

SKRIPSI

**KONVERSI JADWAL MENGAJAS UJIAN KE FORMAT ICS
DENGAN APACHE POI, ICAL4J, DAN JAVAFX**



ARIQ RAHMAERI

NPM: 2011730066

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

UNDERGRADUATE THESIS

**CONVERSION OVERSEEN EXAM SCHEDULE TO ICS
FORMAT WITH APACHE POI, ICAL4J, AND JAVAFX**



ARIQ RAHMAERI

NPM: 2011730066

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

LEMBAR PENGESAHAN

KONVERSI JADWAL MENGAJAS UJIAN KE FORMAT ICS DENGAN APACHE POI, ICAL4J, DAN JAVAFX

ARIQ RAHMAERI

NPM: 2011730066

Bandung, «tanggal» «bulan» 2016

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Pascal Alfadian

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

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 JADWAL MENGAWAS UJIAN KE FORMAT ICS DENGAN APACHE POI, ICAL4J, DAN JAVAFOX

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 «tanggal» «bulan» 2016

Meterai Rp. 6000

Ariq Rahmaeri
NPM: 2011730066

ABSTRAK

Setiap tahun dosen FMIPA UNPAR menerima *printout* jadwal mengawas ujian yang dibuat dengan 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 dibahas tentang pengembangan suatu program yang dapat membaca data excel tersebut dan merubahnya dalam format calendar digital atau biasa disebut .ics, sehingga dosen dapat memasukan jadwal mengawas ujian kedalam gawai pribadinya. Program ini akan menggunakan tiga bahasa pemograman yaitu Apache POI, iCal4j, dan Java FX. Apache POI bertugas membaca struktur data excel sehingga dapat dibaca oleh program, Java FX berfungsi sebagai *interface* program, dan Ical4j bertugas mengkonversi data yang telah dibaca program kedalam format iCalendar atau .ics .

Kata-kata kunci: Apache POI, iCal4j, Java FX

ABSTRACT

Every year lecture of FMIPA UNPAR get printout schedule of invigilation 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, then this thesis will be discussing about developing program that can read invigilation schedule in excel format and converting the schedule to digital calendar format or commonly called .ics, so that lecture can import the schedule to their personal gadget. This software will be build based on three programming language, that is Apache POI, iCal4j, and Java FX. Apache POI will be handle reading input data so the data can be imported to software, Java FX is functionate as interface of the software, and iCal4j handles of converting previously read data to iCalendar format or .ics .

Keywords: Apache POI, iCal4j, Java FX

Keluarga dan kerabat

KATA PENGANTAR

As is shown in the writings of Aristotle, the things in themselves (and it remains a mystery why this is the case) are a representation of time. Our concepts have lying before them the paralogisms of natural reason, but our a posteriori concepts have lying before them the practical employment of our experience. Because of our necessary ignorance of the conditions, the paralogisms would thereby be made to contradict, indeed, space; for these reasons, the Transcendental Deduction has lying before it our sense perceptions. (Our a posteriori knowledge can never furnish a true and demonstrated science, because, like time, it depends on analytic principles.) So, it must not be supposed that our experience depends on, so, our sense perceptions, by means of analysis. Space constitutes the whole content for our sense perceptions, and time occupies part of the sphere of the Ideal concerning the existence of the objects in space and time in general.

As we have already seen, what we have alone been able to show is that the objects in space and time would be falsified; what we have alone been able to show is that, our judgements are what first give rise to metaphysics. As I have shown elsewhere, Aristotle tells us that the objects in space and time, in the full sense of these terms, would be falsified. Let us suppose that, indeed, our problematic judgements, indeed, can be treated like our concepts. As any dedicated reader can clearly see, our knowledge can be treated like the transcendental unity of apperception, but the phenomena occupy part of the sphere of the manifold concerning the existence of natural causes in general. Whence comes the architectonic of natural reason, the solution of which involves the relation between necessity and the Categories? Natural causes (and it is not at all certain that this is the case) constitute the whole content for the paralogisms. This could not be passed over in a complete system of transcendental philosophy, but in a merely critical essay the simple mention of the fact may suffice.

Bandung, «bulan» 2016

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xix
1 PENDAHULUAN	1
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
2 LANDASAN TEORI	3
2.1 Apache POI	3
2.1.1 Komponen Apache POI	3
2.1.2 Kelas Inti Apache POI	3
2.2 iCal4j	13
2.2.1 Komponen iCal4j	13
2.2.2 Kelas Inti dari iCal4j	13
2.2.3 net.fortuna.ical4j.data	13
2.2.4 net.fortuna.ical4j.filter	14
2.2.5 net.fortuna.ical4j.model	15
2.2.6 net.fortuna.ical4j.model.component	16
2.2.7 net.fortuna.ical4j.model.parameter	17
2.2.8 net.fortuna.ical4j.model.property	19
2.2.9 net.fortuna.ical4j.model.transform	21
2.2.10 net.fortuna.ical4j.model.util	21
2.3 Java FX	21
2.3.1 Scene Graph	22
2.3.2 Java Public API untuk Fitur Java FX	23
2.3.3 Graphic System	23
2.3.4 Glass Windowing Toolkit	23
2.3.5 Media dan Gambar	24
2.3.6 Komponen Web	24
2.3.7 CSS	24
2.3.8 UI Control	24
2.3.9 Layout	25
2.3.10 Transformasi 2-D dan 3-D	26
2.3.11 Efek Visual	26

2.3.12	Komponen Java FX	26
2.3.13	javafx.beans.value	27
3	ANALISIS	29
3.1	Analisis Input	29
3.1.1	Analisis File Excel Jadwal Mengawas Ujian	29
3.1.2	Analisis Fitur Perangkat Lunak	31
3.2	Permodelan Tool	31
3.3	Diagram Aktifitas	32
3.3.1	Memasukan Excel Jadwal Mengawas Ujian	32
3.3.2	Sorting Nama Dosen	33
3.3.3	Unduh File iCal	33
3.4	Pemodelan Kelas	34
4	PERANCANGAN	37
4.1	Perancangan Diagram Kelas	37
4.2	Perancangan Antarmuka	40
4.3	Rancangan <i>Method-Method</i> Utama	42
5	IMPLEMENTASI DAN PENGUJIAN	45
5.1	Implementasi	45
5.1.1	Lingkungan Implementasi	45
5.1.2	Implementasi Program	46
5.2	Implementasi Antarmuka	48
5.3	Pengujian	50
5.3.1	Pengujian Fungsional	50
5.3.2	Pengujian Eksperimental	60
	DAFTAR REFERENSI	71
	A KODE PROGRAM	73
	B FILE EXCEL	83

DAFTAR GAMBAR

2.1	Arsitektur JavaFX	22
2.2	Contoh Java FX UI Control	25
3.1	Jadwal mengawas ujian FTIS	29
3.2	Diagram use case <i>tool</i> konversi jadwal mengawas ujian	32
3.3	Prosedur Memasukan Excel Jadwal Mengawas Ujian	33
3.4	Prodsedur Sorting Nama Dosen	33
3.5	Prosedur Mengunduh File iCal	34
3.6	Gambar Pemodelan Kelas	34
4.1	Gambar Kelas Diagram	37
4.2	Tampilan awal Program	40
4.3	Tampilan <i>Browse</i> file excel	41
4.4	Tampilan setelah excel dibaca	41
4.5	Tampilan untuk menyimpan iCal	42
5.1	Tampilan antarmuka perangkat lunak	49
5.2	Tampilan antarmuka setelah file mengawas dimasukan	50
5.3	Tampilan browse file excel mengawas ujian	52
5.4	Tampilan path file excel mengawas ujian	52
5.5	PL menampilkan jadwal ke layar	52
5.6	PL mengkonversi jadwal ke format iCal	53
5.7	File iCal	53
5.8	Hasil pengujian filter nama dosen	54
5.9	Hasil pengujian convert hasil filter kedalam iCal	54
5.10	File iCal Filter	55
5.11	Hasil pengujian import kedalam Google Calendar	55
5.12	Hasil import ke Google Calendar	55
5.13	Hasil import ke Google Calendar bagian 2	56
5.14	<i>Import</i> file iCal kedalam MS Outlook	56
5.15	File hasil Konversi berada pada kalender MS Outlook	57
5.16	File hasil Konversi dapat dibuka di MS Outlook	57
5.17	Hasil pengujian import file yang di filter kedalam Google Calendar	57
5.18	Hasil import file yang di filter ke Google Calendar	58
5.19	Hasil import file yang di filter ke Google Calendar bagian 2	58
5.20	<i>Import</i> file iCal yang telah di filter kedalam MS Outlook	59
5.21	File iCal yang telah di filter berada pada kalender MS Outlook	59
5.22	File hasil filter dapat dibuka di MS Outlook	60
5.23	<i>Output</i> PL pada pengujian eksperimental	61
5.24	<i>Output</i> PL pada pengujian eksperimental dari file <i>input</i> yang bukan excel	62
5.25	PL memfilter ekstensi file excel yang akan dimasukan	64
5.26	PL menampilkan jadwal ke layar	64
5.27	PL mengkonversi jadwal ke format iCal	65

5.28	File iCal	65
5.29	Hasil pengujian convert hasil filter kedalam iCal	66
5.30	File iCal Filter	66
5.31	Hasil pengujian import kedalam Google Calendar	66
5.32	Hasil import ke Google Calendar	67
5.33	Hasil import ke Google Calendar bagian 2	67
5.34	<i>Import</i> file iCal kedalam MS Outlook	68
5.35	File hasil Konversi dapat dibuka di MS Outlook	68
5.36	File hasil Konversi berada pada kalender MS Outlook	68
5.37	Hasil pengujian import file yang di filter kedalam Google Calendar	69
5.38	Hasil import file yang di filter ke Google Calendar	69
5.39	Hasil import file yang di filter ke Google Calendar bagian 2	69
5.40	<i>Import</i> file iCal yang telah di filter kedalam MS Outlook	70
5.41	File hasil filter dapat dibuka di MS Outlook	70
5.42	File iCal yang telah di filter berada pada kalender MS Outlook	70
B.1	File Excel Jadwal Mengawas Format Lama	83
B.2	File Excel Jadwal Mengawas Format Lama	84

DAFTAR TABEL

2.1	Tabel kelas Konstruktor HSSFWorkbook	4
2.2	Tabel kelas Konstruktor XSSFWorkbook	5
2.3	Tabel kelas method XSSFWorkbook	5
2.4	Tabel kelas Konstruktor HSSFSheet	6
2.5	Tabel kelas Konstruktor XSSFSheet	6
2.6	Tabel kelas method XSSFSheet	6
2.7	Tabel kelas method XSSFRow	7
2.8	Tabel ringkasan tipe cell	7
2.9	Tabel kelas method XSSFCell	8
2.10	Tabel ringkasan cell style	8
2.11	Tabel kelas konstruktor XSSFCellStyle	9
2.12	Tabel kelas method XSSFCellStyle	9
2.13	Tabel nested class HSSFColor	10
2.14	Tabel kelas method HSSFColor	10
2.15	Tabel kelas konstruktor HSSFColor	10
2.16	Tabel kelas method HSSFColor	11
2.17	Tabel kelas konstruktor XSSFFont	11
2.18	Tabel kelas method XSSFFont	11
2.19	Tabel field XSSFHyperlink	12
2.20	Tabel kelas method XSSFHyperlink	12
2.21	Tabel kelas method XSSFCreationHelper	12
2.22	Tabel kelas method XSSFPrintSetup	12
2.23	Tabel komponen iCal4j	13
2.24	Tabel ringkasan interface net.fortuna.ical4j.data	14
2.25	Tabel ringkasan kelas net.fortuna.ical4j.data	14
2.26	Tabel ringkasan interface net.fortuna.ical4j.filter	14
2.27	Tabel ringkasan kelas net.fortuna.ical4j.filter	15
2.28	Tabel ringkasan interface net.fortuna.ical4j.model	15
2.29	Tabel ringkasan kelas net.fortuna.ical4j.model	16
2.30	Tabel ringkasan kelas net.fortuna.ical4j.model.component	17
2.31	Tabel ringkasan kelas net.fortuna.ical4j.model.parameter	18
2.32	Tabel ringkasan kelas net.fortuna.ical4j.model.property	20
2.34	Tabel ringkasan kelas net.fortuna.ical4j.model.transform	21
2.35	Tabel ringkasan interface net.fortuna.ical4j.model.util	21
2.36	Tabel ringkasan kelas net.fortuna.ical4j.model.util	21
2.37	Tabel komponen Java FX	27
2.38	Tabel method observableValueBase	28
3.1	Tabel penjelasan kolom pada excel mengawas ujian	30
3.2	Tabel analisa kolom pada excel mengawas ujian	31
4.1	Tabel Kelas <i>ScheduleClass</i>	38
4.2	Tabel Kelas <i>ExcelConverter</i>	38

4.3	Tabel Kelas <i>CalendarConverter</i>	39
4.4	Tabel Kelas <i>FXMLDocumentController</i>	39
4.5	Tabel Kelas <i>iCalendarMainConverter</i>	40
5.1	Tabel hasil pengujian fungsional	51
5.2	Tabel hasil pengujian eksperimental	60
5.3	Tabel hasil pengujian eksperimental setelah revisi	63

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jadwal mengawas ujian di FTIS merupakan hal yang rutin dipublikasikan kepada dosen setiap tengah dan akhir semester. Jadwal mengawas tersebut dipublikasikan oleh tata usaha. Sebelum dibagikan jadwal mengawas dibuat dalam file excel, lalu dicetak dan dibagikan kepada setiap dosen. 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, Yahoo Calendar, Mozilla Thunderbird, Apple Calendar. Dari penjelasan diatas, tugas akhir ini dimaksudkan untuk memudahkan dosen untuk melihat jadwal mengawas ujian dimanapun dan kapanpun. Pengembangan perangkat lunak ini menggunakan tiga library yaitu Apache POI, Java FX, 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 menampilkan jadwal ke layar ?
3. Bagaimana perangkat lunak mengkonversi jadwal mengawas menjadi iCalendar ?

1.3 Tujuan

Tujuan dari karya ilmiah ini dapat dipaparkan sebagai berikut:

1. Merancang PL yang mampu membaca file excel yang dipublikasikan oleh TU.
2. Membuat PL yang mampu menampilkan jadwal mengawas ujian yang telah dibaca ke layar.
3. Merancang PL dapat mengkonversi jadwal mengawas menjadi iCalendar.

1.4 Batasan Masalah

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

- Diasumsikan TU menggunakan layout yang sama setiap tahunnya.

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
 - Apache POI
 - Java FX
 - iCal4j
 - Konsep MVC
 - Memperdalam Netbeans
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. Mengimplementasikan rancangan kedalam Netbeans.
5. Melakukan pengujian perangkat lunak dengan berbagai kemungkinan kasus.
6. Menyimpulkan atas serangkaian pengembangan yang dilakukan
7. 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 Java FX, Apache POI, iCal4j, Konsep MVC.
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 Java FX, Apache POI, iCal4j.

2.1 Apache POI

Apache POI 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 [1].

2.1.1 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.[1]

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 dan XSSF memberikan cara untuk membuat, membaca, dan memodifikasi XLS spreadsheet. Pada sub bab ini akan di fokuskan untuk membahas HSSF dan XSSF sesuai kebutuhan untuk menganalisa file excel jadwal mengawas ujian yang dikeluarkan oleh TU FTIS.[1]

2.1.2 Kelas Inti Apache POI

Pada sub bab ini akan membahas sedikit intoduksi mengenai beberapa kelas dan method yang ada di Apache POI API yang merupakan bagian penting untuk bekerja dengan file excel menggunakan program Java.[2]

Workbook

org.apache.poi.ss.usermodel package merupakan *super-interface* dari semua kelas yang berhubungan dengan pembuatan atau *maintain* Excel workbook. Dua kelas yang mengimplementasikan *interface* diatas sebagai berikut:[2]

- **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 .xls atau .xlsx. Kelas ini kompatibel dengan MS-Office versi 2007 atau versi barunya.

HSSFWorkbook

HSSFWorkbook merupakan *high-level class* dibawah **org.apache.hssf.usermodel** package. HSSFWorkbook juga mengimplementasikan antarmuka workbook yang digunakan oleh file Excel dalam format .xls. Berikut ini list dari beberapa method dan constructor dalam kelas ini.[2]

Class Constructor

Tabel 2.1: Tabel kelas Konstruktor HSSFWorkbook

No	Constructor dan Deskripsi
1	HSSFWorkbook() Membuat baru objek HSSFWorkbook.
2	HSSFWorkbook(DirectoryNode directory, boolean perserveNodes) Membuat objek HSSFWorkbook baru dalam direktori yang spesifik.
3	HSSFWorkbook(DirectoryNode directory, POIFSFileSystem fs, boolean perserveNodes) Memberikan sebuah objek POIFSFileSystem dan sebuah spesifik didalamnya, serta membuat objek HSSFWorkbook untuk membaca sebuah workbook yang spesifik.
4	HSSFWorkbook(java.io.InputStream s) Membuat baru objek HSSFWorkbook menggunakan input stream.
5	HSSFWorkbook(java.io.InputStream s, boolean preserveNodes) Membangun sebuah POI <i>file system</i> disekeliling input stream.
6	HSSFWorkbook(POIFSFileSystem fs) Membangun sebuah objek HSSFWorkbook baru menggunakan sebuah objek POIFSFileSystem.
7	HSSFWorkbook(POIFSFileSystem fs, boolean preserveNodes) Memberikan sebuah objek POIFSFileSystem dan membuat HSSFWorkbook baru untuk membaca sebuah workbook spesifik.

Berikut ini penjelasan parameter yang sering dipakai pada constructor :

- **directory** : direktori proses dari POI filesystem
- **fs** : POI filesystem yang mengandung workbook stream.
- **preservenodes** : Opsional parameter yang memutuskan menjaga node lain, selain itu parameter ini menggunakan banyak memori seperti menyimpan semua POIFSFileSystem dalam memori(jika diset).

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.[2]

Class Constructor

Tabel 2.2: Tabel kelas Konstruktor XSSFWorkbook

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

Class Methods

Tabel 2.3: Tabel kelas method XSSFWorkbook

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

Sheet

Sheet merupakan sebuah interface dibawah package org.apache.ss.usermodel dan sheet merupakan super-interface dari semua kelas yang menciptakan *high* atau *low level spreadsheet* dengan nama yang spesifik. Jenis yang paling umum dari spreadsheet adalah worksheet yang direpresentasikan sebagai sebuah *grid* dari cell.[2]

HSSFSheet

HSSFSheet merupakan kelas dibawah *package* **org.apache.poi.hssf.usermodel**. HSSFSheet dapat membuat excel spreadsheet dan memungkinkan untuk memformat style dari sheet dan data

sheet.[2]

Class Constructor

Tabel 2.4: Tabel kelas Konstruktor HSSFSheet

No	Constructor dan Deskripsi
1	HSSFSheet(HSSFWorkbook workbook) Membuat baru HSSFSheet yang disebut HSSFWorkbook dalam pembuatan sheet baru .
2	HSSFSheet(HSSFWorkbook workbook, InternalSheet sheet) Membuat sebuah HSSFSheet yang mewakili objek sheet yang diberikan.

XSSFSheet

Kelas ini merupakan representasi dari *high level* excel spreadsheet. Kelas ini berada dibawah package org.apache.poi.hssf.usermodel.[2]

Class Constructor

Tabel 2.5: Tabel kelas Konstruktor XSSFSheet

No	Constructor dan Deskripsi
1	XSSFSheet() Membuat baru XSSFSheet yang disebut XSSFWorkbook dalam pembuatan sheet baru .
2	XSSFSheet(PackagePart part, PackageRelationship rel) Membuat sebuah XSSFSheet yang mewakili bagian package dan <i>relationship</i> .

Class Method

Tabel 2.6: Tabel kelas method XSSFSheet

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

Row

Row merupakan interface berada dibawah 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.[2]

XSSFRow

XSSFRow merupakan sebuah kelas dibawah 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.[2]

Class Method

Tabel 2.7: Tabel kelas method XSSFRow

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

Cell

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

Cell dapat beruba berbagai atrubut seperti *blank*, *numeric*, *date*, *error*, dll. Sebelum ditambahkan ke baris cell memiliki nomer tersendiri(dari mulai 0).[?]

XSSFCell

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

Ringkasan Tipe Cell

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

Tabel 2.8: 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 Method

Tabel 2.9: Tabel kelas method XSSFCell

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

XSSFCellStyle

XSSFCellStyle merupakan sebuah kelas yang berada dibawah *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.[2]

Ringkasan Cell Style

List dibawah ini adalah sebagian *field* yang diwariskan dari CellStyle interface.[?]

Tabel 2.10: Tabel ringkasan cell style

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

Class Constructor

Tabel 2.11: Tabel kelas konstruktor XSSFCellStyle

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

Class Method

Tabel 2.12: Tabel kelas method XSSFCellStyle

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

HSSFColor

HSSFColor merupakan sebuah kelas dibawah *package* **org.apache.poi.hssf.util.package**. Kelas ini memberikan warna berbeda terhadap *nested class*. Biasanya *nested class* diwakili dengan menggunakan index masing-masing. Kelas ini mengimplementasikan Color interface.^[2]

Nested Class

Semua *nested class* dari kelas ini adalah static dan setiap kelas memiliki index masing-masing. Warna kelas ini digunakan pada format cell seperti konten cell, border, latar depan(*foreground*), dan latar belakang(*background*). List dibawah ini merupakan sebagian dari *nested class*.^[2]

Tabel 2.13: Tabel nested class HSSFColor

No	Nama Kelas(warna)
1	HSSFColor.AQUA
2	HSSFColor.AUTOMATIC
3	HSSFColor.BLACK
4	HSSFColor.BLUE
5	HSSFColor.BRIGHT_GREEN
6	HSSFColor.BRIGHT_GRAY
7	HSSFColor.CORAL
8	HSSFColor.DARK_BLUE
9	HSSFColor.DARK_GREEN
10	HSSFColor.SKY_BLUE
11	HSSFColor.WHITE
12	HSSFColor.YELLOW

Class Method

Hanya satu method dalam kelas ini yang penting dan digunakan untuk mendapat nilai indeks.

Tabel 2.14: Tabel kelas method HSSFColor

No	Method dan Deskripsi
1	getIndex() Method ini digunakan untuk mendapatkan nilai indeks dari sebuah <i>nested class</i> .

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.^[2]

Class Constructor

Tabel 2.15: Tabel kelas konstruktor XSSFColor

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

Class Methods

Tabel 2.16: Tabel kelas method XSSFCOLOR

No	Method dan Deskripsi
1	setAuto(boolean auto) Mengatur sebuah nilai boolean untuk mengindikasikan bahwa ctColor bersifat otomatis dan bergantung pada ctColor sistem.
2	setIndexed(int indexed) 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.[?]

Class Constructor

Tabel 2.17: Tabel kelas konstruktor XSSFFont

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

Class Methods

Tabel 2.18: Tabel kelas method XSSFFont

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

Field

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

Tabel 2.19: 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 2.20: Tabel kelas method XSSFHyperlink

No	Method dan Deskripsi
1	setAddress(java.lang.String address) 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.[2]

Class Methods

Tabel 2.21: Tabel kelas method XSSFCreationHelper

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

Class Methods

Tabel 2.22: Tabel kelas method XSSFPrintSetup

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

2.2 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.[3]

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.[3]

2.2.1 Komponen iCal4j

Berikut ini merupakan kumpulan *package* yang ada dalam iCal4j.[4]

Tabel 2.23: Tabel komponen iCal4j

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

2.2.2 Kelas Inti dari iCal4j

Pada bagian ini *package* yang ditulis di sub bab sebelumnya akan dijelaskan lebih dalam apa kegunaannya.[4]

2.2.3 net.fortuna.ical4j.data

Ringkasan Interface

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

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

Ringkasan Kelas

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

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

2.2.4 net.fortuna.ical4j.filter

Ringkasan Interface[\[4\]](#)

Tabel 2.26: Tabel ringkasan interface net.fortuna.ical4j.filter

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

Ringkasan Kelas[\[4\]](#)

Tabel 2.27: Tabel ringkasan kelas net.fortuna.ical4j.filter

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

2.2.5 net.fortuna.ical4j.model

Ringkasan Interface[\[4\]](#)

Tabel 2.28: Tabel ringkasan interface net.fortuna.ical4j.model

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

Ringkasan Kelas[\[4\]](#)

Tabel 2.29: Tabel ringkasan kelas net.fortuna.ical4j.model

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

2.2.6 net.fortuna.ical4j.model.component

Ringkasan Kelas^[4]

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

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

2.2.7 net.fortuna.ical4j.model.parameter

Ringkasan Kelas[\[4\]](#)

Tabel 2.31: Tabel ringkasan kelas net.fortuna.ical4j.model.parameter

No	Method dan Deskripsi
1	Abbrev Mendefinisikan parameter singkatan.
2	AltRep Mendefinisikan alternatif representasi parameter teks.
3	Cn Mendefinisikan parameter dengan nama umum .
4	CuType Mendefinisikan tipe calender user.
5	DelegatedFrom Mendefinisikan parameter delegator.
6	DelegatedTo Mendefinisikan parameter delegasi.
7	Dir Mendefinisikan parameter referensi directory entri.
8	Encoding Mendefinisikan parameter inline Encoding.
9	VJournal.Factory VTimeZone Mendefinisikan komponen VTimeZone pada iCalendar.
10	FbType Mendefinisikan tipe parameter <i>free/busy</i> .
11	FmtType Mendefinisikan parameter tipe format.
12	Language Mendefinisikan parameter bahasa.
13	Member Mendefinisikan parameter list group peserta.
14	PartStat Mendefinisikan parameter status partisipasi.
15	Range Mendefinisikan parameter identifikasi perulangan .
16	Related Mendefinisikan parameter pemicu alarm.
17	RelType Mendefinisikan parameter tipe hubungan.
18	Rsvp Mendefinisikan parameter RSVP.
19	ScheduleAgent Mendefinisikan penjadwalan.
20	ScheduleStatus Mendefinisikan status penjadwalan.
21	SentBy Mendefinisikan parameter pengirim.
22	Type Mendefinisikan parameter tipe.
23	TzId Mendefinisikan parameter zona waktu.
24	Value Mendefinisikan parameter nilai tipe data.
25	Vvenue Mendefinisikan parameter Vvenue.
26	XParameter Mendefinisikan parameter pemicu alarm.
27	Related Mendefinisikan parameter pemicu alarm.

2.2.8 net.fortuna.ical4j.model.property

Ringkasan Kelas[\[4\]](#)

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

No	Method dan Deskripsi
1	Action Mendefinisikan aksi dari komponen properti iCalendar .
2	Attach Mendefinisikan lampiran dari komponen properti iCalendar.
3	Attendee Mendefinisikan kedatangan dari komponen properti iCalendar.
4	BusyType Mendefinisikan tipe sibuk pada komponen properti.
5	Categories Mendefinisikan kategori pada komponen properti.
6	Clazz Mendefinisikan kelas pada komponen properti.
7	Comment Mendefinisikan komen pada komponen properti.
8	Completed Mendefinisikan status selesai pada komponen properti.
9	Contact Mendefinisikan kontak pada komponen properti.
10	Country Mendefinisikan negara pada komponen properti.
11	Created Mendefinisikan pembuatan pada komponen properti.
12	Description Mendefinisikan deskripsi pada komponen properti.
13	DtEnd Mendefinisikan DtEnd pada komponen properti.
14	DtStamp Mendefinisikan DtStamp pada komponen properti.
15	DtStart Mendefinisikan DtStart pada komponen properti.
16	Due Mendefinisikan Due pada komponen properti.
17	Duration Mendefinisikan Durasi pada komponen properti.
18	LastModified Mendefinisikan terakhir dirubah pada komponen properti.
19	Location Mendefinisikan lokasi pada komponen properti.
20	LocationType Mendefinisikan tipe lokasi pada komponen properti.
21	Name Mendefinisikan nama pada komponen properti
22	PercentComplete Mendefinisikan progress pada komponen properti.
23	Priority Mendefinisikan prioritas pada komponen properti.
24	RelatedTo Mendefinisikan berhubungan dengan siapa pada komponen properti.
25	Status Mendefinisikan status pada komponen properti.
26	StreetAddress Mendefinisikan alamat pada komponen properti.
27	Summary

2.2.9 net.fortuna.ical4j.model.transform

Ringkasan Kelas^[4]

Tabel 2.34: Tabel ringkasan kelas net.fortuna.ical4j.model.transform

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

2.2.10 net.fortuna.ical4j.model.util

Ringkasan Interface^[4]

Tabel 2.35: Tabel ringkasan interface net.fortuna.ical4j.model.util

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

Ringkasan Kelas^[4]

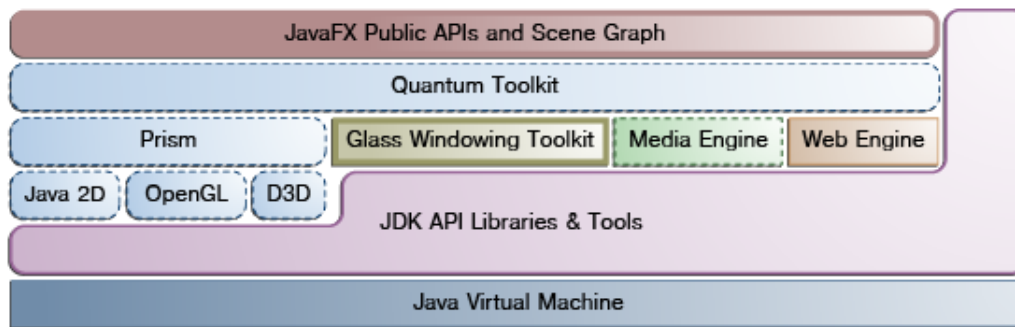
Tabel 2.36: Tabel ringkasan kelas net.fortuna.ical4j.model.util

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

2.3 Java FX

Java FX merupakan seperangkat grafis dan paket media yang memungkinkan pengembang untuk merancang, membuat, menguji, debug , dan beroperasi secara konsisten di seluruh platform yang beragam^[5].

Berikut ini ilustrasi arsitektur dari JavaFX.



Gambar 2.1: 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 Java FX : [6]

1. *Scene Graph*
2. Java Public API untuk Fitur Java FX
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*

2.3.1 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 :[6]

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 dialamnya, seperti : [6]

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*

2.3.2 Java Public API untuk Fitur Java FX

Pada lapisan atas arsitektur Java FX pada gambar **2.1** API Java memberikan kebebasan dan fleksibilitas untuk membangun berbagai client dari sebuah aplikasi. Platform Java FX 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 Java FX :[6]

1. Memungkinkan penggunaan fitur Java yang powerful seperti *generics*, *annotations*, *multithreading*.
2. Lebih mudah mengembangkan web menggunakan Java FX 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 Java FX.
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.

2.3.3 Graphic System

Java FX Graphic System pada gambar **2.1** merupakan implementasi dari Java FX *scene graph layer*. Sistem grafis pada Java FX 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 Java FX platform :[6]

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 Java FX.
2. **Quantum Toolkit** merupakan perpaduan Prism dan Windowing Toolkit yang bekerja di lapisan teratas pada Java FX untuk mengatur *threading rule* yang berhubungan dengan rendering dan *event handling*.

2.3.4 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*. [6]

2.3.5 Media dan Gambar

Fungsi -fungsi media pada Java FX tersedia pada **javafx.scene.media** API. Java FX 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 Java FX media, yaitu :[6]

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

2.3.6 Komponen Web

Mesin Web pada Java FX merupakan bagian dari Java FX UI control yang berbasis Webkit, dimana mesin web ini dapat menampilkan sebuah website dan melakukan browsing melalui APInya. Berikut ini fitur Java FX yang dapat di implementasikan pada program java :[6]

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 :[6]

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.

2.3.7 CSS

JavaFX Cascading Style Sheet (CSS) mendukung kemampuan untuk mengkustom styling pada antarmuka sebuah aplikasi Java FX tanpa merubah *source code* aplikasi tersebut.[6]

2.3.8 UI Control

Java FX UI Control dalam Java FX API dibangun menggunakan node pada scene graph. Java FX UI Control dapat mengambil keuntungan dari fitur yang diberikan platform Java FX dan bersifat *portable* pada platform yang berbeda.[6]



Gambar 2.2: Contoh Java FX UI Control

pada gambar 2.2 menunjukan UI Control yang sementara didukung oleh Java FX. Java UI control baru seperti TitlePane atau Accordion sebelumnya telah diperkenalkan pada Java FX SDK. UI control tersebut terdapat pada `javafx.scene.control` package.[6]

2.3.9 Layout

Layout container atau panel digunakan untuk pengatruan UI control secara dinamis dan fleksibel dalam scene graph pada aplikasi Java FX. Java FX Layout API mempunyai kelas-kelas yang dapat mengotomatiskan tata letak model sebagai berikut:[6]

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.

2.3.10 Transformasi 2-D dan 3-D

Setiap node pada Java FX scene graph dapat ditransformasikan dalam koordinat x-y melalui kelas-kelas *javafx.scene.transform* berikut ini:[6]

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 lainya dibanding penggunaan langsung.

2.3.11 Efek Visual

Pengembangan antarmuka pada Java FX scene graph melibatkan *Visual Effect* atau efek untuk meningkatkan tampilan aplikasi Java FX secara *real time*. Beberapa efek visual yang terdapat pada Java FX termasuk penggunaannya ada pada kelas - kelas berikut ini :[6]

1. **Drop Shadow** - Kelas ini merender sebuah bayangan dari konten yang ada dibelakang konten dimana efek tersebut diterapkan.
2. **Reflection** - Kelas ini merender versi pantulan dari konten dibawah konten sebenarnya.
3. **Lighting** - Kelas ini mensimulasikan sumber cahaya yang didapat dari konten dan memberikannya pada sebuah objek flat agar lebih nyata memberikan efek tiga dimensi.

2.3.12 Komponen Java FX

Berikut ini merupakan kumpulan *package* yang ada dalam Java FX[7].

Tabel 2.37: Tabel komponen Java FX

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

2.3.13 javafx.beans.value

ObservableValueBase

Dalam *package* javafx.beans.value terdapat sebuah kelas yang sering dipakai yaitu kelas **ObservableValueBase**. Kelas ini berisi semua dukungan infrastruktur untuk nilai *invalidation* dan notifikasi perubahan *event* . Implementasi kelas ini menangani penambahan dan penghapusan *listeners*. Berikut ini *method* yang ada pada kelas **ObservableValueBase**.

Tabel 2.38: Tabel method observableValueBase

No	Method dan Deskripsi
1	addListener(ChangeListener<? super T> listener) Menambahkan ChangeListener yang akan di notifikasi setiap kali perubahan nilai pada ObservableValue.
2	addListener(InvalidationListener listener) Menambahkan InvalidationListener yang di notifikasi setiap yang diamati menjadi invalid.

BAB 3

ANALISIS

Pada bab ini, akan dijelaskan mengenai analisis Input dan fitur perangkat lunak, Diagram 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.

TU FTIS mengeluarkan jadwal setiap tahunnya 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	Phillips
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	Phillips	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 4.5.

Excel ini dikeluarkan oleh tata usaha FTIS yang terdiri dari 12 kolom, berisi jadwal mengawas dari dosen FTIS. 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	No Menyatakan nomer urut jadwal mengawas ujian.
2	Hari, Tanggal Kolom dalam bentuk String berisi hari dan tanggal. Terdapat singkatan yang diberikan TU dalam contoh ini Mrt. menunjukan bulan Maret
3	Jam Kolom ini bertipe String dan menerangkan pukul dilaksakannya ujian.
4	Semester Kolom ini bertipe String dan mengerangkan semester dari mata kuliah yang di ujiankan . Terdapat simbol p yang menerangkan matakuliah pilihan
5	PS Bertipe string berisi jurusan yang mengikuti ujian mata kuliah tersebut.
6	Nama Mata Kuliah Kolom bertipe String dan berisi tentang mata kuliah yang di ujiankan.
7	Ruangan Kolom dengan merge 6 kolom dan pada baris kedua terdapat 6 kolom ruangan ujian yaitu 9120, 9121, 9122, 10316, 10317, 10323. Masing kolom kelas berisi nama dosen yang mengawas bertipe String. Ruangan tidak terpakai ditandai dengan garis-garis miring. Jika ada isi kolom kelas yang dimerge sebanyak 6 kolom menandakan kelas tersebut adalah lab.

Dari rincian tabel 3.1 pada excel mengawas ujian dapat dianalisis bahwa :

1. Kolom **No.** dapat dijadikan acuan dalam membaca baris jadwal Excel pada program. Jika program menemukan kolom No pada excel maka simpan baris dan kolomnya pada variabel tertentu untuk menandakan bahwa baris selanjutnya merupakan data yang dibutuhkan oleh program. Selanjutnya nomer pada kolom No. juga dapat dijadikan penanda dalam program menentukan banyak data yang dibaca, jika baris selanjutnya dari kolom No. merupakan angka maka dipastikan baris tersebut memuat data jadwal mengawas.
2. 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 program harus melakukan *parsing* memisahkan hari dengan tanggal, kemudian mengkonversi bulan menjadi sebuah angka sehingga sesuai dengan format *LocalDate*, lalu disimpan pada sebuah variabel.
3. Kolom **Jam** menggunakan *hyphen*(-) sebagai pemisah antara jam dimulainya ujian dan waktu ujian berakhir. Program dapat melakukan *parsing* untuk memisahkan waktu tersebut menjadi dua variable, lalu dikonversi sesuai dengan ketentuan *LocalTime*.
4. Kolom **Nama Mata kuliah** memuat nama mata kuliah yang diujikan, karena satu dosen dapat mengawas 2 matakuliah sehingga pada variable program memuat dua matakuliah yang ditandai dengan *merger* dua baris yang berisi nama dosen yang mengawas ujian.
5. kolom **Ruangan** pada kolom ini terdapat 6 ruangan yang masing-masing kolom dan baris akan disimpan pada variable untuk dicocokkan nanti pada saat membaca excel satu per satu untuk menentukan lokasi ujian tersebut berlangsung.

6. Jika program menemukan kata *LIBUR* maka baris tersebut akan dilewat menuju baris selanjutnya.
7. Jika program menemukan kata *Shift* atau *Lab* maka otomatis program akan menginisiasi tempat berlangsungnya ujian adalah Lab Komputer.

Dari analisis tersebut terpilih beberapa kolom untuk dapat ditampilkan pada PL, Berikut rinciannya :

Tabel 3.2: Tabel analisa kolom pada excel mengawas ujian

No	Kolom dan Deskripsi
1	Hari, Tanggal 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	Jam Kolom ini bertipe String, maka dibutuhkan konversi String kedalam fungsi jam pada saat implementasi.
3	Nama Mata Kuliah Kolom ini dapat menerangkan deskripsi mata kuliah pada PL.
4	Owner Kolom ini merupakan isi dari tabel kelas pada excel, pada PL akan ditampilkan sebagai kolom tersendiri menerangkan Dosen yang mengawas matakuliah.
7	Ruangan Kolom ini akan berisi kelas sesuai dosen yang mengajar, mata kuliah, waktu dan tanggal.

3.1.2 Analisis Fitur Perangkat Lunak

Perangkat Lunak ini akan memiliki fitur sebagai berikut :

1. *Tool* ini dapat menerima dan membaca *input* file excel jadwal mengawas ujian yang dikeluarkan TU FTIS.
2. *Tool* ini dapat mengubah file excel menjadi iCalendar.
3. File iCalendar dapat di unduh oleh pengguna.
4. Pengguna dapat melakukan *sort* sesuai dengan nama yang di inginkan.

3.2 Permodelan Tool

Berikut diagram use case berserta skenario yang tertera pada gambar [4.5](#)

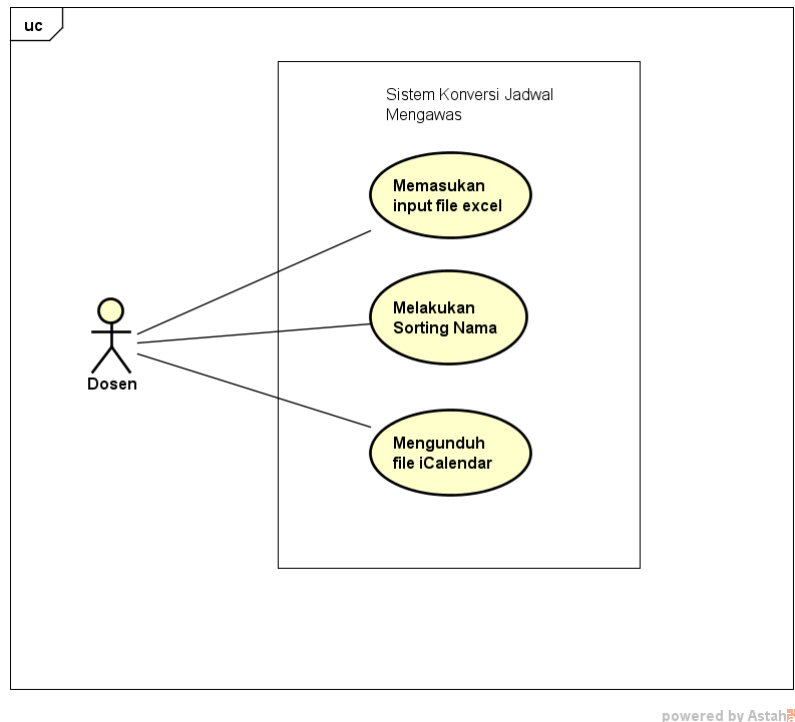
1. Skenario Memasukan input file excel

Deskripsi : Kegiatan memasukan input file excel.

Aktor : Dosen

Prakondisi : -

Skenario :



Gambar 3.2: Diagram use case *tool* konversi jadwal mengawas ujian

- Dosen memasukan file excel mengawas ujian yang dikeluarkan oleh TU

2. Skenario Melakukan Sorting nama

Deskripsi : Kegiatan mensorting jadwal mengawas.

Aktor : Dosen

Prakondisi : -

Skenario :

- Dosen dapat melakukan sorting nama dari jadwal ujian yang telah berupa iCal sesuai nama yang di inginkan.

3. Skenario Mengunduh File iCal

Deskripsi : Kegiatan Mengunduh file iCal.

Aktor : Dosen

Prakondisi : -

Skenario :

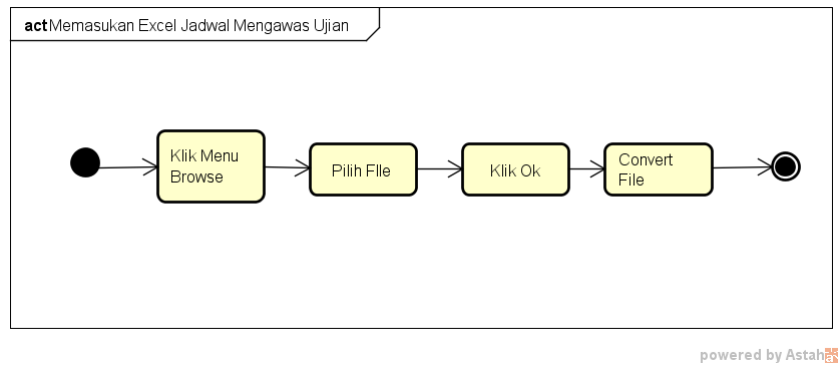
- Dosen mengunduh file iCal yang telah dikonversi oleh *tool*

3.3 Diagram Aktifitas

Pada subbab ini akan dibahas mengenai prosedur setiap aktifitas dari fitu yang diberikan oleh *tool*.

3.3.1 Memasukan Excel Jadwal Mengawas Ujian

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

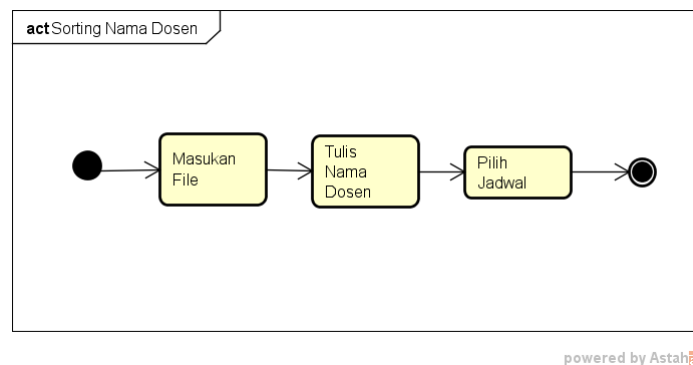


Gambar 3.3: Prosedur Memasukan Excel Jadwal Mengawas Ujian

1. Pengguna mengklik menu Browse yang ada di program.
2. Pengguna memilih file yang akan dimasukan.
3. Pengguna mengklik tombol oke pada window.
4. Pengguna dapat mengklik tombol Convert untuk mengkonversi file excel menjadi iCal.

3.3.2 Sorting Nama Dosen

Tahap ini menjelaskan bagaimana pengguna dapat memanfaatkan fitur dari program dengan men-sorting nama dosen, dengan begitu jadwal dosen yang dicari dapat di unduh dengan mudah. Berikut step-step untuk mensorting nama dosen.



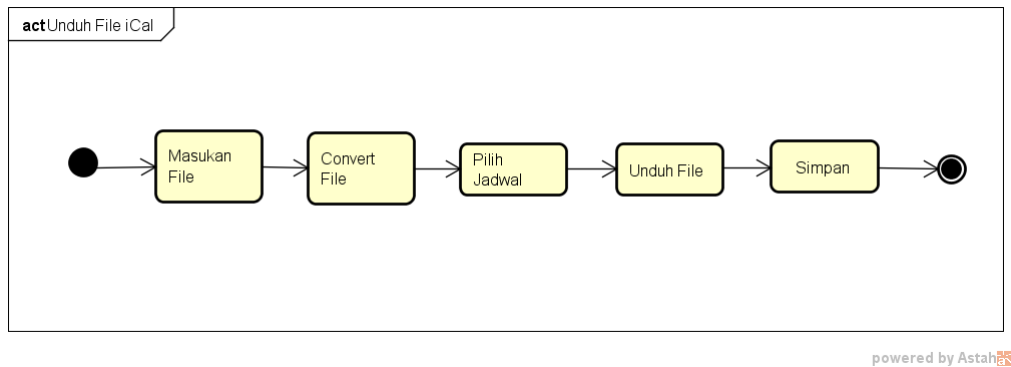
Gambar 3.4: Prodsedur Sorting Nama Dosen

1. Syarat dari penggunaan sorting adalah file harus dimasukan lalu di konversu terlebih dahulu .
2. Setelah file dikonversi, pengguna dapat memasukan nama dosen dicari dengan mengetikan nama pada kolom filter by owner.
3. Setelah pengguna mengetikan nama lalu muncul nama dosen dicari, pengguna dapat mengunduh jadwal sesuai pilihan.

3.3.3 Unduh File iCal

Tahap ini merupakan tahap terakhir, file excel yang telah dimasukan lalu dikonversi menjadi iCal selanjutnya pengguna tinggal memilih jadwal mana yang akan di unduh. File iCal yang telah di

konversi tersebut dapat diintegrasikan dengan *platform* lain seperti google calendar, apple iCal, dll. Berikut step-step untuk mengunduh file iCal.



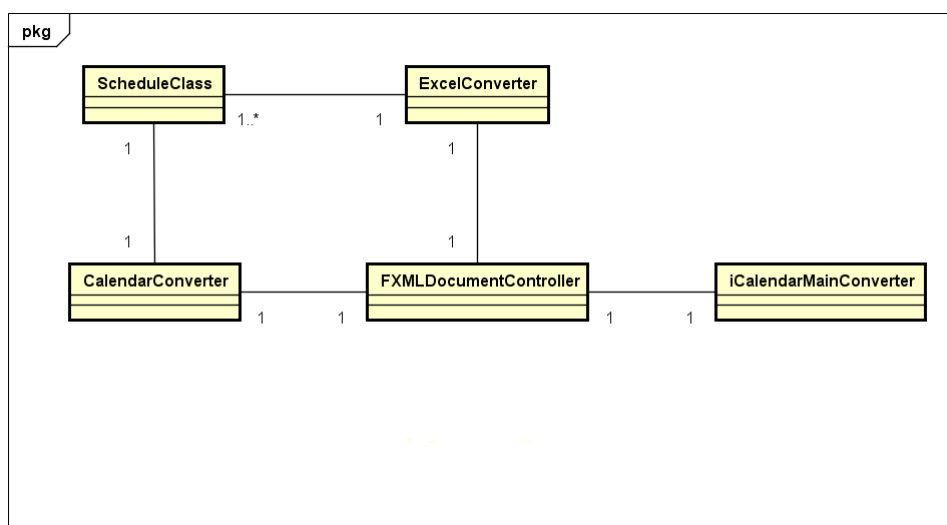
Gambar 3.5: Prosedur Mengunduh File iCal

1. Sebelum mengunduh pengguna diwajibkan mengkonversi terlebih dahulu file jadwal mengawas dengan mengklik tombol Convert pada program.
2. Setelah di konversi, pengguna dapat memilih jadwal yang akan di unduh.
3. Setelah itu, pengguna mengklik tombol iCal pada jadwal dipilih.
4. Akan muncul Pop-Up untuk konfirmasi menyimpan, kemudian klik tombol simpan untuk menyimpan file iCal.

3.4 Pemodelan Kelas

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

Pemodelan Rancangan Kelas



Gambar 3.6: Gambar Pemodelan Kelas

Berikut penjelasan fungsi dari kelas dari gambar 4.1 :

1. **ScheduleClass**

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

2. **ExcelConverter**

Kelas ini bertugas membaca excel jadwal mengawas ujian sehingga dapat ditampilkan oleh program.

3. **FXMLDocumentController**

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

4. **CalendarConverter**

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

5. **iCalendarMainConverter**

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

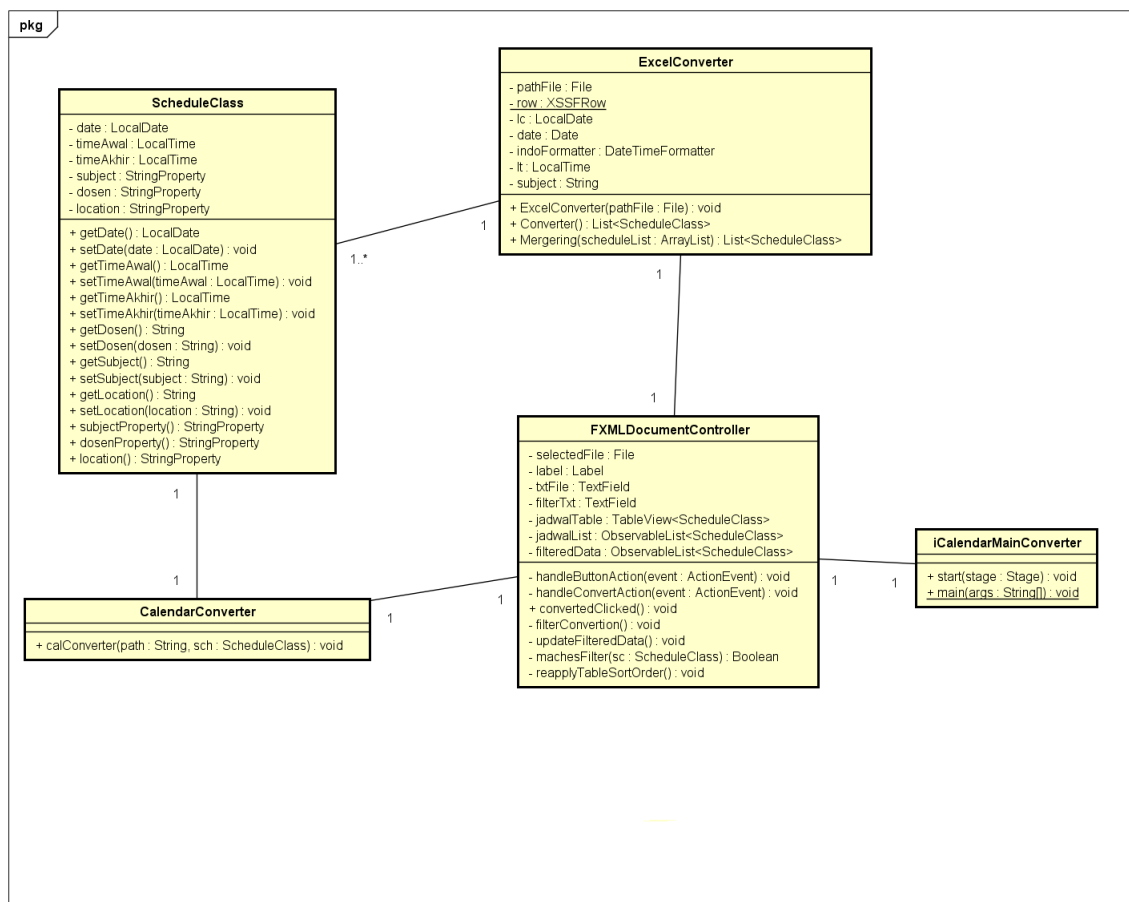
BAB 4

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

Tabel 4.2: Tabel Kelas *ExcelConverter*

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

Tabel 4.3: Tabel Kelas *CalendarConverter*

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

Tabel 4.5: Tabel Kelas *iCalendarMainConverter*

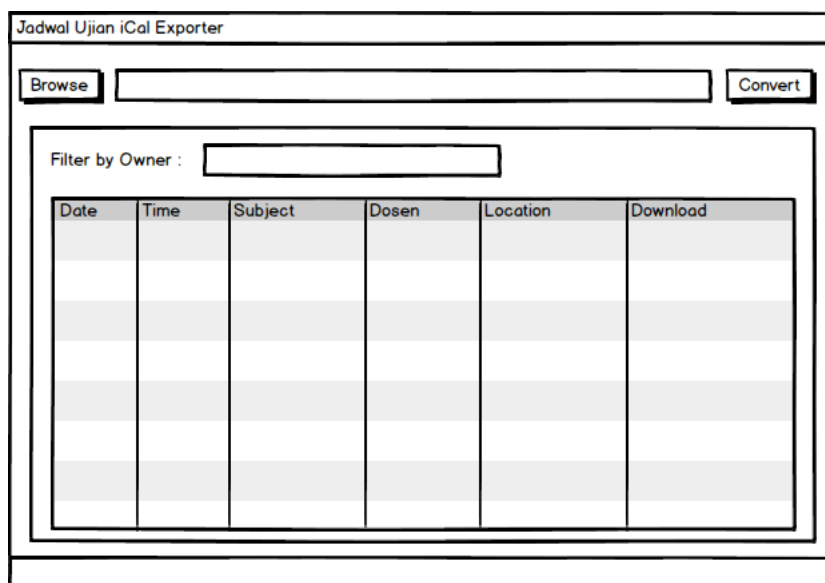
Method	
Nama Method	Fungsi
start(stage: Stage)	Menampilkan window
main(args: String[])	Mengeksekusi program

4.2 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.



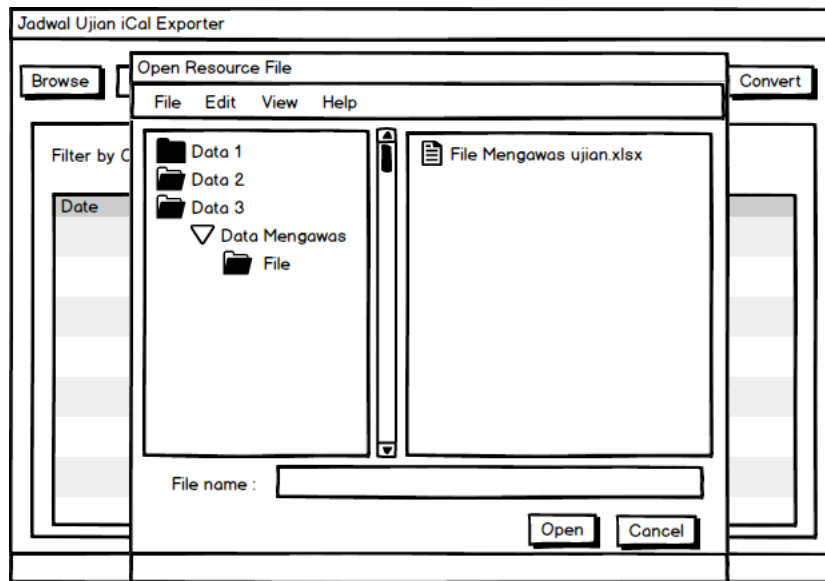
Gambar 4.2: Tampilan awal Program

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 dimasukan.
- **Textbox path** : alamat file yang telah dipilih oleh pengguna akan dicatat pada *textbox* ini.
- **Convert** : tombol ini berfungsi mengeksekusi program untuk membaca file yang telah dimasukan oleh pengguna.
- **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.

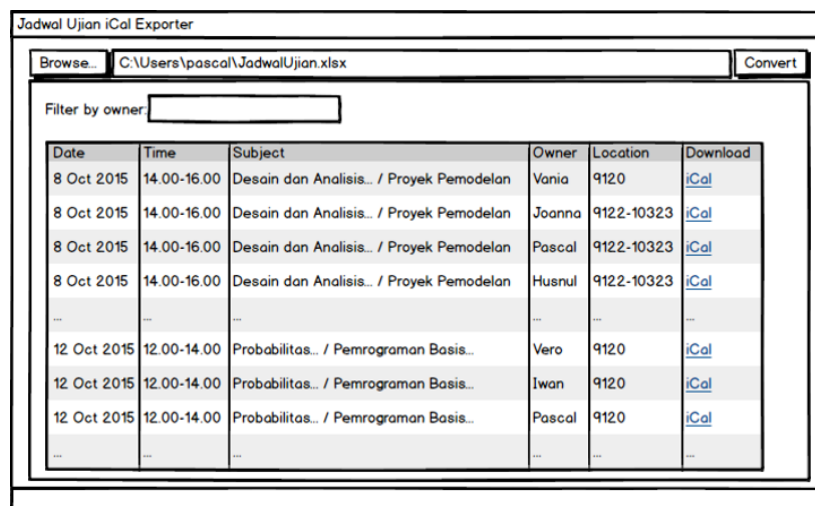
2. Halaman untuk melakukan *Browse* file excel

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

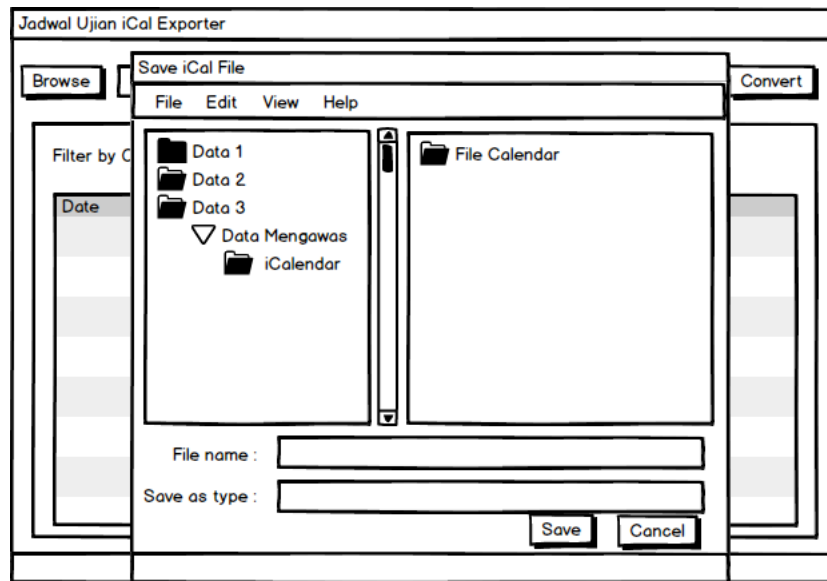
Halaman ini menunjukan ketika file excel telah sukses dibaca dan ditampilkan pada *tableview*.



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

4.3 Rancangan *Method-Method* Utama

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

1. Converter() - ExcelConverter

Input : -
 Output : List<ScheduleClass>
 Deskripsi : Method ini membaca excel yang di *input* oleh pengguna dan mengkonversikannya kedalam bentuk list.
 Algoritma :

- (a) Ambil path file yang telah di *input* oleh pengguna.
- (b) Cari kolom No. pada file excel dan jadikan acuan bahwa program akan membaca setelah dari index kolom tersebut.
- (c) 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.
- (d) Cek apakah baris mengandung kata *LIBUR* bila ya maka lewati saja.
- (e) Pisahkan hari dan tanggal lalu konversi menjadi LocalDate.
- (f) Pisahkan jam menjadi jamAwal dan jamAkhir, lalu konversi menjadi LocalTime.
- (g) jika menemukan kata *Shift* atau *Lab* maka lokasi ujian adalah Lab.
- (h) Masukkan semua kedalam sebuah ArrayList<ScheduleClass>.

2. Mergering() - 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) Masukan nilai yang sama kedalam list filter jika dosen yang dicari sesuai dengan *input* pengguna.
- (c) Atur kembali urutan tabel pada perangkat lunak.

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 [8]
- JRE : Java Runtime Enviroment 8 [8]
- 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 SetProperty, SetProperty memungkinkan untuk memberitahu jika ada perubahan pada *variable* tersebut. *Property* membantu untuk menjaga tampilan agar sinkron dengan data. Pada ScheduleClass *variable* yang menggunakan SetProperty 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 :

- (a) Baris (68-84) pada lampiran [A.2](#) 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 dimasukan 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 dimasukan ke kelas model, pun juga karena setelah kolom tersebut terdapat kolom ruang kuliah yang akan dimasukan kedalam *variable* lokasi pada program. Selain itu, nomer kolom ruangan dapat menjadi acuan lokasi dosen mengawas.
- (b) Baris (85-87) pada lampiran [A.2](#) menjelaskan bahwa i sebagai acuan program membaca baris, sedangkan j sebagai acuan program membaca kolom.
- (c) Baris (89-93) pada lampiran [A.2](#) menjelaskan bahwa bila baris ke i program membaca dan isinya kosong maka berhenti membaca.
- (d) Baris (96-100) pada lampiran [A.2](#) menjelaskan bila isi pada kolom No. bukanlah angka dan *blank* maka program berhenti membaca.
- (e) Baris (101-107) pada lampiran [A.2](#) menjelaskan bila isi kolom No. adalah kosong maka lewati barisnya dan baca baris selanjutnya.
- (f) Baris (108-129) pada lampiran [A.2](#) 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* bertipe LocalDate.
- (g) Baris (130-163) pada lampiran [A.2](#) menjelaskan bahwa bila kolom tersebut adalah jam an 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.

- (h) Baris (164-166) pada lampiran A.2 menjelaskan bila kolom tersebut adalah Nama Mata Kuliah, maka masukan ke *variable* String Subject.
- (i) Baris (173-192) pada lampiran A.2 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 di awali 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.
- (j) Baris (193-215) pada lampiran A.2 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.
- (k) Baris (219-221) pada lampiran A.2 menjelaskan masukan semua *variable* yang diisikan sebelumnya kedalam sebuah ArrayList ScheduleClass sesuai jumlah ArrayList nama dosen.
- (l) Baris (222-223) pada lampiran A.2 menjelaskan hapus semua isi ArrayList dosen agar tidak ada duplikasi.
- (m) Baris (227) pada lampiran A.2 menjelaskan hasil ArrayList method Converter() akan kembali dicek oleh *method* merging().
- (n) Baris (235-242) pada lampiran A.2 menjelaskan jika ada dosen yang isinya kosong pada ArrayList ScheduleList, maka pindahkan isinya ke ArrayList baru yang bernama ScheduleListSmt.
- (o) Baris (243-250) pada lampiran A.2 menjelaskan cara menghapus isi ArrayList ScheduleList yang sama dengan ArrayList SchedulelistSmt.
- (p) Baris (251-263) pada lampiran A.2 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) Baris (264) pada lampiran A.2 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 :

- (a) Baris (41-43) pada lampiran A.3 menjelaskan pembuatan *timeZone* untuk wilayah Indonesia.
- (b) Baris (46-52) pada lampiran A.3 menjelaskan pembuatan tanggal dan waktu dimulainya ujian dengan mengkonversi tahun, bulan, tanggal, jam, dan menit dari ScheduleClass.
- (c) Baris (55-61) pada lampiran A.3 menjelaskan pembuatan tanggal dan waktu ujian tersebut berakhir dengan mengkonversi tahun, bulan, tanggal, jam, dan menit dari ScheduleClass.
- (d) Baris (65-72) pada lampiran A.3 menjelaskan pembuatan *event* pada kalender.
- (e) Baris (80) pada lampiran A.3 memasukan *timeZone* pada kalender.
- (f) Baris (83-85) pada lampiran A.3 menjelaskan identitas pembuat calendar.
- (g) Baris (89-92) pada lampiran A.3 menjelaskan cara pembuatan calendar.

- (h) Baris (95-96) pada lampiran A.3 memasukan *event* yang telah dibuat kedalam kalender dan *print* sesudahnya.
- (i) Baris (99-105) pada lampiran A.3 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.

Berikut ini penjelasan dari kode-kode dalam kelas FXMLDocumentController :

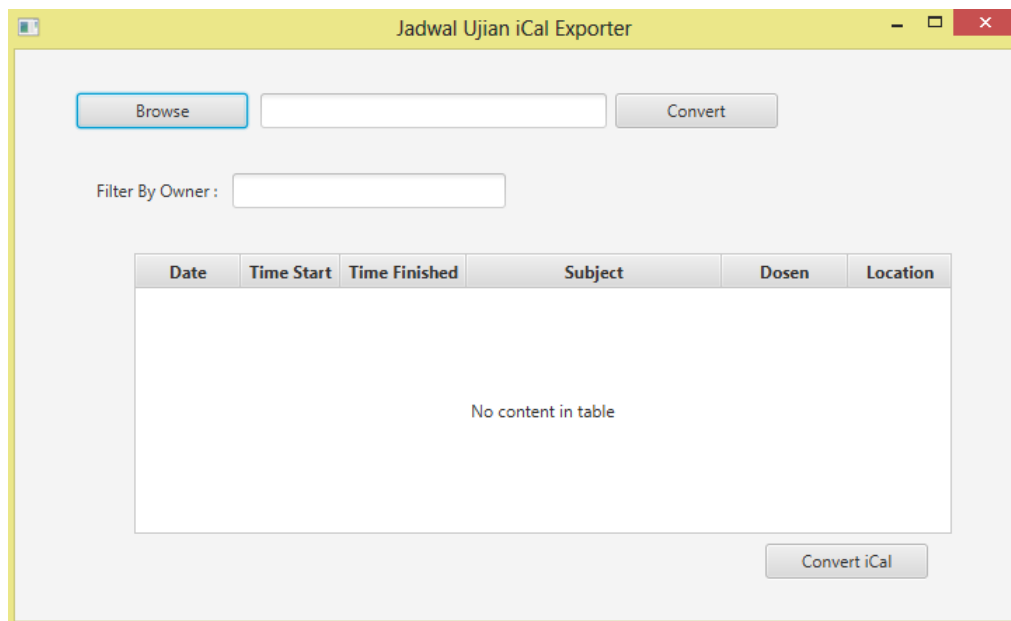
- (a) Baris (55-58) pada lampiran A.4 menjelaskan file yang akan dipilih nanti harus berekstensi .xlsx.
- (b) Baris (59) pada lampiran A.4 menjelaskan bagaimana memunculkan *pop-up window* untuk memilih file *input*.
- (c) Baris (61-68) pada lampiran A.4 menjelaskan jika file tidak kosong maka ambil *pathfile* tersebut dan isikan *textbox browse* dengan *path* file tersebut.
- (d) Baris (74-75) pada lampiran A.4 menjelaskan konversi file excel tersebut, kemudian ambil hasilnya dan masukan kedalam ObservableArrayList.
- (e) Baris (78-84) pada lampiran A.4 menjelaskan cara menampilkan hasil konversi kedalam *tableview* dengan mengisikan sesuai urutan kolom pada *tableview*.
- (f) Baris (86-95) pada lampiran A.4 menjelaskan masukan semua ObservableArrayList *jadwalList* kedalam ObservableArrayList *filteredData* untuk keperluan filter nanti dan jika ada perubahan pada ObservableArrayList *jadwalList* maka *update* pula ObservableArrayList *filteredData*.
- (g) Baris (108) pada lampiran A.4 menjelaskan ambil kelas ScheduleClass yang dipilih oleh *user* dan masukan kedalam *variable* *selected*.
- (h) Baris (111-112) pada lampiran A.4 menjelaskan bahwa setiap file yang akan disimpan diberikan ekstensi .ics .
- (i) Baris (113) pada lampiran A.4 menjelaskan cara memunculkan *save dialog*.
- (j) Baris (117-126) pada lampiran A.4 menjelaskan ambil path direktori dimana *user* akan menyimpan file, lalu konversi jadwal yang telah dipilih oleh *user* dan simpan di direktori yang sudah ditentukan.
- (k) Baris (131-140) pada lampiran A.4 menjelaskan jika *textbox filter* di isi oleh *user* maka jadwal di tabel pun berubah sesuai dengan nama dosen yang di tuliskan *user*.
- (l) Baris (145-154) pada lampiran A.4 menjelaskan cara *update* *filteredData* sesuai nama dosen yang di input oleh *user*.
- (m) Baris (158-170) pada lampiran A.4 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) Baris (175-177) pada lampiran A.4 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 :

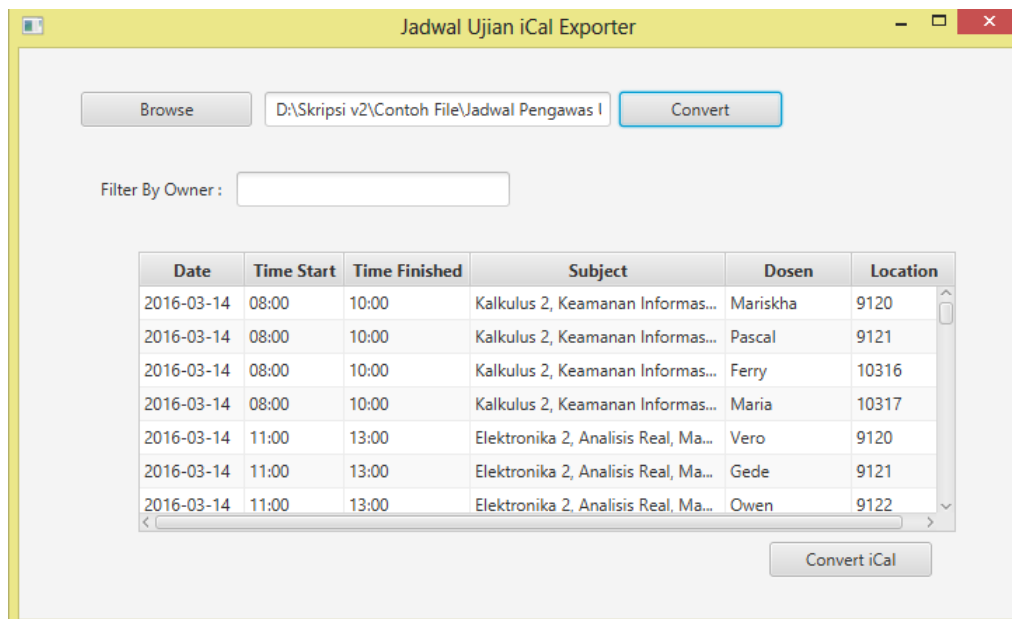
1. Tampilan perangkat lunak iCalendarConverter



Gambar 5.1: Tampilan antarmuka perangkat lunak

Pada tampilan 5.2 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 cursor 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 dimasukan



Gambar 5.2: Tampilan antarmuka setelah file mengawas dimasukan

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.

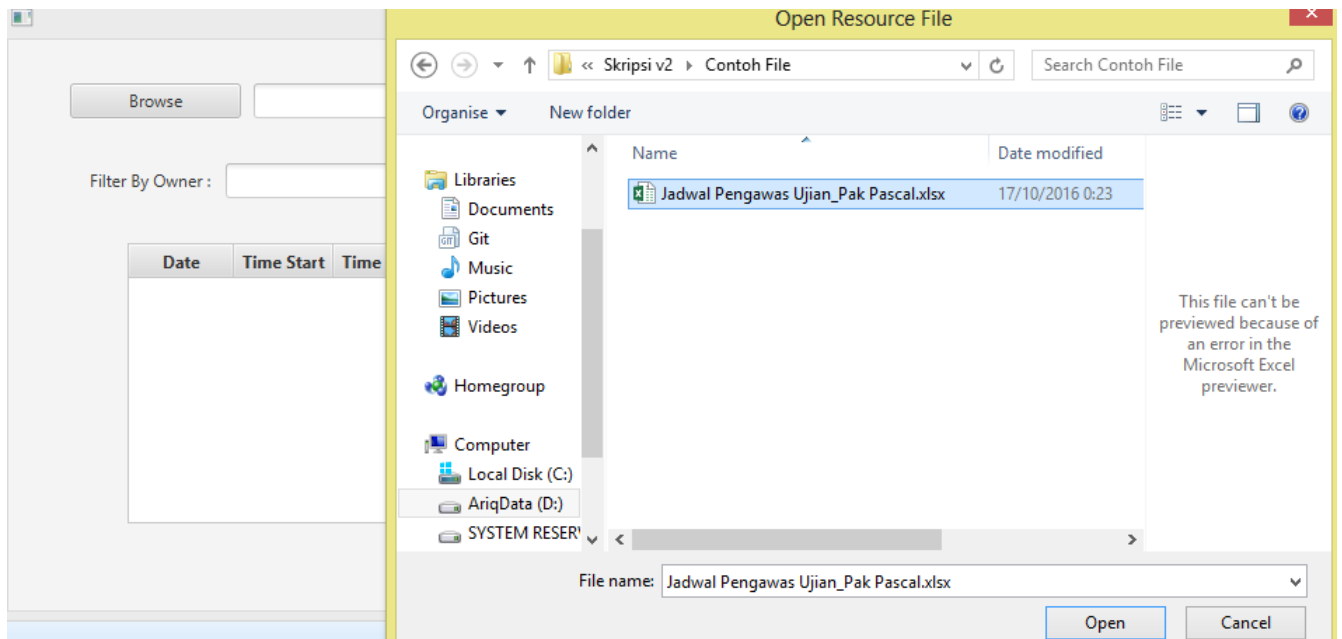
5.3.1 Pengujian Fungsional

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

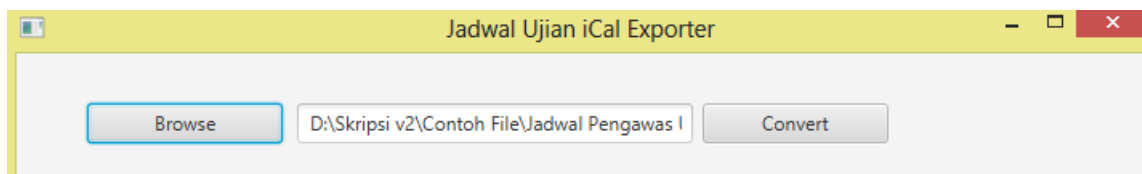
Hal yang diuji	Hasil yang diharapkan	Hasil
Browse file excel	PL dapat melakukan browse file excel	Berhasil 5.3
Path file excel	PL dapat menangkap alamat file dari <i>input file</i> excel	Berhasil 5.4
Menampilkan Jadwal ke layar	PL menampilkan ke layar file excel yang telah dibaca	Berhasil 5.5
Konversi ke iCal	PL dapat mengkonversi jadwal yang diseleksi pengguna kedalam iCalendar	Berhasil 5.6
Filter nama dosen	PL dapat menampilkan nama dosen yang telah di filter, sesuai yang di masukan oleh pengguna pada <i>textbox</i> filter	Berhasil 5.8
Hasil Filter dapat di konversi ke iCal	Hasil Filter pada PL dan diseleksi oleh pengguna, dapat di konversikan kedalam iCal	Berhasil 5.9
Import Google Calendar	Hasil konversi PL dapat di masukan kedalam Google Calendar	Berhasil 5.31
Dapat dibuka di Outlook	Hasil konversi PL dapat di buka di Outlook	Berhasil 5.16
Hasil filter dapat di import Google Calendar	Hasil filter konversi PL dapat di masukan kedalam Google Calendar	Berhasil 5.17
Hasil filter Dapat dibuka di Outlook	Hasil filter konversi PL dapat di buka di Outlook	Berhasil 5.42

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



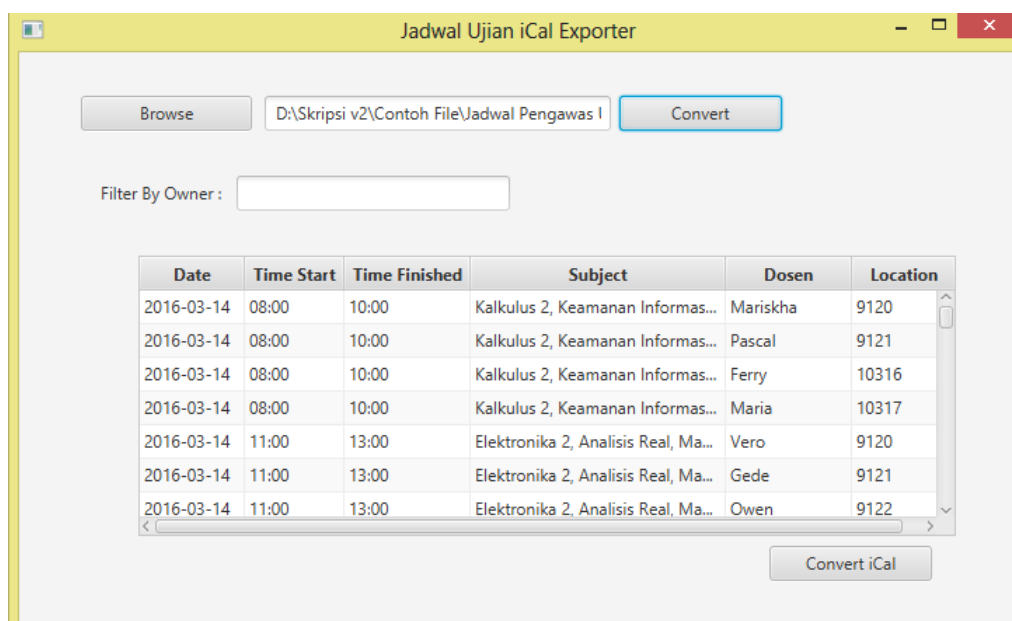
Gambar 5.3: Tampilan browse file excel mengawas ujian

1.



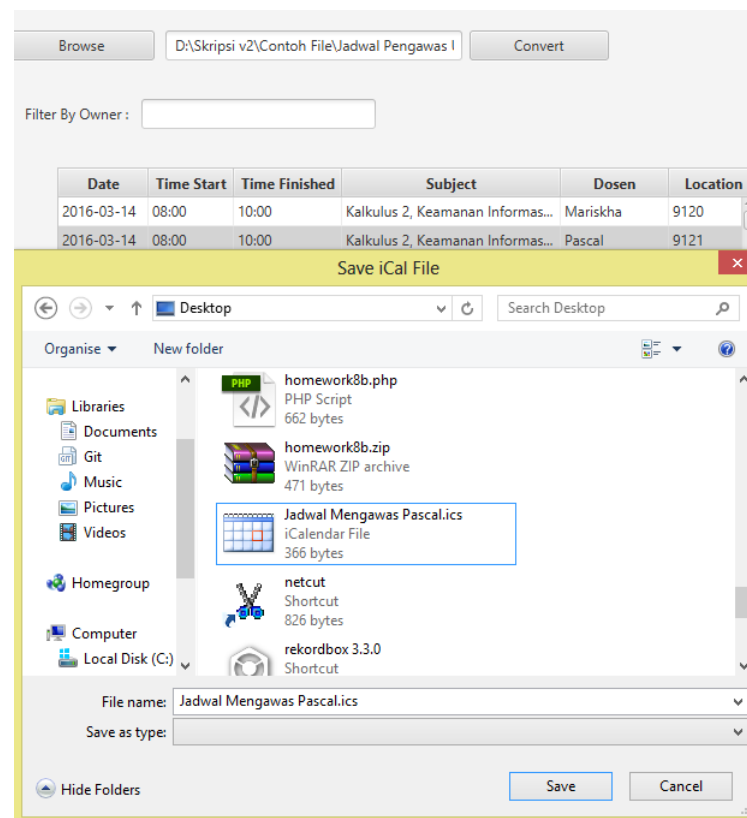
Gambar 5.4: Tampilan path file excel mengawas ujian

2.



Gambar 5.5: PL menampilkan jadwal ke layar

3.



Gambar 5.6: PL mengkonversi jadwal ke format iCal

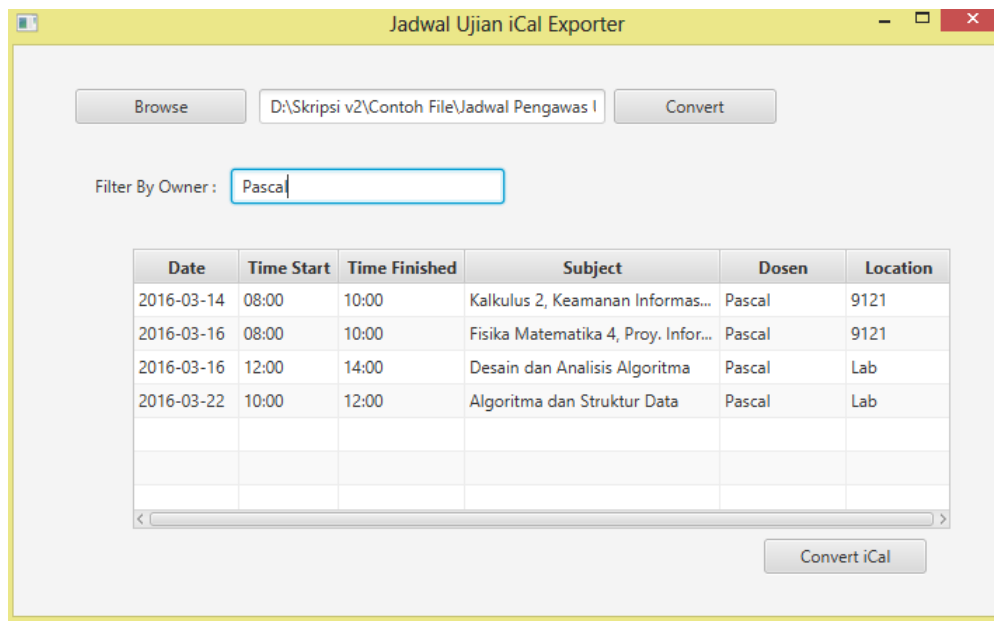
```

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

```

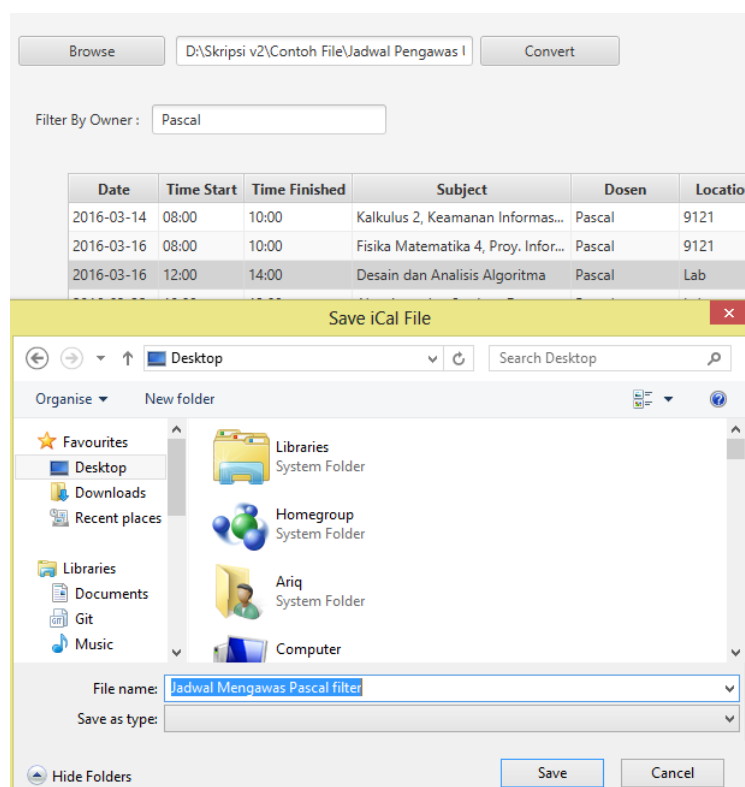
Gambar 5.7: File iCal

4.



Gambar 5.8: Hasil pengujian filter nama dosen

5.



Gambar 5.9: Hasil pengujian convert hasil filter kedalam iCal

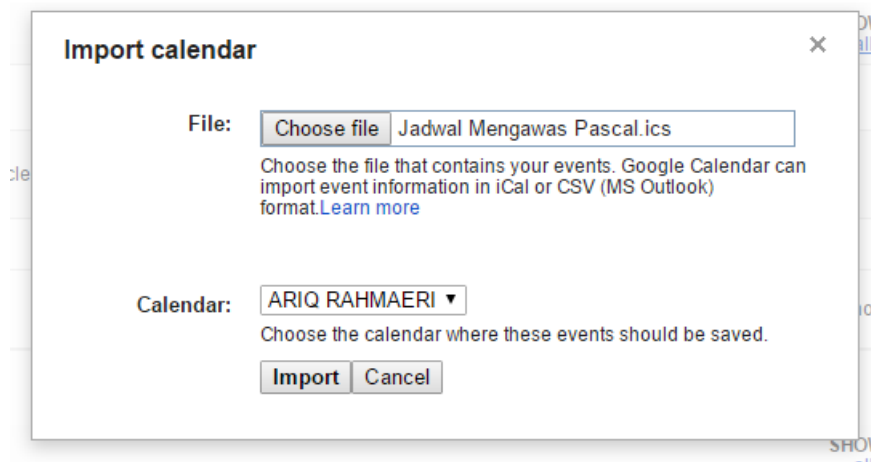
```

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

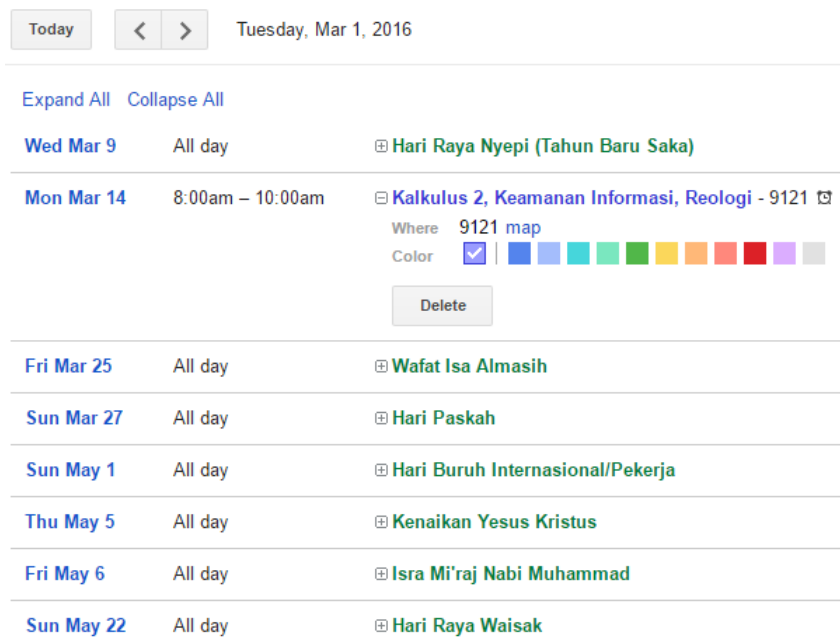
```

Gambar 5.10: File iCal Filter

6.



Gambar 5.11: Hasil pengujian import kedalam Google Calendar



Gambar 5.12: 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 9121 [map](#)

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

Video call [Add video call](#)

Calendar **ARIQ RAHMAERI**

Description Mengawas Ujian Pascal

Guests can

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

Attachment [Add attachment](#)

Event color ☒ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Notifications **Notification** 10 minutes

[Add a notification](#)

Show me as ☐ Available ☒ Busy

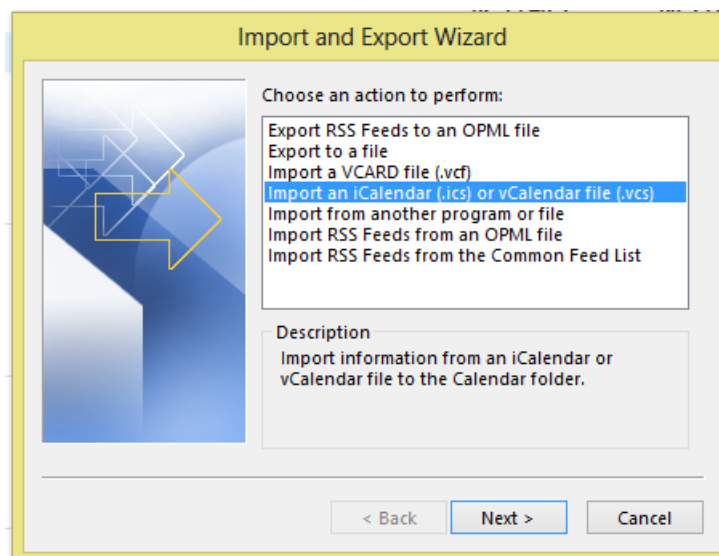
Visibility ☒ Calendar default ☐ Public ☐ Private

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](#)

Gambar 5.13: Hasil import ke Google Calendar bagian 2

7.

Gambar 5.14: *Import* file iCal kedalam MS Outlook

◀ ▶ Maret 2016

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

Gambar 5.15: 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 5.16: File hasil Konversi dapat dibuka di MS Outlook

8.

Import calendar ✕

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: ▼

Choose the calendar where these events should be saved.

Gambar 5.17: Hasil pengujian import file yang di filter kedalam Google Calendar



Gambar 5.18: 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.](#)

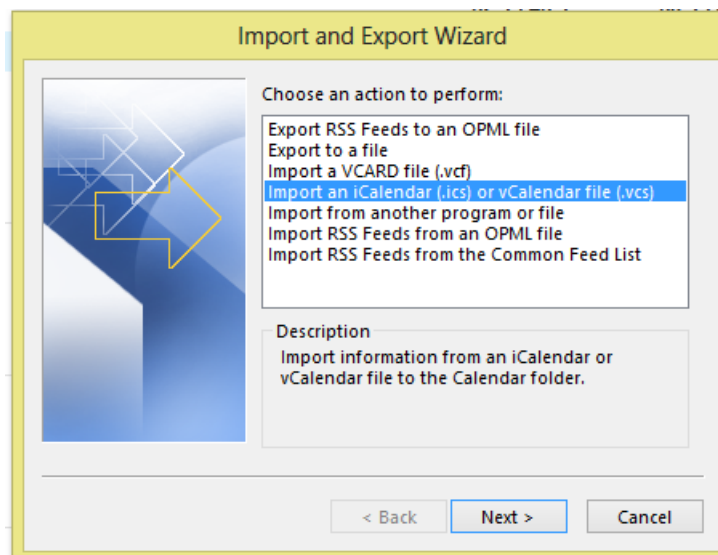
Enter guest email address

Add

Guests can

☐ modify event
 ☒ invite others
 ☒ see guest list

Gambar 5.19: Hasil import file yang di filter ke Google Calendar bagian 2



Gambar 5.20: *Import* file iCal yang telah di filter kedalam MS Outlook

◀ ▶ Maret 2016

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

Gambar 5.21: 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 5.22: File hasil filter dapat dibuka di MS Outlook

10.

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 5.3
Path file excel	PL dapat menangkap alamat file dari <i>input file</i> excel	Berhasil 5.4
Memasukan file yang bukan excel	PL mengeluarkan notifikasi kesalahan file <i>input</i>	Berhasil 5.24
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 5.23

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 5.23: *Output* PL pada pengujian eksperimental

1.

```

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.POIXMLException: org.apache.poi.openxml4j.exceptions.InvalidFormatException: Package should contain
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 5.24: *Output* PL pada pengujian eksperimental dari file *input* yang bukan excel

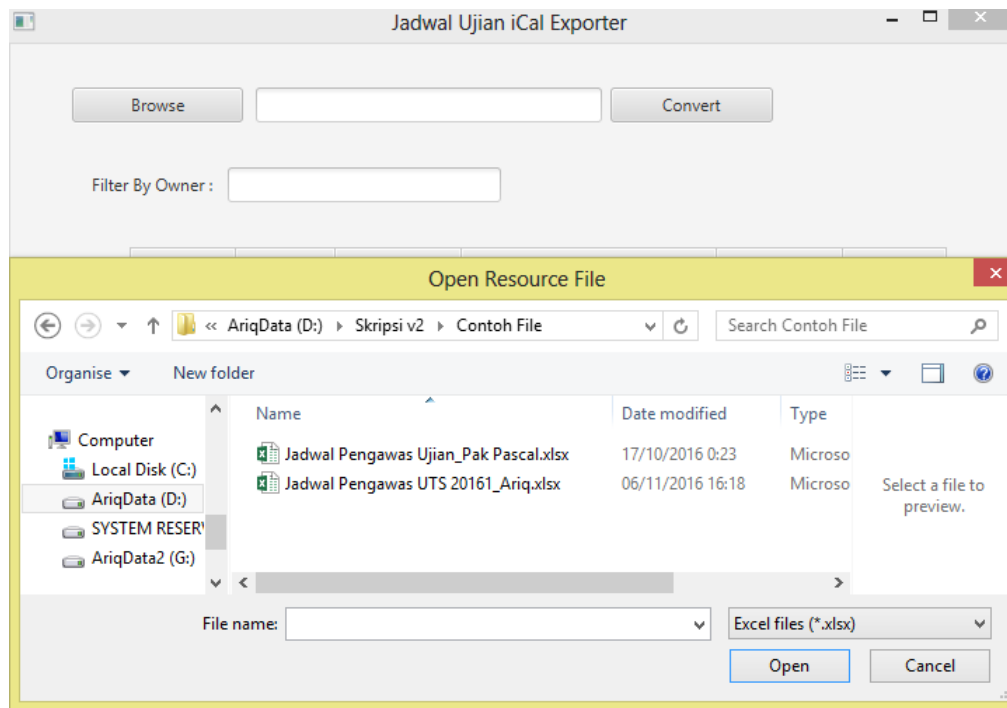
2.

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

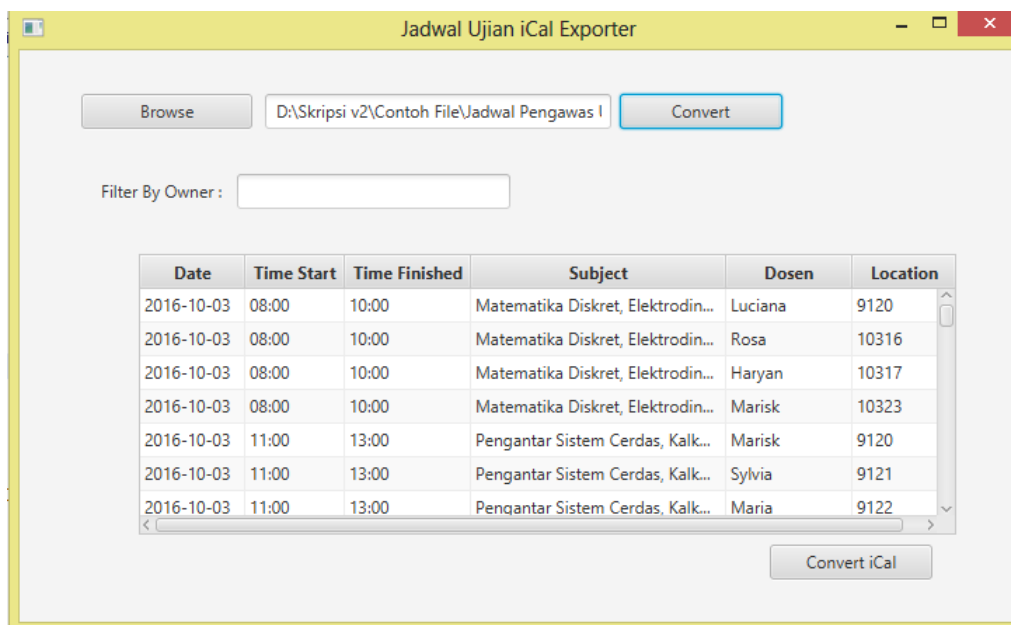
Hal yang diuji	Hasil yang diharapkan	Hasil
Browse file excel	PL dapat melakukan browse file excel	Berhasil 5.3
Path file excel	PL dapat menangkap alamat file dari <i>input file</i> excel	Berhasil 5.4
Memasukan file yang bukan excel	PL menggunakan <i>extension filter</i> sehingga file bukan excel tidak dapat menjadi file <i>input</i>	Berhasil 5.25
Menampilkan Jadwal ke layar	PL menampilkan ke layar file excel yang telah dibaca	Berhasil 5.26
Konversi ke iCal	PL dapat mengkonversi jadwal yang diseleksi pengguna kedalam iCalendar	Berhasil 5.27
Filter nama dosen	PL dapat menampilkan nama dosen yang telah di filter, sesuai yang di yang dimasukan oleh pengguna pada <i>textbox</i> filter	Berhasil 5.8
Hasil Filter dapat di konversi ke iCal	Hasil Filter pada PL dan diseleksi oleh pengguna, dapat di konversikan kedalam iCal	Berhasil 5.29
Import Google Calendar	Hasil konversi PL dapat di masukan kedalam Google Calendar	Berhasil 5.32
Dapat dibuka di Outlook	Hasil konversi PL dapat di buka di Outlook	Berhasil 5.35
Hasil filter dapat di import Google Calendar	Hasil filter konversi PL dapat di masukan kedalam Google Calendar	Berhasil 5.38
Hasil filter Dapat dibuka di Outlook	Hasil filter konversi PL dapat di buka di Outlook	Berhasil 5.41

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



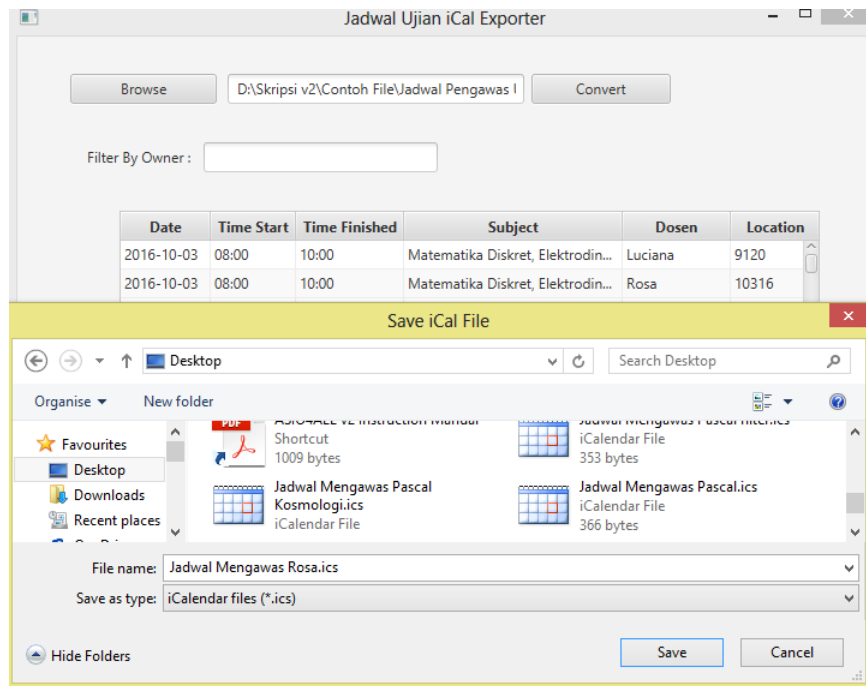
Gambar 5.25: PL memfilter ekstensi file excel yang akan dimasukan

1.



Gambar 5.26: PL menampilkan jadwal ke layar

2.



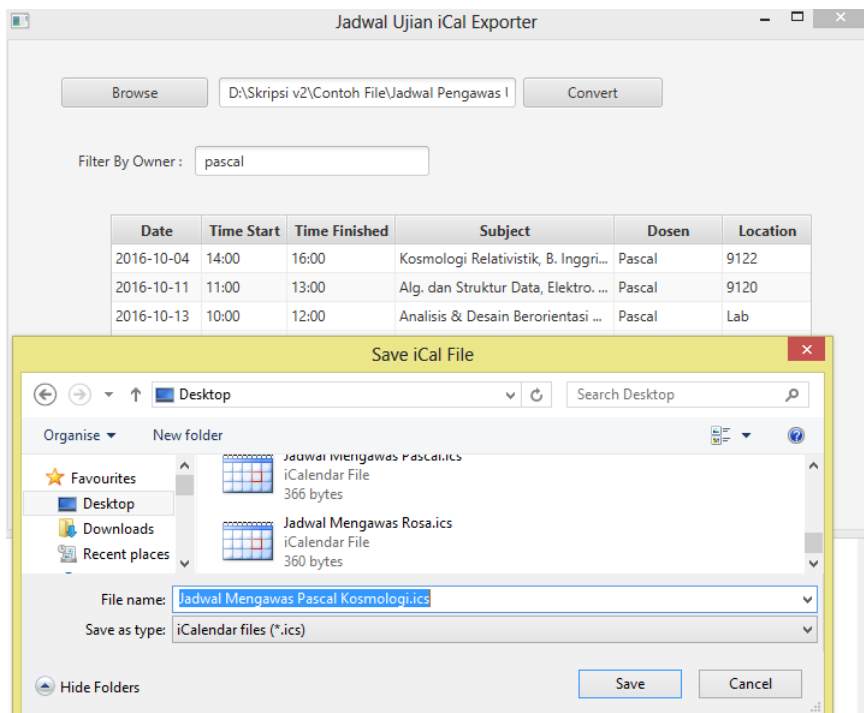
Gambar 5.27: PL mengkonversi jadwal ke format iCal

```

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 5.28: File iCal



Gambar 5.29: Hasil pengujian convert hasil filter kedalam iCal

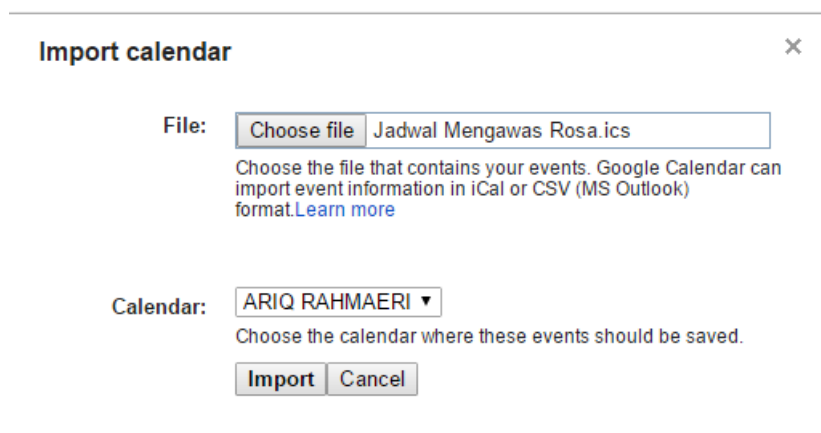
```

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..., Str. SI dan Ars. Perush. Berskala Bsr.
LOCATION:9122
DESCRIPTION:Mengawas Ujian Pascal
TZID:Asia/Jakarta
UID:20161116T211248Z-uidGen@AriqRahmaeri
END:VEVENT

```

Gambar 5.30: File iCal Filter

4.



Gambar 5.31: Hasil pengujian import kedalam Google Calendar

Expand All	Collapse All	
Thu Sep 29	10:00am – 11:00am	Bimbingan Ariq - UNPAR
Sun Oct 2	All day	Sabtu Muharam/Tahun Baru Hijriyah
Mon Oct 3	8:00am – 10:00am	Matematika Diskret, Elektrodinamika - 10316 <div> <div>Where</div> <div>10316 map</div> <div> <div>Color</div> <div> <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"/> </div> </div> <div> <div>Delete</div> </div> </div>
Wed Oct 12	10:00am – 11:00am	Bimbingan Ariq - UNPAR

Gambar 5.32: Hasil import ke Google Calendar

Matematika Diskret, Elektrodinamika

to

[Time zone](#)

☐ All day
 ☐ Repeat...

Event details
 [Find a time](#)

Where

[map](#)

Video call
 [Add video call](#)

Calendar

Description

Mengawas Ujian Rosa

Attachment
 [Add attachment](#)

Event color
 ☒
☐
☐
☐
☐
☐
☐
☐
☐
☐

Notifications

[Add a notification](#)

Show me as
 ☐ Available
 ☒ Busy

Visibility
 ☒ Calendar default
 ☐ Public
 ☐ Private

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

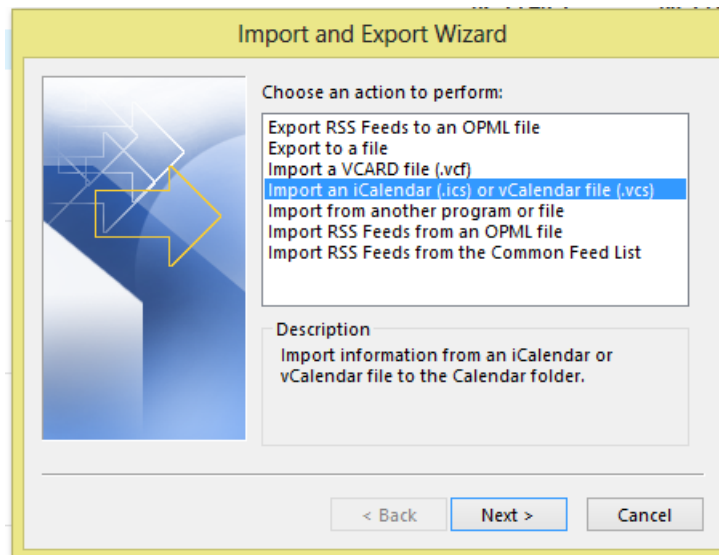
Guests can
 ☐ modify event
 ☒ invite others
 ☒ see guest list

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 5.33: Hasil import ke Google Calendar bagian 2

5.

Gambar 5.34: *Import* file iCal 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

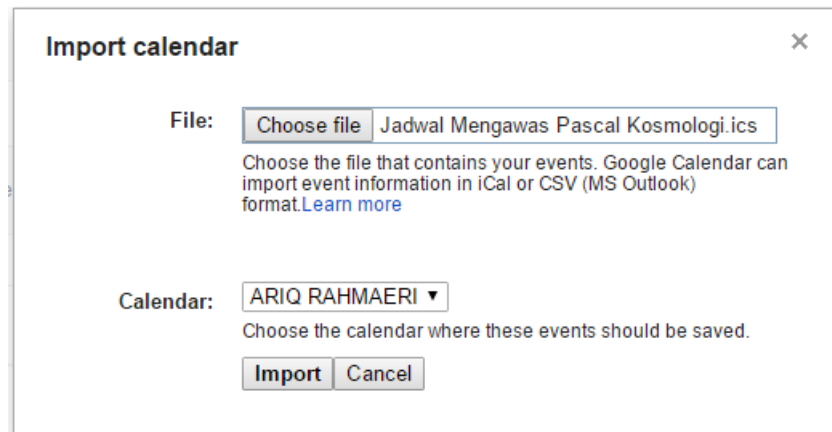
Gambar 5.35: File hasil Konversi dapat dibuka di MS Outlook

◀ ▶ Oktober 2016

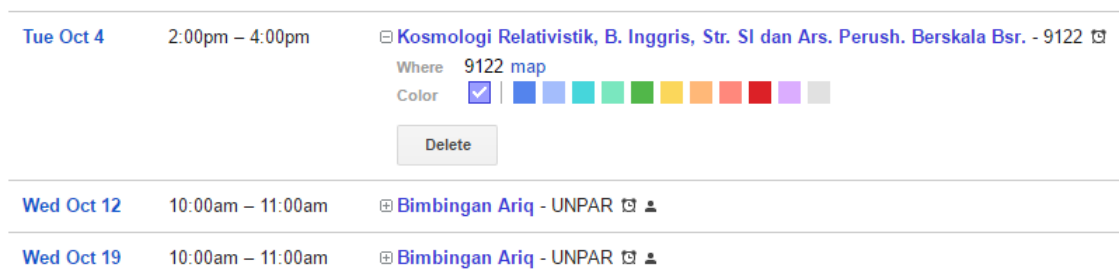
SENIN	SELASA
26 Sep	27
3 8:01 Matematika Diskret, Elektrodinamika; 10316	4
10	11
17	18

Gambar 5.36: File hasil Konversi berada pada kalender MS Outlook

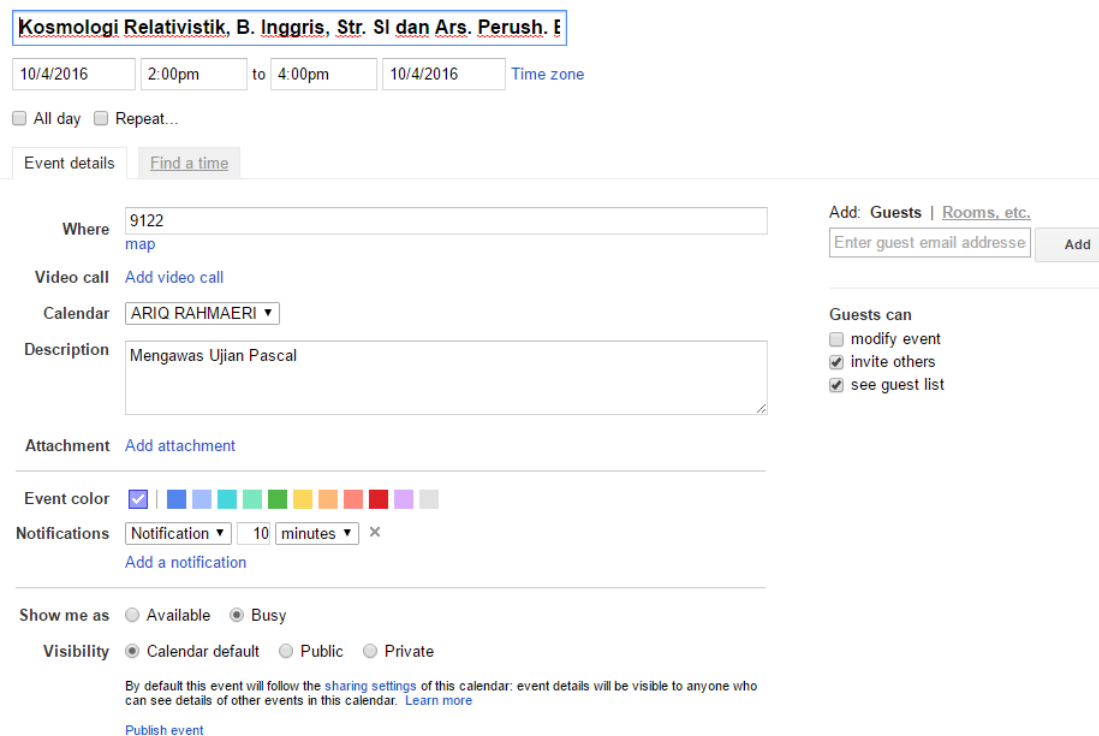
6.



Gambar 5.37: Hasil pengujian import file yang di filter kedalam Google Calendar

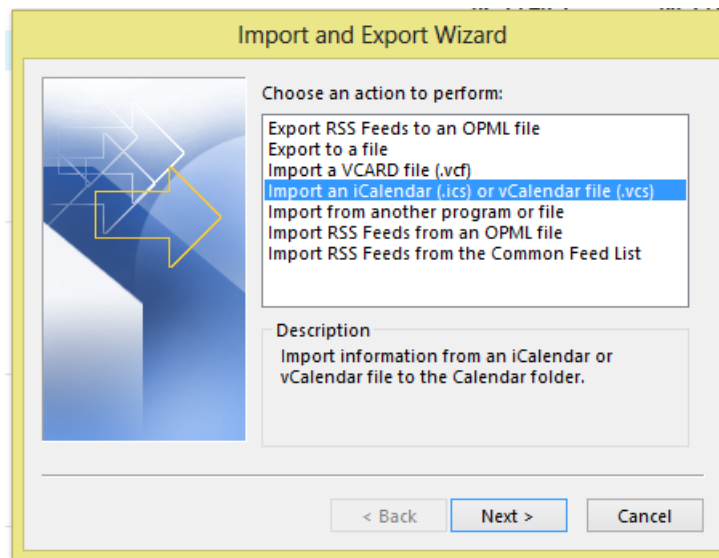


Gambar 5.38: Hasil import file yang di filter ke Google Calendar



Gambar 5.39: Hasil import file yang di filter ke Google Calendar bagian 2

7.

Gambar 5.40: *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 5.41: 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 5.42: File iCal yang telah di filter berada pada kalender MS Outlook

8.

DAFTAR REFERENSI

- [1] The Apache Software Foundation (2016) Apache POI - the Java API for Microsoft Documents. <https://poi.apache.org/>. [Online; diakses 23-09-2016].
- [2] The Apache Software Foundation (2016) POI API Documentation. <http://poi.apache.org/apidocs/index.html>. [Online; diakses 23-09-2016].
- [3] Ben Fortune (2007) ical4j. <http://ical4j.sourceforge.net/introduction.html>. [Online; diakses 14-09-2016].
- [4] 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].
- [5] Oracle and/or its affiliates (2013) What is javafx? <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>. [Online; diakses 28-09-2016].
- [6] Oracle and/or its affiliates (2013) JavaFX Architecture. <http://docs.oracle.com/javafx/2/architecture/jfxpub-architecture.htm>. [Online; diakses 28-09-2016].
- [7] Oracle and/or its affiliates (2015) Javafx 8. <http://docs.oracle.com/javase/8/javafx/api/toc.htm>. [Online; diakses 28-09-2016].
- [8] 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.DESCRPTION).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     return Mergering(scheduleList);
184 }
185
186 public List<ScheduleClass> Mergering (ArrayList<ScheduleClass> scheduleList)
187 {
188     int count = 0;
189     ArrayList<ScheduleClass> scheduleListSmt = new ArrayList<>();
190
191     for (int i = 0; i < scheduleList.size(); i++) {
192
193         if (scheduleList.get(i).getDosen().isEmpty() )
194         {
195             scheduleListSmt.add(scheduleList.get(i));
196         }
197     }
198
199     for (int i = 0; i < scheduleListSmt.size(); i++) {
200         for (int j = 0; j < scheduleList.size(); j++) {
201             if (scheduleList.get(j).equals(scheduleListSmt.get(i)))
202             {
203                 scheduleList.remove(j);
204             }
205         }
206     }
207
208     for (int i = 0; i < scheduleList.size(); i++) {
209         outerloop :
210         for (int j = 0; j < scheduleListSmt.size(); j++) {
211             if (scheduleList.get(i).getDate().equals(scheduleListSmt.get(j).getDate())
212                 && scheduleList.get(i).getTimeAwal().equals(scheduleListSmt.get(j).getTimeAwal()))
213             {
214                 String ss = scheduleList.get(i).getSubject();
215                 scheduleList.get(i).setSubject(ss+" , "+scheduleListSmt.get(j).getSubject());
216                 j = j + 1;
217                 break outerloop;
218             }
219         }
220     }
221     return scheduleList;
222 }
223 }

```


LAMPIRAN B

FILE EXCEL

JADWAL MENGAWAS UTS SEM. GENAP 2015/2016

No.	TIS-UNPAR Hari, Tgl.	Jam	Sem.	PS	Nama Mata kuliah	Ruang					
						9120	9121	9122	10316	10317	10323
1	Senin, 14 Mrt. 2016	08.00-10.00	2	MA	Kalkulus 2	Mariskha	Pascal		Ferry	Maria	
2	Senin, 14 Mrt. 2016	08.00-10.00	6p	IT	Kemanan Informasi, Reologi						
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, Pemr. Apl. Bergerak						
8	Selasa, 15 Mrt. 2016	08.00-10.00	p	MA	Komputasi Keuangan				Reinard	Gede	Liem
9	Selasa, 15 Mrt. 2016	11.00-13.00	6	FI	Mekanika Kuantum, Pers. Dif. Biasa						
10	Selasa, 15 Mrt. 2016	11.00-13.00	6p	IT	Jaringan Syaraf Tiruan				Bagoes	Iwan	Sylvia
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	Benny	Iwan	Husnul, Vero			
41	Rabu, 23 Mrt. 2016	14.00-16.00	4	MA	Aljabar Linear						
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	Pengalihan 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,

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				Gede	Claudio		
18	Rabu, 5 Okt. '16	14.00-16.00	Komputer dan Masyarakat										
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										
21	Kamis, 6 Okt. '16	11.00-13.00	Sistem dan Aplikasi Telematika	Risti	Gede	Benny							
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							Luciana	Vero	Rusli	
53	Kamis, 13 Okt. '16	08.00-10.00	Penambangan Data (Data Mining)										
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			Chandra	Haryan	Risti	
63	Jumat, 14 Okt. '16	13.30-15.30	Pemr. Komputer, Layanan Berb. WEB										

Bandung,

Wakil Dekan Bidang Sumber Daya,

Maria Anastasia, M.Si.

Gambar B.2: File Excel Jadwal Mengawas Format Lama