

TUGAS AKHIR

PERANGKAT LUNAK SIMULASI FRS



Fransiskus Anugrah Putra Nusantara Witjaksono

NPM: 6181901012

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2024

FINAL PROJECT

FRS SIMULATION SOFTWARE



Fransiskus Anugrah Putra Nusantara Witjaksono

NPM: 6181901012

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2024**

LEMBAR PENGESAHAN

PERANGKAT LUNAK SIMULASI FRS

Fransiskus Anugrah Putra Nusantara Witjaksono

NPM: 6181901012

Bandung, 21 Juni 2024

Menyetujui,

Pembimbing Utama
Digitally signed
by Mariskha Tri
Adithia

Mariskha Tri Adithia, P.D.Eng

Pembimbing Pendamping
Digitally signed
by Raymond
Chandra Putra

Raymond Chandra Putra, M.T.

Ketua Tim Penguji
Digitally signed
by Husnul
Hakim

Husnul Hakim, M.T.

Anggota Tim Penguji
Digitally signed
by Pascal
Alfadian Nugroho

Pascal Alfadian, Nugroho, M.Comp.

Mengetahui,

Ketua Program Studi

Digitally signed
by Lionov

Lionov, Ph.D.

PERNYATAAN

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

PERANGKAT LUNAK SIMULASI FRS

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 21 Juni 2024



Fransiskus Anugrah Putra Nusantara Witjaksono
NPM: 6181901012

ABSTRAK

Pengisian Form Rencana Studi (FRS) merupakan hal yang perlu dilakukan oleh setiap mahasiswa pada masa studinya di suatu universitas. FRS merupakan sebuah gambaran dari kegiatan perkuliahan setiap mahasiswa. Di Universitas Katolik Parahyangan (UNPAR), pengisian FRS dilakukan pada sebuah sistem informasi yang bernama *Student Portal*. Namun, mahasiswa sering mengalami kesulitan untuk memilah setiap jadwal untuk mata kuliah yang diambil, dikarenakan pada sistem yang sudah ada belum dapat mengidentifikasi bentrok antara jadwal satu dengan yang lainnya. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah perangkat lunak yang dapat mensimulasikan kegiatan pengisian FRS.

Pada tugas akhir ini dilakukan analisis bagaimana alur dari awal sebuah mata kuliah dibuka, hingga mata kuliah diambil oleh mahasiswa untuk diikuti pada semester yang akan datang. Pada hasil analisis yang didapat, dilakukan perancangan terkait perangkat lunak yang dapat mensimulasikan kegiatan pengisian FRS yang memiliki fitur utama yakni pengecekan bentrok jadwal baik sesi perkuliahan dan ujian serta pencetakan jadwal yang sudah benar. Fitur pengecekan bentrok jadwal sesi dan jadwal ujian dikerjakan dengan bantuan sebuah algoritma bernama *Schedule Spread*. Lalu, pembangunan perangkat lunak juga dilakukan pada sebuah *Framework Next.js* dengan beberapa *library* pendukung seperti *Nextauth.js* dan *Papaparse*.

Perangkat lunak yang sudah dibangun juga sudah melalui tahap pengujian baik fungsional dan eksperimental. Pada pengujian fungsional dilakukan pengujian terkait fitur utama yakni pengecekan bentrok jadwal sesi kuliah, praktikum, maupun responsi, jadwal ujian serta pengujian fitur pencetakan jadwal yang sudah benar. Lalu dari sisi pengujian eksperimental, dilakukan juga pengujian dengan melibatkan beberapa responden untuk mencoba perangkat lunak secara langsung. Hasil dari pengujian fungsional menunjukkan hasil yang baik sementara pengujian eksperimental memberikan beberapa masukan, namun pada umumnya menunjukkan bahwa perangkat lunak dapat memudahkan mahasiswa dalam melakukan pengecekan bentrok jadwal secara otomatis.

Kata-kata kunci: Formulir Rencana Studi(FRS), pencetakan jadwal, *Student Portal*, *Framework Next.js*, *Schedule Spread*, *Papaparse*

ABSTRACT

The completion of the Study Plan Form (FRS) is a necessary task for every student during their studies at a university. The FRS provides an overview of each student's coursework activities. At Parahyangan Catholic University (UNPAR), the FRS is completed through an information system called the Student Portal. However, students often face difficulties in sorting out the schedules for their courses, as the existing system does not identify conflicts between schedules. Therefore, this research aims to develop software that can simulate the FRS completion process.

In this thesis, an analysis is conducted on the workflow from the moment a course is opened until it is taken by students for the upcoming semester. Based on the analysis results, a software design is created to simulate the FRS completion process, featuring the primary capability of checking schedule conflicts for both lecture and exam sessions and printing the finalized schedule. The schedule conflict checking for sessions and exams is handled using an algorithm called Schedule Spread. The software development is carried out using the Next.js framework, along with several supporting libraries such as Nextauth.js and Papaparse.

The developed software has undergone both functional and experimental testing. Functional testing involves examining the main features, such as the conflict checking for lecture, lab, and tutorial sessions, as well as exam schedules and the printing of the finalized schedule. Experimental testing includes involving several respondents to try the software directly. The results from functional testing show satisfactory outcomes, while the experimental testing provides some feedback, but generally indicates that the software can facilitate students in automatically checking for schedule conflicts.

Keywords: Study Plan Form (FRS), printing the schedule, Student Portal, Framework, Next.js, Schedule Spread, Papaparse

Penulis mempersembahkan Tugas Akhir ini untuk keluarga tercinta.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang tulus kepada semua pihak yang telah memberikan bantuan, dukungan, dan bimbingan selama proses pengerjaan tugas akhir ini.

1. Pertama, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada ayah penulis yang senantiasa menemani dan memberikan dukungan tak henti-hentinya.
2. Penulis juga sangat berterima kasih kepada Ibu Mariskha Tri Adithia dan Bapak Raymond Chandra Putra selaku pembimbing utama dan pendamping yang selalu memberikan masukan berharga serta arahan selama proses pengerjaan tugas akhir ini.
3. Penulis juga ingin menyampaikan terima kasih yang mendalam kepada Brigitta Stacia Maharani yang selalu memberikan dukungan dan semangat, serta menghibur penulis sehingga penulis dapat melalui setiap tantangan dengan lebih baik.
4. Ucapan terima kasih juga penulis sampaikan kepada Martin Sentoriyan, Dionisius Herdian, dan Ruben Remalya, sebagai teman dekat yang selalu memberikan masukan dan dorongan selama proses pengerjaan tugas akhir ini.
5. Semua pihak yang tidak dapat penulis sebutkan satu per satu namun telah memberikan kontribusi dan dukungan yang berarti dalam menyelesaikan tugas akhir ini.

Tujuan dari tugas akhir ini adalah untuk menemukan manfaat dari apa yang telah diteliti dan kemudian dapat membantu atau mempermudah pekerjaan orang lain. Penulis menyadari bahwa dalam tugas akhir ini masih banyak terdapat kesalahan dan kekurangan, untuk itu penulis memohon maaf yang sebesar-besarnya dan sangat mengharapkan kritik serta saran yang membangun. Semoga tugas akhir ini dapat memberikan manfaat, menjadi inspirasi bagi peneliti berikutnya, dan menambah pengetahuan bagi para pembacanya.

Bandung, Juni 2024

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE PROGRAM	xxviii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	4
1.4 Batasan Masalah	4
1.5 Metodologi	4
2 LANDASAN TEORI	5
2.1 Formulir Rencana Studi	5
2.2 Proses Perwalian dengan Dosen Wali ^[1]	6
2.3 Pengisian Form Rencana Studi (FRS) ^[1]	6
2.4 Pemrograman Basis Web ^[2]	7
2.4.1 <i>Hypertext Markup Language (HTML)</i>	7
2.4.2 <i>Cascading Style Sheets</i>	8
2.4.3 <i>Javascript</i>	8
2.5 <i>React</i> ^[3]	9
2.5.1 <i>Component</i>	9
2.5.2 <i>React Hooks</i>	10
2.5.3 <i>State dan useState</i>	10
2.5.4 <i>Effect dan useEffect</i>	10
2.6 <i>Next.js</i>	11
2.7 <i>Papaparse</i>	12
2.8 <i>PostgreSQL</i>	12
2.9 <i>NextAuth.js</i>	13
2.10 Perangkat Lunak Basis Web yang <i>User Friendly</i>	14
2.11 Algoritma <i>Schedule Spread</i>	15
3 ANALISIS	17
3.1 Analisis Masalah	17
3.2 Spesifikasi Kebutuhan	18
3.3 Modifikasi Algoritma	20
3.3.1 Modifikasi Algoritma untuk Jadwal Kuliah	20
3.3.2 Studi Kasus Jadwal Kuliah Bentrok Sebagian di Depan	22

3.3.3	Studi Kasus Jadwal Kuliah Bentrok Keseluruhan	23
3.3.4	Studi Kasus Jadwal Kuliah Bentrok Sebagian di Belakang	25
3.3.5	Hasil Modifikasi Algoritma untuk Jadwal Ujian	25
3.3.6	Studi Kasus Jadwal Ujian Bentrok Keseluruhan	27
3.3.7	Studi Kasus Jadwal Ujian Bentrok Sebagian di Depan	27
3.3.8	Studi Kasus Jadwal Kuliah Bentrok Sebagian di Belakang	28
3.4	Analisis Proses Bisnis	29
3.4.1	Proses Pembukaan Mata Kuliah	29
3.4.2	Pemilihan Mata Kuliah	31
3.4.3	Proses Perwalian dengan Dosen Wali	32
3.4.4	Proses Pengisian FRS	33
3.5	Analisis Data Jadwal Mata Kuliah	34
3.6	Analisis Kebutuhan <i>Template File CSV</i> dan Penyiapan Data	35
3.6.1	Kebutuhan <i>Template CSV</i> Jadwal Kuliah	35
3.6.2	Kebutuhan <i>Template CSV</i> untuk Jadwal Ujian	37
3.7	Analisis Pemilihan Teknologi	39
3.7.1	<i>Next.js</i>	39
3.7.2	<i>NextAuth.js</i>	39
3.7.3	<i>PostgreSQL</i>	40
3.7.4	<i>Papaparse</i>	40
3.8	Analisis Perangkat Lunak	40
3.8.1	Analisis Pengguna	41
3.8.2	Diagram <i>Use Case</i>	41
3.8.3	<i>Use Case Scenario</i>	42
3.8.4	<i>Entity Relationship Diagram</i>	49
3.8.5	<i>Swimlane Diagram</i>	51
4	PERANCANGAN	53
4.1	Perancangan Modul Pembangunan	53
4.2	Perancangan Basis Data	55
4.3	Perancangan Antarmuka	56
4.3.1	Perancangan Antarmuka Halaman	56
4.3.2	Perancangan Antarmuka Komponen	60
4.4	Perancangan <i>Endpoint</i>	68
4.5	Perancangan Halaman dan Komponen	71
4.5.1	Perancangan Halaman	71
4.5.2	Perancangan Komponen	73
5	IMPLEMENTASI DAN PENGUJIAN	83
5.1	Implementasi	83
5.1.1	Lingkungan Implementasi	83
5.1.2	Implementasi Basis Data	83
5.1.3	Implementasi Antarmuka	85
5.2	Pengujian	99
5.2.1	Pengujian Fungsional	99
5.2.2	Pengujian <i>Usability</i>	109
5.2.3	Analisis Hasil Pengujian	111
6	KESIMPULAN DAN SARAN	113
6.1	Kesimpulan	113
6.2	Saran	113

DAFTAR REFERENSI	115
A KODE PROGRAM	117
B HASIL SURVEI <i>Testing</i> PERANGKAT LUNAK SIMULASI FRS	153

DAFTAR GAMBAR

1.1	Halaman <i>Home Student Portal</i>	1
1.2	Contoh Daftar Jadwal Kuliah	2
2.1	Contoh penggunaan komponen pada <i>React</i> ¹	10
2.2	Sistem <i>Routing App Router</i>	11
2.3	Hasil konversi dari <i>file csv</i> ke format <i>array json</i>	13
2.4	Diagram Alur dari Algoritma <i>Schedule Spread</i> sederhana.	16
3.1	Diagram <i>swimlane</i> dari perangkat lunak	18
3.2	Diagram Alur Hasil Modifikasi dari Algoritma <i>Schedule Spread</i>	21
3.3	Hasil pemasukan jam mulai dan jam selesai ke <i>Array startToken dan endToken</i>	22
3.4	Hasil pemasukan jam mulai dan jam selesai ke <i>Array startToken dan endToken</i>	23
3.5	Hasil pemasukan tanggal ujian UTS ke dalam <i>Array tanggalUjian</i>	27
3.6	Hasil pemasukan tanggal ujian UTS ke dalam <i>Array tanggalUjian</i>	28
3.7	Hasil pemasukan tanggal ujian UTS ke dalam <i>Array tanggalUjian</i>	28
3.8	Contoh daftar jadwal mata kuliah dalam format <i>file excel</i>	30
3.9	Contoh keterangan warna untuk setiap program studi pada Fakultas Teknologi Informasi dan Sains	30
3.10	Tiga contoh kasus terjadinya bentrok pada jadwal kuliah.	32
3.11	Contoh isian untuk mata kuliah wajib dan hasil nilai akhirnya	32
3.12	Contoh isi formulir untuk konsultasi daftar mata kuliah yang akan diambil	33
3.13	Contoh FRS pada <i>Student Portal</i>	33
3.14	Diagram Alur Proses Pengisian FRS.	34
3.15	Contoh kode warna jadwal untuk setiap program studi pada suatu fakultas. Gambar (a) merupakan kode warna untuk semester genap 2022-2023, sementara untuk Gambar (b) merupakan kode warna untuk semester ganjil 2023-2024.	34
3.16	Contoh jadwal praktikum dari Mata Kuliah Dasar Pemrograman dan Pemrograman Berorientasi Objek yang dilaksanakan secara paralel.	35
3.17	Contoh jadwal Ujian Tengah Semester dan Ujian Akhir Semester untuk Mata Kuliah Algoritma dan Struktur Data.	35
3.18	Bentuk awal data mata kuliah yang didapatkan ber format <i>File Spreadsheet</i>	36
3.19	Hasil pembersihan data yang sudah didapat	36
3.20	<i>Template CSV</i> yang dibutuhkan untuk pemasukan data jadwal kuliah.	37
3.21	Bentuk awal data jadwal ujian yang didapatkan ber format <i>File Spreadsheet</i>	38
3.22	Hasil pembersihan data jadwal ujian yang sudah didapat	38
3.23	<i>Template CSV</i> yang dibutuhkan untuk pemasukan data jadwal ujian	39
3.24	Diagram <i>Use Case</i> dari Perangkat Lunak Simulasi FRS	41
3.25	<i>Entity Relationship Diagram</i> dari Perangkat Lunak Simulasi FRS	49
3.26	Diagram <i>swimlane</i> lengkap dari perangkat lunak	52
4.1	Pembagian Modul Pembangunan Perangkat Lunak Simulasi FRS	53
4.2	Rancangan Halaman <i>Login</i>	57
4.3	Rancangan halaman <i>home</i> untuk pengguna dengan peran admin	57

4.4	Rancangan Halaman <i>Home</i> untuk pengguna dengan peran mahasiswa	58
4.5	Rancangan Halaman Jadwal Ujian	59
4.6	Rancangan Halaman Pengguna	59
4.7	Rancangan Halaman Simulasi FRS	60
4.8	Rancangan Komponen <i>layoutAdmin</i>	60
4.9	Rancangan Komponen <i>layoutUser</i>	61
4.10	Rancangan Komponen <i>addMataKuliah</i> untuk admin	61
4.11	Rancangan Komponen <i>csvUpload</i>	62
4.12	Rancangan Komponen <i>deleteAllMataKuliah</i> untuk admin	62
4.13	Rancangan Komponen <i>updateMataKuliah</i>	62
4.14	Rancangan Komponen <i>deleteMataKuliah</i>	63
4.15	Rancangan Komponen <i>searchBar</i>	63
4.16	Rancangan Komponen <i>addJadwalKuliah</i>	63
4.17	Rancangan Komponen <i>csvUploadUjian</i>	64
4.18	Rancangan Komponen <i>deleteAllUjian</i>	64
4.19	Rancangan Komponen <i>updateJadwalUjian</i>	65
4.20	Rancangan Komponen <i>deleteJadwalUjian</i>	65
4.21	Rancangan Komponen <i>addAdmin</i>	65
4.22	Rancangan Komponen <i>deleteAdmin</i>	66
4.23	Rancangan Komponen <i>deleteMahasiswaSingle</i>	66
4.24	Rancangan Komponen <i>deleteMahasiswa</i>	66
4.25	Rancangan Komponen <i>addMataKuliah</i> untuk mahasiswa	67
4.26	Rancangan Komponen <i>cekBentrok</i>	67
4.27	Rancangan Komponen <i>deleteMataKuliah</i> untuk Mahasiswa	68
4.28	Rancangan Komponen <i>modalPrint</i>	68
5.1	Hasil Implementasi halaman <i>login</i>	85
5.2	Halaman Pilih Akun Google	85
5.3	Hasil Implementasi halaman <i>Home Admin</i>	86
5.4	Hasil Implementasi Komponen <i>addMataKuliah</i> untuk Admin	87
5.5	Hasil Implementasi Komponen <i>csvUpload</i>	87
5.6	Hasil Implementasi Komponen <i>deleteAllMataKuliah</i>	88
5.7	Hasil Implementasi Komponen <i>updateMataKuliah</i>	88
5.8	Hasil Implementasi Komponen <i>deleteMataKuliah</i>	89
5.9	Hasil Implementasi Halaman Jadwal Ujian	89
5.10	Hasil Implementasi Komponen <i>addJadwalUjian</i>	90
5.11	Hasil Implementasi Komponen <i>csvUploadUjian</i>	90
5.12	Hasil Implementasi Komponen <i>deleteAllUjian</i>	91
5.13	Hasil Implementasi Komponen <i>updateJadwalUjian</i>	91
5.14	Hasil Implementasi Komponen <i>deleteJadwalUjian</i>	92
5.15	Hasil Implementasi Halaman Pengguna	92
5.16	Hasil Implementasi Komponen <i>addAdmin</i>	93
5.17	Hasil Implementasi Komponen <i>deleteMahasiswa</i>	93
5.18	Hasil Implementasi Komponen <i>deleteAdmin</i>	94
5.19	Hasil Implementasi Komponen <i>deleteMahasiswaSingle</i>	94
5.20	Hasil Implementasi Halaman <i>Home</i> untuk Mahasiswa	95
5.21	Hasil Implementasi Halaman <i>Simulasi</i>	95
5.22	Hasil Implementasi Komponen <i>addMataKuliah</i> untuk Mahasiswa	96
5.23	Hasil Implementasi Komponen <i>deleteMataKuliah</i> untuk Mahasiswa	96
5.24	Hasil Implementasi Komponen <i>cekBentrok</i>	97
5.25	Hasil Implementasi Komponen <i>modalPrint</i>	97
5.26	Hasil Cetak PDF Simulasi FRS	98

5.27	Tampilan “ <i>pop-up</i> ” dari Komponen <i>addMataKuliah</i> yang sudah terisi data jadwal mata kuliah baru	99
5.28	Hasil pemasukan data jadwal mata kuliah baru	100
5.29	Pengunggahan <i>file CSV</i> jadwal kuliah	100
5.30	Hasil pemasukan data jadwal mata kuliah baru dengan menggunakan <i>file CSV</i>	100
5.31	Tampilan “ <i>pop-up</i> ” dari Komponen <i>addJadwalUjian</i> yang sudah terisi data jadwal ujian baru	101
5.32	Hasil pemasukan data jadwal ujian secara manual	101
5.33	Pengunggahan data jadwal ujian baru dengan menggunakan <i>file CSV</i>	102
5.34	Hasil pemasukan data jadwal ujian dengan menggunakan <i>file CSV</i>	102
5.35	Hasil penambahan data jadwal Mata Kuliah Tugas Akhir 2	103
5.36	Hasil penambahan data jadwal Mata Kuliah Desain dan Analisis Algoritma	103
5.37	Hasil keluaran algoritma untuk jadwal mata kuliah	104
5.38	Hasil penambahan Mata Kuliah Algoritma dan Pemrograman, Dasar Pemrograman dan Arsitektur dan Organisasi Komputer	104
5.39	Hasil keluaran algoritma untuk jadwal mata kuliah jika terdapat lebih dari satu bentrok	105
5.40	Hasil pemasukan data jadwal Mata Kuliah Statistika untuk Komputasi dan Analisis Big Data	106
5.41	Hasil keluaran algoritma untuk jadwal ujian	106
5.42	Hasil pemasukan data jadwal Mata Kuliah Komputasi Geometris dan Proyek Data Science 1	107
5.43	Hasil keluaran algoritma untuk jadwal ujian jika terdapat lebih dari satu bentrok jadwal ujian	107
5.44	Hasil pemasukan data jadwal beberapa mata kuliah untuk pengecekan tidak bentrok	108
5.45	Hasil keluaran algoritma untuk jadwal ujian jika tidak terdapat bentrok jadwal	108
5.46	Pemasukan beberapa mata kuliah untuk pengujian cetak PDF	108
5.47	Hasil cetak PDF	109
5.48	Persentase respon mahasiswa terhadap kemudahan navigasi antarmuka Perangkat Lunak Simulasi FRS	110
5.49	Persentase respon mahasiswa terhadap kejelasan penyediaan informasi pada Perangkat Lunak Simulasi FRS	110
5.50	Persentase respon mahasiswa terhadap responsifitas Perangkat Lunak Simulasi FRS	111
5.51	Persentase respon mahasiswa terhadap kepuasan penggunaan Perangkat Lunak Simulasi FRS	111
B.1	Hasil survei terkait kemudahan navigasi antarmuka perangkat lunak	153
B.2	Hasil survei terkait kejelasan informasi yang disajikan perangkat lunak	154
B.3	Hasil survei terkait responsifitas perangkat lunak	154
B.4	Hasil survei terkait kepuasan para mahasiswa terhadap perangkat lunak	155

DAFTAR TABEL

2.1	Peraturan Rektor UNPAR nomor III/PRT/2018-09/137 Terkait Pengambilan Jumlah SKS	6
3.1	Hasil pemasukan data jadwal Mata Kuliah “Tugas Akhir 1” ke dalam <i>array weeklySchedule</i>	22
3.2	Hasil pemasukan semua data jadwal mata kuliah	23
3.3	Hasil pemasukan data jadwal Mata Kuliah “Tugas Akhir 1” ke dalam <i>array weeklySchedule</i>	24
3.4	Hasil pemasukan semua data jadwal mata kuliah	24
3.5	Hasil pemasukan semua data jadwal mata kuliah	25
3.6	Hasil pemasukan jadwal ujian yang memiliki tanggal ujian yang sama	27
3.7	Hasil pemasukan jadwal ujian yang memiliki tanggal ujian yang sama	28
3.8	Hasil pemasukan jadwal ujian yang memiliki tanggal ujian yang sama	29
3.9	Skenario untuk autentikasi dan otorisasi mahasiswa	42
3.10	Skenario untuk memilih jadwal mata kuliah	43
3.11	Skenario untuk cek bentrok jadwal kuliah	43
3.12	Skenario untuk cek bentrok jadwal ujian	43
3.13	Skenario untuk menghapus jadwal kuliah	44
3.14	Skenario untuk cetak hasil simulasi FRS	44
3.15	Skenario untuk autentikasi dan otorisasi admin	45
3.16	Skenario untuk memasukkan jadwal kuliah	45
3.17	Skenario untuk memasukkan jadwal kuliah dengan bantuan <i>file CSV</i>	46
3.18	Skenario untuk memasukkan jadwal ujian	46
3.19	Skenario untuk memasukkan jadwal ujian dengan bantuan <i>file CSV</i>	47
3.20	Skenario untuk mengunduh <i>file template CSV</i> untuk jadwal kuliah/ujian	47
3.21	Skenario untuk Mengelola akun pengguna/admin	48
3.22	Skenario untuk memperbarui/menghapus jadwal kuliah/jadwal ujian	48
4.1	Modul Pembangunan Perangkat Lunak Simulasi FRS	54
4.2	Rancangan Tabel <i>User</i>	55
4.3	Rancangan Tabel Jadwal Mata Kuliah	55
4.4	Rancangan Tabel Jadwal Ujian	56
4.5	Rancangan Tabel Jadwal Mahasiswa	56
4.6	Rancangan Tabel Jadwal Ujian Mahasiswa	56
4.7	<i>State</i> Halaman Login	72
4.8	Fungsi pada Halaman <i>Home Admin</i>	72
4.9	Fungsi pada Halaman Simulasi	73
4.10	<i>Hooks</i> pada Komponen <i>layoutUser.js</i>	74
4.11	<i>State</i> Komponen <i>addMataKuliah.js</i>	74
4.12	Fungsi Komponen <i>csvUpload.js</i>	75
4.13	<i>State</i> Komponen <i>csvUpload.js</i>	75
4.14	<i>State</i> Komponen <i>deleteAllMataKuliah.js</i>	75
4.15	<i>State</i> Komponen <i>deleteMataKuliah.js</i>	76

4.16	State Komponen <i>updateMataKuliah.js</i>	76
4.17	State Komponen <i>addJadwalUjian.js</i>	77
4.18	Fungsi Komponen <i>csvUploadUjian.js</i>	77
4.19	State Komponen <i>csvUploadUjian.js</i>	77
4.20	State Komponen <i>deleteAllJadwalUjian.js</i>	78
4.21	State Komponen <i>deleteJadwalUjian.js</i>	78
4.22	State Komponen <i>updateJadwalUjian.js</i>	79
4.23	State Komponen <i>addAdmin.js</i>	79
4.24	State Komponen <i>deleteAdmin.js</i>	79
4.25	State Komponen <i>deleteMahasiswa.js</i>	80
4.26	State Komponen <i>deleteMahasiswaSingle.js</i>	80
4.27	State Komponen <i>addMataKuliah.js</i> untuk Mahasiswa	81
4.28	State Komponen <i>deleteMataKuliah.js</i> untuk Mahasiswa	81
4.29	State Komponen <i>modalPrint.js</i>	81

DAFTAR KODE PROGRAM

2.1	Contoh penulisan sebuah <i>script HTML</i> sederhana	7
2.2	Contoh penggunaan <i>CSS</i> pada sebuah halaman <i>HTML</i>	8
2.3	Contoh kode <i>CSS</i> yang dibuatkan pada <i>file</i> terpisah	8
2.4	Contoh pemanggilan <i>file CSS</i> pada sebuah halaman <i>HTML</i>	8
2.5	Contoh penggunaan <i>Javascript</i> pada sebuah kode <i>HTML</i>	9
2.6	Contoh penggunaan dan pengimplementasian sebuah <i>component</i>	9
2.7	Contoh penggunaan <i>state</i> dan <i>useState</i>	10
2.8	Contoh penggunaan <i>state</i> dan <i>useState</i>	10
2.9	Contoh penggunaan <i>useEffect</i>	11
2.10	Contoh penggunaan <i>Library Papaparse</i>	12
2.11	Contoh Pembuatan <i>Pool</i>	13
2.12	Contoh Penggunaan <i>Pool</i> untuk Query	13
2.13	Contoh Inisiasi <i>NextAuth.js</i>	14
5.1	Hasil Implementasi Tabel <i>User</i>	83
5.2	Hasil Implementasi Tabel Jadwal Mata Kuliah.	84
5.3	Hasil Implementasi Tabel Jadwal Mata Kuliah.	84
5.4	Hasil Implementasi Tabel Jadwal Ujian	84
5.5	Hasil Implementasi Tabel Jadwal Mahasiswa	84
5.6	Hasil Implementasi Tabel Jadwal Ujian Mahasiswa	84
A.1	ScheduleSpread.java	117
A.2	ScheduleSpreadKuliah.java	117
A.3	ScheduleSpreadUjian.java	119
A.4	/api/auth/[...nextauth]/route.js	119
A.5	/api/homeAdmin/search/[keyword]/route.js	120
A.6	/api/homeAdmin/ujian/route.js	121
A.7	/api/homeAdmin/ujianAdd/route.js	122
A.8	/api/jadwalKuliah/route.js	122
A.9	/api/simulasi/[...user]/route.js	122
A.10	/api/ujian/[...user]/route.js	123
A.11	/api/usersAdmin/route.js	123
A.12	/home/page.js	124
A.13	/homeAdmin/page.js	124
A.14	/homeAdmin/search/[keyword]/page.js	125
A.15	/homeAdmin/components/addMataKuliah.js	126
A.16	/homeAdmin/components/csvUpload.js	127
A.17	/homeAdmin/components/deleteAllMataKuliah.js	129
A.18	/homeAdmin/components/deleteMataKuliah.js	129
A.19	/homeAdmin/components/searchBar.js	130
A.20	/homeAdmin/components/updateMataKuliah.js	130
A.21	/jadwalUjian/page.js	131

A.22	/jadwalUjian/components/addJadwalUjian.js	132
A.23	/jadwalUjian/components/csvUploadUjian.js	133
A.24	/jadwalUjian/components/deleteAllUjian.js	135
A.25	/jadwalUjian/components/deleteJadwalUjian.js	135
A.26	/jadwalUjian/components/updateJadwalUjian.js	136
A.27	/simulasi/[...user]/page.js	137
A.28	/simulasi/addMataKuliah.js	140
A.29	/simulasi/cekBentrok.js	141
A.30	/simulasi/deleteMataKuliah.js	142
A.31	/simulasi/modalPrint.js	142
A.32	/usersAdmin/page.js	144
A.33	/usersAdmin/components/addAdmin.js	145
A.34	/usersAdmin/components/deleteAdmin.js	146
A.35	/usersAdmin/components/deleteMahasiswa.js	147
A.36	/usersAdmin/components/deleteMahasiswaSingle.js	147
A.37	layout.js	148
A.38	layoutAdmin.js	148
A.39	layoutUser.js	149
A.40	page.js	150
A.41	db.js	150

menjadi tidak sesuai dengan aturan yang berlaku. Lalu, mahasiswa juga perlu memastikan bahwa setiap jadwal baik kuliah, praktikum, responsi dan ujiannya tidak terdapat bentrok satu dengan yang lainnya. Aturan ini diberlakukan agar seorang mahasiswa dapat mengikuti kegiatan perkuliahan dengan baik dan tidak terjadi bentrok jadwal diantara setiap mata kuliah yang diambil, sehingga dapat menghasilkan nilai yang memuaskan. Jika seorang mahasiswa melanggar aturan ini, maka mahasiswa tersebut akan dikenakan sanksi seperti jika terdapat bentrok, seorang mahasiswa akan diminta untuk membatalkan pengambilan salah satu mata kuliah tersebut, atau akan dibatalkan secara acak oleh kaprodi. Namun, apabila mata kuliah yang bentrok baru ditemui di pertengahan atau pun akhir semester, maka nilai mata kuliah tersebut akan dibuat E—tidak lulus oleh kaprodi. Dengan demikian seorang mahasiswa memilih jadwal yang tepat merupakan hal penting dan wajib dilakukan.

Namun, terdapat sebuah kesulitan ketika seorang mahasiswa memastikan bahwa setiap jadwal merupakan jadwal yang benar dan tidak ada yang bentrok. Hal ini dikarenakan daftar jadwal mata kuliah yang dibagikan dengan tempat mahasiswa melakukan FRS tidak saling terintegrasi, sehingga mengakibatkan sulitnya memilah jadwal. Oleh karena itu perlu dibangun sebuah perangkat lunak di mana mahasiswa bisa melakukan simulasi pengisian FRS, dan ketika mahasiswa memilih jadwal yang bentrok, maka perangkat lunak akan mengeluarkan peringatan bahwa terdapat bentrok jadwal pada mata kuliah tersebut. Hal ini akan sangat membantu mahasiswa dan mempersingkat waktu dalam melakukan FRS.

Untuk menangani terjadinya bentrok pada jadwal yang dipilih, akan digunakan sebuah algoritma sederhana yang bernama *Schedule Spread*. Algoritma ini merupakan pendekatan sederhana dengan memanfaatkan *array* dua dimensi. *Array* tersebut akan digunakan untuk penyimpanan data jadwal yang akan diperiksa apakah terjadi bentrok atau tidak.

Perangkat lunak simulasi FRS, dapat dibangun dengan menggunakan *React Framework*. Dikutip dari website *React.dev*[3] pengertian *React* merupakan sebuah *library javascript* yang memiliki fungsi untuk menampilkan *User Interface*. *User Interface* sendiri terdiri dari komponen-komponen kecil seperti tombol, text, ataupun gambar. Dengan menggunakan *React*, seorang pengembang dapat dipermudah dalam penggunaan komponen-komponen tersebut secara *reusable*.

Untuk menggunakan *React* dalam pengembangan sebuah perangkat lunak berbasis web, dibutuhkan juga *framework* yang dapat mengatasi bagian *back-end* dari aplikasi tersebut. Salah satu *React framework* yang banyak digunakan untuk menangani hal tersebut adalah *next.js*. *Next.js* sendiri merupakan sebuah *React framework* yang diperuntukkan untuk membangun sebuah *full-stack web application* [5].

Pada Tugas Akhir ini akan dibangun sebuah perangkat lunak simulasi FRS berbasis web dengan menggunakan *React Framework* yaitu *Next.js*, yang nantinya dapat melakukan autentikasi pengguna menggunakan *Library NextAuth.js*. Perangkat lunak juga dapat menerima *input* berupa jadwal mata kuliah secara manual maupun dengan bantuan *file csv*, dan jadwal ujian secara manual yang tersedia pada semester terkait. Lalu mahasiswa dapat melakukan simulasi FRS dengan tujuan untuk memeriksa apakah jadwal dari mata kuliah yang diambil terdapat bentrok atau tidak dengan bantuan algoritma *Schedule Spread*. Hasil dari simulasi tersebut juga dapat disimpan dalam bentuk PDF untuk nantinya digunakan sebagai bahan konsultasi dengan dosen wali masing-masing mahasiswa.

1.2 Rumusan Masalah

Rumusan masalah dalam pengerjaan skripsi ini dipaparkan sebagai berikut:

1. Bagaimana langkah-langkah untuk menentukan apakah terjadi bentrok jadwal antara dua mata kuliah?
2. Bagaimana cara kerja *React Framework* yaitu *Next.js* dalam membangun perangkat lunak basis web?

3. Bagaimana cara mengimplementasikan perangkat lunak simulasi FRS dengan menggunakan *React Framework Next.js*

1.3 Tujuan

Tujuan yang ingin dicapai pada pengerjaan skripsi ini adalah:

1. Mempelajari langkah-langkah untuk menentukan apakah kedua mata kuliah terjadi bentrok jadwal atau tidak.
2. Mempelajari cara kerja *React Framework* yaitu *Next.js* dalam membuat perangkat lunak basis web.
3. Mengimplementasikan perangkat lunak simulasi FRS dengan menggunakan *React Framework Next.js*.

1.4 Batasan Masalah

Terdapat beberapa batasan yang dibuat untuk pengerjaan skripsi ini adalah:

1. Perangkat lunak yang akan dibuat hanya menangani jadwal yang terdapat pada Program Studi Informatika.
2. Hasil dari pengecekan bentrok antara jadwal hanya memberikan notifikasi/peringatan terkait bentrok pada jadwal apa, tidak memberikan saran untuk mengambil jadwal yang tidak bentrok.

1.5 Metodologi

Langkah-langkah yang dilakukan untuk membangun perangkat lunak simulasi FRS ini adalah :

1. Mempelajari tentang apa itu Formulir Rencana Studi (FRS) secara umum, dan bagaimana mekanismenya untuk mengisi FRS di UNPAR.
2. Melakukan studi literatur terkait jadwal yang diterima oleh setiap mahasiswa Informatika UNPAR dan mempelajari polanya.
3. Melakukan studi literatur terkait *React*, *Next.js*, *NextAuth.js*, *PostgreSQL* dan *Papaparse*.
4. Melakukan studi literatur dan uji coba terkait algoritma yang akan digunakan untuk memecahkan masalah bentrok jadwal yaitu Algoritma *Schedule Spread*.
5. Melakukan studi literatur terkait pembangunan perangkat lunak basis web yang *user friendly*.
6. Melakukan penyiapan data jadwal agar dapat digunakan untuk perangkat lunak yang akan dibuat.
7. Menganalisis perangkat lunak yang akan dibuat dari hasil studi literatur jadwal, studi literatur algoritma, dan penyiapan data.
8. Melakukan perancangan perangkat lunak dengan membuat rancangan tampilan antar muka, menentukan setiap komponen dari perangkat lunak seperti mahasiswa, *admin*, jadwal kuliah, jadwal ujian dan lain sebagainya, serta menentukan hubungan di antara komponen-komponen tersebut.
9. Membangun perangkat lunak simulasi pengisian Formulir Rencana Studi basis web, dengan menggunakan *React Framework* dengan memperhitungkan *user experiencenya*.
10. Mengimplementasikan algoritma yang sudah dibangun dalam perangkat lunak simulasi pengisian Formulir Rencana Studi untuk memecahkan masalah bentrok jadwal.
11. Melakukan pengujian fungsional dan eksperimental terhadap perangkat lunak simulasi FRS yang sudah dibuat. Pengujian eksperimental dilakukan dengan meminta beberapa mahasiswa untuk menggunakan perangkat lunak simulasi FRS secara langsung.
12. Menyusun dokumen tugas akhir.