

FRONTEND DEVELOPMENT OF COURSE SCHEDULING SYSTEM INTEGRATED SIA AT ENGINEERING FACULTY UNIVERSITY OF JENDERAL SOEDIRMAN USING DEVOPS METHOD

Herfina Intan Yuanita^{*1}, Bangun Wijayanto², Teguh Cahyono³

^{1,2,3}Informatika, Fakultas Teknik, Universitas Jenderal Soedirman, Indonesia

Email: herfina.yuanita@mhs.unsoed.ac.id, bangun.wijayanto@unsoed.ac.id, teguh.cahyono@unsoed.ac.id

(Naskah masuk: 18 Maret 2022, Revisi : 25 Maret 2021, diterbitkan: 25 April 2022)

Abstract

The Course Scheduling System is a website-based system that has the function of compiling course schedules at the Faculty of Engineering, Jenderal Sudirman University. This system is used by several users, namely Bapendik/admin and vice dean, while the results of the scheduling will be inputted into SIA and viewed by SIA users. This system has features to manage lecturer data, course data, room data, slot data, day data, and arrange course scheduling. Previously, the preparation of courses was still using the manual method, namely by entering the course data one by one. Based on the existing problems, the author tries to build a frontend of the Course Scheduling System to facilitate users in compiling a computerized and automatic course schedule through the system. The system is built using the DevOps development method and the javascript library, namely React Js. The results of system testing show that all functions on the system can run according to user needs.

Keywords: *DevOps, Frontend, React Js, Scheduling, System.*

PERANCANGAN FRONT-END SISTEM PENJADWALAN MATA KULIAH TERINTEGRASI SIA DI FAKULTAS TEKNIK UNIVERSITAS JENDERAL SOEDIRMAN DENGAN METODE DEVOPS

Abstrak

Sistem Penjadwalan Mata Kuliah adalah sebuah sistem berbasis website yang memiliki fungsi untuk menyusun jadwal mata kuliah di Fakultas Teknik Universitas Jenderal Soedirman. Sistem ini digunakan oleh beberapa pengguna yaitu bapendik/admin dan wakil dekan, adapun hasil dari penjadwalan akan diinputkan ke SIA dan dilihat oleh pengguna SIA. Sistem ini memiliki fitur untuk mengelola data dosen, data mata kuliah, data ruangan, data slot, data hari, dan menyusun penjadwalan mata kuliah. Sebelumnya penyusunan mata kuliah masih menggunakan metode manual yaitu dengan cara memasukkan satu per satu data mata kuliah. Berdasarkan masalah yang ada, penulis mencoba untuk membangun frontend Sistem Penjadwalan Mata Kuliah untuk memudahkan pengguna dalam penyusunan jadwal mata kuliah secara terkomputerisasi dan otomatis melalui sistem. Sistem dibuat menggunakan metode pengembangan DevOps dan library javascript yaitu React Js. Pada hasil pengujian sistem menunjukkan bahwa seluruh fungsi pada sistem dapat berjalan sesuai dengan kebutuhan pengguna.

Kata kunci: *DevOps, Frontend, Penjadwalan, React Js, Sistem.*

1. PENDAHULUAN

Kemajuan teknologi informasi yang semakin pesat berpengaruh terhadap kehidupan manusia sehari-hari. Berbagai bidang kini dituntut untuk maju dan mengikuti perkembangan teknologi informasi yang bertujuan untuk membantu menyelesaikan permasalahan yang muncul dalam keseharian. Salah satunya pada bidang pendidikan, khususnya perguruan tinggi yang pembelajarannya memiliki keunikan yang berbeda dengan instansi

pendidikan lainnya. Pada jenjang perguruan tinggi mahasiswa dituntut untuk mandiri dalam memenuhi berbagai kebutuhannya selama sama kuliah. Seperti dalam memilih mata kuliah yang akan diambil setiap semester, mahasiswa akan mengakses sistem layanan dan fasilitas yang telah disediakan secara interaktif oleh pihak fakultas.

Jadwal kuliah merupakan salah satu aspek terpenting dalam perkuliahan. Penjadwalan mata kuliah adalah kegiatan administratif yang paling utama di universitas. Dalam masalah penjadwalan

mata kuliah, sejumlah mata kuliah yang dialokasikan ke sejumlah ruang yang tersedia dan sejumlah slot waktu disertai dengan constraints [1]. Dalam menyusun jadwal mata kuliah ada batasan-batasan tertentu yang harus dipenuhi, contohnya satu mata kuliah yang di jadwalkan pada sebuah ruang dan waktu diampu oleh seorang dosen tidak boleh bersamaan dengan mata kuliah lain yang diampu oleh dosen lain. Batasan-batasan seperti itulah yang membuat sulitnya menyusun jadwal mata kuliah, terlebih lagi jika komponen yang akan dijadwalkan berjumlah banyak maka proses penjadwalan akan semakin rumit.

Fakultas Teknik Universitas Jenderal Soedirman setiap semesternya melakukan proses penyusunan mata kuliah untuk lima program studi yang ada didalamnya. Namun, selama ini penyusunan mata kuliah yang dilakukan masih menemui beberapa kendala, seperti proses yang memakan waktu lama serta beberapa jadwal yang masih bentrok. Oleh karena itu, sebuah sistem penjadwalan mata kuliah sangat dibutuhkan di Fakultas Teknik Universitas Jenderal Soedirman.

Pada penelitian ini akan dirancang sebuah front-end sistem penjadwalan mata kuliah yang akan membantu mempermudah civitas akademika Fakultas Teknik Universitas Jenderal Soedirman dalam mengoperasikan sistem penjadwalan mata kuliah terotomatisasi. Output dari sistem ini berupa informasi jadwal mata pelajaran untuk prodi yang ada di Fakultas Teknik. Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun masa yang akan datang [2]. Metode pengumpulan data pada penelitian ini dilakukan dengan studi pustaka, wawancara dan studi literatur sejenis. Untuk pengembangan sistem dilakukan dengan metode DevOps dimana pengembangan (*development*) dilakukan sembari melakukan *operations*, dengan memanfaatkan sebuah *version control tools* yaitu gitlab selama proses pengembangan sistem. Front-end yaitu aplikasi yang akan menjadi representasi informasi yang akan dihadapkan oleh end-user, frontend terhubung dengan back-end yang menjadi penyedia informasi yang dibutuhkan [3]. Back-end sistem penjadwalan dibuat menggunakan Django. Data yang akan ditampilkan di front-end berasal dari API yang telah disediakan oleh bagian back-end dari sistem penjadwalan. API merupakan pemanggilan fungsi melalui Hyper Text Transfer Protocol (HTTP) dan mendapatkan respon berupa Extensible Markup Language (XML) atau JavaScript Object Notation (JSON), tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda [4]. Penelitian ini hanya akan berfokus pada perancangan front-end sistem penjadwalan dengan memanfaatkan sebuah *library javascript*, yaitu React Js.

2. METODE PENELITIAN

2.1. Metode Pengumpulan Data

2.1.1. Studi Pustaka

Studi Pustaka dilakukan melalui pengumpulan data dan informasi dengan cara membaca buku, jurnal maupun artikel yang dapat dijadikan acuan pembahasan topik yang diangkat dalam penelitian ini. Adapun pustaka yang dibutuhkan dalam penelitian ini diantaranya yaitu teori tentang konsep dasar sistem, jurnal tentang penjadwalan mata kuliah, frontend, javasript, react js, JSON, API, metode DevOps (Development and Operations), serta konsep dasar UML.

Sistem yaitu suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu [5], sistem penjadwalan mata kuliah ini pun memiliki beberapa fitur (seperti menu dashboard, master data dan menu pembuatan jadwal) dan memiliki tujuan (untuk membuat penjadwalan mata kuliah yang dilakukan secara otomatis) serta memiliki batasan. Penjadwalan mata kuliah harus memperhatikan batasan yaitu tidak boleh ada jadwal yang bertabrakan baik dosen yang sama atau waktu yang sama.

Frontend sistem ini dikembangkan melalui sebuah *library javascript* yaitu React Js, dengan bahasa *javascript*, yaitu bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode HTML dan di proses di sisi *client* dan digunakan dalam pembuatan *website* agar lebih interaktif dengan memberikan kemampuan tambahan terhadap HTML melalui eksekusi perintah di sisi *browser* [6].

React Js memiliki fitur *reusability* komponen yang memungkinkan *developer* untuk menggunakan kembali komponen yang telah dikembangkan ke aplikasi lain yang menggunakan fungsi yang sama [7]. Komponen di Reactjs dapat dibuat dengan dua cara yaitu dengan menggunakan fungsi dan dengan menggunakan class. Komponen yang dibuat dengan fungsi disebut juga dengan *function components* dan yang menggunakan class disebut *class components* (*class component*, cara membuatnya harus melakukan *extends* dari *class React.Component*). Pada *functionnal component* kita bisa mengganti *lifecycle method componentDidMount()* dan *componentDidUpdate()* yang dipakai pada *Class Component* dengan *React Hook useEffect()* [8]. Dalam React Js, terdapat 2 jenis komponen, yaitu *stateful component* dan *stateless component*. *Stateful components* adalah komponen yang menggunakan *state*. Sedangkan *Stateless components* adalah komponen yang tidak menggunakan *state*. *State* sendiri merupakan objek yang digunakan untuk menyimpan data di dalam komponen [9].

2.1.2. Wawancara

Wawancara adalah salah satu teknik pengumpulan data dengan cara mengajukan pertanyaan langsung kepada informan. Wawancara pada penelitian ini dilakukan pada awal kegiatan MBKM Fakultas Teknik dimulai, yaitu bulan September 2021, dengan narasumber yaitu dosen pembimbing kelompok MBKM yaitu Bapak Hari Siswanto, S.T., M.T., Ph.D. Dari wawancara yang dilakukan sudah cukup untuk mendapatkan *requirement* dari sistem penjadwalan secara keseluruhan.

2.1.3. Studi Literatur Sejenis

Studi literatur sejenis dilakukan dengan tujuan untuk mendapatkan topik tentang penelitian yang dilakukan dengan memahami suatu perbedaan-perbedaan dengan penelitian sejenis yang lain. Dari studi literatur sejenis yang telah dilakukan didapatkan beberapa informasi yang berguna untuk peneliti merancang frontend sistem penjadwalan. Metode pengembangan yang diambil oleh peneliti pun dibandingkan dengan metode-metode yang digunakan oleh literatur sejenis lainnya untuk mendapatkan metode yang paling sesuai untuk sistem yang akan dikembangkan.

2.2. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam mengembangkan sistem ini adalah metode DevOps (Developments & Operations). Metode ini mampu mempersingkat waktu antara pengembangan dan pengoperasian perangkat lunak tanpa mengurangi kualitas perangkat lunak [10]. Metode DevOps dibagi dalam beberapa tahapan yang terhubung satu sama lain dalam sebuah lifecycle, diantaranya: *plan*, *develop*, *build*, *test*, *deploy*, *operate*, dan *monitor*. Pada penelitian ini tahapan yang dilakukan hanya sampai dengan tahap *test*.

2.2.1. Tahap Perencanaan (*Plan*)

Pada tahap perencanaan, dilakukan proses identifikasi tujuan dan persyaratan untuk merancang dan mengembangkan perangkat lunak. Kegiatan lain yang dilakukan pada tahapan ini, diantaranya yaitu merencanakan manajemen proyek (merancang jadwal pembuatan proyek), studi literatur, studi pustaka, melakukan wawancara, menganalisis kebutuhan sistem dan kebutuhan pengguna, membuat desain UML (*use case*, *activity diagram* dan *sequence diagram*), serta membuat desain *mock up*.

2.2.2. Tahap Pengembangan (*Develop*)

Pada tahap ini dilakukan proses penulisan kode berdasarkan hasil dari tahap perencanaan. Kode front-end sistem dikembangkan dengan menggunakan bahasa pemrograman javascript

melalui library React Js dan dikelola dengan menggunakan sistem kontrol versi gitlab.

2.2.3. Tahap Pembangunan (*Build*)

Pada tahap ini dilakukan integrasi dari berbagai modul perangkat lunak dilakukan untuk membuat file yang dapat dieksekusi untuk fitur produk atau produk yang dikembangkan sepenuhnya. Pada tahap ini kode frontend sudah dihubungkan dengan backend melalui django rest API.

2.2.4. Tahap Pengujian (*Test*)

Pada tahap ini pengujian otomatis dilakukan secara terus-menerus untuk memastikan kualitas perangkat lunak yang telah dikembangkan. Pengujian pada penelitian ini menggunakan pengujian *system testing* menggunakan metode *blackbox testing*. *System testing* dilakukan untuk memastikan bahwa keseluruhan sistem berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (*user requirement*).

2.2.5. Tahap Deploy

Fase ini berfokus pada implementasi ulang perangkat lunak yang berkelanjutan di lingkungan produksi. Tahap ini melibatkan manajemen konfigurasi *platform* dan sumber daya target. Jika berhasil diimplementasikan, fitur atau produk yang telah dikembangkan siap untuk dirilis.

2.2.6. Tahap Operasional (*Operate*)

Pada fase ini, melibatkan instalasi infrastruktur, perubahan skalabilitas infrastruktur, manajemen dan konfigurasi infrastruktur, perencanaan kapasitas, manajemen kapasitas dan sumber daya, pengecekan keamanan, *recovery* data, manajemen *log/backup*, dan manajemen basis data.

2.2.7. Tahap Kontrol (*Monitor*)

Pada fase ini, kinerja sistem yang diimplementasikan dipantau dengan mengumpulkan dan menganalisis data penggunaan yang dapat membantu mendeteksi dan mengidentifikasi anomali dan memberikan umpan balik reguler untuk peningkatan perangkat lunak.

3. HASIL DAN PEMBAHASAN

3.1. Tahap Perencanaan (*Plan*)

Pada tahap ini dilakukan identifikasi tujuan, mendefinisikan kebutuhan dari sistem yang akan dikembangkan, membuat desain UML (*use case*, *activity diagram* dan *sequence diagram*), blok diagram arsitektur sistem, serta membuat desain *mock up*. Pengidentifikasi tujuan dan pendefinisian kebutuhan sistem didapatkan melalui metode wawancara dengan narasumber dari pihak

Fakultas Teknik Universitas Jenderal Soedirman. Pengumpulan data dan informasi ditujukan untuk menyusun *user requirement* dan *system requirement*.

3.1.1. Identifikasi Pengguna

Sistem Penjadwalan Mata Kuliah di Fakultas Teknik ini digunakan oleh 3 (tiga) jenis pengguna, yaitu:

a. Admin

Admin dari sistem ini adalah pegawai bapendik Fakultas Teknik Universitas Jenderal Soedirman. Admin diberi hak akses untuk membuat jadwal mata kuliah serta melihat data jadwal mata kuliah, dosen, ruangan, mata kuliah, slot jadwal dan hari.

b. Wakil Dekan Bidang Akademik

Wakil Dekan Bidang Akademik diberi akses untuk melihat jadwal mata kuliah dan diagram chart statistik data.

c. SIA

SIA dalam penelitian ini termasuk dalam kategori pengguna, karena hasil dari generate jadwal mata kuliah pada sistem penjadwalan mata kuliah nantinya akan diinput kemudian diakses oleh dosen dan mahasiswa melalui SIA.

3.1.2. Identifikasi Kebutuhan Pengguna

Berdasarkan identifikasi pengguna dan hasil wawancara maka didapat daftar kebutuhan pengguna dan prosedur yang dapat dilakukan oleh masing-masing pengguna. Tabel 1 menunjukkan kebutuhan pengguna dari sistem penjadwalan.

Tabel 1. *User Requirement*

No	Pengguna	Kebutuhan	User Requirement
1	Admin	Melakukan login	UR-Sijama-01
		Melihat dashboard	UR-Sijama-02
		Melihat daftar dosen	UR-Sijama-03
		Melihat daftar mata kuliah	UR-Sijama-05
		Mengupdate daftar mata kuliah	UR-Sijama-06
		Melihat daftar ruangan	UR-Sijama-07
		Mengupdate daftar ruangan	UR-Sijama-08
		Melihat daftar slot jadwal	UR-Sijama-09
		Mengupdate daftar slot jadwal	UR-Sijama-10
		Melihat daftar hari	UR-Sijama-11
		Mengupdate daftar hari	UR-Sijama-12
		Membuat jadwal mata kuliah	UR-Sijama-13
		Melihat beberapa tabel model jadwal mata kuliah yang telah dibuat	UR-Sijama-14
		Mengunduh file jadwal mata	UR-Sijama-15

2	Wakil Dekan Bidang Akademik	kuliah dalam ekstensi excel	
		Melakukan logout	UR-Sijama-16
		Melakukan login	UR-Sijama-17
		Melihat dashboard	UR-Sijama-18
		Melihat jadwal mata kuliah	UR-Sijama-19
3	SIA	Mengunduh file Jadwal mata kuliah dalam ekstensi excel	UR-Sijama-20
		Melakukan logout	UR-Sijama-21
		Mendapatkan hasil jadwal mata kuliah yang telah dibuat	UR-Sijama-22

3.1.3. Identifikasi Kebutuhan Sistem

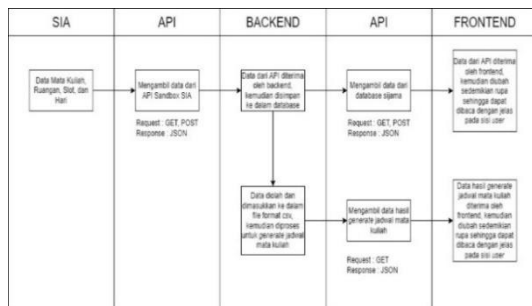
Setelah mengetahui analisis dari kebutuhan pengguna, maka dapat dirumuskan kebutuhan dari sistem yang akan dikembangkan. Kebutuhan sistem berisi proses-proses yang akan dijalankan oleh sistem saat dijalankan. Kebutuhan sistem dapat dilihat pada tabel 2.

No	Pengguna	Kebutuhan	System Requirement
1	Admin	Sistem menyediakan verifikasi login.	SRSijama.01
		Sistem menyediakan fasilitas untuk memberikan info-info yang ada melalui fitur dashboard.	SRSijama.02
		Sistem menyediakan fitur untuk melihat daftar dosen.	SRSijama.03
		Sistem menyediakan fitur untuk melihat daftar mata kuliah.	SRSijama.05
		Sistem menyediakan fitur untuk mengupdate daftar mata kuliah.	SRSijama.06
		Sistem menyediakan fitur untuk melihat daftar ruangan.	SRSijama.07
		Sistem menyediakan fitur untuk mengupdate daftar ruangan.	SRSijama.08
		Sistem menyediakan fitur untuk melihat daftar slot.	SRSijama.09
		Sistem menyediakan fitur untuk mengupdate daftar slot.	SRSijama.10
		Sistem menyediakan fitur untuk melihat daftar hari.	SRSijama.11
		Sistem menyediakan fitur untuk mengupdate daftar hari.	SRSijama.12
		Sistem menyediakan fitur untuk membuat jadwal mata kuliah secara otomatis.	SRSijama.13
		Sistem menyediakan fitur untuk melihat beberapa tabel model jadwal mata kuliah yang telah dibuat	SRSijama.14
		Sistem menyediakan fitur untuk mengunduh file Jadwal mata kuliah dalam ekstensi excel	SRSijama.15

		Sistem menyediakan fitur untuk logout.	SRSijama.16
		Sistem menyediakan verifikasi login	SRSijama.17
		Sistem menyediakan fasilitas untuk memberikan info-info/ statistik data yang ada melalui fitur dashboard	SRSijama.18
2	Wakil Dekan Bidang Akademik	Sistem menyediakan fitur untuk melihat jadwal mata kuliah.	SRSijama.19
		Sistem menyediakan fitur untuk mengunduh file Jadwal mata kuliah dalam ekstensi excel	SRSijama.20
		Sistem menyediakan fitur untuk logout.	SRSijama.21
3	SIA	Sistem menyediakan hasil generate jadwal yang nantinya digunakan pengguna untuk di inputkan ke SIA	SRSijama.22

3.1.4. Blok Diagram Arsitektur Sistem

Pada penelitian ini hanya berfokus pada bagian blok frontend seperti yang terlihat pada gambar 1, dimana prosesnya adalah bagian sisi *client* (frontend) melakukan *HTTP Request* dengan method *GET* dan atau *POST* ke *server* (backend) yang menyediakan *resource* melalui URI atau sering disebut dengan *endpoint*. Kemudian *REST Server* memberikan umpan balik berupa *HTTP Response* sesuai permintaan dari sisi *client*. *Server* (Backend) akan mengembalikan *response* kepada *client* dengan format *JSON*. Keluaran yang dihasilkan berformat *JSON* tersebut akan diubah sedemikian rupa sehingga dapat dibaca dengan jelas pada sisi *client* (frontend). Blok diagram arsitektur sistem penjadwalan mata kuliah dapat dilihat pada Gambar 1.



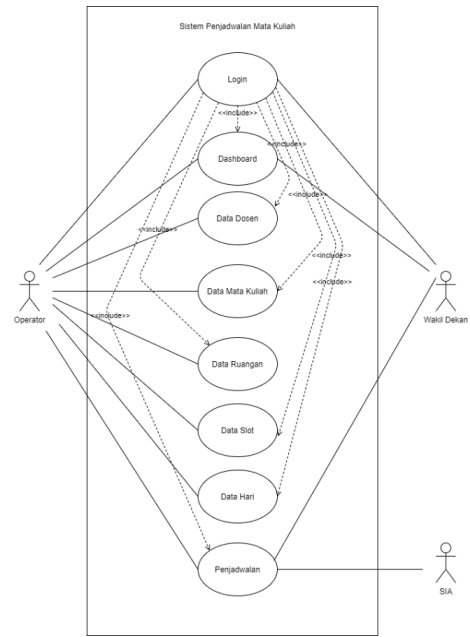
Gambar 1. Blok Diagram Arsitektur Sistem

3.1.5. Desain UML

Desain UML dibuat sebagai gambaran tentang bagaimana sistem akan dikembangkan UML adalah sebuah standarisasi bahasa pemodelan yang digunakan untuk pembangunan perangkat lunak yang dibangun menggunakan teknik pemrograman berorientasi objek [15]. Pada penelitian ini desain UML yang dibuat diantaranya yaitu *Use Case*, *Activity Diagram* serta *Sequence Diagram*.

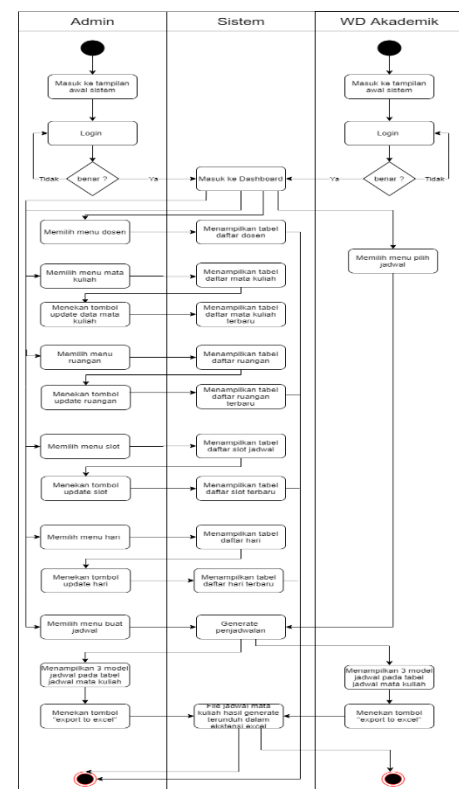
a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat, use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan *actor* yang berhak menggunakan fungsi - fungsi tersebut. Use Case Diagram sistem penjadwalan mata kuliah dapat dilihat pada Gambar 2.



Gambar 2. Use Case Diagram

b. *Activity Diagram*



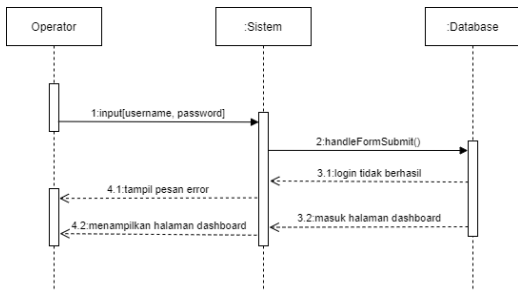
Gambar 3. Activity Diagram

Activity Diagram adalah gambaran aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak. Activity Diagram menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor. Activity Diagram sistem penjadwalan mata kuliah dapat dilihat pada Gambar 3.

c. *Sequence Diagram*

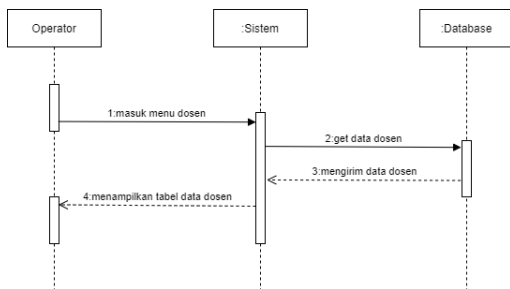
Sequence Diagram merupakan sebuah diagram yang menggambarkan interaksi dari objek-objek. Diagram ini akan menunjukkan pesan yang akan ditukarkan antara objek dalam sistem penjadwalan ketika saling berinteraksi dalam melakukan tugas tertentu. Berikut adalah sequence diagram dari Sistem Penjadwalan Mata Kuliah di Fakultas Teknik Unsoed:

- 1) Sequence Diagram Dashboard Admin
Sequence Diagram Dashboard Admin dapat dilihat pada gambar 4.



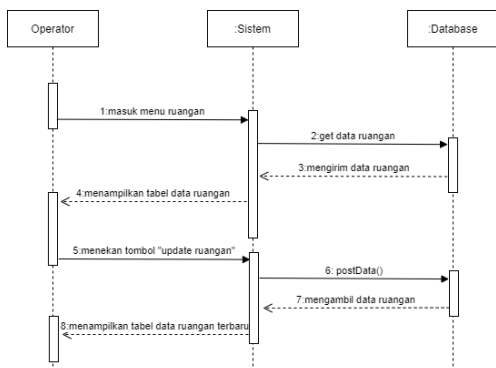
Gambar 4. *Sequence Diagram* Dashboard Admin

- 2) Sequence Diagram Dosen
Sequence Diagram Dosen dapat dilihat pada gambar 5.



Gambar 5. *Sequence Diagram* Dosen

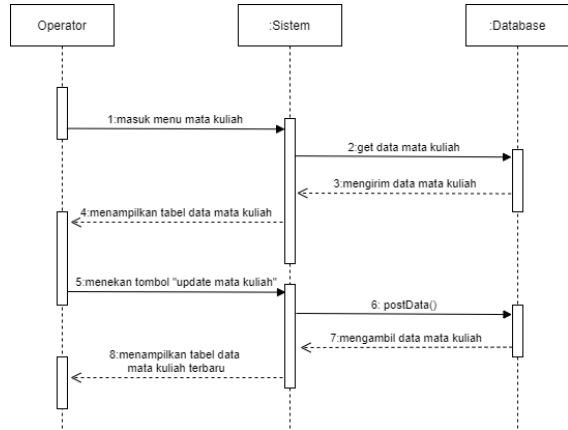
- 3) Sequence Diagram Ruang



Gambar 6. *Sequence Diagram* Ruang

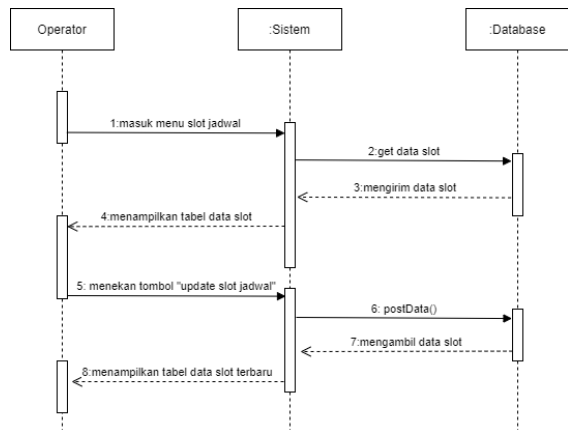
Sequence Diagram Ruang dapat dilihat pada gambar 6.

- 4) Sequence Diagram Mata Kuliah
Sequence Diagram Mata Kuliah dapat dilihat pada gambar 7.



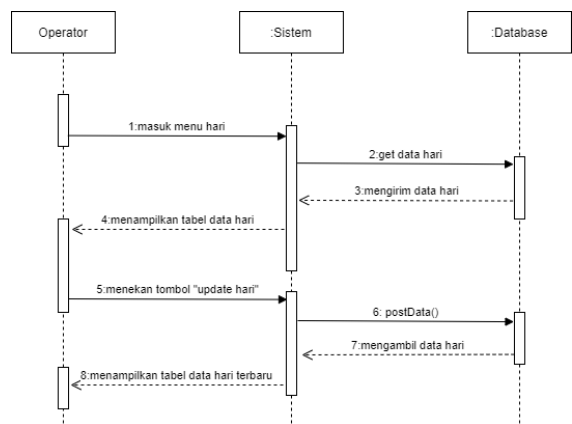
Gambar 7. *Sequence Diagram* Mata Kuliah

- 5) Sequence Diagram Slot Jadwal
Sequence Diagram Slot Jadwal dapat dilihat pada gambar 8.



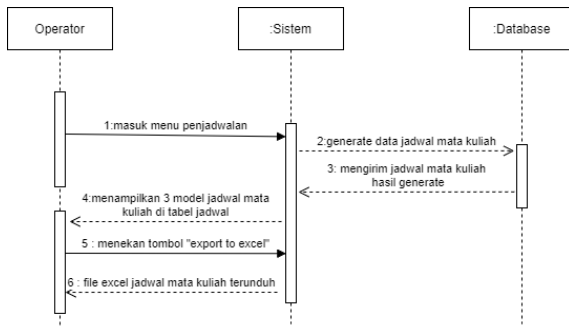
Gambar 8. *Sequence Diagram* Slot Jadwal

- 6) Sequence Diagram Hari
Sequence Diagram Hari dapat dilihat pada gambar 9.



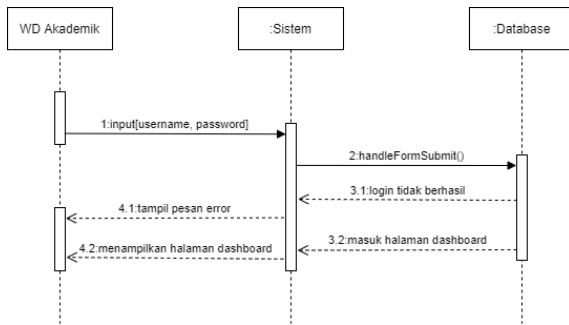
Gambar 9. *Sequence Diagram* Hari

7) Sequence Diagram Buat Jadwal
Sequence Diagram Buat Jadwal dapat dilihat pada gambar 10.



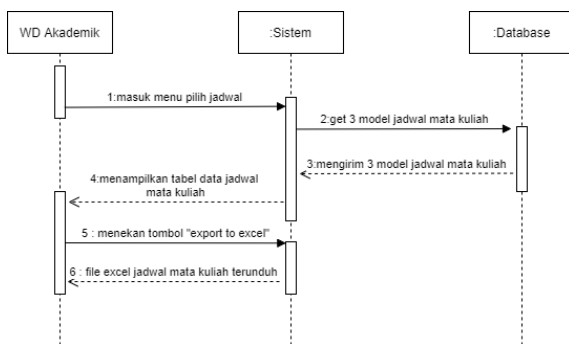
Gambar 10. Sequence Diagram Buat Jadwal

8) Sequence Diagram Dashboard WD Akademik
Sequence Diagram Dashboard WD Akademik dapat dilihat pada gambar 11.



Gambar 11. Sequence Diagram Dashboard WD Akademik

9) Sequence Diagram Lihat Jadwal Untuk WD
Sequence Diagram Lihat Jadwal dapat dilihat pada gambar 12.



Gambar 12. Sequence Diagram Lihat Jadwal

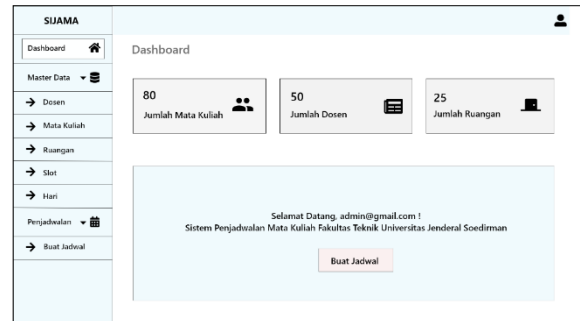
3.1.6. Perancangan Antarmuka Pengguna

Perancangan antarmuka dilakukan dengan mengacu *sequence diagram* yang telah dibuat. Rancangan antarmuka dibuat untuk menggambarkan hasil jadi dari sistem dari segi tampilan antarmuka sistem yang berinteraksi langsung dengan pengguna.

a. Perancangan Halaman Dashboard Admin

Gambar 13 merupakan halaman dashboard admin yang tampil setelah pengguna melakukan login menggunakan akun yang terdaftar sebagai

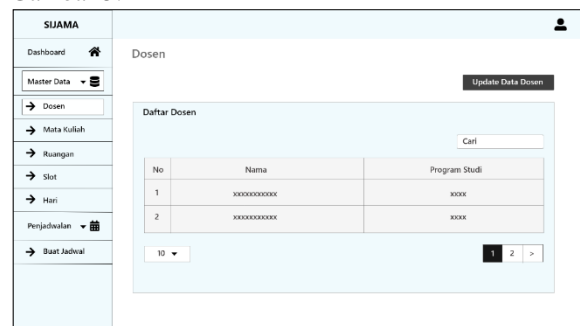
babendik. Halaman ini berisi jumlah data dosen, ruangan, slot, hari dan mata kuliah, serta menu pembuatan jadwal mata kuliah. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Dashboard Admin yang terdapat pada Gambar 4



Gambar 13. Halaman Dashboard Admin

b. Perancangan Halaman Master Data Dosen

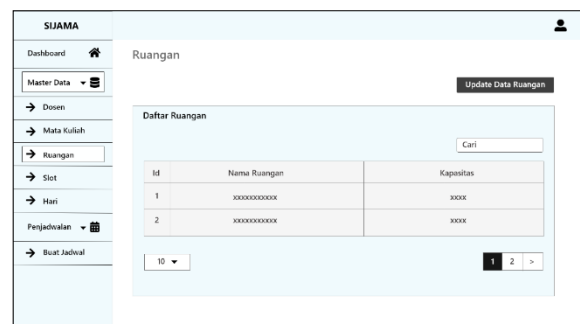
Gambar 14 merupakan halaman dosen yang menampilkan tabel daftar dosen di Fakultas Teknik Unsoed. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Dosen yang terdapat pada Gambar 5.



Gambar 14. Halaman Master Data Dosen

c. Perancangan Halaman Master Data Ruangan

Gambar 15 merupakan halaman ruangan yang menampilkan tabel daftar ruangan di Fakultas Teknik Unsoed. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Ruangan yang terdapat pada Gambar 6.



Gambar 15. Halaman Master Data Ruangan

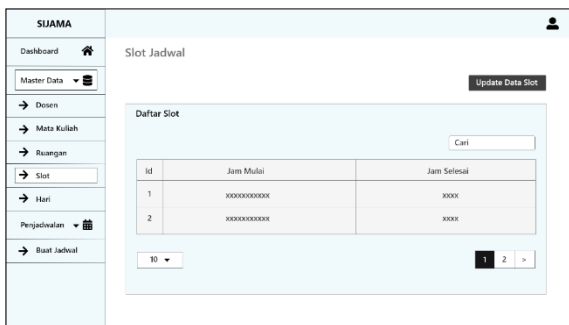
d. Perancangan Halaman Master Data Mata Kuliah

Gambar 16 merupakan halaman mata kuliah yang menampilkan tabel daftar mata kuliah. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Mata Kuliah yang terdapat pada Gambar 7.



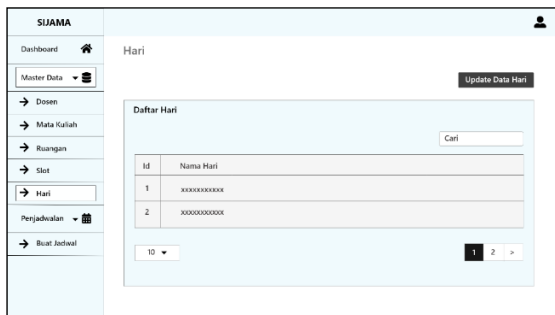
Gambar 16. Halaman Master Data Mata Kuliah

e. Perancangan Halaman Master Data Slot Jadwal
 Gambar 17 merupakan halaman slot jadwal yang menampilkan tabel daftar slot jam yang tersedia untuk jadwal mata kuliah. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Slot yang terdapat pada Gambar 8.



Gambar 17. Halaman Master Data Slot Jadwal

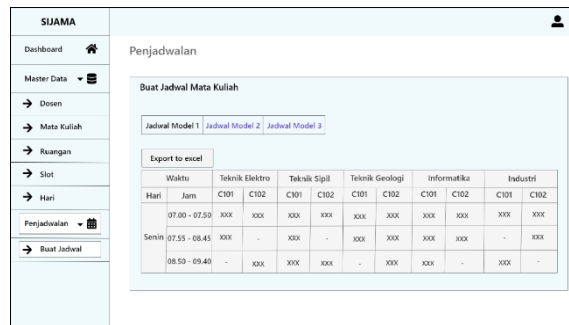
f. Perancangan Halaman Hari
 Gambar 18 merupakan halaman slot jadwal hari yang menampilkan tabel daftar hari. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Hari yang terdapat pada Gambar 9.



Gambar 18. Halaman Master Data Hari

g. Perancangan Halaman Buat Jadwal
 Gambar 19 merupakan halaman buat jadwal yang digunakan untuk membuat, melihat hasil olah data jadwal mata kuliah yang terdiri dari 3 model jadwal mata kuliah yang ada di dalam tabel. Rancangan antarmuka ini mengacu pada *Sequence*

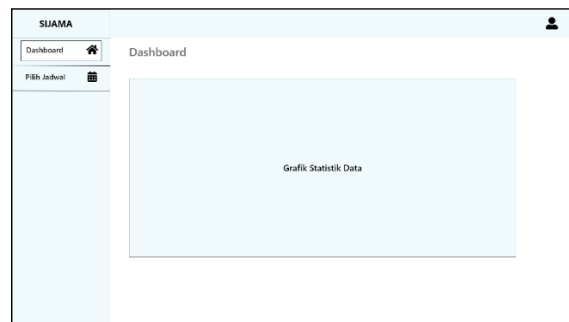
Diagram Buat Jadwal yang terdapat pada Gambar 10.



Gambar 19. Halaman Master Data Hari

h. Perancangan Halaman Dashboard WD Akademik

Gambar 20 merupakan halaman dashboard WD Akademik yang tampil setelah pengguna melakukan login menggunakan akun yang terdaftar sebagai wakil dekan II bidang akademik. Halaman ini berisi *chart* statistik jumlah master data tertentu. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Buat Jadwal yang terdapat pada Gambar 11.



Gambar 20. Halaman Master Data Hari

i. Perancangan Halaman Lihat Jadwal Untuk WD Akademik

Gambar 21 merupakan halaman lihat jadwal untuk wakil dekan bidang akademik yang tampil setelah pengguna membuka menu lihat jadwal. Halaman ini berisi 3 model jadwal mata kuliah yang sudah tergenerate yang kemudian akan dipilih oleh WDA. Rancangan antarmuka ini mengacu pada *Sequence Diagram* Lihat Jadwal yang terdapat pada Gambar 12.



Gambar 21. Halaman Master Data Hari

3.2. Tahap Pengembangan (Develop) dan Tahap Pembangunan (Build)

Pada tahap pengembangan dan pembangunan dilakukan proses pengkodean sistem penjadwalan mata kuliah. Dalam proses pengembangan dan pembangunan sistem, digunakan sebuah version control yaitu gitlab untuk mengelola setiap perubahan dari sistem baik dari sisi *back-end* developer ataupun sisi peneliti yang mengerjakan bagian *front-end*. *Version control* memungkinkan developer untuk berbagi file dengan cara yang mudah dan meminimalisir terjadinya saling mengambil, berbenturan dan menimpa perubahan kode yang mungkin telah diselesaikan *developer* lain. Karena *frontend* sistem dibangun dengan sebuah library javascript yaitu React Js, maka hal yang pertama dilakukan dalam membuat *frontend* sistem adalah menginstall React Js. Untuk menjalankan sistem, maka cukup membuka command prompt lalu masuk ke folder projectnya kemudian mengetik “*npm start*”.

Tampilan antarmuka yang sudah di koding akan terhubung dengan database melalui sebuah API yang telah disediakan oleh bagian *back-end* dari sistem penjadwalan. Berikut ini merupakan list tabel API yang digunakan untuk mengambil data yang akan ditampilkan dibagian *front-end* (Keterangan: semua *endpoint* diawali dengan <http://127.0.0.1:8000>):

Tabel 3. List API

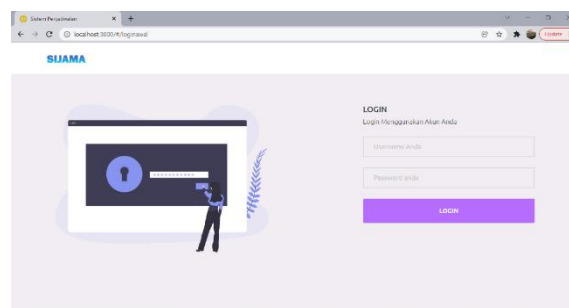
Kode	Endpoint	Method	Deskripsi
E1	/api/slot-mulai/	Get	Digunakan untuk mengambil data slot dari database sijama
E2	/api/api-slot/	Post	Digunakan untuk menyimpan data slot dari sandbox SIA ke dalam database sijama
E3	/api/ruangan/	Get	Digunakan untuk mengambil data ruangan dari database sijama
E4	/api/api-ruangan/	Post	Digunakan untuk menyimpan data ruangan dari sandbox SIA ke dalam database sijama
E5	/api/hari/	Get	Digunakan untuk mengambil data hari dari database sijama
E6	/api/api-hari/	Post	Digunakan untuk menyimpan data hari dari sandbox SIA ke dalam database sijama
E7	/api/mata-kuliah/	Get	Digunakan untuk mengambil data mata kuliah dari database sijama
E8	/api/api-matkul/	Post	Digunakan untuk menyimpan data mata kuliah dari sandbox SIA ke dalam database sijama
E9	/api/dosen/	Get	Digunakan untuk mengambil data dosen dari database sijama
E10	/api/schedule2/	Get	Digunakan untuk mengenerate/ generate ulang Jadwal mata

E11	/api/user/	Post	kuliah dari backend sijama Digunakan untuk mengirim data username dan password pada fitur login
E12	/auth/	Post	Digunakan untuk mengirim data username dan password agar mendapatkan token untuk fitur login

Pada tahap pengembangan dan pembangunan, desain sistem diimplementasikan kedalam bentuk perintah – perintah yang dimengerti komputer dengan mempergunakan sebuah *library javascript* yaitu *React Js*. Berikut ini adalah hasil tampilan sistem penjadwalan mata kuliah :

a. Tampilan Login

Untuk masuk ke dalam sistem penjadwalan mata kuliah dibuat sebuah fitur login dengan menginputkan *username* dan *password* yang sesuai dengan data *user* yang ada di database sistem penjadwalan. Gambar 22 merupakan tampilan halaman login sedangkan Gambar 23 merupakan kode pemanggilan api user (E11 pada tabel list API) untuk mengirim data username dan password pada fitur login serta api auth dengan method POST (E12 pada tabel list API) untuk mengirim data *username* dan *password* agar mendapatkan token untuk validasi pada fitur login.



Gambar 22. Antarmuka Login

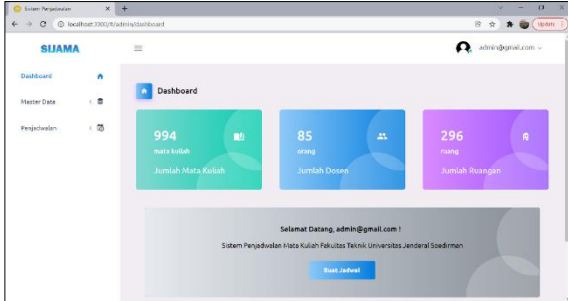
```
const handleFormSubmit = (event) => {
  axios
    .post(api + "/api/user/", requestBody,
  config)
    .then(async (res) => {
      const data_token = await axios.post(
        api + "/auth/",
        qs.stringify(requestBody),
        config
      );
    })
}
```

Gambar 23. Kode Pemanggilan API pada Bagian Login

b. Tampilan Dashboard Admin

Setelah admin berhasil login, maka tampilan pertama yang akan muncul adalah halaman dashboard admin. Dalam halaman ini, terdapat informasi mengenai jumlah mata kuliah, dosen dan ruangan serta jadwal mata kuliah. Gambar 24

merupakan tampilan halaman dashboard admin sedangkan Gambar 25 merupakan kode pemanggilan api mata kuliah, dosen dan ruangan (E7, E9, E3 pada tabel list API) untuk mengambil jumlah data mata kuliah, dosen dan ruangan dari *database*.



Gambar 24. Antarmuka Dashboard Admin

```

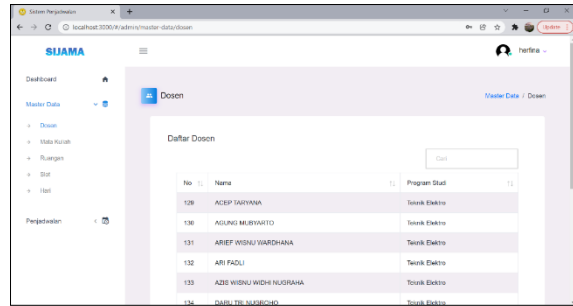
let getMatkul = async () => {
  let response = await fetch("/api/mata-kuliah/",
  {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  });
  let datamatkul = await response.json();
  setMatkul(datamatkul);
};
let getDosen = async () => {
  let response = await fetch("/api/dosen/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  });
  let datadosen = await response.json();
  setDosen(datadosen);
};
let getRuangan = async () => {
  let response = await fetch("/api/ruangan/",
  {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  });
  let dataruangan = await response.json();
  setRuangan(dataruangan);
};

```

Gambar 25. Kode Pemanggilan API pada Dashboard Admin

c. Tampilan Master Data Dosen

Ketika membuka menu master data dosen, maka akan muncul tabel daftar dosen. Gambar 26 merupakan tampilan halaman master data dosen sedangkan Gambar 27 merupakan kode pemanggilan api dosen (E9 pada tabel list API) untuk mengambil data dosen dari *database* agar tampil di *frontend* sistem.



Gambar 26. Antarmuka Master Data Dosen

```

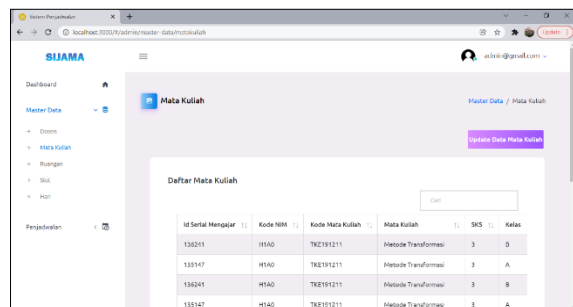
refreshList() {
  fetch("/api/dosen/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then((response) => response.json())
  .then((data) => {
    this.setState({ dosen: data });
  });
}

```

Gambar 27. Kode Pemanggilan API pada master data dosen

d. Tampilan Master Data Mata Kuliah

Ketika membuka menu master data kuliah, maka akan muncul tabel daftar mata kuliah. Dalam menu mata kuliah, data bisa diperbarui dengan mengklik tombol “update data mata kuliah”. Gambar 28 merupakan tampilan halaman master data mata kuliah sedangkan Gambar 29 merupakan kode pemanggilan api mata kuliah dengan method GET (E7 pada tabel list API) untuk mengambil data mata kuliah dari *database* agar tampil di *frontend* sistem serta api mata kuliah dengan method POST (E8 pada tabel list API) untuk menyimpan data mata kuliah dari *sandbox* SIA ke dalam *database* saat tombol “update data mata kuliah” ditekan.



Gambar 28. Antarmuka Master Data Mata Kuliah

```

refreshList() {
  fetch("/api/mata-kuliah/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
}

```

```

.then((response) => response.json())
.then((data) => {
  this.setState({ matkul: data });
});
}

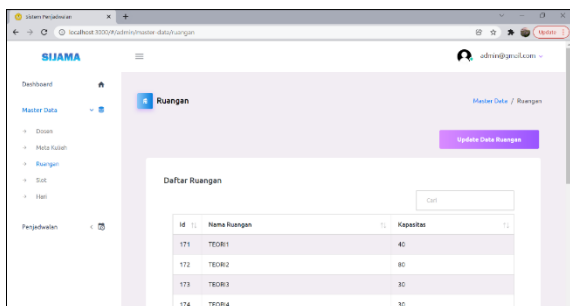
postData() {
  this.setState({ isLoading: true });
  fetch("/api/api-matkul/", {
    method: "POST",
    headers: {
      Accept: "application/json",
      "Content-Type": "application/json",
    },
    body: JSON.stringify(),
  })
}

```

Gambar 29. Kode Pemanggilan API pada master data mata kuliah

e. Tampilan Master Data Ruang

Ketika membuka menu master ruangan, maka akan muncul tabel daftar ruangan. Dalam menu mata kuliah, data bisa diperbarui dengan mengklik tombol “update data ruangan”. Gambar 30 merupakan tampilan halaman master data ruangan sedangkan Gambar 31 merupakan kode pemanggilan api ruangan dengan method GET (E3 pada tabel list API) untuk mengambil data ruangan dari *database* agar tampil di *frontend* sistem serta api ruangan dengan method POST (E4 pada tabel list API) untuk menyimpan data ruangan dari *sandbox* SIA ke dalam *database* saat tombol “update data ruangan” ditekan.



Gambar 30. Antarmuka Master Data Ruang

```

refreshList() {
  fetch("/api/ruangan/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then((response) => response.json())
  .then((data) => {
    this.setState({ ruangan: data });
  });
}

postData() {
  this.setState({ isLoading: true });
  fetch("/api/api-ruangan/", {
    method: "POST",

```

```

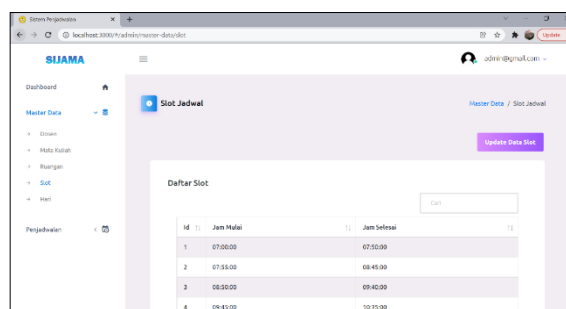
headers: {
  Accept: "application/json",
  "Content-Type": "application/json",
},
body: JSON.stringify(),
})

```

Gambar 31. Kode Pemanggilan API pada master data ruangan

f. Tampilan Master Data Slot Jadwal

Ketika membuka menu master slot, maka akan muncul tabel daftar slot. Dalam menu slot, data bisa diperbarui dengan mengklik tombol “update data slot”. Gambar 32 merupakan tampilan halaman master data slot jadwal sedangkan Gambar 33 merupakan kode pemanggilan api slot jadwal dengan method GET (E1 pada tabel list API) untuk mengambil data slot jadwal dari *database* agar tampil di *frontend* sistem serta api slot jadwal dengan method POST (E2 pada tabel list API) untuk menyimpan data slot jadwal dari *sandbox* SIA ke dalam *database* saat tombol “update data slot” ditekan.



Gambar 32. Antarmuka Master Data Slot Jadwal

```

refreshList() {
  fetch("/api/slot-mulai/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then((response) => response.json())
  .then((data) => {
    this.setState({ slot: data });
  });
}

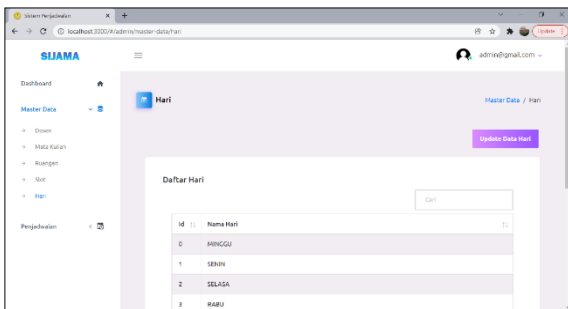
postData() {
  this.setState({ isLoading: true });
  fetch("/api/api-slot/", {
    method: "POST",
    headers: {
      Accept: "application/json",
      "Content-Type": "application/json",
    },
    body: JSON.stringify(),
  })
  .then((res) => res.json())

```

Gambar 33. Kode Pemanggilan API pada master data slot jadwal

g. Tampilan Master Data Hari

Ketika membuka menu master hari, maka akan muncul tabel daftar hari. Dalam menu master data hari, data bisa diperbarui dengan mengklik tombol "update data hari". Gambar 34 merupakan tampilan halaman master data hari sedangkan Gambar 35 merupakan kode pemanggilan api hari dengan method GET (E5 pada tabel list API) untuk mengambil data hari dari *database* agar tampil di *frontend* sistem serta api hari dengan method POST (E6 pada tabel list API) untuk menyimpan data hari dari *sandbox* SIA ke dalam *database* saat tombol "update data hari" ditekan.



Gambar 34. Antarmuka Master Data Hari

```
refreshList() {
  fetch("/api/hari/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
      // 'Authorization': `Token
      ${token['token']}`,
    },
  })
  .then((response) => response.json())
  .then((data) => {
    this.setState({ hari: data });
  });
}

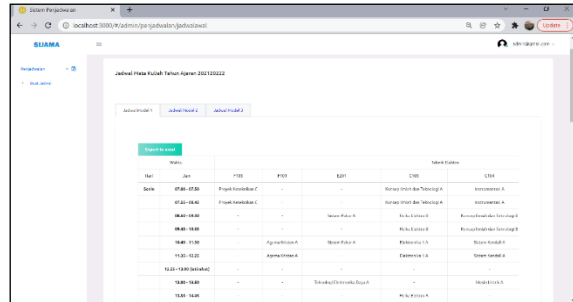
postData() {
  this.setState({ isLoading: true });
  fetch("/api/api-hari/", {
    method: "POST",
    headers: {
      Accept: "application/json",
      "Content-Type": "application/json",
    },
    body: JSON.stringify(),
  })
}
```

Gambar 35. Kode Pemanggilan API pada master data slot jadwal

h. Tampilan Buat Jadwal

Pada halaman ini dilakukan *generate* jadwal secara otomatis dari sistem, jadwal akan ditampilkan dalam sebuah tabel jadwal mata kuliah. Gambar 36 merupakan tampilan halaman buat jadwal sedangkan Gambar 37 merupakan kode pemanggilan api schedule (E10 pada tabel list API) untuk mengambil

hasil generate jadwal mata kuliah agar tampil di *frontend* sistem.



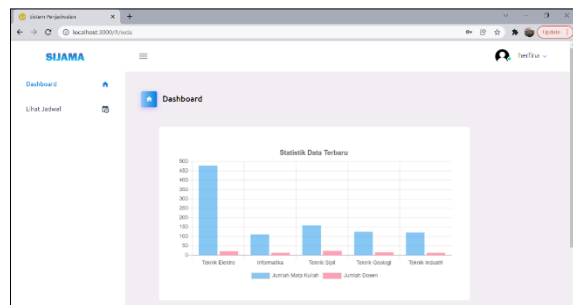
Gambar 36. Antarmuka Buat Jadwal

```
refreshList() {
  fetch("/api/schedule2/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then((response) => response.json())
  .then((data) => {
    this.setState({ jadwal: data });
  });
}
```

Gambar 37. Kode Pemanggilan API pada bagian buat jadwal

i. Tampilan Dashboard Wakil Dekan Bidang Akademik

Setelah WDA berhasil login, maka tampilan yang akan muncul adalah halaman dashboard wda. Dalam halaman ini, terdapat informasi mengenai chart statistic data tertentu. Gambar 38 merupakan tampilan halaman dashboard WDA sedangkan Gambar 39 merupakan kode pemanggilan api mata kuliah dan dosen (E7 dan E9 pada tabel list API) untuk mengambil jumlah data mata kuliah dan dosen agar tampil di *frontend* sistem.



Gambar 38. Antarmuka Dashboard WDA

```
const chart = () => {
  fetch("/api/mata-kuliah", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
      Authorization: `Token
      ${token["token"]}`,
    },
  })
}
```

```

    })

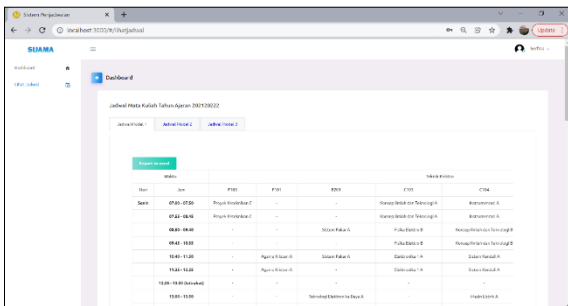
    fetch("/api/dosen", {
      method: "GET",
      headers: {
        "Content-Type": "application/json",
        Authorization: `Token
        ${token["token"]}`,
      },
    })
  }
}

```

Gambar 39. Kode Pemanggilan API pada Dashboard WDA

j. Tampilan Lihat Jadwal Untuk Wakil Dekan Bidang Akademik

Dalam menu lihat jadwal, akan menampilkan 3 model jadwal mata kuliah hasil *generate* pada tabel jadwal mata kuliah. Gambar 40 merupakan tampilan halaman lihat jadwal sedangkan Gambar 41 merupakan kode pemanggilan api *schedule* (E10 pada tabel list API) untuk mengambil hasil *generate* jadwal mata kuliah agar tampil di frontend sistem.



Gambar 40. Antarmuka Lihat Jadwal

```

refreshList() {
  fetch("/api/schedule2/", {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  })
  .then((response) => response.json())
  .then((data) => {
    this.setState({ jadwal: data });
  });
}

```

Gambar 41. Kode Pemanggilan API pada bagian lihat Jadwal

3.3. Tahap Pengujian (Test)

Pada tahap ini pengujian dilakukan secara terus-menerus untuk memastikan kualitas perangkat lunak yang telah dikembangkan. Pengujian pada penelitian ini menggunakan pengujian *system testing* menggunakan metode *blackbox testing*. *System testing* dilakukan untuk memastikan bahwa keseluruhan sistem berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (*user requirement*). Berikut ini *test case* yang sudah dibuat dan diujikan

kepada pengguna. Dari pengujian yang dilakukan diterima hasil valid/berhasil untuk setiap fitur pada sistem penjadwalan, hal ini berarti sistem sudah sesuai dengan *user requirement*.

User	Type	Scenario	Test Case	Procedure	Expected Result	Test Result
Bapendik (Admin)	Positive	Mengecek fungsionalitas login	Login berhasil dengan mengisi username dan password yang benar	User membuka sistem, User mengisi username dan password dengan benar dan mengklik tombol login	User berhasil masuk ke halaman dashboard	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Login gagal karena mengisi username yang salah dan password yang benar	User membuka sistem, User mengisi username yang salah dan password yang benar kemudian mengklik tombol login	Muncul pesan peringatan bahwa username/passw ord yang diisi salah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Login gagal karena mengisi username yang benar dan password yang salah	User membuka sistem, User mengisi username yang benar dan password yang salah kemudian mengklik tombol login	Muncul pesan peringatan bahwa username/passw ord yang diisi salah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas dashboard admin	Berhasil melihat informasi yang ada di dashboard	User berhasil login dan masuk ke halaman dashboard	User dapat melihat informasi jadwal mata kuliah serta jumlah data dosen, mata kuliah dan ruangan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas master data dosen	Berhasil melihat tabel daftar dosen	User membuka menu master data dosen	User dapat melihat tabel yang berisi informasi daftar dosen	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Berhasil melihat tabel daftar mata kuliah	User membuka menu master data mata kuliah	User dapat melihat tabel yang berisi informasi daftar mata kuliah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas master data mata kuliah	Berhasil mengupdate tabel daftar mata kuliah	User menekan tombol update data mata kuliah	Muncul pesan (toast) berhasil di update dan daftar mata kuliah terbaru muncul pada tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Gagal mengupdate tabel daftar mata kuliah	User menekan tombol update data mata kuliah	Muncul pesan (toast) gagal di update	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas master data ruangan	Berhasil melihat tabel daftar ruangan	User membuka menu master data ruangan	User dapat melihat tabel yang berisi informasi daftar ruangan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Berhasil mengupdate tabel daftar ruangan	User menekan tombol update data ruangan	Muncul pesan (toast) berhasil di update dan daftar ruangan terbaru muncul pada tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Gagal mengupdate tabel daftar ruangan	User menekan tombol update data ruangan	Muncul pesan (toast) gagal di update	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Berhasil melihat tabel daftar slot	User membuka menu master data slot	User dapat melihat tabel yang berisi informasi slot	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas master slot	Berhasil mengupdate tabel daftar slot	User menekan tombol update data slot	Muncul pesan (toast) berhasil di update dan daftar slot terbaru muncul pada tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Gagal mengupdate tabel daftar slot	User menekan tombol update data slot	Muncul pesan (toast) gagal di update	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Berhasil melihat tabel daftar hari	User membuka menu master data hari	User dapat melihat tabel yang berisi informasi hari	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas master hari	Berhasil mengupdate tabel daftar hari	User menekan tombol update data hari	Muncul pesan (toast) berhasil di update dan daftar hari terbaru muncul pada tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Gagal mengupdate tabel daftar hari	User menekan tombol update data hari	Muncul pesan (toast) gagal di update	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

Wakil Dekan Bidang Akademik	Positive	Mengecek fungsionalitas menu buat jadwal	Berhasil membuat jadwal mata kuliah	User membuka menu buat jadwal	3 model jadwal mata kuliah terbuat dan muncul pada tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Berhasil mengunduh jadwal mata kuliah dalam ekstensi file excel	User menekan tombol "export to excel"	File excel Jadwal mata kuliah berhasil terunduh	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Login berhasil dengan mengisi username dan password yang benar	User membuka sistem, User mengisi username dan password dengan benar dan mengklik tombol login	User berhasil masuk ke halaman dashboard	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative	Mengecek fungsionalitas login	Login gagal karena mengisi username yang salah dan password yang benar	User membuka sistem, User mengisi username yang salah dan password yang benar kemudian mengklik tombol login	Muncul pesan peringatan bahwa username/passw ord yang diisi salah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Negative		Login gagal karena mengisi username yang benar dan password yang salah	User membuka sistem, User mengisi username yang benar dan password yang salah kemudian mengklik tombol login	Muncul pesan peringatan bahwa username/passw ord yang diisi salah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas dashboard WDA	Berhasil melihat informasi yang ada di dashboard	User berhasil login dan masuk ke halaman dashboard	User dapat melihat informasi chart statistik data tertentu	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive	Mengecek fungsionalitas Menu Lihat Jadwal	Berhasil melihat tabel Jadwal mata kuliah	User berhasil login dan masuk ke halaman Lihat Jadwal	User dapat melihat informasi 3 model jadwal mata kuliah pada tabel	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil
	Positive		Berhasil mengunduh jadwal mata kuliah dalam ekstensi file excel	User menekan tombol "export to excel"	File excel Jadwal mata kuliah berhasil terunduh	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Tidak Berhasil

Gambar 42. Test Case Sistem Penjadwalan Mata Kuliah

4. KESIMPULAN

Penelitian ini menunjukkan bahwa sistem penjadwalan mata kuliah berhasil mengatasi permasalahan penjadwalan yang sebelumnya ada, hal ini ditunjukkan dengan sistem yang berhasil mengenerate tiga model jadwal mata kuliah. Penelitian ini juga menunjukkan bahwa dengan membangun *frontend* sistem menggunakan sebuah *library javasript* yaitu *react js* akan meminimalkan penulisan kode dengan adanya fitur *reusable component*. Namun disisi lain, *react js* memiliki dua cara membuat *component* yaitu *function component* dan *class component* yang sangat berbeda penulisan *code*-nya, sehingga peneliti cukup memakan waktu yang tidak sebentar untuk mempelajari keduanya. *Frontend* sistem penjadwalan berhasil dibuat dan sudah sesuai dengan *user requirement*. Hal ini dapat diketahui dengan melihat hasil *test case* pada tabel *Test Case Sistem Penjadwalan Mata Kuliah*, pada Tabel 4, yang telah diisi pengguna/responden yang akan menggunakan sistem tersebut. Peneliti menyadari bahwa sistem penjadwalan yang sudah dibangun masih perlu dikembangkan lebih lanjut dengan menambah fitur yang akan membuat sistem lebih kompleks dan semakin memudahkan proses penyusunan jadwal mata kuliah di Fakultas Teknik Universitas Jenderal Soedirman. Adapun saran yang bisa peneliti sampaikan untuk pengembangan sistem di masa mendatang diantaranya yaitu fitur buat

jadwal pada sistem penjadwalan mata kuliah masih perlu perkembangan lebih lanjut, diharapkan adanya fitur upload hasil generate jadwal mata kuliah ke SIA Unsoed pada pengembangan selanjutnya serta login SSO kori unsoed sebaiknya diterapkan pada perkembangan selanjutnya.

DAFTAR PUSTAKA

- [1] Tyas, "Analisis Sistem Informasi Penjadwalan Universitas 'Aisyiyah Yogyakarta," Jurnal Ilmiah NERO, vol. 4, no. 2, pp. 135-143, 2019.
- [2] N. Khaerunnisa and Nofiyati, "Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Web Studi Kasus Desa Sidakangen Purbalingga," Jurnal Teknik Informatika (JUTIF), vol. 1, no. 1, pp. 25-33, 2020.
- [3] H. Fakhrol, "Pengembangan Front-End Aplikasi Uiiportofolio Berbasis Web," Universitas Islam Indonesia, Yogyakarta, 2018.
- [4] Hasanudin, H. Asgar and B. Hartono, "Rancang Bangun Rest Api Aplikasi Weshare Sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan," Jurnal Informatika Tekonologi dan Sains (JINTEKS), vol. 4, no. 1, pp. 8-14, 2022.
- [5] Jogyanto, Analisis dan Desain Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis, Yogyakarta: Andi.
- [6] Chlaruman, "Rancang Bangun Frontend Pada Aplikasi Troubleshoot Kendaraan Roda Empat "OTOKU" Berbasis Android," Politeknik Negeri Jakarta, Jakarta, 2020.
- [7] B. University, "Apa itu React Js," 2019. [Online]. Available: <https://socs.binus.ac.id/2019/12/30/apa-itu-react-js/>. [Accessed 1 Oktober 2021].
- [8] Y. Agae, "React Class Component vs Function Component," 2020. [Online]. Available: <https://medium.com/devsaurus-class/react-class-component-vs-function-component-10937b8fc513>. [Accessed 5 Maret 2022].
- [9] I. Musyaffa, "Metode Pengembangan DevOps," 2021. [Online]. Available: <https://agus-hermanto.com/blog/detail/metode-pengembangan-devops>. [Accessed 1 Oktober 2021].
- [10] F. Ismayadi, "Implementasi Application Programmning Interface Untuk Integrasi Data Toko Cahaya Meuble Dengan Toko Selaparang Meubel," Universitas Mataram, Mataram, 2021.
- [11] A. Kadir, Pengenalan Sistem Informasi,

- Yogyakarta: Andi.
- [12] A. Muhandian, "Tutorial Reactjs #3: Segala Hal yang Harus Kamu Ketahui tentang Components di React," 2021. [Online]. Available: <https://www.petanikode.com/reactjs-komponen/>. [Accessed 1 Oktober 2021].
 - [13] U. Tohirin, Widiyanto and Mauludyans, "Implementasi DevOps pada Pengembangan Aplikasi e-Skrining Covid," *Jurnal Mutinetics*, vol. 1, no. 6, pp. 15-20, 2020.
 - [14] A. S. Oktriwina, 2020. [Online]. Available: <https://glints.com/id/lowongan/reactjs-adalah/#.YiG22p5Bw2x>. [Accessed 1 Oktober 2021].
 - [15] R. B. Pamungkas, "Apa itu React JS? Panduan Lengkap untuk Pemula," 2020. [Online]. Available: <https://www.niagahoster.co.id/blog/react-js-adalah/>. [Accessed 1 Oktober 2021].
 - [16] M. A. Prastetiyo and A. Triawan, "Penerapan Web Service (XML dan JSON) Untuk Meningkatkan Performance Pada Informasi Pembayaran Uang Kuliah," *Jurnal Ilmiah Teknologi-Informasi dan Sains (TeknoIS)*, vol. 1, no. 8, pp. 62-77, 2018.
 - [17] I. A. Ramadhani, "Pengembangan Sistem Informasi Penjadwalan Mata Kuliah Berbasis Web Di Fakultas Teknik Universitas Negeri Makassar," *Jurnal Pendidikan*, vol. 2, no. 9, pp. 1-17, 2018.
 - [18] A. S. Rosa and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, Bandung: Informatika.
 - [19] R. K. Safitri and H. P. Putro, "Implementasi REST API untuk Komunikasi Antara ReactJS dan NodeJS (Studi Kasus : Modul Manajemen User Solusi247)," *Jurnal Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia*, pp. 1-5, 2021.
 - [20] R. Sayyidati, Muri and S. U. Hendrik, "Search Engine Get Application Programming Interface," *Jurnal Sains dan Informatika*, vol. 2, no. 5, pp. 88-97, 2019