

SKRIPSI

PENGEMBANGAN LANJUT SISTEM INFORMASI
PENJADWALAN KULIAH DENGAN MENGGUNAKAN
FRAMEWORK VUE.JS



Deddy Chandra

NPM: 6181801007

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2022

UNDERGRADUATE THESIS

**ADVANCED DEVELOPMENT OF LECTURE SCHEDULING
INFORMATION SYSTEM USING VUE.JS FRAMEWORK**



Deddy Chandra

NPM: 6181801007

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2022**

LEMBAR PENGESAHAN

PENGEMBANGAN LANJUT SISTEM INFORMASI PENJADWALAN KULIAH DENGAN MENGGUNAKAN *FRAMEWORK VUE.JS*

Deddy Chandra

NPM: 6181801007

Bandung, 1 Juli 2022

Menyetujui,

Pembimbing

Digitally signed
by Raymond
Chandra Putra

Raymond Chandra Putra, M.T.

Ketua Tim Penguji
Digitally signed
by Maria V.
Claudia M.

Maria Veronica, M.T.

Anggota Tim Penguji
Digitally signed
by Husnul
Hakim

Husnul Hakim, M.T.

Mengetahui,

Ketua Program Studi
Digitally signed
by Mariskha Tri
Adithia

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

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

PENGEMBANGAN LANJUT SISTEM INFORMASI PENJADWALAN KULIAH DENGAN MENGGUNAKAN *FRAMEWORK VUE.JS*

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 1 Juli 2022



Deddy Chandra
NPM: 6181801007

ABSTRAK

Sistem informasi penjadwalan kuliah adalah sebuah aplikasi berbasis web yang dibuat untuk melakukan penjadwalan kuliah pada sebuah semester secara otomatis dengan algoritma genetik maupun dengan cara memasukan aktivitas secara manual. Sistem informasi penjadwalan kuliah ini telah selesai dibuat dengan menggunakan *framework* Laravel, *Springboot*, Bootstrap, dan *library* jQuery oleh pengembang sebelumnya. Sistem informasi penjadwalan ini telah selesai dibuat, tetapi terdapat beberapa masalah pada sistem ini di antaranya adalah kode program yang tidak terstruktur dengan baik atau seluruh kode program digabung pada sebuah *file* sehingga kode menjadi rumit (dalam sebuah *file* terdapat HTML, PHP, CSS, JavaScript) dan terdapat beberapa bagian kode yang sama tetapi disalin ulang pada beberapa *file* yang berbeda.

Sistem informasi penjadwalan kuliah tersebut dibangun ulang dengan tujuan agar kode pada sistem tersebut menjadi lebih rapi dan terstruktur dengan baik. Bagian *frontend* pada sistem informasi ini dibangun ulang dengan menggunakan paradigma *Component-Based Software Engineering* (CBSE) dengan bantuan *framework* *Vue.js*. *Framework* *Vue.js* ini dapat digunakan untuk membangun ulang bagian *frontend* dengan menggunakan komponen dan beberapa fungsi yang telah disediakan. Fitur pada sistem yang dibangun ulang akan sama dengan sistem sebelumnya. Beberapa fitur yang sama pada halaman berbeda dibuat menjadi sebuah komponen, sehingga komponen tersebut dapat digunakan kembali pada semua halaman yang memiliki fitur tersebut. Penyebab terjadinya kode program yang menjadi sangat rumit ini dikarenakan terdapat banyak fitur pada sebuah halaman dan seluruh fitur tersebut digabung pada sebuah *file*, terdapat juga beberapa fitur yang sama tetapi kode program pada fitur tersebut ditulis berulang kali pada *file* yang berbeda. Dengan adanya komponen, maka fitur-fitur tersebut dapat dibuat menjadi masing-masing komponen dan komponen tersebut dapat digunakan kembali. Agar kode program menjadi lebih terstruktur dan rapi, penempatan kode dibagi menjadi tiga bagian yaitu bagian *style*, bagian tampilan, dan bagian logika.

Hasil dari pengujian yang dilakukan menyimpulkan bahwa sistem informasi penjadwalan kuliah yang dikembangkan ulang dengan menggunakan *framework* *Vue.js* menjadi *clean code*, mudah dilakukan pemeliharaan dan terstruktur dengan baik. Pengujian fungsionalitas juga menunjukkan bahwa fitur pada sistem informasi yang telah dikembangkan masih memiliki fungsi yang sama dengan sistem informasi sebelumnya.

Kata-kata kunci: Penjadwalan, *Frontend*, *Framework* *Vue.js*, Komponen, *Component-Based Software Engineering*.

ABSTRACT

The lecture scheduling information system is a web-based application created for schedule lectures in a semester automatically with a genetic algorithm or by entering the activity manually. This lecture scheduling information system is built using the Laravel, Springboot, Bootstrap, and jQuery libraries by previous developers. This scheduling information system has been completed created, but there are several problems with this system including the program code that is not structured properly or all the program code is combined in a file so that code becomes complicated (in a file there is HTML, PHP, CSS, JavaScript) and there are some parts of the same code but copied in several different files.

The lecture scheduling information system is rebuilt with the aim of making the code in the system more neat and well structured. The frontend of this information system is rebuilt using the CBSE (Component-Based Software Engineering) paradigm with the help of the Vue.js framework. This Vue.js framework can be used to rebuild the frontend using components and some functions which has been provided. The features on the rebuilt system will be the same as the previous system. Some of the same features on different pages are made into a component, so that the component can be reused on all pages that have this feature. The cause of the program being very complicated is because there are many features on a page and all of these features are combined in a file, there are also some features that are the same but the program code for these features is written repeatedly in different files. With the components, these features can be made into individual components and these components can be reused. In order for the program code to be more structured and neat, the placement of the code is divided into three sections that is style section, display or view section, and the logic section.

The results of the tests carried out concluded that the lecture scheduling information system that redeveloped using the Vue.js framework became clean code, easy to maintain and well structured. Functionality testing also shows that the features in the information system that have been developed still have the same function as the previous information system.

Keywords: Timetable, Frontend, Vue.js framework, Component, Component-Based Software Engineering.

*Dipersembahkan kepada papa, mama, cece dan happy family yang
selalu mendukung penulis dalam pembuatan skripsi*

KATA PENGANTAR

Puji syukur penulis mengucapkan kepada Tuhan Yang Maha Esa atas segala karunia dan kasihNya sehingga skripsi yang berjudul “Pengembangan Lanjut Sistem Informasi Penjadwalan Kuliah Dengan Menggunakan *Framework Vue.js*” ini dapat diselesaikan dengan baik dan tepat waktu. Skripsi ini telah melalui pengujian dilakukan pada hari Jumat tanggal 1 Juli 2022. Skripsi ini dibuat dan diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar Sarjana pada Program studi Informatika, Universitas Katolik Parahyangan dan memperbaiki sistem informasi penjadwalan kuliah dengan menggunakan *framework Vue.js*. Selama penulisan skripsi ini, penulis menyadari bahwa skripsi ini tidak dapat diselesaikan tanpa bantuan dan dukungan berbagai pihak. Sehingga penulis mengucapkan rasa terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah mendengar doa-doa penulis dan selalu memberkati penulis dengan penuh kekuatan dan kemampuan.
2. Keluarga (Papa, Mama, Cece), terima kasih selalu memberikan dukungan baik dukungan mental maupun finansial, perhatian dan nasehat, serta doa-doa agar skripsi ini diselesaikan dengan baik.
3. Semua anggota Happy Family, terima kasih sudah selalu mendukung penulis dan selalu mendoakan penulis sehingga skripsi ini berjalan dengan lancar dan baik.
4. Bapak Raymond Chandra Putra, M.T. selaku pembimbing penulis dari awal hingga skripsi ini berjalan dengan baik, terima kasih buat perhatian, nasehat, dan masukan serta dukungan yang telah Bapak berikan kepada penulis.
5. Ibu Maria Veronica Claudia, M.T. dan Bapak Husnul Hakim, M.T selaku dosen penguji yang telah meyidang, memberikan kritik dan saran yang membangun untuk skripsi ini.
6. Seluruh Bapak/Ibu dosen Informatika yang telah memberikan pengetahuan dan pelajaran selama penulis berkuliah di UNPAR, terutama Pak Chandra Wijaya yang memberikan kesempatan dalam pembuatan proyek SI dan Pak Pascal Alfadian Nugroho yang memberi kesempatan untuk *intern* di DNArtworks.
7. Tim FS Traveloka bagian Engineer yang mendukung dalam pengisian kuisisioner skripsi ini.
8. Tim DNArtworks dan teman-teman dari Tim Engineer yang selalu mendukung progress pengerjaan skripsi ini.
9. Teman-teman yang mendukung dalam pengerjaan skripsi ini (MF, WorstGeneration, Quack-Quack, J&L, dan lain-lain yang tidak dapat disebutkan satu persatu).
10. Teman-teman seperjuangan selama perkuliahan: Bryan Heryanto, Jiang Han, Daniel Alexander Kefas, Chris Ardiansyah, Gerald Akira Surya, Florentia Kezia Kurniawan, Obed Kristiaji, Warren Mazmur, Ivan Limosi, Clement Rivandi, Bernadus Lukas, Julyus Andreas Binoto dan lain-lain.

Bandung, Juli 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR KODE PROGRAM	xxiv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Metodologi	4
1.6 Sistematika Pembahasan	4
2 LANDASAN TEORI	5
2.1 Penjadwalan	5
2.2 <i>Clean Code</i>	5
2.2.1 Nama yang bermakna	6
2.2.2 Fungsi	6
2.3 Rekayasa Perangkat Lunak	6
2.3.1 Arsitektur <i>Model View Controller</i>	7
2.3.2 <i>Component-Based Software Engineering</i> (CBSE)	7
2.3.3 <i>Object-Oriented Design</i> SOLID	8
2.4 <i>Library</i> dan <i>Framework</i>	8
2.4.1 <i>Vue.js</i>	9
2.4.2 <i>Laravel</i>	10
2.4.3 Spring boot	10
2.4.4 jQuery	11
2.5 Algoritma Genetik	11
3 ANALISIS	13
3.1 Analisis Sistem Informasi Penjadwalan Kuliah	13
3.1.1 Analisis Sistem Kini	13
3.1.2 Arsitektur Sistem	19
3.1.3 Struktur <i>Folder</i>	19
3.2 Analisis Masalah	21
3.3 Analisis Pemilihan <i>Framework Vue.js</i>	24
3.4 Analisis Perancangan Perangkat Lunak	24
3.4.1 Analisis Perancangan Komponen	24

3.4.2	Analisis Pemisahan dan Penempatan Kode Program	27
3.5	Analisis Dampak	28
3.5.1	jQuery	28
3.5.2	Table Controller	29
3.5.3	Kode program dan <i>file</i>	30
4	PERANCANGAN	33
4.1	Perancangan Komponen	33
4.2	Perancangan <i>Services</i>	43
4.3	Perancangan Model pada Komponen	44
4.4	Perancangan Perubahan Halaman	44
5	IMPLEMENTASI DAN PENGUJIAN	47
5.1	Lingkungan Perangkat Lunak	47
5.2	Implementasi	47
5.2.1	Implementasi Pendaftaran Komponen	47
5.2.2	Implementasi Pada Layout Bagian Menu	47
5.2.3	SubNavController	48
5.2.4	Implementasi <i>Shared_components</i>	49
5.2.5	Implementasi Pada Modul Penjadwalan	50
5.2.6	Implementasi Pada Halaman CRUD	54
5.3	Pengujian	58
5.3.1	Pengujian Fungsional	58
5.3.2	Pengujian Experimental	68
5.3.3	Analisis Hasil Pengujian Fungsional Dan Experimental Pada Sistem Kini Dan Sistem Usulan	69
6	KESIMPULAN DAN SARAN	71
6.1	Kesimpulan	71
6.2	Saran	71
	DAFTAR REFERENSI	73
	A KODE PROGRAM	75

DAFTAR GAMBAR

1.1	Contoh penjadwalan kuliah fakultas FTIS di UNPAR	1
3.1	Halaman input aktivitas	14
3.2	Tangkapan layar pada halaman penjadwalan	15
3.3	Tangkapan layar pada saat melakukan penjadwalan secara manual	15
3.4	Halaman hasil penjadwalan	16
3.5	Contoh <i>file</i> hasil export	16
3.6	Contoh <i>file</i> hasil unduh printable	17
3.7	Contoh halaman lihat hasil penjadwalan	17
3.8	Ilustrasi arsitektur pada sistem kini	19
3.9	Tangkapan layar pada <i>folder view</i>	20
3.10	Tangkapan layar pada <i>folder controller</i>	20
3.11	Tangkapan layar pada <i>folder models</i>	21
3.12	Potongan sebagian kode program pada halaman <i>index</i> penjadwalan	22
3.13	Potongan sebagian kode program pada halaman <i>show</i> penjadwalan	22
3.14	Contoh halaman buat aktivitas	25
3.15	Contoh halaman buat pengaturan penjadwalan	25
3.16	Contoh <i>header</i> tabel pada modul input aktivitas	26
3.17	Contoh <i>header</i> tabel pada modul mata kuliah	26
3.18	Contoh tabel pada penjadwalan	26
4.1	Contoh komponen AllNavController	34
4.2	Contoh komponen NavController	34
4.3	Contoh komponen SubNavController	35
4.4	Contoh komponen BreadCrumbComponent	35
4.5	Contoh komponen LoaderComponent	36
4.6	Contoh komponen TableComponent	36
4.7	Contoh komponen DownloadComponent	37
4.8	Contoh komponen UploadComponent	37
4.9	Contoh komponen ValidateComponent	37
4.10	Contoh komponen ActivitiesFieldsComponent	38
4.11	Contoh komponen ActivitiesShowFieldsComponent	38
4.12	Contoh komponen MultipleTimeSlotComponent	39
4.13	Contoh komponen CustomCheckComponent	40
4.14	Contoh komponen MultipleInputComponent	40
4.15	Contoh komponen CourseHoverComponent	41
4.16	Contoh komponen DaysComponent	41
4.17	Contoh komponen EditTableComponent	41
4.18	Contoh komponen EditTableHoverComponent	42
4.19	Contoh komponen ForceInsertModalComponent	42
4.20	Contoh komponen TimeTableComponent	43
4.21	Contoh komponen TimeTablingComponent	43

5.1 Hasil kuisisioner dengan pertanyaan berbentuk skala	69
-------------------------------------------------------------------	----

DAFTAR TABEL

3.1	Daftar fitur pada setiap modul	23
4.1	Perancangan komponen	33
4.2	Komponen pada masing-masing halaman	45
5.1	Hasil pengujian pada modul Input Aktivitas	59
5.2	Hasil pengujian pada modul pengaturan penjadwalan	60
5.3	Hasil pengujian pada modul penjadwalan mode utama	60
5.4	Hasil pengujian pada modul penjadwalan mode ubah tabel	61
5.5	Hasil pengujian pada modul hasil penjadwalan	62
5.6	Hasil pengujian pada modul pengajar	63
5.7	Hasil pengujian pada modul mata kuliah	64
5.8	Hasil pengujian pada modul ruangan	65
5.9	Hasil pengujian pada modul semester	65
5.10	Hasil pengujian pada modul program studi	66
5.11	Hasil pengujian pada modul tipe ruangan	67
5.12	Hasil pengujian pada modul tipe aktivitas	67
5.13	Hasil pengujian pada modul pengguna	68

DAFTAR KODE PROGRAM

2.1	Contoh komponen sederhana <i>Vue</i>	9
2.2	Contoh kode pada <i>root Vue</i>	9
2.3	Contoh penggunaan komponen	9
3.1	Contoh kode program yang menggunakan <i>jQuery</i>	28
3.2	Contoh kode program yang menggunakan <i>Vue.js</i>	28
3.3	Contoh salah satu <i>function</i> di Table Controller pada sistem kini	29
3.4	Contoh salah satu <i>function</i> di Table Controller pada sistem usulan	30
3.5	Contoh kode program <i>breadcrumb</i> pengguna pada sistem kini	30
3.6	Contoh kode program <i>breadcrumb</i> lecturer pada sistem kini	31
3.7	Contoh kode program <i>breadcrumb</i> pada sistem usulan	31
3.8	Contoh fungsi kode program yang tidak pernah digunakan	31
A.1	Kode halaman utama pada penjadwalan	75
A.2	Pendaftaran komponen	88
A.3	Kode menu yang sudah menggunakan komponen <i>vue.js</i>	90
A.4	Model <i>NavComponent</i>	91
A.5	kode komponen <i>AllNavComponent</i>	91
A.6	kode komponen <i>NavComponent</i>	92
A.7	kode komponen <i>SubNavComponent</i>	92
A.8	Kode komponen <i>Breadcrumb</i>	93
A.9	Kode komponen <i>Loader</i>	93
A.10	Kode komponen <i>Upload</i>	94
A.11	Kode model <i>Modal</i>	94
A.12	Kode <i>UploadServices</i>	95
A.13	Kode komponen <i>Download</i>	95
A.14	Kode komponen <i>Validate</i>	95
A.15	Kode <i>ValidateServices</i>	96
A.16	Kode komponen <i>Tabel</i>	96
A.17	Model <i>TableComponent</i>	97
A.18	Kode <i>SemesterServices</i>	98
A.19	Halaman utama penjadwalan pada sistem usulan	98
A.20	Kode <i>TimeTableServices</i>	98
A.21	kode komponen <i>TimeTablingComponent</i>	106
A.22	kode komponen <i>TimeTableComponent</i>	108
A.23	kode komponen <i>CourseHoverComponent</i>	111
A.24	kode komponen <i>DaysComponent</i>	112
A.25	kode komponen <i>EditTableComponent</i>	112
A.26	kode komponen <i>EditTableHoverComponent</i>	114
A.27	kode komponen <i>ForceInsertModalComponent</i>	115
A.28	Halaman utama input aktivitas pada sistem usulan	115
A.29	Halaman buat input aktivitas pada sistem usulan	116

A.30 Halaman ubah input aktivitas pada sistem usulan	117
A.31 kode komponen ActivitiesFieldsComponent	117
A.32 Kode ActivitiesServices	119
A.33 Halaman lihat input aktivitas pada sistem usulan	119
A.34 kode komponen ActivitiesShowFieldsComponent	120
A.35 Halaman utama pengaturan penjadwalan pada sistem usulan	121
A.36 Halaman buat pengaturan penjadwalan pada sistem usulan	121
A.37 Halaman ubah pengaturan penjadwalan pada sistem usulan	122
A.38 kode komponen SettingsFieldsComponent	122
A.39 kode komponen MultipleInputComponent	124
A.40 kode komponen CustomCheckComponent	124
A.41 Halaman hasil penjadwalan pada sistem usulan	125
A.42 Halaman lihat detail hasil sebuah penjadwalan pada sistem usulan	125
A.43 Halaman utama pengajar pada sistem usulan	125
A.44 Halaman buat pengajar pada sistem baru	126
A.45 Halaman ubah pengajar pada sistem baru	126
A.46 kode komponen LecturersFieldsComponent	127
A.47 kode komponen MultipleTimeSlotComponent	128
A.48 Halaman show pengajar pada sistem baru	129
A.49 kode komponen LecturersShowFieldsComponent	129
A.50 Halaman utama matakuliah pada sistem usulan	131
A.51 Halaman buat matakuliah pada sistem usulan	131
A.52 Halaman ubah matakuliah pada sistem usulan	132
A.53 kode komponen CoursesFieldsComponent	132
A.54 Kode CourseServices	133
A.55 Halaman lihat matakuliah pada sistem usulan	133
A.56 kode komponen CoursesShowFieldsComponent	134
A.57 Halaman utama ruangan pada sistem baru	134
A.58 Halaman buat ruangan pada sistem usulan	135
A.59 Halaman ubah ruangan pada sistem usulan	136
A.60 kode komponen RoomsFieldsComponent	136
A.61 Halaman lihat ruangan pada sistem baru	138
A.62 kode komponen RoomsShowFieldsComponent	138
A.63 Halaman utama program studi pada sistem baru	140
A.64 Halaman buat program studi pada sistem usulan	140
A.65 Halaman ubah program studi pada sistem usulan	141
A.66 kode komponen JurusansFieldsComponent	141
A.67 kode komponen ConsentrationComponent	142
A.68 Halaman utama tipe ruangan pada sistem usulan	143
A.69 Halaman buat tipe ruangan pada sistem baru	143
A.70 Halaman ubah tipe ruangan pada sistem baru	143
A.71 kode komponen RoomTypesFieldsComponent	144
A.72 Halaman utama tipe aktivitas pada sistem baru	144
A.73 Halaman buat tipe aktivitas pada sistem baru	145
A.74 Halaman ubah tipe aktivitas pada sistem baru	145
A.75 kode komponen ActivityTypesFieldsComponent	146
A.76 Halaman utama Pengguna pada sistem usulan	146
A.77 Halaman buat pengguna pada sistem usulan	146
A.78 Halaman edit pengguna pada sistem usulan	147
A.79 kode komponen UserFieldsComponent	147

BAB 1

PENDAHULUAN

Pada bab ini, dibahas mengenai latar belakang penelitian, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan yang dilakukan pada penelitian ini.

1.1 Latar Belakang

Universitas Katolik Parahyangan (UNPAR) merupakan salah satu universitas ternama yang terletak di Bandung. UNPAR memiliki 7 fakultas dengan 17 program studi jenjang sarjana dan diploma. Salah satu fakultas di UNPAR adalah Fakultas Teknologi Informasi dan Sains (FTIS) yang memiliki program studi Teknik Informatika, Matematika dan Fisika. Salah satu kegiatan inti adalah proses belajar mengajar. Agar proses belajar mengajar berjalan dengan baik maka dibutuhkan sebuah penjadwalan. Pada umumnya penjadwalan kuliah dibuat dan dikelompokkan berdasarkan fakultas. Gambar 1.1 merupakan contoh hasil penjadwalan matakuliah pada fakultas FTIS. Penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan dalam jadwal¹.

JADWAL KULIAH SEMESTER GANJIL 2021/2022										
SENIN										
Kuang/kap R.Kuliah	07-08	08 - 09	09 - 10	10 - 11	11 - 12	12 - 13	13 - 14	14 - 15	15 - 16	16 - 17
9120/60 R.Kuliah	Teori Suku Bunga (A) - Maria Anastasia		Kalkulus Vektor (A) - Erwinna		Kalkulus Vektor (B) - Erwinna		Model - Survival (B) - Farah K.	Pemecahan Masalah Matematika (B) - Jonathan Hoseana		
9121/60 R.Kuliah	Teori Suku Bunga (C) - Robyn Irawan		Kalkulus Vektor (C) - Daniel Salim		Kalkulus Vektor (D) - Daniel Salim					
9122/60 R.Kuliah			Statistika Elementer (C) - Rizky Reza		Seminar - Livia Owen			Matematika Keuangan- Erwinna Chendra		
10316/60 R.Kuliah			Fungsi Kompleks (A) - Wono		Fungsi Kompleks (B) - Wono		Persamaan Diferensial Parsial (A) - Andreas Parama			
10317/60 R.Kuliah	Pemodelan Matematika (A) - Jonathan Hoseana		Pemrograman Berorientasi Objek (A) - Pascal		Matematika Diskret (A) - Mariskha			Keamanan Informasi - Mariskha / Pascal		
10323/60 R.Kuliah			Matematika Dasar (B) - Rosa		Matematika Diskret (B) - Natalia			Teori Peluang (B) - Ferry J.P.		
Lab.Komp. (SB-9013)	Dasar-dasar Pemrograman - Husnul		Matematika Dasar (A) - Luciana					Pencarian dan Temu Kembali Informasi - Kristopher		
SB-9014/30 R. Kuliah		Pengantar Sistem Cerdas (A) - Lionov			Teori Peluang (A) - Ferry J.P.			Pengantar Sistem Cerdas (B) - Lionov		
Lab.Komp. (SB-9015)	Basis Data & Pemrograman SQL untuk Big Data - Maria Veronica				Struktur Diskret (A) - Heni		Algoritma dan Struktur Data (B) - Vania			
Lab.Komp. (SB-9016)		Sertifikasi DevOps - Chandra			Struktur Diskret (B) - Luciana		Algoritma dan Struktur Data (A) - Husnul			
Lab.Komp. (SB-9017)			Skripsi 1 - Mariskha		Pemrograman Pada Perangkat Bergerak (A) - Raymond					
Lab.Komp. (SB-9018)				Skripsi 2 - Mariskha	Pemrograman Pada Perangkat Bergerak (B) - Keenan					
10307/10 R.Kuliah	Fisika Dasar (A) - Sylvia		Mekanika 2 - Paulus		Elektrodinamika - Paulus		Fisika Matematika 1 - Responsi - Elvina Octora		Elektrodinamika (Responsi) - Michael Hidayat	
10308/10 Lab.Fisika					Matematika Asuransi (A) - Felivia			Teori Relativitas - Haryanto		
10314 Lab.Fisika	Proyek Bersama - Janto				Pemrograman Komputer - Janto					

Gambar 1.1: Contoh penjadwalan kuliah fakultas FTIS di UNPAR

Penjadwalan perkuliahan termasuk dalam golongan jenis *timetabling*, yaitu suatu daftar waktu mengenai kapan suatu kegiatan dilakukan, terutama terkait waktunya. Permasalahan *timetabling*

¹<https://kbbi.kemdikbud.go.id/entri/penjadwalan> (diakses 20 Juli 2021)

dapat digolongkan sebagai *NP-Hard Problem* (*Nondeterministic Polynomial Time*). *NP-Hard Problem* adalah permasalahan optimasi yang membutuhkan exponential time untuk diselesaikan secara optimal (Talbi, 2009) Apabila seluruh kombinasi alternatif solusi dari *NP-Hard Problem* dicobakan untuk diuji, maka waktu yang dibutuhkan untuk mencari solusi yang *feasible* dari permasalahan meningkat secara eksponensial [1].

Perkuliahan pada FTIS terdapat beberapa aktivitas pada sebuah matakuliah contohnya kuliah, praktikum, dan responsi. Aktivitas pada matakuliah tersebut dibuat menjadi sebuah penjadwalan kuliah. Terdapat juga beberapa aturan yang tidak boleh dilanggar ketika melakukan penjadwalan perkuliahan sehingga untuk melakukan penjadwalan kuliah menjadi semakin susah atau sulit. Berikut adalah beberapa contoh aturan dalam melakukan penjadwalan aktivitas kuliah:

1. Seorang dosen tidak boleh mengajar secara paralel pada jam yang sama.
2. Sebuah ruangan atau kelas tidak boleh digunakan secara bersama (terdapat dua atau lebih proses belajar dan mengajar pada matakuliah yang berbeda di ruangan yang sama).
3. Jika sebuah matakuliah terdiri lebih dari satu kelas, maka kelas itu dapat berjalan secara paralel dengan pengajar yang berbeda.
4. Kapasitas mahasiswa pada sebuah matakuliah harus lebih kecil atau sama dengan kapasitas sebuah ruangan.
5. Matakuliah yang terdiri lebih dari satu pertemuan atau terdapat lebih dari satu aktivitas (kuliah, responsi, praktikum, dan sebagainya) dalam satu minggu, tidak boleh dijadwalkan pada hari yang sama.

Sistem informasi ini sebenarnya telah selesai dibuat dengan menggunakan *framework Laravel*, *Spring framework*, *Bootstrap* dan *jQuery*. Tetapi kode program pada sistem informasi ini terlalu rumit (jumlah baris pada kode yang sangat panjang, yang sebenarnya kode tersebut dapat dipisahkan), kode program tidak terstruktur dengan baik, dan beberapa kode program yang sama ditulis berulang pada *file* yang berbeda. Hal ini membuat pengembang menjadi sangat sulit untuk melakukan pemeliharaan pada sistem informasi penjadwalan kuliah ini. Maka dari itu terdapat usulan untuk melakukan pengembangan pada sistem informasi ini, yaitu menggunakan *framework Vue.js*. Pada sistem informasi ini terdapat beberapa modul yang setiap modul tersebut memiliki halaman utama, halaman tambah, halaman ubah, dan beberapa halaman lainnya. Pengembangan ulang pada sistem informasi ini hanya dilakukan pada bagian *frontend* saja. Bagian *frontend* yang dibangun ulang adalah bagian yang kode tersebut sangat rumit dan ditulis berulang kali pada *file* yang berbeda.

Vue.js adalah sebuah *framework* dan ekosistem yang mencakup sebagian besar fitur umum yang diperlukan dalam pengembangan *frontend*. *Vue.js* juga didesain untuk menjadi fleksibel dan dapat diadopsi secara bertahap contohnya komunitas seperti *framework Laravel*. *Vue.js* dapat digunakan dalam berbagai cara tergantung pada kasus, berikut adalah beberapa cara yang dapat digunakan pada *Vue.js*:

1. Meningkatkan HTML statis tanpa *build step* atau langkah pembuatan.
2. Menanamkan atau *Embedding* komponen *vue* di halaman mana pun.
3. Aplikasi satu halaman atau sering disebut *Single-Page Application* (SPA).
4. *Fullstack* / *Server-Side-Rendering* (SSR).
5. *Jamstack* / *Static-Site-Generation* (SSG).
6. Menargetkan desktop, seluler, WebGL, atau bahkan terminal.

Vue.js dibangun di atas HTML standar, CSS dan JavaScript yang menyediakan model pemrograman deklaratif dan berbasis komponen yang membantu pada saat pengembangan antarmuka pengguna secara efisien, baik itu sederhana atau kompleks². Pemrograman deklaratif merupakan suatu pendekatan dalam membuat struktur dan elemen dari aplikasi dengan mengekspresikan apa (what) yang sistem harus lakukan untuk menyelesaikan permasalahan, dibandingkan bagaimana (how) mengimplementasikan logika program untuk menyelesaikan permasalahan pada sistem [2]

Kode program yang baik pada sebuah perangkat lunak adalah kode yang dapat dipahami dengan mudah oleh pengembang. Salah satu prinsip untuk membuat kode yang baik adalah dengan

²<https://vuejs.org/guide/introduction.html> (diakses 26 Mei 2022)

menggunakan prinsip *Clean code*. *Clean code* adalah kode di dalam *software* yang formatnya benar dan disusun dengan baik sehingga pengembang lain dapat dengan mudah membaca atau memodifikasi kode tersebut³. Kelebihan dari menggunakan *clean code* adalah sebagai berikut:

1. *Understandability*, kode dapat dipahami dengan mudah.
2. *Readability*, kode dapat dibaca dengan mudah⁴.
3. *Changeability*, kode dapat dengan mudah diubah.
4. *Maintainability*, kode dapat dengan mudah dirawat atau dilakukan pengembangan.

Menurut Bjarne Stroustrup seorang penemu bahasa pemrograman C++, *clean code* adalah sebuah kode yang elegant dan efisien. Dimana *logic* harus lugas untuk membuatnya sulit untuk tersembunyi sebuah bug, dependensi minimal untuk kemudahan perawatan, dan kinerja yang optimal agar tidak membuat kode berantakan dengan optimasi yang tidak berprinsip [3]. Selain *clean code* terdapat juga sebuah prinsip lain yaitu DRY (*Don't Repeat Yourself*). Prinsip DRY ini untuk menghindari penulisan kode yang sama berulang kali pada tempat yang berbeda, karena ketika kode tersebut ingin diubah, maka seluruh kode yang ditulis berulang tersebut harus diubah. Hal ini sangat tidak efisien dan berpotensi untuk terjadinya kekurangan pembaruan pada beberapa bagian kode dan inkonsistensi [4].

Pada penelitian ini, sistem informasi penjadwalan kini dikembangkan ulang pada beberapa bagian, terutama pada bagian *frontend* dengan menggunakan *framework Vue.js*. Tujuan dari pengembangan ini adalah kode program yang sangat rumit dan kode yang ditulis berulang kali dapat digantikan dengan menggunakan komponen dari *Vue.js*. Sehingga kode menjadi *clean code*, menghindari pengulangan penulisan kode yang sama atau DRY, dan kode program menjadi lebih terstruktur dengan baik. Fitur pada sistem informasi penjadwalan yang dilakukan pengembangan ulang, tetap sama dengan sistem informasi sebelumnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah pada makalah ini adalah sebagai berikut:

1. Bagaimana cara memperbaiki sistem informasi penjadwalan kini pada bagian *frontend* agar menjadi *clean code*?
2. Bagaimana cara memanfaatkan *framework Vue.js* untuk memperbaiki komponen-komponen yang perlu diperbaiki?
3. Bagaimana membangun perangkat lunak sistem informasi yang merupakan perbaikan dari sistem informasi yang lama?

1.3 Tujuan

Berdasarkan rumusan masalah yang telah disebut, tujuan dari skripsi ini adalah sebagai berikut:

1. Memperbaiki sistem informasi penjadwalan kini pada bagian *frontend* agar menjadi *clean code*.
2. Memanfaatkan *framework Vue.js* untuk memperbaiki komponen-komponen yang perlu diperbaiki.
3. Membangun perangkat lunak sistem informasi yang merupakan perbaikan dari sistem informasi yang lama.

³<https://sis.binus.ac.id/2014/04/12/clean-code/> (diakses 26 Mei 2021)

⁴<https://www.oxfordlearnersdictionaries.com/definition/english/readability?q=readability> (diakses 7 Juni 2022)

1.4 Batasan Masalah

Batasan masalah pada skripsi adalah sebagai berikut:

1. *Browser* utama yang digunakan saat melakukan pengujian adalah *Google Chrome* sehingga belum diuji *compatibility* pada browser lain.
2. Perbaikan sistem informasi ini hanya pada bagian *frontend* saja dengan menggunakan *framework Vue.js*.

1.5 Metodologi

Metodologi masalah pada makalah ini adalah sebagai berikut:

1. Melakukan studi literatur mengenai penjadwalan, *framework Vue.js*, *framework Laravel*, *Spring framework*, algoritma genetik, *clean code* dan standar kode yang rapi dan terstruktur.
2. Mempelajari *framework Vue.js* yang digunakan dan fitur-fitur yang digunakan untuk menyelesaikan masalah.
3. Merancang ulang sistem informasi sebelumnya dengan menggunakan *framework Vue.js* pada bagian *frontend*
4. Melakukan pengujian fungsional dan pengujian experimental pada sistem informasi penjadwalan kuliah yang telah selesai dirancang dengan komponen *Vue.js*.
5. Menganalisa dan membandingkan sistem informasi penjadwalan kuliah sebelumnya dengan sistem informasi penjadwalan kuliah yang telah selesai dirancang dengan komponen *Vue.js*.
6. Mengambil kesimpulan dan saran yang diperoleh dari pengembangan sistem informasi penjadwalan kuliah.
7. Menulis dokumen skripsi.

1.6 Sistematika Pembahasan

Pembahasan penelitian dilakukan secara sistematis dengan detail sebagai berikut:

1. Bab 1 Pendahuluan
Bab ini membahas tentang gambaran umum pada skripsi ini. Isi dari bab ini berupa latar belakang pengembangan sistem informasi penjadwalan kuliah, rumusan masalah, tujuan, batasan masalah, metodologi serta sistematika pembahasan.
2. Bab 2 Landasan Teori
Bab ini membahas tentang landasan teori dari Penjadwalan, *Clean code*, Rekayasa perangkat lunak, *library* dan *Framework*, dan *Algoritma Genetik* yang menjadi landasan untuk membantu analisis penelitian sistem informasi penjadwalan kuliah ini.
3. Bab 3 Analisis
Bab ini membahas tentang pembahasan analisis sistem informasi penjadwalan kuliah, analisis masalah yang ingin diselesaikan dari analisis sistem masa kini, analisis pemilihan *framework*, analisis perancangna perangkat lunak dan analisis dampak.
4. Bab 4 Perancangan
Bab ini membahas tentang perancangan komponen, perancangan services, perancangan model pada komponen, dan perancangan perubahan halaman pada sistem informasi penjadwalan kuliah.
5. Bab 5 Implementasi dan Pengujian
Bab ini membahas tentang pembahasan lingkungan perangkat lunak, implementasi, pengujian fungsional dan pengujian experimental terhadap sistem informasi penjadwalan kuliah.
6. Bab 6 Kesimpulan dan Saran
Bab ini membahas tentang kesimpulan dan saran dari penelitian sistem informasi penjadwalan kuliah berdasarkan perancangan, implementasi dan pengujian yang telah dilakukan.