13.00	4	IT	Manajemen Informasi dan Basis Data						
5	Senin, 14 Mrt. 2016	14.00-16.00	2	IT	Matematika Informatika	Maria	Haryanto	Janto			
6	Senin, 14 Mrt. 2016	14.00-16.00	2	IT	Matematika Informatika						
7	Selasa, 15 Mrt. 2016	08.00-10.00	2	FI	Fisika Dasar 2, Pemn. Apl. Bergerak				Reinard	Gede	Liem
8	Selasa, 15 Mrt. 2016	08.00-10.00	p	MA	Komputasi Keuangan						
9	Selasa, 15 Mrt. 2016	11.00-13.00	6	FI	Mekanika Kuantum, Pers. Dif. Biasa				Bagoes	Iwan	Sylvia
10	Selasa, 15 Mrt. 2016	11.00-13.00	6p	IT	Jaringan Syarif Tiruan						
11	Selasa, 15 Mrt. 2016	14.00-16.00	8p	IT	Majemen Pengetahuan	Taufik	Janto				
12	Selasa, 15 Mrt. 2016	14.00-16.00	8p	IT	Majemen Pengetahuan						
13	Rabu, 16 Mrt. 2016	08.00-10.00	4	FI	Fisika Matematika 4, Proy. Inform., Logika Inf.	Risti	Pascal	Mariskha	Philips	Vania	Benny
14	Rabu, 16 Mrt. 2016	08.00-10.00	p	MA	Pengantar Matematika Asuransi						
15	Rabu, 16 Mrt. 2016	10.00-12.00	4	IT	Desain dan Analisis Algoritma	Shift 1: Luciana, Flaviana					
16	Rabu, 16 Mrt. 2016	12.00-14.00	4	IT	Desain dan Analisis Algoritma	Shift 2: Pascal, Luciana					
17	Rabu, 16 Mrt. 2016	11.00-13.00	2	MA	Aljabar Matriks	Taufik	Sylvia		Heni	Rusli	Haryanto
18	Rabu, 16 Mrt. 2016	11.00-13.00	8/8	FI/IT	Etika Profesi						
19	Rabu, 16 Mrt. 2016	14.00-16.00	6	IT	Penulisan Ilmiah	Lab.: Anung, Vania					
20	Kamis, 17 Mrt. 2016	08.00-10.00	2	FI	Fisika Matematika 2, AIK 2, AIK 4	Iwan	Dharma	Liem	Lab.: Chandra, Flaviana		
21	Kamis, 17 Mrt. 2016	08.00-10.00	4	MA	Optimasi, Kapsel, Fisika Instrumentasi						
22	Kamis, 17 Mrt. 2016	11.00-13.00	6	FI	Optika				Haryanto	Liem	
23	Kamis, 17 Mrt. 2016	11.00-13.00	6p	IT	Pemodelan dan Simulasi, Geometri						
24	Kamis, 17 Mrt. 2016	14.00-16.00	6	IT	Bahasa Inggris	Elok					
25	Senin, 21 Mrt. 2016	08.00-10.00	2	FI	Fisika Modern, Tata Surya, Mat. Diskret	Luciana	Gede	Joanna	Haryanto	Owen	Iwan
26	Senin, 21 Mrt. 2016	08.00-10.00	4	IT	Interaksi Manusia Komputer						
27	Senin, 21 Mrt. 2016	11.00-13.00	p	FI	Fisika Laser				Risti	Joanna	
28	Senin, 21 Mrt. 2016	11.00-13.00	7p	IT	Pencarian dan Temu Kembali Informasi						
29	Senin, 21 Mrt. 2016	14.00-16.00	6	FI	Pengantar Fisika Material	Elok	Janto				
30	Senin, 21 Mrt. 2016	14.00-16.00	6p	IT	E-Commerce						
31	Selasa, 22 Mrt. 2016	08.00-10.00	6p	IT	Pemodelan Formal, Listrik Magnet	Bagoes	Lab.: Benny, Owen				
32	Selasa, 22 Mrt. 2016	08.00-10.00	4	MA	Komputasi Matematika						
33	Selasa, 22 Mrt. 2016	10.00-12.00	2	IT	Algoritma dan Struktur Data	Shift 1: Sylvia, Husnul, Pascal					
34	Selasa, 22 Mrt. 2016	12.00-14.00	2	IT	Algoritma dan Struktur Data	Shift 2: Chandra, Husnul, Luciana					
35	Rabu, 23 Mrt. 2016	08.00-10.00	8p	IT	Pengendalian dan Audit Teknologi Informasi	Gede	Dharma	Reinard	Chandra	Vania	Bagoes
36	Rabu, 23 Mrt. 2016	08.00-10.00	p	FI	Fisika Kedokteran, Sistem Operasi						
37	Rabu, 23 Mrt. 2016	08.00-10.00	2	FI	Fisika Komputasi	Lab. Komp. : Elok					
38	Rabu, 23 Mrt. 2016	11.00-13.00	6/6/2	MA/PI/IT	Bahasa Indonesia	Sylvia	Philips	Heni	Owen	Lab: Lucian, Taufik	
39	Rabu, 23 Mrt. 2016	11.00-13.00	2	MA	Pemrograman Komputer						
40	Rabu, 23 Mrt. 2016	14.00-16.00	1	PI/IT	Pemrograman Berorientasi Objek						
41	Rabu, 23 Mrt. 2016	14.00-16.00	4	MA	Aljabar Linear	Benny	Iwan	Husnul, Vero			
42	Kamis, 24 Mrt. 2016				LIBUR						
43	Jumat, 25 Mrt. 2016										
44	Senin, 28 Mrt. 2016	08.00-10.00	4	FI	Fisika Statistik, Pengendalian Mutu	Lab. Komp. : Joanna				Reinard	Dharma
45	Senin, 28 Mrt. 2016	08.00-10.00	6p	IT	Komputasi Paralel						
46	Senin, 28 Mrt. 2016	11.00-13.00	8p	IT	Manajemen Proses Bisnis	Gede					
47	Senin, 28 Mrt. 2016	14.00-16.00	p	FI	Nanoteknologi dan Lingkungan	Elok	Lab.: Haryanto, Joanna				
48	Senin, 28 Mrt. 2016	14.00-16.00	8p	IT	Pengolahan Citra						
49	Selasa, 29 Mrt. 2016	08.00-10.00	6p	IT	Pemrograman Basis Data, T. Manaj. Investasi	Lab: Rein, Van	Rusli	Owen	Vania	Joanna	Bagoes
50	Selasa, 29 Mrt. 2016	08.00-10.00	p	FI	Dunia Digital dan Sains, Sistem Digital						
51	Selasa, 29 Mrt. 2016	14.00-16.00	4	MA	Statistika Matematika, Mekanika	Benny	Risti		Anung	Mariskha	Reinard
52	Selasa, 29 Mrt. 2016	14.00-16.00	4	IT	Rekayasa Perangkat Lunak						

Pengawas mohon hadir paling lambat 20 menit sebelum ujian dimulai. Bagi pengawas yang berhalangan hadir dimohon mencari pengantinya sendiri dan memberitahukan ke KTU paling lambat 1 (satu) hari sebelum pelaksanaan ujian.

Bandung, .....

Bandung, .....

Maria Anastasia, M.Si.

Gambar B.1: File Excel Jadwal Mengawas Format Lama

No.	Hari, Tanggal	Pukul	Nama Mata Kuliah	Ruang											
				9120	9121	9122	9014	9017	9018	10316	10317	10323			
1	Senin, 3 Okt. '16	08.00-10.00	Matematika Diskret	Luciana											
2	Senin, 3 Okt. '16	08.00-10.00	Elektrodinamika												
3	Senin, 3 Okt. '16	11.00-13.00	Pengantar Sistem Cerdas	Marisk	Sylvia	Maria							Iwan	Reinard	Anung
4	Senin, 3 Okt. '16	11.00-13.00	Kalkulus 1, Fisika Matematika 1												
5	Senin, 3 Okt. '16	14.00-16.00	Proses Stokastik	Dharma	Risti										
6	Senin, 3 Okt. '16	14.00-16.00	Hamburan dan Spektroskopi												
7	Selasa, 4 Okt. '16	08.00-10.00	Desain Antarmuka Grafis	Vero	Farica								Husnul	Fla	
8	Selasa, 4 Okt. '16	08.00-10.00	Kalkulus Vektor, Gelombang												
9	Selasa, 4 Okt. '16	11.00-13.00	Pemodelan Matematika	Sylvia									Chand	Bagoes	Philips
10	Selasa, 4 Okt. '16	11.00-13.00	Arsitektur dan Organisasi Komputer												
11	Selasa, 4 Okt. '16	14.00-16.00	Kosmologi Relativistik, B. Inggris	Rosa	Chand	Pascal	Rusli	Taufik	Gede	Claudio	Vania	Bagoes			
12	Selasa, 4 Okt. '16	14.00-16.00	Str. SI dan Ars. Perush. Berskala Bsr.												
13	Rabu, 5 Okt. '16	08.00-10.00	Pemecahan Masalah Matematika	Rosa	Vero	Iwan							Benny	Dharma	
14	Rabu, 5 Okt. '16	08.00-10.00	P. Sistem Informasi, Fisika Kuantum												
15	Rabu, 5 Okt. '16	11.00-13.00	Fungsi Kompleks										Sylvia	Philips	
16	Rabu, 5 Okt. '16	11.00-13.00	Kecerdasan Bisnis												
17	Rabu, 5 Okt. '16	14.00-16.00	Bahasa Indonesia, Fisika Mat. 3	Benny	Reinard										
18	Rabu, 5 Okt. '16	14.00-16.00	Komputer dan Masyarakat									Gede	Claudio		
19	Kamis, 6 Okt. '16	08.00-10.00	Struktur Diskret										Heni	Liem	
20	Kamis, 6 Okt. '16	11.00-13.00	Etika Profesi, Teori Peluang	Risti	Gede	Benny									
21	Kamis, 6 Okt. '16	11.00-13.00	Sistem dan Aplikasi Telematika												
22	Kamis, 6 Okt. '16	14.00-16.00	Geofisika, Graf. Komp. Pemr. Stokast												
23	Kamis, 6 Okt. '16	14.00-16.00	Desain dan Analisis Algoritma	Taufik	Liem				Lab. : Joanna				Risti		Bagoes
24	Jum'at, 7 Okt. '16	08.00-09.00	Pendidikan Kewarganegaraan												
25	Jum'at, 7 Okt. '16	09.30-10.30	Etika												
26	Jum'at, 7 Okt. '16	13.00-14.00	Logika												
27	Jum'at, 7 Okt. '16	14.30-15.30	Estetika												
28	Senin, 10 Okt. '16	08.00-10.00	Probabilitas dan Statistika	Iwan	Janto	Mariskha								Reinard	
29	Senin, 10 Okt. '16	08.00-10.00	Metoda Numerik												
30	Senin, 10 Okt. '16	11.00-13.00	Dasar-Dasar Pemrograman Java	Farica	Haryan	Sylvia				Lab. Gede, Philips					
31	Senin, 10 Okt. '16	11.00-13.00	Fisika Dasar 1, Fisika Dasar												
32	Senin, 10 Okt. '16	Shift 1	Pemrograman Berbasis WEB				Lab. : Joanna								
33	Senin, 10 Okt. '16	12.00-14.00													
34	Senin, 10 Okt. '16	Shift 2	Pemrograman Berbasis WEB				Lab. : Joanna								
35	Senin, 10 Okt. '16	14.00-16.00													
36	Senin, 10 Okt. '16	14.00-16.00	Lab. Fisika Instrumentasi	Janto											
37	Senin, 10 Okt. '16														
38	Selasa, 11 Okt. '16	08.00-10.00	Kapsel. Fis. Kedokt., Str. Alj. Lanjut	Farica			Lab. : Chand, Luciana, Gede								
39	Selasa, 11 Okt. '16	08.00-10.00	Adm Jar. Komp. 1, Adm Jar. Komp. 3												
40	Selasa, 11 Okt. '16	11.00-13.00	Alg. dan Struktur Data, Elektro. 1	Pascal			Joanna	Lab. Komp. Husnul, Liem, Haryan							
41	Selasa, 11 Okt. '16	11.00-13.00	Pemr. Kompetitif, Komp. Statistika												
42	Selasa, 11 Okt. '16	13.00-14.00	Pend.Pancasila												
43	Selasa, 11 Okt. '16	14.30-15.30	Agama Katolik/Fenom												
44	Rabu, 12 Okt. '16	08.00-10.00	Metoda Matematika										Maria	Benny	
45	Rabu, 12 Okt. '16	08.00-10.00	Pengantar Fisika Nuklir												
46	Rabu, 12 Okt. '16	10.00-12.00	Pemrograman Berorientasi Objek				Lab. : Husnul, Iwan, Vania								
47	Rabu, 12 Okt. '16	12.00-14.00	Pemrograman Berorientasi Objek				Lab. : Claudio, Bagoes, Vania								
48	Rabu, 12 Okt. '16	11.00-13.00	Penulisan ilmiah	Sylvia		Lab. : Anung									
49	Rabu, 12 Okt. '16	11.00-13.00	Kimia Dasar												
50	Rabu, 12 Okt. '16	14.00-16.00	Analisis Deret Waktu										Vero	Bagoes	
51	Rabu, 12 Okt. '16	14.00-16.00	Algoritma kriptografi												
52	Kamis, 13 Okt. '16	08.00-10.00	Teori Suku Bunga, PPS Fisika												
53	Kamis, 13 Okt. '16	08.00-10.00	Penambangan Data (Data Mining)										Luciana	Vero	Rusli
54	Kamis, 13 Okt. '16	10.00-12.00	Analisis & Desain Berorientasi Obj.				Lab. : Pascal, Vania								
55	Kamis, 13 Okt. '16	12.00-14.00	Analisis & Desain Berorientasi Obj.				Lab. : Pascal, Vania								
56	Kamis, 13 Okt. '16	11.00-13.00	Pengolahan Bahasa Alami	Dharma	Haryan										
57	Kamis, 13 Okt. '16	11.00-13.00	Teori Relativitas												
58	Kamis, 13 Okt. '16	14.00-16.00	Statistika Elementer, Fisika Thermal				Lab.: Heni, Reinard					Rosa	Janto	Fla	
59	Kamis, 13 Okt. '16	14.00-16.00	Pemrograman Fungsional												
60	Jumat, 14 Okt. '16	08.00-10.00	Jaringan Komputer	Ferry	Farica								Chand	Claudio	
61	Jumat, 14 Okt. '16	08.00-10.00	Matematika Asuransi												
62	Jumat, 14 Okt. '16	13.30-15.30	Pengantar Informatika	Rosa	Benny	Liem	Lab. : Taufik								
63	Jumat, 14 Okt. '16	13.30-15.30	Pemr. Komputer, Layanan Berb. WEB									Chandra	Haryan	Risti	

Bandung, .....

Wakil Dekan Bidang Sumber Daya,

Maria Anastasia, M.Si.

Gambar B.2: File Excel Jadwal Mengawas Format Lama