

**IMPLEMENTASI REACT.JS PADA PENGEMBANGAN
FRONTEND SISTEM INFORMASI MANAJEMEN
KADER PARTAI**



Disusun Oleh:

N a m a : Ahmad Raihan Akhdani
NIM : 18523216

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2020**

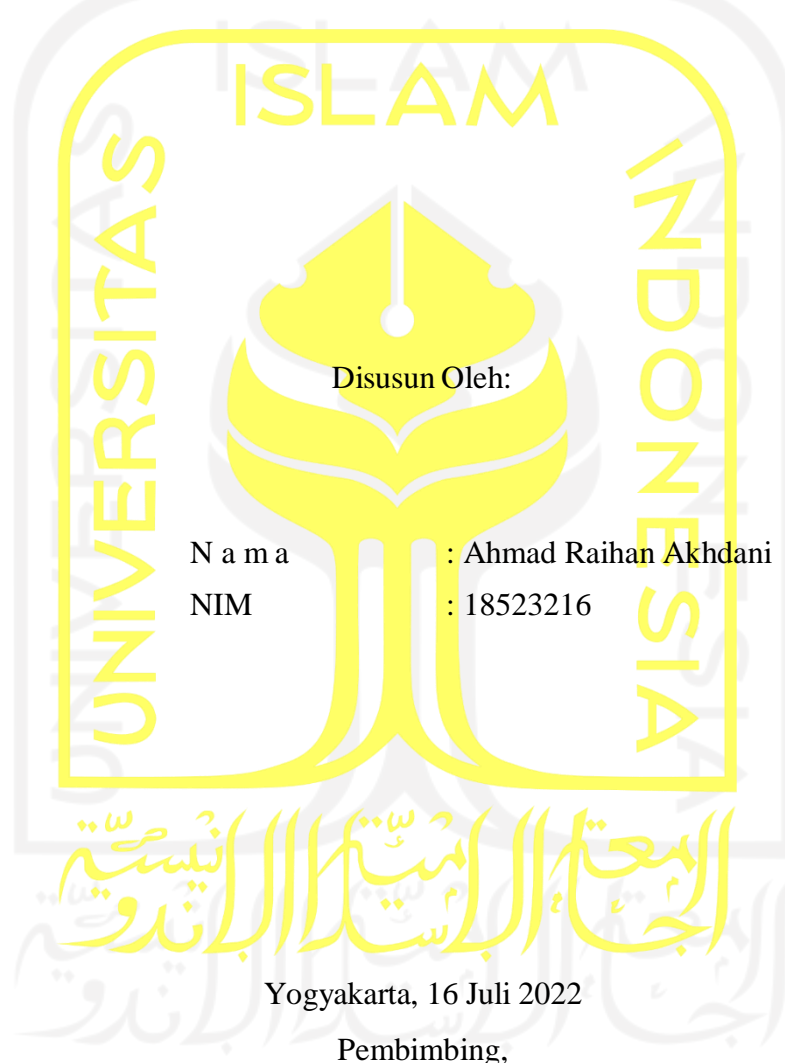
HALAMAN PENGESAHAN DOSEN PEMBIMBING

IMPLEMENTASI REACT.JS PADA PENGEMBANGAN

FRONTEND SISTEM INFORMASI MANAJEMEN

KADER PARTAI

TUGAS AKHIR JALUR MAGANG



(Erika Ramadhani, S.T., M.Eng.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**IMPLEMENTASI REACT.JS PADA PENGEMBANGAN
FRONTEND SISTEM INFORMASI MANAJEMEN
KADER PARTAI**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 28 Juli 2022

Tim Penguji

Erika Ramadhani, S.T., M.Eng.

Anggota 1

Dr. Ahmad Luthfi, S.Kom., M.Kom.

Anggota 2

Chandra Kusuma Dewa, S.Kom., M.Cs.,
Phd.


 Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Ahmad Raihan Akhdani

NIM : 18523216

Tugas akhir dengan judul:

IMPLEMENTASI REACT.JS PADA PENGEMBANGAN FRONTEND SISTEM INFORMASI MANAJEMEN KADER PARTAI

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 16 Juli 2022



(Ahmad Raihan Akhdani)

HALAMAN PERSEMBAHAN

Alhamdulillah puji dan syukur penulis panjatkan kepada kehadiran Allah SWT yang telah memberikan rahmat dan hidayahnya sehingga penulis dapat melewati masa perkuliahan hingga sampailah pada kesempatan untuk menyelesaikan tugas akhir ini sebagai syarat untuk mendapatkan gelar sarjana.

Tugas akhir ini penulis persembahkan untuk kedua orang tua penulis yang selalu memberikan dukungan, doa, dan motivasinya hingga saat ini. Penulis juga mengucapkan terimakasih banyak kepada semua pihak yang membantu dalam penulisan tugas akhir ini.



HALAMAN MOTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

Q.S. Al-Baqarah ayat 286

“Perbaikilah sholatmu, maka Allah akan memperbaiki hidupmu”



KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, puji syukur selalu penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah dan inayah-Nya sehingga penulis dapat melaksanakan proses magang dan menyelesaikan laporan ini dengan baik tanpa halangan apapun. Laporan ini dibuat sebagai salah satu syarat untuk memenuhi persyaratan tugas jalur magang di program studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Kelancaran kegiatan magang ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terimakasih kepada semua pihak yang telah membantu, diantaranya:

1. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Ibu Erika Ramadhani, S.T., M.Eng., selaku Dosen Pembimbing yang telah memberikan saran, nasehat dan bimbingan dalam penyusunan laporan ini.
3. Pimpinan dan seluruh staf Geekgarden Software House yang telah menerima dan membimbing penulis dengan baik dalam proses pelaksanaan magang.
4. Kedua orang tua dan teman-teman penulis yang telah mendoakan, memotivasi dan mendukung penulis.

Dalam penulisan tugas akhir ini penulis menyadari bahwa masih banyak kekurangan dan kesalahan, maka dari itu penulis membutuhkan saran dan kritik guna menyempurnakan tugas akhir ini. Akhir kata, penulis berharap semoga tugas akhir ini dapat bermanfaat bagi penulis serta bagi para pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 16 Juli 2022



(Ahmad Raihan Akhdani)

SARI

GeekGarden Software House merupakan salah satu perusahaan teknologi informasi yang terletak di Yogyakarta dengan fokus pada layanan jasa pengembangan perangkat lunak. Layanan yang GeekGarden sediakan antara lain IT Project Management, IT Consultant, Analysis System, Design, dan Code Construction. Dengan layanan yang ditawarkan tersebut, partai PDIP mempercayakan pada GeekGarden dalam pengembangan sebuah aplikasi sistem informasi manajemen berbasis website yang disebut SIMKADER PDIP. Aplikasi tersebut digunakan untuk membantu dalam mengawasi, mengatur, dan mengelola informasi para kader, sehingga nantinya dapat dipergunakan untuk menjalankan fungsi perancangan dan perencanaan. Dalam sebuah aplikasi, tentu saja tampilan antarmuka menjadi suatu hal yang penting. Oleh karena itu, dipilihlah React.Js sebagai library yang akan digunakan pada sisi frontend untuk membantu dalam pembangunan tampilan antarmuka. Alasannya, karena React.Js menawarkan berbagai fitur dan keunggulan yang akan memudahkan pengembang dalam membangun tampilan antarmuka penggunaannya. Pengembangan dilakukan menggunakan metode Agile dengan tahapannya dimulai dari analisis, desain, implementasi, pengujian, hingga deployment. Setelah melalui beberapa tahapan tersebut dan berdasarkan hasil pengujian, dihasilkan sebuah sistem informasi manajemen yang digunakan dalam mengelola dan memajemen kader partai PDIP.

Kata kunci: GeekGarden, aplikasi SIMKADER PDIP, React.js.

GLOSARIUM

<i>Library</i>	sekumpulan kode atau script siap pakai yang digunakan untuk tujuan tertentu. Biasanya digunakan pengembang untuk menyederhanakan aplikasi.
<i>Front-end</i>	bagian dari website yang dapat dilihat dan berinteraksi langsung dengan pengguna.
<i>Open Source</i>	kode perangkat lunak yang dirancang untuk dapat diakses dan dimodifikasi oleh siapapun.
<i>React.Js</i>	library javascript yang digunakan untuk membangun tampilan antarmuka pengguna.
<i>Response</i>	tanggapan, jawaban, atau umpan balik dari suatu komunikasi.
<i>Request</i>	mengirimkan suatu permintaan terhadap server.
<i>Mandatory</i>	wajib untuk diisi.
<i>Bug</i>	kerusakan atau kecacatan yang membuat aplikasi tidak berjalan dengan seharusnya.
<i>Push</i>	perintah yang digunakan untuk mengirimkan perubahan kode dari local repository ke remote repository.
<i>Branch</i>	cabang selain main branch dari main repository yang memiliki history yang berbeda dan dapat digabungkan menjadi satu dengan main branch.
<i>Merge</i>	merupakan suatu command dalam git yang berfungsi untuk menggabungkan branch yang bercabang menjadi satu.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR.....	vii
SARI.....	viii
GLOSARIUM.....	ix
DAFTAR ISI	x
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Ruang Lingkup	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA	4
2.1 Sistem Informasi Manajemen	4
2.2 React.Js.....	4
2.3 JSX	5
2.4 Postman	5
2.5 Gitlab.....	5
2.6 Agile Development Method.....	6
2.7 Acceptance Test Driven Development	7
2.8 Tinjauan Pustaka.....	7
BAB III PELAKSANAAN MAGANG	12
3.1 Inisiasi Proyek.....	12
3.2 Perencanaan Proyek	12
3.3 Pelaksanaan Proyek.....	14
3.3.1 Analisis.....	15
3.3.2 Desain	15
3.3.3 Implementasi	15
3.3.4 Pengujian.....	33
3.3.5 Deployment	46
3.4 Pemantauan Proyek.....	47
3.5 Penutupan Proyek	48
BAB IV REFLEKSI PELAKSANAAN MAGANG.....	50
4.1 Relevansi Akademik	50
4.2 Pembelajaran Magang	53
BAB V PENUTUP	57
5.1 Kesimpulan.....	57
5.2 Saran.....	58
DAFTAR PUSTAKA	59
LAMPIRAN	62

DAFTAR TABEL

Tabel 2.1 Tabel perbandingan penelitian terkait	9
Tabel 3.1 Tabel Aktivitas Pengerjaan Aplikasi.....	13
Tabel 3.2 Pengujian halaman login	34
Tabel 3.3 Pengujian kegagalan halaman login.....	34
Tabel 3.4 Pengujian halaman kader.....	35
Tabel 3.5 Pengujian tambah kader	36
Tabel 3.6 Pengujian kegagalan tambah kader.....	36
Tabel 3.7 Pengujian ubah data kader.....	37
Tabel 3.8 Pengujian kegagalan ubah data kader	37
Tabel 3.9 Pengujian lihat detail kader	37
Tabel 3.10 Pengujian hapus data kader	38
Tabel 3.11 Pengujian kegagalan hapus data kader.....	38
Tabel 3.12 Pengujian halaman biografi	38
Tabel 3.13 Pengujian ubah biografi	39
Tabel 3.14 Pengujian kegagalan ubah biografi.....	39
Tabel 3.15 Pengujian halaman jabatan	39
Tabel 3.16 Pengujian tambah jabatan.....	40
Tabel 3.17 Pengujian kegagalan tambah jabatan	41
Tabel 3.18 Pengujian ubah jabatan	41
Tabel 3.19 Pengujian kegagalan ubah jabatan	41
Tabel 3.20 Pengujian hapus jabatan	42
Tabel 3.21 Pengujian kegagalan hapus jabatan.....	42
Tabel 3.22 Pengujian tambah anak jabatan	42
Tabel 3.23 Pengujian kegagalan tambah anak jabatan	43
Tabel 3.24 Pengujian halaman pengguna	43
Tabel 3.25 Pengujian tambah pengguna.....	44
Tabel 3.26 Pengujian kegagalan tambah pengguna	44
Tabel 3.27 Pengujian ubah pengguna.....	45
Tabel 3.28 Pengujian kegagalan ubah pengguna	45
Tabel 3.29 Pengujian hapus pengguna	45
Tabel 3.30 Pengujian kegagalan hapus pengguna.....	46
Tabel 3.31 Pengujian penentuan hak akses	46

Tabel 4.1 Perbandingan teori dengan pelaksanaan lapangan..... 52



DAFTAR GAMBAR

Gambar 2.1 Tahapan metode agile pada pengembangan SIMKADER PDIP	6
Gambar 3.1 Tampilan React.Js Proyek SIMKADER PDIP	14
Gambar 3.2 Tampilan Postman Proyek SIMKADER PDIP.....	14
Gambar 3.3 Tampilan Figma Proyek SIMKADER PDIP.....	15
Gambar 3.4 Instalasi <i>Next.Js</i>	16
Gambar 3.5 Instalasi <i>library Ant Design</i>	16
Gambar 3.6 Proses <i>import component skeleton</i> dari <i>library Ant Design</i>	16
Gambar 3.7 <i>Setup</i> tata letak halaman	17
Gambar 3.8 Menggunakan <i>component sider</i> dari <i>library Ant Design</i> untuk membuat <i>sidebar</i>	18
Gambar 3.9 Mengatur setiap menu yang ada pada sidebar	18
Gambar 3.10 Membuat <i>Header</i> dengan memanfaatkan <i>component layout header</i> dari <i>library Ant Design</i>	20
Gambar 3.11 Instalasi <i>library Axios</i>	20
Gambar 3.12 <i>Import library Axios</i>	20
Gambar 3.13 Proses <i>request</i> dengan menggunakan <i>library Axios</i>	21
Gambar 3.14 Instalasi <i>library MobX</i>	21
Gambar 3.15 Penerapan <i>MobX</i> pada management state Data Kader	22
Gambar 3.16 Menampilkan data pada tabel	24
Gambar 3.17 Tampilan Menu Data Kader	24
Gambar 3.18 Tampilan Tambah Data	25
Gambar 3.19 Tampilan Detail Data	25
Gambar 3.20 Tampilan Edit Data	26
Gambar 3.21 Tampilan Hapus Data.....	26
Gambar 3.22 Tampilan Halaman Login	27
Gambar 3.23 Tampilan Halaman Biografi	28
Gambar 3.24 Tampilan Halaman Jabatan.....	28
Gambar 3.25 Tampilan Tambah Jabatan	29
Gambar 3.26 Tampilan Edit Jabatan	29
Gambar 3.27 Tampilan Hapus Jabatan.....	30
Gambar 3.28 Tampilan Tambah <i>Child</i> Jabatan	31
Gambar 3.29 Tampilan Halaman Pengguna	31

Gambar 3.30 Tampilan Tambah Pengguna	32
Gambar 3.31 Tampilan Edit Pengguna	32
Gambar 3.32 Tampilan Hapus Pengguna.....	33
Gambar 3.33 Tampilan Penentuan Hak Akses	33
Gambar 3.34 Tampilan Trello Proyek SIMKADER PDIP.....	48
Gambar 3.35 Tampilan GitLab Proyek SIMKADER PDIP.....	48



BAB I

PENDAHULUAN

1.1 Latar Belakang

Geekgarden Software House adalah sebuah perusahaan *software house* yang terletak di Yogyakarta. Geekgarden fokus pada jasa pengembangan perangkat lunak dengan memberikan layanan pada perencanaan, pengembangan, dan pengelolaan sistem baik berskala kecil, menengah maupun besar. Layanan yang Geekgarden sediakan yaitu jasa *IT Project Management, IT Consultant, Analysis System, Design, dan Code Construction*. Dengan layanan yang ditawarkan tersebut, banyak klien yang mempercayakan pengerjaan proyek IT pada Geekgarden, salah satunya yaitu partai PDIP dengan proyek yang bernama SIMKADERPDIP. Seperti namanya, SIMKADER PDIP merupakan sebuah sistem informasi manajemen yang akan mengelola data kader dari partai PDIP.

Sistem tersebut digunakan agar mempermudah partai PDIP dalam melakukan pengelolaan dan perencanaan untuk mencapai tujuan. Seperti yang diketahui bahwa perkembangan teknologi dari tahun ke tahun sangat pesat dan cepat, hal ini membuat kebutuhan akan informasi menjadi semakin penting yang harus dimiliki setiap partai dalam melakukan kegiatan operasionalnya agar dapat bersaing dengan partai-partai lainnya. Setiap partai pasti memiliki informasi yang berbeda dan beraneka ragam. Informasi-informasi tersebut yang nantinya akan menjadi komponen utama sebagai dasar dalam menetapkan suatu keputusan. Untuk kasus ini, informasi yang ingin dikelola adalah data para kader yang memiliki berbagai macam informasi, apalagi data akan terus bertambah setiap kali pemilihan akan dilaksanakan. Dengan banyaknya informasi tersebut, tentu saja akan membuat keputusan menjadi lebih tepat.

Maka dibutuhkan sebuah sistem informasi manajemen yang dapat membantu dalam mengelola data para kader menjadi lebih mudah dan cepat. Dengan begitu, berbagai manfaat seperti perancangan, perencanaan, pengawasan, dan pengarahan menjadi lebih efektif dan efisien.

Sistem informasi manajemen yang dikembangkan akan berbasis website. Pemilihan website sebagai basis aplikasi dikarenakan website memiliki berbagai keunggulan. Dengan berbasis website, akan membantu dalam kecepatan dan kualitas penyampaian informasi. Selain itu, informasi data dapat diakses setiap saat dengan tempat yang tidak ditentukan asalkan

terkoneksi dengan jaringan internet. (Setiyawan, Purnama, dan Sukadi, 2013). Sedangkan metode pengembangan sistem menggunakan metode agile dengan beberapa tahap yaitu analisis, desain, implementasi, pengujian, dan deployment.

Pada pembangunan aplikasi, tampilan antarmuka menjadi suatu hal yang penting untuk diperhatikan, karena bukan hanya tentang tampilan aplikasi saja, tetapi juga berkaitan dengan seberapa mudah menggunakan aplikasi tersebut. Karena itulah pada aplikasi SIMKADER PDIP, pembuatan tampilan antarmuka dibangun dengan menggunakan framework React.Js. Alasannya adalah karena React.Js merupakan framework yang populer dan banyak digunakan hingga saat ini. React.js sebenarnya merupakan JavaScript *library* yang menawarkan kemudahan bagi pengembang untuk membuat tampilan antarmuka yang responsif dan interaktif. Salah satunya yaitu component based, artinya pengembang dapat memecah seluruh bagian aplikasi menjadi komponen-komponen kecil. Tujuannya agar dapat digunakan kembali berulang kali tanpa mengulang kode dari awal serta memudahkan bagi pengembang untuk memperbaiki kesalahan-kesalahan.

1.2 Ruang Lingkup

Penulis melaksanakan magang sebagai *Front-end Developer* di GeekGarden Software House, yaitu sebuah perusahaan teknologi informasi yang terletak di Yogyakarta. Magang dilaksanakan selama 6 bulan yang dimulai dari pertengahan bulan September 2021 hingga pertengahan bulan Maret 2022. Selama magang berlangsung, penulis telah berkontribusi pada pengerjaan 2 proyek pengembangan aplikasi, salah satunya yaitu pengembangan aplikasi SIMKADER PDIP dari partai PDIP. Adapun aktivitas yang penulis lakukan pada proyek ini adalah sebagai berikut:

- a. Mempelajari teknologi yang digunakan
- b. Mengembangkan halaman login
- c. Mengembangkan halaman menu Data Kader
- d. Mengembangkan halaman menu Biografi
- e. Mengembangkan halaman menu Jabatan
- f. Mengembangkan halaman menu Pengguna

1.3 Tujuan

Tujuan dari proyek pengembangan aplikasi PDIP adalah untuk menghasilkan aplikasi frontend SIMKADER PDIP yang digunakan untuk mengatur dan mengelola data kader partai PDIP.

1.4 Manfaat

Manfaat dari proyek pengembangan aplikasi PDIP adalah untuk membantu partai PDIP dalam melakukan pengelolaan data partainya termasuk para kadernya.

1.5 Sistematika Penulisan

Penulisan tugas akhir ini terdiri dari lima bab dengan susunan sistematika penulisan sebagai berikut:

a. **BAB I PENDAHULUAN**

Bab ini menguraikan tentang latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan.

b. **BAB II DASAR TEORI**

Bab ini menguraikan tentang teori-teori yang memiliki keterkaitan dengan topik laporan untuk digunakan penulis sebagai landasan teori dalam penyusunan laporan.

c. **BAB III PELAKSANAAN MAGANG**

Bab ini menguraikan tentang kegiatan dan tugas apa saja yang dikerjakan selama proses pelaksanaan magang.

d. **BAB IV REFLEKSI PELAKSANAAN MAGANG**

Bab ini menguraikan tentang hasil refleksi yang penulis dapat selama proses pelaksanaan magang.

e. **BAB V KESIMPULAN DAN SARAN**

Bab ini menguraikan tentang kesimpulan dari laporan yang telah ditulis. Bagian ini juga berisi tentang saran penulis.

BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) adalah sebuah sistem yang digunakan dalam pengambilan keputusan dengan memberikan dukungan informasi dan pengolahan. Secara umum, sistem informasi manajemen dapat dikatakan sebagai sebuah sistem yang terdapat pada organisasi dengan mengintegrasikan manusia dan sistem dalam menyediakan informasi yang berfungsi untuk mendukung operasi manajemen dan penentuan alternatif tindakan. Dalam operasinya, sistem informasi manajemen menggunakan perangkat keras, perangkat lunak, prosedur, model manajemen, dan keputusan (Gaol, 2008).

Menurut Gordon B. Davis (1984), sistem informasi manajemen adalah sebuah sistem antar manusia dengan sistem yang terintegrasi dalam memberikan informasi untuk mendukung operasi, manajemen, dan pengambilan keputusan dalam organisasi.

2.2 React.Js

React, biasa dikenal sebagai React atau React.Js, merupakan sebuah *library* JavaScript dengan sifat *open source* yang digunakan untuk mempermudah dalam membangun tampilan antarmuka pengguna (UI) menjadi lebih responsive dan interaktif. React biasa digunakan untuk mengembangkan tampilan pada *Single Page Application* (SPA) dan *mobile application* (Tung Khat, 2018).

React diciptakan oleh seorang insinyur perangkat lunak dari Facebook yang bernama Jordan Walke pada tahun 2011 dan secara resmi digunakan oleh Facebook pada bagian *news feed* mereka. Selain itu, Instagram juga memilih React untuk digunakan pada sistem mereka. Sejak saat itu, React telah berkembang menjadi salah satu *library* JavaScript yang banyak digunakan hingga saat ini (Anup Satyal, 2020).

React hanya merepresentasikan bagian *view* dari konsep MVC (*Model-View-Controller*) (Archana Bhalla, 2020). Artinya, React hanya digunakan untuk membuat tampilan antarmukanya saja tanpa memiliki kemampuan untuk berhubungan dengan basis data.

2.3 JSX

React.Js memiliki aturan tersendiri dalam menuliskan kode pada setiap komponennya, yaitu dengan menggunakan JSX yang memungkinkan developer untuk menuliskan sintaks HTML di dalam JavaScript. Dengan mengkombinasikan dua hal tersebut antara HTML dan JavaScript, penulisan kode menjadi lebih sederhana dan ringkas dalam pendeskripsian tampilan antarmuka.

JSX sendiri merupakan ekstensi sintaks untuk ECMAScript tanpa ada definisi semantik tertentu. Pada dasarnya React.Js bisa digunakan tanpa JSX, namun tanpa JSX penulisan kode menjadi lebih panjang dan sulit. Sedangkan dengan menggunakan JSX dapat memudahkan pengembang dalam penulisan kode dan membantu saat melakukan *error debugging* (Tung Khuat, 2018).

2.4 Postman

Postman merupakan sebuah aplikasi yang membantu para *developer* untuk melakukan pengujian terhadap endpoint atau API (Application Programming Interface) yang telah dibuat dengan berbagai fungsi di dalamnya seperti *GET*, *POST*, *PUT*, dan *DELETE* (Bahtyar, 2019). Aplikasi ini memiliki tampilan yang lengkap untuk memudahkan dan mempercepat *developer* dalam mengembangkan API, mulai dari merancang API, membangun API, *testing* API, dan mendokumentasikan API (Rachmat Fajrin, 2017). Maka dari itu, dengan menggunakan Postman, *developer* dapat dengan mudah mensimulasikan API dan memastikan bahwa API tersebut dapat berjalan dengan baik.

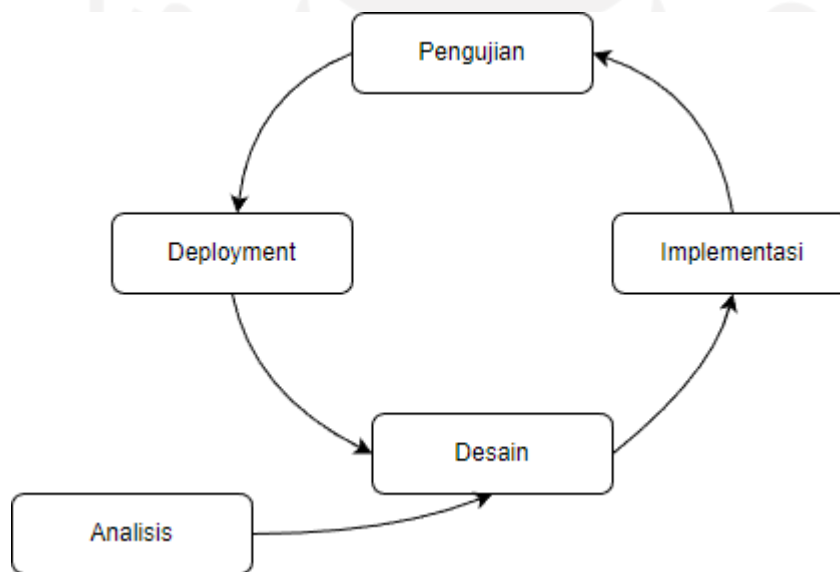
2.5 Gitlab

Gitlab adalah suatu sistem *cloud* berbasis web yang menyimpan semua perubahan *code* para *developer*. Dengan menggunakan Gitlab, antar *developer* akan lebih mudah dalam melakukan kolaborasi (misalnya *frontend* dan *backend*) tanpa takut akan terjadi bentrok pada perubahan *code* yang sudah mereka buat. Gitlab sendiri merupakan salah satu dari VCS (*Version Control Systems*) yang populer digunakan saat ini. Sedangkan VCS adalah sebuah sistem yang digunakan untuk mengontrol dan merekam setiap perubahan file yang dilakukan setiap *developer* serta dapat memanggil kembali pada versi yang diinginkan. Dengan menggunakan VCS, *developer* dapat dengan mudah melacak perubahan *code* dan membandingkannya dengan versi sebelumnya (Adam O'Grady, 2018). Dalam proses pengembangan aplikasi, VCS merupakan suatu hal yang penting, karena dengan menggunakan

sistem tersebut, dapat mengurangi terjadinya kesalahan pada kinerja semua anggota tim developer.

2.6 Agile Development Method

Agile development method merupakan salah satu metode pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip pengembangan sistem jangka pendek yang membutuhkan adaptasi cepat dalam mengatasi segala bentuk perubahan (Haryana, K. S., 2019). Artinya, bahwa agile development method memiliki keunggulan dalam hal fleksibilitas dan cepat tanggap terhadap berbagai perubahan yang terjadi selama proses pengembangan. Adapun tahapan yang dilakukan pada pengembangan sistem informasi manajemen kader yaitu analisis, desain, implementasi, pengujian, dan deployment. Tahapan tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tahapan metode agile pada pengembangan SIMKADER PDIP

1. Analisis

Analisis merupakan tahap pertama dalam metode waterfall. Tahap ini memerlukan komunikasi dengan klien agar dapat menganalisis kebutuhan apa saja yang diinginkan klien terhadap sistem yang akan dikembangkan.

2. Desain

Tahap desain dilakukan untuk membuat desain aplikasi berdasarkan analisis yang telah dilakukan pada tahap sebelumnya. Tahap ini bertujuan untuk memberikan gambaran

lengkap dan jelas mengenai tampilan antarmuka aplikasi yang akan dikembangkan sebelum dieksekusi oleh programmer.

3. Implementasi

Desain yang telah dibuat pada tahap sebelumnya, akan diterjemahkan dalam bentuk code oleh programmer atau tim developer dengan menggunakan tools dan bahasa pemrograman yang telah direncanakan pada tahap awal. Tahap implementasi pada pengembangan sistem informasi manajemen kader partai PDIP dibagi menjadi 2 tim, yaitu front-end developer yang bertanggungjawab dalam membangun user interface dan back-end developer yang bertugas untuk mengelola data di database.

4. Pengujian

Setelah implementasi telah selesai sepenuhnya, tahap selanjutnya yaitu tahap pengujian. Tahap ini bertujuan untuk memastikan bahwa fungsionalitas aplikasi yang dikembangkan dapat berjalan dengan baik dan sesuai dengan spesifikasi yang telah didefinisikan sebelumnya sehingga akan mencegah adanya error atau bug sebelum berlanjut ke tahap deployment.

5. Deployment

Setelah seluruh tahap sebelumnya telah selesai mulai dari analisis hingga pengujian, maka aplikasi sudah siap untuk digunakan oleh customer secara langsung dengan melakukan deployment ke server.

2.7 Acceptance Test Driven Development

Acceptance Test Driven Development (ATDD) merupakan salah satu metode pengujian dalam metode Agile Development Method. Pengujian ini melibatkan proses acceptance testing sebelum proses pembangunan kode, sehingga akan meningkatkan waktu pengembangan serta kesesuaian antara kebutuhan dengan aplikasi yang dibangun (Pugh, 2010). Pengujian ini biasanya dilakukan secara manual dengan menggunakan user-level blackbox testing sebagai pengganti unit test (Sauvé, Abath Neto, & Cirne, 2006).

2.8 Tinjauan Pustaka

Penelitian ini dapat diselesaikan tidak lepas dari penelitian terdahulu yang terkait. Adapun penelitian terdahulu akan berhubungan dengan topik penelitian yaitu tentang implementasi react.js pada pengembangan sistem informasi manajemen kader. Nantinya, penelitian terdahulu akan menjadi sumber referensi dan sebagai bahan pembanding.

Penelitian yang dilakukan oleh Faisal F. N, dkk, dengan judul “Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan *ReactJS* dan *React Native* Menggunakan *Prototype* (Studi Kasus: Toko Uda Fajri)” (2020) di latar belakang karena pengelolaan barang dagang di toko tersebut masih manual dan belum memanfaatkan teknologi. Karena itu, hasil dari penelitian tersebut adalah sistem informasi pengelolaan barang untuk mempermudah pedagang dalam mengelola dagangannya tanpa harus menggunakan kertas untuk pencatatannya. Peneliti menggunakan metode *prototype* dalam pengembangannya dan sistemnya dikembangkan dengan arsitektur MVVM (*Model View ViewModel*) menggunakan *react.js* untuk sistem berbasis website dan *react native* untuk aplikasi yang dijalankan pada *device android*. Tahapan pengujian dilakukan dengan 3 metode, yaitu *whitebox testing*, *blackbox testing*, dan *user acceptance testing*.

Muhammad Fakhri A. H. (2021) melakukan penelitian *Pengembangan Front End Sistem Informasi Manajemen Data Patroli Pencegahan Karhutla Di Wilayah Sumatera*. Penelitian tersebut dilakukan untuk mengembangkan modul aplikasi berbasis web yang mampu untuk memonitoring data patroli dan memanajemen data pengguna patroli karhutla. Metode yang digunakan pada penelitian tersebut adalah metode prototyping dengan lima tahapan yaitu *communication*, *quick plan*, *modelling quick design*, *construction of prototype*, dan *deployment, delivery, and feedback*. Menggunakan *reactjs* pada pengembangan modul aplikasinya karena memiliki kemampuan untuk memecah tampilan yang kompleks menjadi komponen-komponen kecil.

Penelitian serupa dilakukan oleh Alvin Afandi tentang pengembangan modul *frontend* pada sistem informasi manajemen unit kandang hewan percobaan di pusat studi Biofarmaka Tropika pada tahun 2019. Permasalahan dalam penelitian tersebut disebabkan beberapa hal, diantaranya sulitnya memonitor penelitian akibat banyaknya penelitian yang dilakukan oleh pihak UKHP (Unit Kandang Hewan Percobaan), proses administrasi yang belum maksimal karena pembayaran masih dilakukan dengan cara pencatatan melalui *spreadsheet*, dan pendataan dokumen inventaris bahan dan alat masih harus dihitung secara manual. Berdasarkan masalah tersebut, akan dihasilkan sebuah sistem informasi manajemen UKHP (SIM-UKHP) yang akan memudahkan dalam pemantauan penelitian, pembayaran, serta pembuatan laporan inventaris hewan. Pengembangan modul *frontend* menggunakan framework Laravel dengan metode yang digunakan ialah *prototyping*. Tahapan yang diterapkan pada metode prototyping yaitu komunikasi, perencanaan cepat, pemodelan rancangan cepat, pembuatan prototipe, serta penyebaran, pengujian, dan umpan balik.

Penelitian lain dilakukan oleh I Wayan Aditya A.P, dkk. (2019) tentang pengembangan frontend untuk sistem informasi manajemen absensi pegawai di perusahaan Avatar Solutions. Penelitian tersebut menghasilkan sistem yang mempermudah pegawai dalam melakukan absensi guna mengatur jam kerja dan waktu istirahat per harinya. Sistem yang dikembangkan menggunakan framework AngularJs sebagai tools untuk membangun antarmukanya. Sementara metode yang diterapkan adalah metode waterfall dengan beberapa tahapan yaitu, analisis kebutuhan sistem, perancangan sistem sesuai definisi kebutuhan sistem, implementasi rancangan sistem, dan pengujian sistem.

Afif N. S, dkk, (2018) dalam penelitiannya yang berjudul Pengembangan Sistem Informasi Manajemen Tesis (Studi Kasus: Program Studi Magister Administrasi Bisnis Fakultas Ilmu Administrasi UB) menyebutkan bahwa terdapat masalah yang timbul pada alur pengerjaan tesis di Program Studi Magister Administrasi Bisnis, Fakultas Administrasi Bisnis, Universitas Brawijaya, yaitu masih manualnya pengecekan dan penandatanganan formulir kertas di setiap tahapan pengerjaan tesis, sehingga akan memperlambat alur pengerjaan tesis serta memiliki beberapa resiko yaitu hilang, rusak, atau diambil oleh pihak yang tidak berwenang. Berdasarkan masalah tersebut, solusi yang peneliti tawarkan adalah mengembangkan sebuah sistem informasi manajemen tesis untuk mengurangi penggunaan formulir isian dalam setiap tahapan pengerjaan tesis dan menggantinya menjadi formulir digital. Metode penelitian yang digunakan berupa pengumpulan data, pemodelan proses bisnis, analisis kebutuhan, perancangan, implementasi, dan pengujian. Sistem dikembangkan menjadi 2 bagian yaitu *backend* dan *frontend*. *Backend* terdiri dari *web service* dengan menggunakan Bahasa pemrograman PHP, sedangkan *frontend* menerapkan konsep *single page application* dengan bantuan *framework* Vue.Js. Pengujian yang digunakan adalah pengujian *black-box* dan pengujian *compatibility*.

Adapun perbandingan antara penelitian terdahulu dengan penelitian ini apabila disajikan dalam bentuk tabel dapat dilihat pada **Error! Reference source not found..**

Tabel 2.1 Tabel perbandingan penelitian terkait

No	Peneliti (Tahun)	Judul	Metode Pengembangan	Teknologi	Pengujian
1	Faisal Fajar Nursaid, Adam Hendra Brata, Agi Putra	Pengembangan Sistem Informasi Pengelolaan Persediaan Barang	Metode <i>SDLC</i> (Software Development Life Cycle) <i>Prototype</i> .	<i>Framework ReactJs</i> dan <i>React Native</i> , bahasa pemrograman <i>JavaScript</i> ,	<i>Whitebox testing</i> , <i>blackbox testing</i> , dan <i>user</i>

	Kharisma (2020).	Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus: Toko Uda Fajri).		penyimpanan basis data menggunakan <i>Postgresql</i> .	<i>acceptance testing</i> .
2	Muhammad Fakhri Alauddin Hidayat (2021)	Pengembangan <i>Front End</i> Sistem Informasi Manajemen Data Patroli Pencegahan Karhutla Di Wilayah Sumatera	Metode pengembangan <i>prototyping</i>	<i>Framework ReactJs</i> , bahasa pemrograman <i>TypeScript</i>	Metode <i>blackbox testing</i>
3	Alvin Afandi (2019)	Pengembangan Modul Frontend Pada Sistem Informasi Manajemen Unit Kandang Hewan Percobaan Di Pusat Studi Biofarmaka Tropika	Metode pengembangan <i>prototyping</i>	<i>Framework Laravel</i>	Metode <i>user acceptance testing</i>
4	Afif Nandya Saputra, Satrio Agung Wicaksono, Fajar Pradana (2018)	Pengembangan Sistem Informasi Manajemen Tesis (Studi Kasus: Program Studi Magister Administrasi Bisnis Fakultas Ilmu Administrasi UB)	Metode pengembangan berupa berupa pengumpulan data, pemodelan proses bisnis, analisis kebutuhan, perancangan, implementasi, dan pengujian	<i>Framework VueJs</i>	Metode <i>blackbox testing</i> dan <i>compatibility testing</i>
5	I Wayan Aditya Anggara Putra, I Gusti	Pengembangan <i>Front End</i> Sistem Informasi	Metode pengembangan <i>waterfall</i>	<i>Framework AngularJs</i>	Tidak disebutkan

	Agung Gede Arya Kadyanan (2019)	Manajemen Absensi Pegawai Avatar Solutions			
--	--	--	--	--	--



BAB III

PELAKSANAAN MAGANG

Dalam manajemen proyek pada pengembangan sistem informasi manajemen kader partai PDIP terdapat beberapa tahapan yang dilakukan, yaitu inisiasi proyek, perencanaan proyek, pelaksanaan proyek, pemantauan proyek, dan penutupan proyek.

3.1 Inisiasi Proyek

Inisiasi proyek merupakan tahap awal yang dilakukan dalam proses pengembangan sistem informasi manajemen kader PDIP. Pada tahap inisiasi, project manager akan melakukan diskusi bersama klien untuk mengidentifikasi dan menganalisis permasalahan serta kebutuhan apa saja yang diinginkan klien terhadap sistem yang akan dikembangkan. Berdasarkan diskusi tersebut, dapat diketahui ruang lingkup, anggaran, waktu, dan tujuan proyek.

3.2 Perencanaan Proyek

Berdasarkan tahap inisiasi dengan melakukan koordinasi bersama klien, berlanjut ke tahap perencanaan proyek. Pada tahap ini, akan menetapkan beberapa hal yang diperlukan untuk terselesainya proyek dari awal sampai akhir. Beberapa hal tersebut diantaranya milestone, timeline pengerjaan proyek, spesifikasi sistem yang dibutuhkan, dan posisi atau peran apa saja yang dibutuhkan untuk menyelesaikan proyek tersebut. Adapun posisi yang diperlukan untuk tercapainya tujuan yang telah direncanakan dalam memenuhi kebutuhan pengembangan proyek sesuai dengan perannya masing-masing adalah sebagai berikut:

- *Project Manager*, memiliki peran untuk menentukan arah pengembangan agar proyek dapat terselesaikan sesuai dengan rencana. Peran yang biasa dilakukan oleh *project manager* yaitu, melakukan koordinasi dengan klien, merancang *timeline* pengerjaan proyek, memberikan tugas kepada setiap anggota tim sesuai dengan perannya masing-masing, dan memantau pengerjaan proyek.
- *UI/UX*, berperan dalam membuat desain tampilan antarmuka berdasarkan flow atau dokumen kebutuhan serta menentukan bagaimana desain tampilan antarmuka tersebut mudah untuk dipahami dan digunakan oleh pengguna.
- *Front-End Developer*, berperan untuk menerjemahkan desain tampilan antarmuka yang sebelumnya telah dibuat oleh tim UI/UX menjadi sejumlah baris kode.

Sejumlah kode tersebut pada akhirnya akan membentuk sebuah aplikasi yang dapat berinteraksi dengan pengguna.

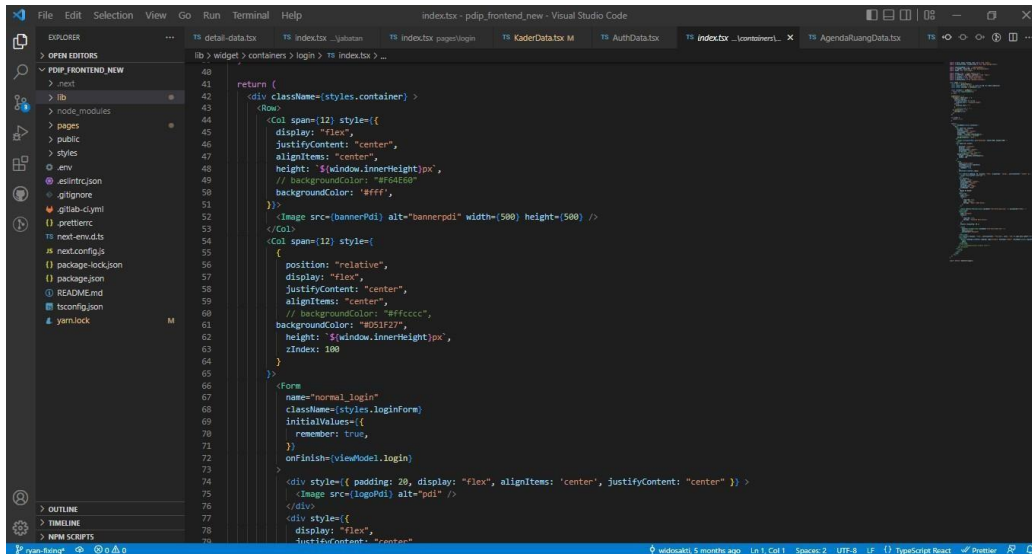
- *Back-End Developer*, berperan dalam menangani seluruh pengelolaan dan penyimpanan data dalam hubungannya dengan *front-end developer* agar aplikasi yang dibuat bersifat dinamis.
- *QA/QC*, berperan dalam melakukan pengujian atau pengecekan terhadap sistem yang telah dibuat oleh *front-end developer* maupun *backend developer*, membuat test case, serta mencatat bug yang ditemukan dan memberikannya kepada developer untuk diperbaiki agar sistem yang dihasilkan berkualitas baik sesuai dengan kebutuhan.
- *DevOps*, berperan untuk mempermudah dalam mengkomunikasikan seluruh anggota tim developer dari *front-end*, *back-end*, hingga *QA/QC*. Tugas yang biasanya dikerjakan oleh *DevOps* dalam pengembangan aplikasi SIMKADER PDIP yaitu membuat repository sebagai pusat proyek, mengawasi kode program dengan merancang script pengujian otomatis, membuat *staging area*, dan melakukan deployment.

Selain itu, pengembangan sistem melalui beberapa tahap, yaitu analisis, desain, implementasi, pengujian, dan deploy. Adapun pembagian waktu pengerjaan pengembangan aplikasi SIMKADER PDIP dapat dilihat pada **Error! Reference source not found.**

Tabel 3.1 Tabel Aktivitas Pengerjaan Aplikasi

No	Aktivitas	Waktu
1	Halaman Login	1 Minggu
2	Halaman Data kader	2 Minggu
3	Halaman Biografi	1 Minggu
5	Halaman Jabatan	2 Minggu
6	Halaman Pengguna	2 Minggu

Terdapat sejumlah teknologi yang akan digunakan bagi *front-end developer* untuk membuat tampilan antarmuka sistem informasi manajemen kader partai PDIP, yaitu library *React.js*, *HTML*, *JavaScript*, *Node.js*, *CSS* dan *Postman*. *React.js* dipilih karena menyediakan berbagai fitur yang memudahkan bagi seorang *front-end developer* dalam membuat tampilan antarmuka, seperti *component based* dan *JSX*. Tampilan file *react.js* pada proyek SIMKADER PDIP dapat dilihat pada Gambar 3.1.



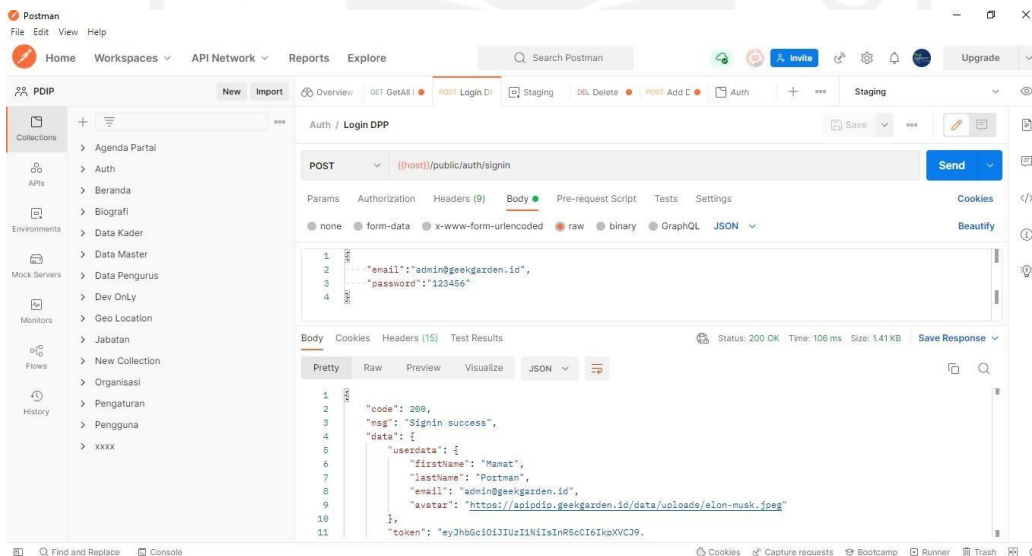
```

return (
  <div className={styles.container}>
    <div style={{
      display: "flex",
      justify-content: "center",
      align-items: "center",
      height: `${window.innerHeight}px`,
      backgroundColor: "#ffff",
    }}>
      <Image src={bannerPdip} alt="bannerpdip" width={500} height={500} />
    </div>
    <div style={{
      position: "relative",
      display: "flex",
      justify-content: "center",
      align-items: "center",
      backgroundColor: "#ffffff",
      height: `${window.innerHeight}px`,
      zIndex: 100
    }}>
      <Form
        name="normal_login"
        className={styles.loginForm}
        initialValues={{
          remember: true,
        }}
        onFinish={viewModel.login}
      />
      <div style={{ padding: 20, display: "flex", align-items: "center", justify-content: "center" }}>
        <Image src={logoPdip} alt="pdip" />
      </div>
    </div>
  </div>
)

```

Gambar 3.1 Tampilan React.Js Proyek SIMKADER PDIP

Sedangkan aplikasi Postman digunakan untuk melakukan percobaan dan pengujian terhadap endpoint API yang telah dibuat oleh back-end developer. Tampilan Postman pada proyek SIMKADER PDIP dapat dilihat pada Gambar 3.2.



Gambar 3.2 Tampilan Postman Proyek SIMKADER PDIP

3.3 Pelaksanaan Proyek

Pengembangan sistem informasi manajemen kader partai PDIP menerapkan metode Waterfall. Adapun tahapan yang dilakukan yaitu analisis, desain, implementasi, pengujian, dan deploy.

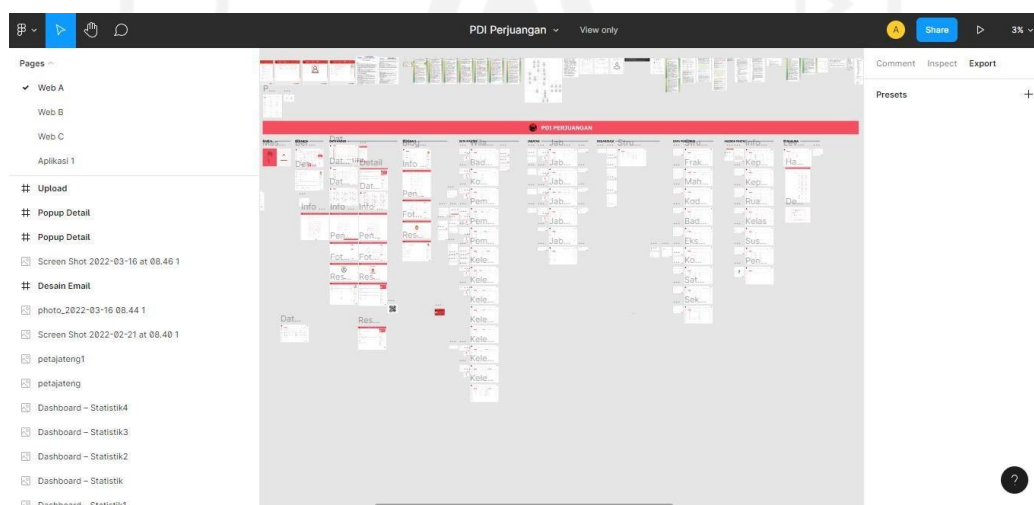
3.3.1 Analisis

Tahap analisis merupakan tahap yang dilakukan untuk mencari dan menganalisis kebutuhan apa saja yang diperlukan dalam proses pengembangan aplikasi. Informasi-informasi yang diperoleh, didapatkan dengan melakukan diskusi bersama klien yang dilakukan oleh Project Manager (PM). Pada tahap ini akan mendapatkan informasi tentang spesifikasi kebutuhan sistem yang ingin dikembangkan serta fitur-fitur yang diinginkan oleh klien. Nantinya, informasi tersebut akan menjadi dasar untuk membuat desain tampilan antarmuka dan desain untuk kebutuhan back-end.

3.3.2 Desain

Berdasarkan dokumen yang didapatkan dari tahap analisis, pengembangan aplikasi dilanjutkan dengan merancang desain tampilan antarmuka pengguna. Tujuannya adalah untuk mendefinisikan alur kerja aplikasi agar mempermudah anggota tim developer dalam melakukan pengkodean. Tahap ini dilakukan oleh tim UI/UX dengan menggunakan salah satu alat desain dan prototipe terpopuler saat ini yaitu Figma. Pemilihan figma dikarenakan memberikan kemudahan dalam kolaborasi antar anggota tim.

Desain yang telah selesai kemudian akan dilakukan evaluasi dengan mendiskusikannya bersama klien untuk mendapatkan umpan balik apakah sudah memenuhi kebutuhan atau belum. Apabila belum memenuhi, maka akan dilakukan perbaikan dan mendiskusikannya ulang hingga kebutuhan klien terpenuhi. Apabila sudah memenuhi, desain kemudian akan diserahkan kepada tim front-end untuk dilakukan pengkodean. Gambar 3.3 memperlihatkan desain antarmuka pengguna aplikasi SIMKADER PDIP pada aplikasi Figma.



Gambar 3.3 Tampilan Figma Proyek SIMKADER PDIP

3.3.3 Implementasi

a. Implementasi React.Js pada Halaman Data Kader

Install package Next.Js

Next.Js merupakan sebuah *framework* yang dibuat sebagai penyempurna *React.Js*. *Next.Js* sudah menyediakan bundel *package React.Js*, sehingga ketika melakukan penginstalan *Next.Js* tidak perlu lagi untuk menginstal *React.Js*. Salah satu kelebihan *Next.Js* adalah tentang kemudahan dalam membuat sistem *routing*. *Routing* dalam *Next.Js* bersifat *page-based*, artinya nama file pada setiap halaman juga merupakan alamat url pada halaman tersebut. Misalnya terdapat file bernama *home.js*, maka alamat urlnya *localhost/home*. Proses instalasi *package Next.Js* dapat dilihat pada Gambar 3.4.

```
$ npm install -g create-next-app
$ create-next-app pdip_frontend
```

Gambar 3.4 Instalasi *Next.Js*

Install library Ant Design

Dalam membantu untuk mempermudah pembangunan tampilan antarmuka yang interaktif, maka digunakan lah *library Ant Design*. *Ant Design* sendiri menyediakan berbagai *component* yang menarik dan mudah untuk diimplementasikan. Untuk proses instalasi *library Ant Design* dapat dilihat pada Gambar 3.5. Setelah proses instalasi selesai, hal yang selanjutnya dilakukan adalah mengimport *component Ant Design* pada *react component* yang telah dibuat. Contoh mengimport salah satu *component Ant Design* yaitu *skeleton* dapat dilihat pada Gambar 3.6.

```
$ npm install antd
```

Gambar 3.5 Instalasi *library Ant Design*

```
import { Skeleton } from "antd";
```

Gambar 3.6 Proses *import component skeleton* dari *library Ant Design*

Mengatur Layout

Hal yang pertama dilakukan adalah mengatur tata letak untuk menentukan bagian *sidebar*, *header*, dan *content*. Dapat dilihat pada Gambar 3.7 merupakan proses mengatur tata letak. Memanggil beberapa child component seperti *SidebarMobile* untuk lebar kurang dari 700, component *Sidebar* untuk lebar lebih dari 700, dan component *HeaderMain* sebagai *Header*.

```
return (
  <Layout style={{ backgroundColor: "red" }}>
    {window.innerWidth < 700 ? (
      <SidebarMobile
        setShowSidebar={setShowSidebar}
        showSidebar={showSidebar}
      />

```

```

) : (
  <Affix offsetTop={0}>
    <SideBar />
  </Affix>
)}
<Layout onClick={() => setShowSidebar(false)}>
  <div ref={container}>
    <HeaderMain show={show} setShow={setShow} />
  </div>
  <Content className={styles.content}>
    <div
      style={{
        padding: 24,
        opacity: showSidebar ? "0.3" : null
      }}
    >
      {children}
    </div>
  </Content>
</Layout>
</Layout>
);

```

Gambar 3.7 Setup tata letak halaman

Membuat Sidebar

Gambar 3.8 merupakan proses pembuatan *Sidebar* dengan memanfaatkan salah satu *component Ant Design* yaitu *Sider*. *Component Sider* memiliki beberapa *property* dan yang digunakan yaitu *collapsible*, *collapsed*, dan *onCollapse*. *Property* tersebut digunakan untuk mengatur *collapse (minimize)* pada *sidebar*. Sementara itu, agar baris kode tidak terlalu panjang, tampilan setiap menu dibuat di *child component* yaitu *MenuItem*. *MenuItem* memiliki sejumlah *props* yang digunakan untuk komunikasi antara *child component* dengan *parent component*.

```

return (
  <Sider
    collapsible
    collapsed={collapse}
    onCollapse={(e) => {
      setCollapse(e);
    }}
    width={130}
    style={{
      overflow: "auto",
      height: "100vh",
    }}
  >
    <div className={styles.logo}>
      <Image
        src={logoPdi}
        alt="pdi"
        width={40}
        height={40}
        objectFit="contain"
      />
    </div>
  </Sider>
);

```



```

    </div>
    <Skeleton loading={load}>
      <MenuItem
        title="Data Kader"
        keys="datakader"
        image={profilIcon}
        imageActive={profilIconActive}
        href="/data-kader"
        collapse={collapse}
      />
    </Skeleton>
  </Sider>
);

```

Gambar 3.8 Menggunakan *component sider* dari *library Ant Design* untuk membuat *sidebar*

Pada component `MenuItem`, data yang diperoleh dari parent component dengan menggunakan props, digunakan untuk menentukan title menu, gambar/icon menu, kondisi menu tersebut sedang *active* atau tidak, link menu, serta kondisi menu sedang *collapse* atau tidak. Proses pembuatan tampilan antarmuka untuk setiap menu dapat dilihat pada Gambar 3.9.

```

return (
  <a onClick={(e) => handleClick(e)} className={style.link}>
    <div>
      <Image
        src={asPath === href ? imageActive : image}
        alt="menu_logo"
        width={20}
        height={20}
      />
    </div>
    <p
      style={{ fontSize: "12px" }}
      className={asPath === href ? style.active : style.title}
    >
      {title}
    </p>
  </a>
);

```

Gambar 3.9 Mengatur setiap menu yang ada pada sidebar

Membuat Header

Pembuatan header memanfaatkan component yang disediakan oleh library Ant Design yaitu `Header` dan `Button`. `Header` merupakan child component dari `MainLayout` sehingga untuk komunikasi antar component menggunakan props. Dapat dilihat pada Gambar 3.10 bahwa terdapat props yang digunakan untuk menampilkan dan menutup box dengan tombol logout apabila pengguna melakukan klik pada foto profil.

```

import { Header } from "antd/lib/layout/layout";
import { Button } from "antd";
import { AuthCookieRemove, getUserData } from "@helper/cookies";
import { useRouter } from "next/router";

type props = {

```



```

    show?: boolean;
    setShow?: any;
  }

  const HeaderMain = ({ show, setShow }: props) => {
    const user = getUserData();
    const urlImg =
      user !== undefined &&
      user.user.avatar == null
        ?
        "https://t3.ftcdn.net/jpg/03/46/83/96/360_F_346839683_6nAPzbhpSkIpb8pmAwufk
        C7c5eD7wYws.jpg"
        : user == undefined ? "" : user.user.avatar;
    const { asPath } = useRouter();

    return (
      <Header
        style={{
          display: "flex",
          alignItems: "center",
          padding: asPath === "/data-kader" ? "30px" : null,
        }}
      >
        {
          window.innerWidth < 700 ? <<</> :
          <div style={{ margin: "0 auto 10px auto" }} >
            <p style={{ font: "bold", margin: "0 auto 0px auto" }} >
              {user == undefined ? "" : user.user["firstName"]} {user ==
              undefined ? "" : user.user["lastName"]}
            </p>
          </div>
        }
        <div>
          <div
            style={{
              display: show ? null : "none",
              position: "absolute",
              zIndex: 999,
              width: "120px",
              right: 30,
              margin: "60px 0 0 0",
            }}
          >
            <div
              style={{
                backgroundColor: "#fff",
                borderRadius: "10px",
                boxShadow:
                  "0 2px 2px 0 rgb(0 0 0 / 14%), 0 3px 1px -2px rgb(0 0 0 /
                  12%), 0 1px 5px 0 rgb(0 0 0 / 20%)",
              }}
            >
              <i className="pop-over-bottom"></i>
              <Button
                onClick={() => {
                  AuthCookieRemove();
                  setShow(false);
                }}
                danger={true}
                style={{ margin: "0 -23px 0 23px" }}
              >

```

```

    >
      Logout
    </Button>
  </div>
</div>
<div style={{ position: "relative" }}>
  <img
    src={urlImg}
    width="30"
    onClick={() => {
      if (show) {
        setShow(false);
      } else {
        setShow(true);
      }
    }}
    height="30"
    alt="profil"
    style={{ objectFit: "contain", borderRadius: "50%" }}
  />
</div>
</Header>
);
}

```

Gambar 3.10 Membuat *Header* dengan memanfaatkan *component layout header* dari library *Ant Design*

Install library Axios

Proses komunikasi dengan tim back-end menggunakan API. Untuk mengakses API tersebut, digunakanlah *library Axios* dalam membantu melakukan permintaan dan menerima tanggapan dari server berupa data JSON. Data tersebut yang nantinya akan ditampilkan dalam aplikasi. Proses instalasi *library Axios* dapat dilihat pada Gambar 3.11, sedangkan proses mengimport *library Axios* pada *react component* dapat dilihat pada Gambar 3.12.

```
$ npm install axios
```

Gambar 3.11 Instalasi *library Axios*

```
import axios from "axios";
```

Gambar 3.12 *Import library Axios*

Melakukan Request

Gambar 3.13 merupakan proses *request* ke back-end menggunakan *library Axios* dengan *method get*. *Request* tersebut dilakukan untuk mendapatkan data kader yang akan ditampilkan pada tabel di menu Data Kader.

```
import ApiConfig from "@config/api";
import axios from "axios";
```

```

import { KaderDataInterface } from "../interface/KaderDataInterface";

class KaderData {
  url: string;

  constructor() {
    this.url = process.env.NEXT_PUBLIC_API_URL || "";
  }

  getData = (): Promise<KaderDataInterface[]> =>
    new Promise(async (resolve, reject) => {
      try {
        let res = await axios.get(
          this.url + "/api/admin/organisasi/kader?limit=100&offset=0",
          ApiConfig.headerWithToken
        );

        if (res.status !== 200) {
          return reject(res.statusText);
        }

        return resolve(res.data);
      } catch (error) {
        return reject(error);
      }
    });
}

export default new KaderData();

```

Gambar 3.13 Proses *request* dengan menggunakan *library Axios*

Membuat *state management* menggunakan *MobX*

Dalam pengelolaan *state* atau manajemen *state* memanfaatkan *library MobX*. *MobX* digunakan untuk melakukan manajemen *state* yang bersifat *global* sehingga dapat diakses di semua *component* antarmuka. Dengan menggunakan *MobX*, tidak perlu lagi untuk memindahkan *state* dari satu *component* ke *component* yang lain. Penerapan *MobX* sendiri lebih sederhana dan lebih mudah dipelajari daripada *library state management* yang lain seperti *Redux*. Proses instalasi *library MobX* dapat dilihat pada Gambar 3.14 sedangkan Gambar 3.15 merupakan proses *management state* data kader dengan menggunakan *MobX*.

```
$ npm install mobx
```

Gambar 3.14 Instalasi *library MobX*

```

import { KaderDataInterface } from "@data/KaderData/interface/KaderDataInterface";
import KaderData from "@data/KaderData/KaderData";
import { action, makeObservable, observable } from "mobx";

export default class DaftarKaderViewModel {
  @observable dataTable: KaderDataInterface[] = [];

```

```

    @action setDataTable = (val: KaderDataInterface[]) => {
      this.dataTable = val;
    };

    @observable loading: boolean = false;
    @action setLoading = (val: boolean) => {
      this.loading = val;
    };

    constructor() {
      makeObservable(this);
    }

    initData = async () => {
      try {
        this.setLoading(true);

        await KaderData.getData()
          .then(async (val: any) => {
            this.setLoading(false);
            this.setDataTable(val.data);
            this.setLoading(false);
            message.error("Terjadi Kesalahan");
          })
          .catch((err) => {
            this.setLoading(false);
            this.setDataTable([]);
            message.error(err.response.data.errorMessage);
          });
      } catch (error) {
        message.error("Terjadi Kesalahan");
      }
    };
  };
};

```

Gambar 3.15 Penerapan *MobX* pada management state Data Kader

Menampilkan data pada tabel halaman Data Kader

Setelah melakukan *request* untuk mendapatkan data dan dilanjutkan dengan melakukan *management state* dari data tersebut, kemudian yang berikutnya dilakukan adalah menampilkan data tersebut. Dapat dilihat pada Gambar 3.16 merupakan proses menampilkan data pada tabel yang diimport dari *library Ant Design*. Data yang diperoleh dari *management state* akan dimasukkan di salah satu *property* tabel yaitu *dataSource*. Dapat dilihat juga pada gambar tersebut bahwa perlu untuk menentukan data yang akan ditampilkan pada setiap kolom dengan memasukkan *variable column* pada *property columns*. *Variable column* tersebut juga dapat digunakan untuk melakukan kustomisasi *style* sesuai dengan desain. Sementara itu, tampilan antarmuka menu Data Kader dapat dilihat pada Gambar 3.17.

```

return (
  <>
  <div style={{ marginTop: -50 }}>

```

```

<Table
  rowKey="uuid"
  loading={viewModel.loading}
  size="small"
  pagination={{ pageSize: pageSize }}
  className="time-table-row-select"
  columns={column}
  scroll={{ x: 960 }}
  dataSource={viewModel.dataTable.filter(
    (x: any) =>
      JSON.stringify(x).toLowerCase().includes(searchStr) &&
      x.status_kader === "Aktif"
    )}
  onChange={handleChange}
/>
</div>
</>
);

const column = [
  {
    title: "NAMA",
    dataIndex: "name",
    width: 170,
    key: "name",
    render: (text: any, record: any, index: any) => {
      return (
        <>
          <Row>
            <Col span={24}>
              <p style={{ fontSize: "12px", fontWeight: "900"
}}>{text}</p>
            </Col>
            <Col span={24}>
              <p
                style={{
                  fontSize: "12px",
                  marginTop: "-10px",
                  color: "#F64E60",
                }}
              >
                {record.gender}
              </p>
            </Col>
          </Row>
        </>
      );
    },
    filteredValue: isfilteredInfo.name || null,
    onFilter: (value: any, record: any) => record.name.includes(value),
    sorter: (a: any, b: any) => a.name.localeCompare(b.name),
    sortOrder: issortedInfo.columnKey === "name" && sortedInfo.order,
  },
]

```

Gambar 3.16 Menampilkan data pada tabel

	NAMA	No.KTA	NIK	TTL	EMAIL/NO.TELP	#
+	Mawar Melati Perempuan	31740910012112900182	317409611290****	21-12-1990	mawar@gmail.com 082112345678	👁️ ⚡️ 🗑️
+	Melani Nur Perempuan	32043210011504870193	320432550487****	15-04-1987	melani@gmail.com 08213351761	👁️ ⚡️ 🗑️
+	Bambang Jeteng Laki-laki	33010120031609990202	330801160999****	16-09-1999	bdjjeteng@gmail.com 0123123131	👁️ ⚡️ 🗑️
+	Ina Wroditen Perempuan	33010120031609990203	330801160999****	16-09-1999	dwroditeng@gmail.com 0123123131	👁️ ⚡️ 🗑️
+	Cipta Jaya Laki-laki	33052120010204840186	330521020484****	02-04-1984	ciptaj@gmail.com 082112345678	👁️ ⚡️ 🗑️
+	Jhon Malik Laki-laki	33060802010909900255	330608090990****	09-09-1990	bjryw@rytetome 08211111222	👁️ ⚡️ 🗑️
+	Mahendra Prabowo Laki-laki	33060802012003980199	340608020398****	02-03-1998	mahendrasrabow0@gmail... 082133549218	👁️ ⚡️ 🗑️
	Wido Inconigto				wido.inconigto@ptpp.ba...	👁️ ⚡️ 🗑️

Gambar 3.17 Tampilan Menu Data Kader

Halaman tambah data kader

Pengguna harus melalui 4 step untuk dapat menambahkan data kader. Step pertama yaitu Informasi Personal, pengguna harus mengisi *input mandatory* yang ditandai dengan simbol bintang berwarna merah. Saat mengakses step pertama, dilakukan request untuk mendapatkan data yang digunakan sebagai pilihan pada *input* provinsi, kabupaten, kecamatan, dan kelurahan. Step kedua yaitu Pendidikan & Pekerjaan. Disediakan 2 input yaitu input Pendidikan dan Pekerjaan. Kedua input tersebut tidak bersifat *mandatory* sehingga tidak wajib untuk diisi. Step 3 yaitu Foto Diri. Pengguna dapat mengupload *image* dan *image* tersebut akan ditampilkan sebagai *preview*. Dan terakhir yaitu Pratinjau. Sistem akan menampilkan semua data yang sebelumnya telah diisi dari step pertama hingga step ketiga. Pengguna dapat mengecek data yang dimasukkan sebelumnya apakah terdapat sesuatu yang tidak tepat atau tidak, jika iya maka pengguna dapat kembali dengan menekan tombol sebelumnya. Apabila sudah tepat, pengguna dapat menekan tombol simpan dan data yang dimasukkan pada step 1 hingga 3 akan dikirim ke back-end dengan menggunakan library Axios untuk kemudian disimpan pada database. Selanjutnya halaman akan otomatis berpindah ke halaman data kader dengan data yang sudah terupdate pada tabel. Tampilan halaman tambah data step pertama dapat dilihat pada Gambar 3.18.

Gambar 3.18 Tampilan Tambah Data

Halaman detail data kader

Halaman ini digunakan untuk menginformasikan lebih detail mengenai data kader. Data yang ditampilkan didapat dari melakukan request ke back-end dengan id data yang dipilih sebagai parameternya. Didapatkan *response* data dalam bentuk JSON yang kemudian akan ditampilkan pada halaman detail data kader. Selain itu, terdapat pula dua tombol yang berada di bagian kanan atas yaitu ikon *left arrow* dan ikon *pencil*. Tombol ikon *left arrow* berfungsi untuk kembali ke halaman sebelumnya sedangkan ikon *pencil* berfungsi untuk pindah ke halaman edit data kader. Tampilan halaman detail data kader dapat dilihat pada Gambar 3.19.

Gambar 3.19 Tampilan Detail Data

Halaman edit data kader

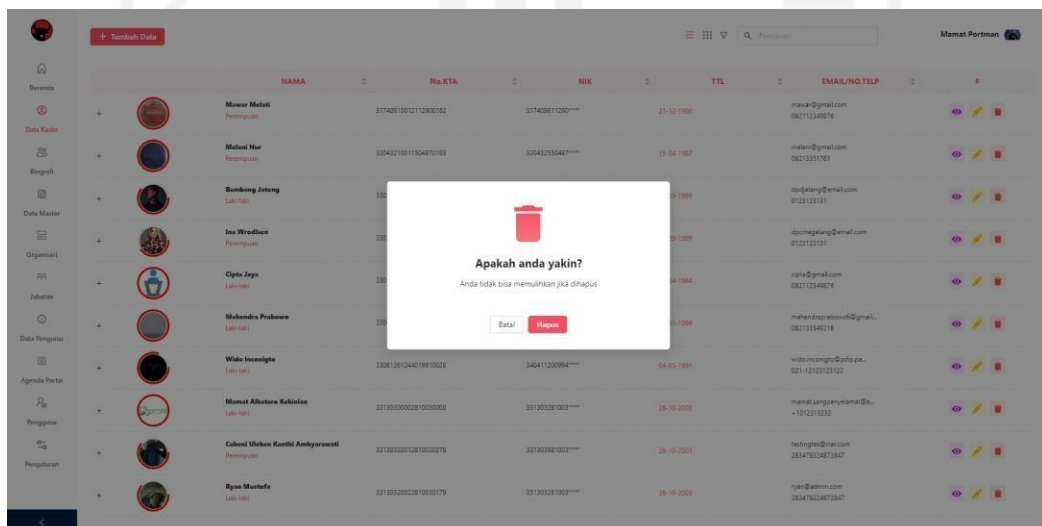
Halaman edit data kader digunakan untuk menyimpan perubahan data yang ingin dirubah. Dapat dilihat pada Gambar 3.20 bahwa tidak ada perbedaan antara tampilan antarmuka

halaman edit dengan tampilan antarmuka halaman tambah. Saat mengakses halaman ini, sistem akan melakukan *request* ke back-end dengan membawa uuid yang didapatkan dari parameter url. Data yang diperoleh dari proses *request* akan ditampilkan sebagai nilai dari setiap *input*.

Gambar 3.20 Tampilan Edit Data

Halaman hapus data kader

Memfaatkan komponen hapus dengan memanggilnya ketika ikon hapus berwarna merah pada tabel ditekan oleh pengguna. Kemudian akan menampilkan pop up dengan 2 pilihan tombol yaitu batal atau hapus seperti pada Gambar 3.21. Tombol batal akan menutup pop up tanpa menghapus data sedangkan tombol hapus akan menghapus data dengan melakukan *request* ke back-end untuk kemudian dilakukan penghapusan pada database.



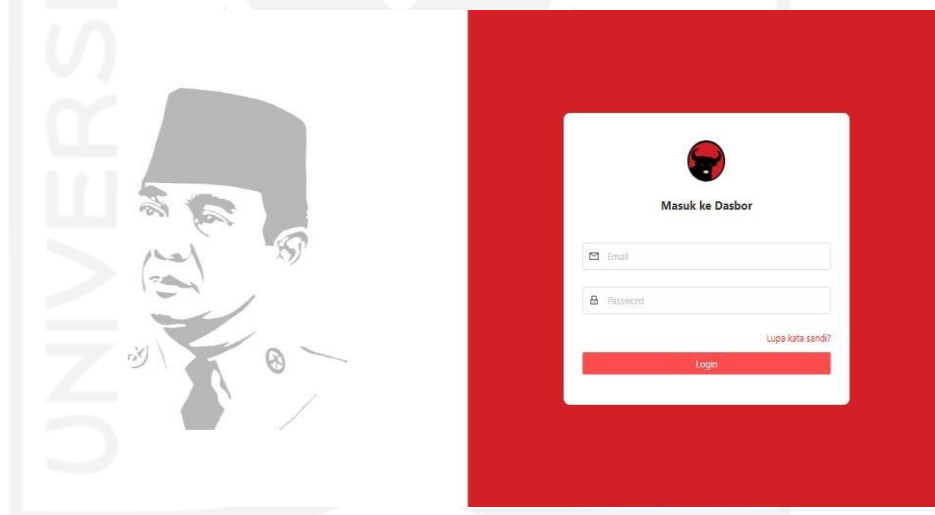
Gambar 3.21 Tampilan Hapus Data

b. Tampilan Hasil Implementasi

Adapun hasil halaman yang telah dibangun pada sistem informasi manajemen kader partai PDIP selain halaman kader yaitu halaman login, biografi, jabatan, data pengurus, dan pengguna.

Halaman Login

Halaman login merupakan tampilan awal aplikasi SIMKADER PDIP yang digunakan bagi pengguna untuk melakukan proses autentikasi agar dapat mengakses sistem. Dapat dilihat pada Gambar 3.22 terdapat dua input yang harus diisi pengguna, yaitu *email* dan *password*. Apabila kedua input tersebut sudah diisi, pengguna dapat menekan tombol login. Email dan password kemudian dikirimkan dengan melakukan *request* ke back-end. Aktivitas *request* memanfaatkan *library* Axios dengan menggunakan API yang telah disediakan oleh tim back-end. Setelah melakukan *request*, didapatkan beberapa *response* data, salah satunya yaitu *token* yang memiliki sifat unik dan disimpan ke dalam *local storage* agar pengguna tidak perlu login kembali apabila menutup tab ataupun browser.



Gambar 3.22 Tampilan Halaman Login

Halaman Biografi

Halaman biografi merupakan halaman yang menampilkan biografi pengguna yang sudah disimpan pada database. Data yang ditampilkan didapatkan dengan melakukan request ke back-end dengan menggunakan API. Kemudian data akan ditampilkan pada tampilan antarmuka pengguna dan disesuaikan dengan desain yang ada di Figma. Pada Gambar 3.23 terdapat pula tombol Ubah Biografi yang digunakan untuk mengubah informasi biografi pengguna. Jika pengguna menekan tombol tersebut, pengguna akan diarahkan ke halaman Ubah Biografi yang *flow* dan desainnya serupa dengan aktivitas tambah Kader.

The screenshot shows a user profile page for 'Mamat Portman'. The page is divided into several sections:

- ALAMAT:**
 - SEGIAN KTP: [Redacted]
 - DOMISILI: JALAN YODHAKARTASING 1 PATUREJO GRABAG PURWOREJO JAWA TENGAH
- PENDIDIKAN:**
 - SD: SD NISAH 2 (2009 - 2015)
 - SMP: SMP NISAH 2 (2015 - 2016)
 - SMA: SMA NISAH 2 (2016 - 2018)
 - S1: UNIVERSITAS MERDEKA, PALEMBANG (2018 - 2019)
- PEKERJAAN:**
 - Karyawan Swasta: PTSAJDI A STAFF (2012 - SEKARANG)
 - PNS: KIR STAFF (2012 - 2018)
- MEDIA SOSIAL:**
 - facebook: mamat.manusa.kesri
- INFORMASI PERSONAL:**
 - Nama Lengkap: Mamat Iskandar Koluman
 - KTA: 311000000210030000
 - NIK: 311002010030006
 - Tempat, Tanggal Lahir: TELUK WOTOGAMBA 28-10-2003
 - Jenis Kelamin: L
 - Golongan Darah: A
 - Status Perkawinan: Belum Kawin
 - Nomor Telepon: +628121232
 - Email: mamat.sandamwamat@gmail.com

Gambar 3.23 Tampilan Halaman Biografi

Halaman Jabatan

Halaman jabatan berfungsi untuk menentukan apa saja jabatan yang dibutuhkan pada setiap bagian. Menggunakan library Axios untuk mendapatkan data yang akan ditampilkan pada tabel. Dapat dilihat pada Gambar 3.24 bahwa tabel pada halaman jabatan memiliki 3 kolom yaitu nama jabatan, level, dan komposisi. Terdapat 4 *action* yaitu tambah, edit, hapus, dan tambah *child* jabatan. Sedangkan pada *child* jabatan memiliki 2 *action* yaitu edit dan hapus. Selain itu juga terdapat *field search* yang digunakan untuk mencari data tertentu.

The screenshot shows the 'Struktur Partai' page with a table of job positions. The table has the following columns: NAMA JABATAN, LEVEL, and KOMPOSISI. There are also columns for 'Aksi' (Add, Edit, Delete) and 'Child' (Add, Edit, Delete).

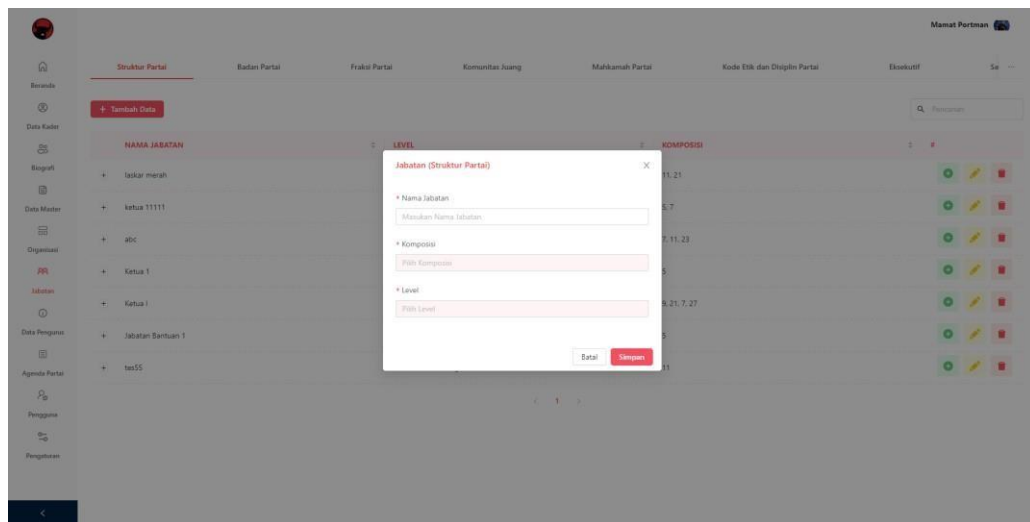
NAMA JABATAN	LEVEL	KOMPOSISI	Aksi	Child
+ laskar merah	Ranting	11, 21	[Add] [Edit] [Delete]	
+ ketua 11111	DRP, DPC, PAC, Ranting, Anak Ranting, PPLN, DRPN	5, 7	[Add] [Edit] [Delete]	
+ abc	DRPN, Anak Ranting, Ranting	7, 11, 23	[Add] [Edit] [Delete]	
+ Ketua 1	DRP	5	[Add] [Edit] [Delete]	
- Ketua 1	DRP, DPC, PAC, Ranting	9, 21, 7, 27	[Add] [Edit] [Delete]	
Bendahara 1	DRP, DPC, PAC, Ranting	9, 21, 7, 27	[Add] [Edit] [Delete]	
+ Jabatan Bertuan 1	DRP	5	[Add] [Edit] [Delete]	
+ tes55	PPLN, Anak Ranting	11	[Add] [Edit] [Delete]	

Gambar 3.24 Tampilan Halaman Jabatan

- Tambah

Terdapat 3 input yang disediakan pada form tambah jabatan yaitu nama jabatan, level, dan komposisi. *Input* nama jabatan bertipe *text* sedangkan *input* level dan komposisi bertipe *multiple select*. Masukan pada setiap input akan dikirimkan ke back-end apabila pengguna menekan tombol tambah. Kemudian pop up akan

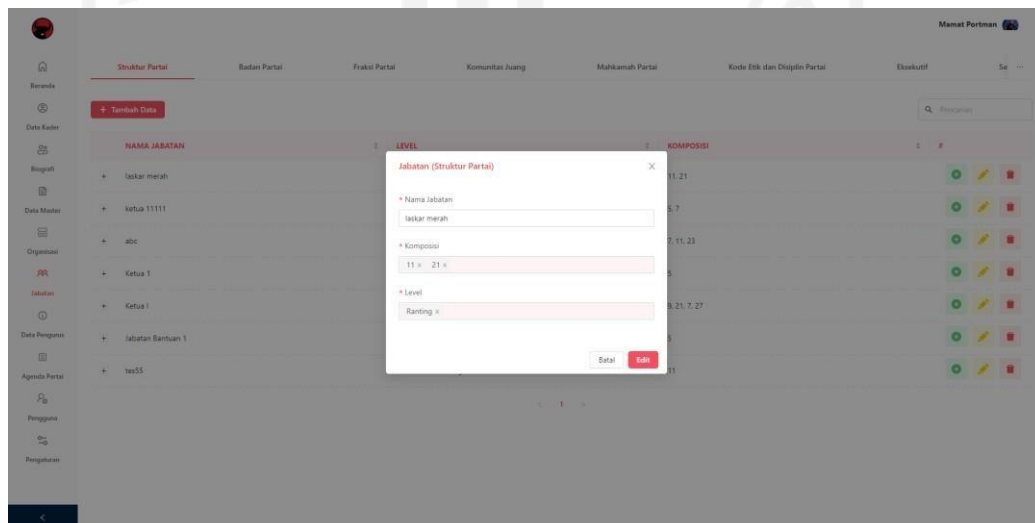
menutup secara otomatis dan data akan terupdate dengan menampilkan *toast* selama 3 detik. Tampilan pop up tambah data dapat dilihat pada Gambar 3.25.



Gambar 3.25 Tampilan Tambah Jabatan

- Edit

Sama seperti *form* tambah jabatan, *form* edit jabatan juga memiliki 3 inputan yaitu nama jabatan, level, dan komposisi. Selain masukan pengguna, id pada data yang dipilih akan menjadi parameter yang akan dikirimkan ke back-end dengan menggunakan API. Tampilan pada edit jabatan dapat dilihat pada Gambar 3.26.

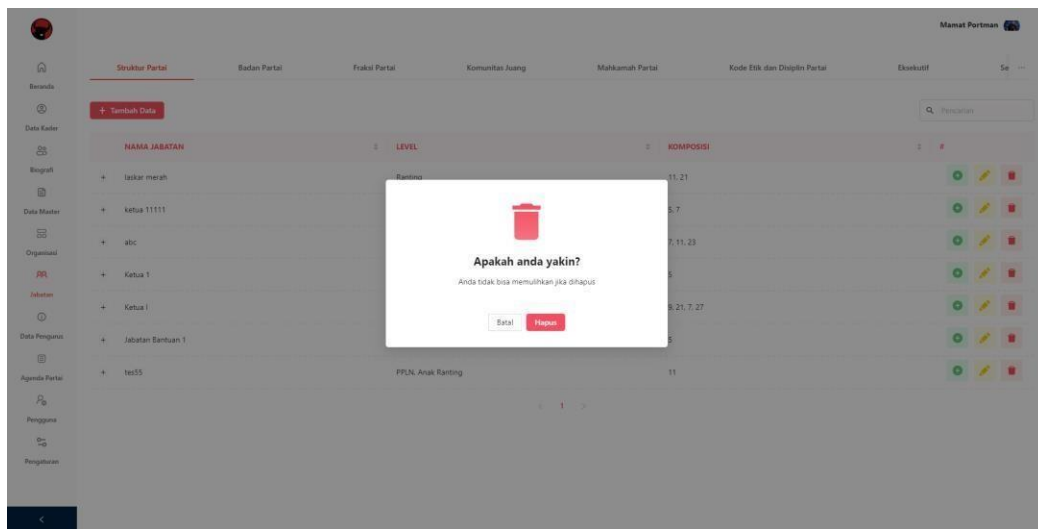


Gambar 3.26 Tampilan Edit Jabatan

- Hapus

Komponen hapus data akan dipanggil ketika pengguna menekan tombol ikon sampah pada tabel. Apabila pengguna menekan tombol hapus, maka id data yang dipilih akan dikirim ke back-end untuk dilakukan penghapusan pada database.

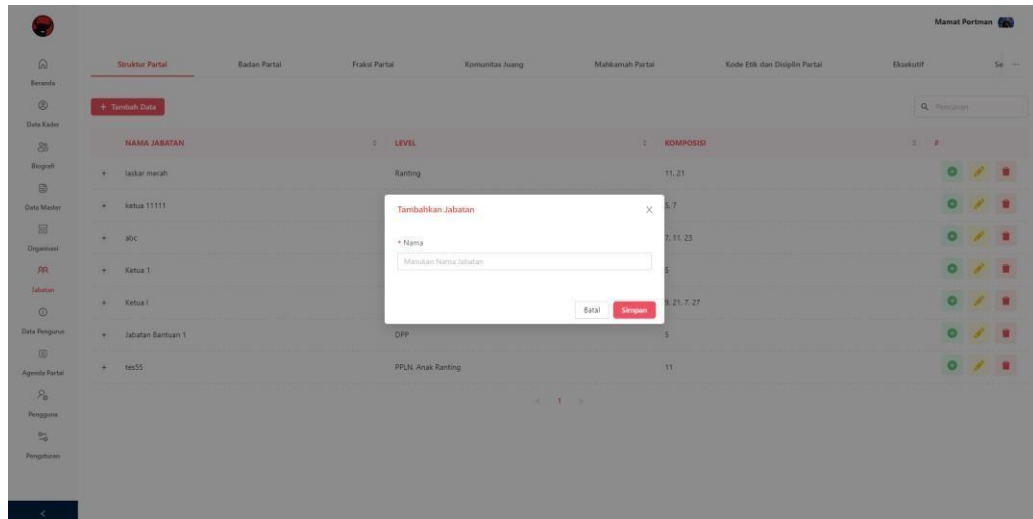
Kemudian pop up akan tertutup dan data akan terupdate. Tampilan pop up hapus data dapat dilihat pada Gambar 3.27.



Gambar 3.27 Tampilan Hapus Jabatan

- **Tambah Child Jabatan**

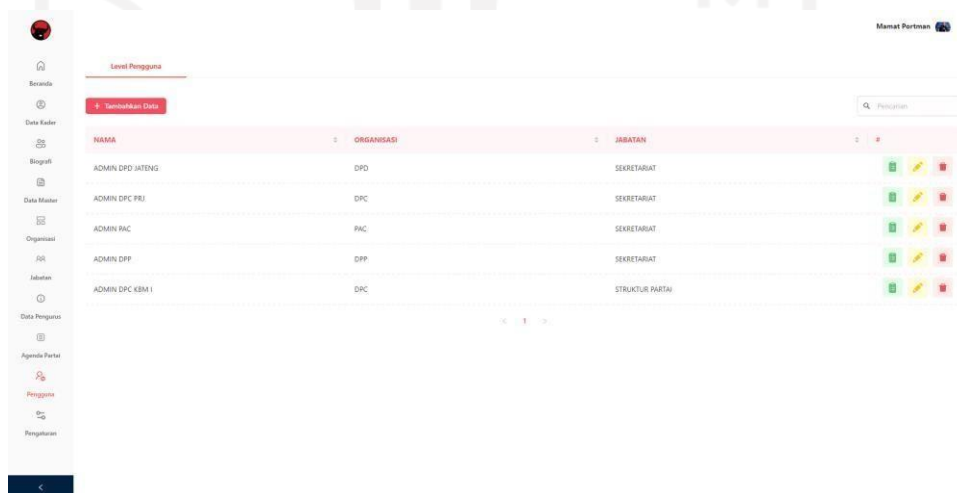
Pop up akan ditampilkan apabila pengguna menekan tombol ikon tambah berwarna hijau pada tabel. Tambah child jabatan berfungsi untuk menambah data jabatan yang memiliki posisi di bawah jabatan yang dipilih. Dapat dilihat pada Gambar 3.28 bahwa *form* tambah *child* jabatan hanya memiliki 1 inputan yaitu nama. Masukan pengguna pada *input* nama dan id dari induk jabatan akan menjadi parameter yang dikirimkan ke back-end melalui API. Data yang ditambahkan akan terupdate dan dapat dilihat apabila pengguna menekan tombol ikon tambah yang berada di samping kolom jabatan. Maka akan menampilkan data *child* jabatan di bawahnya.



Gambar 3.28 Tampilan Tambah *Child* Jabatan

Halaman Pengguna

Halaman pengguna merupakan halaman yang digunakan untuk melakukan aktivitas manajemen pengguna. Berfungsi untuk mengatur dan menentukan hak akses setiap pengguna. Tampilan pada halaman pengguna dapat dilihat pada Gambar 3.29. Data diperoleh dari proses *request* dengan menggunakan API dan ditampilkan dalam bentuk tabel dengan 3 kolom yaitu nama, organisasi, dan jabatan. Terdapat *field search* yang digunakan untuk fungsi pencarian dan 4 *action*, masing-masingnya yaitu tambah, edit, hapus, serta penentuan hak akses.

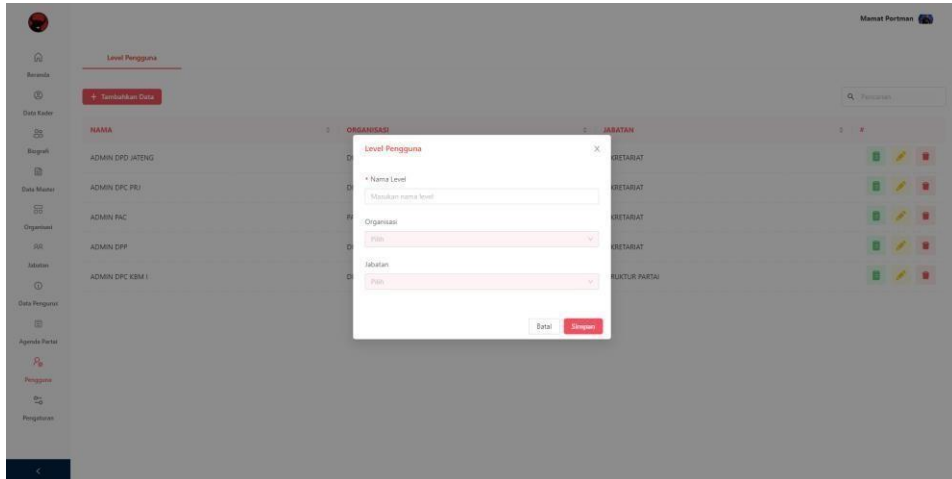


Gambar 3.29 Tampilan Halaman Pengguna

- Tambah

Form tambah pengguna ditampilkan dalam bentuk pop up modal dengan 3 inputan yaitu nama, organisasi, dan jabatan. *Field* nama dengan *input type text* sedangkan *field* organisasi dan jabatan dengan *input type select*. Data yang menjadi pilihan pada *input type select* diperoleh dari proses request ke back-end. Masukan pengguna pada

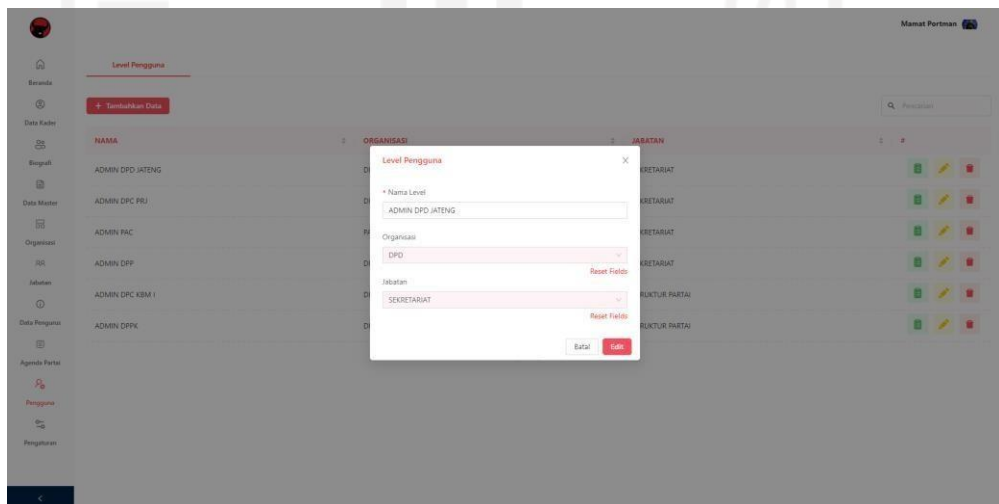
setiap *input* serta token yang berada di *local storage* akan dikirimkan ke back-end dengan menggunakan library Axios. Setelah melakukan *request*, pop up modal akan otomatis tertutup kemudian menampilkan *toast* dan data akan terupdate. Tampilan pada *form* tambah pengguna dapat dilihat pada Gambar 3.30.



Gambar 3.30 Tampilan Tambah Pengguna

- Edit

Dapat dilihat pada Gambar 3.31 bahwa tampilan antarmuka form edit menyerupai tampilan *form* tambah. Perbedaannya bahwa *form* edit memiliki fitur *reset field* pada *input type select* yang berfungsi untuk mereset masukan pengguna.

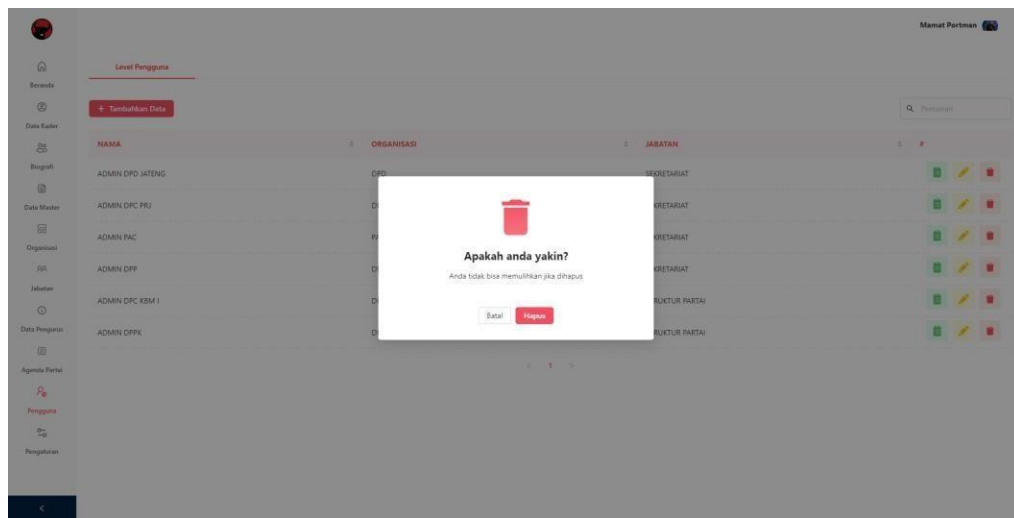


Gambar 3.31 Tampilan Edit Pengguna

- Delete

Tampilan action hapus memanfaatkan component hapus yang telah dibuat sebelumnya. Jika pengguna menekan tombol *delete* maka akan melakukan *request*

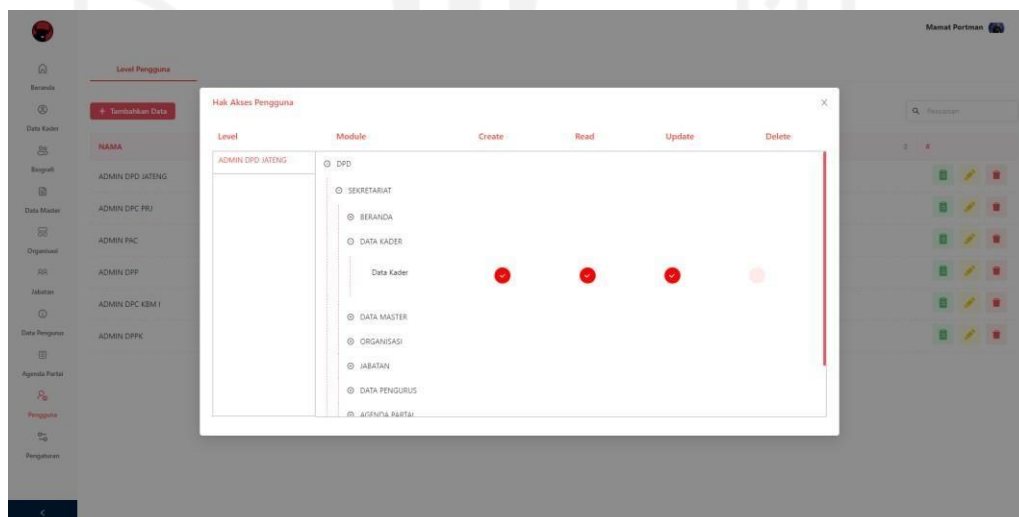
ke back-end dengan membawa id data dan token yang diambil dari *local storage*. Tampilan pop up hapus pengguna bisa dilihat pada Gambar 3.32.



Gambar 3.32 Tampilan Hapus Pengguna

- Penentuan hak akses

Tampilan pop up pada Gambar 3.33 dapat muncul apabila pengguna menekan tombol yang berwarna hijau pada tabel pengguna. Pop up ini berfungsi untuk menentukan hak akses setiap pengguna dalam melakukan *action create, read, update, dan delete* pada setiap menu.



Gambar 3.33 Tampilan Penentuan Hak Akses

3.3.4 Pengujian

Pengujian pada sistem informasi manajemen kader partai PDIP menggunakan salah satu teknik pengujian dalam metode agile yaitu Acceptance Test Driven Development (ATDD). Pengujian ini bertujuan agar fungsionalitas sistem atau perilaku sistem berfungsi dengan baik sesuai dengan kebutuhan klien yang telah ditentukan. Pengujian dengan ATDD akan

menggambarkan kriteria penerimaan fitur dari aplikasi yang dikembangkan dan menguji perilaku sistem apakah sesuai dengan harapan atau tidak dari sudut pandang penggunaannya. Pengujian bisa dilakukan secara manual maupun otomatis, namun dalam penelitian ini pengujian dilakukan secara manual.

Tahapan-tahapan pengujian dengan teknik ATDD yaitu *write the acceptance test*, *fail the acceptance test*, *write the unit test*, *fail the unit test*, *write functional code*, dan *refactor the code* (Anggarwal, V., & Singh, M., 2014). Dalam penelitian ini, pengujian yang dikerjakan hanya *write the acceptance test* dan *fail the acceptance test*. *Write the acceptance test* sendiri adalah tahapan yang dilakukan dengan menulis semua kemungkinan dan harapan dari input dan output sistem. Tahap ini dilakukan untuk memastikan apakah keluaran dari sistem sesuai dengan yang diharapkan. Berbeda dengan *write the acceptance test*, pada tahap *fail the acceptance test* akan menguji kegagalan berdasarkan input dan output sistem. Adapun hasil pengujian menggunakan Acceptance Test Driven Development dapat dilihat pada tabel dibawah ini.

1. Halaman Login

Tabel 3.2 Pengujian halaman login

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi <i>email</i> dan <i>password</i> dengan benar.	Berhasil login dan tampil halaman dashboard.	Berhasil login dan tampil halaman dashboard.	Terpenuhi
2	Salah mengisi <i>email</i> dan mengisi <i>password</i> dengan benar.	Tampil toast dengan pesan bahwa pengguna tidak ditemukan.	Tampil toast selama 3 detik dengan pesan "User not found".	Terpenuhi

Tabel 3.3 Pengujian kegagalan halaman login

No	Input	Hasil yang diharapkan	Output	Hasil
1	Salah mengisi <i>password</i> dan mengisi <i>email</i> dengan benar.	Tampil pesan bahwa email atau password salah.	Tampil toast selama 3 detik dengan pesan "Incorrect email / password".	Terpenuhi
2	Mengetikkan <i>email</i> yang tidak valid.	Tampil pesan error bahwa email tidak valid.	Tampil pesan error bahwa email tidak valid dibawah input.	Terpenuhi

3	Menekan tombol login tanpa mengisi <i>email</i> dan <i>password</i> .	Tampil pesan error bahwa email tidak valid dan password harus diisi.	Tampil pesan error bahwa email tidak valid dan password harus diisi dibawah input.	Terpenuhi
---	---	--	--	-----------

2. Halaman Kader

Tabel 3.4 Pengujian halaman kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengakses halaman kader.	Menampilkan data kader dalam bentuk tabel.	Menampilkan data kader dalam bentuk tabel.	Terpenuhi
2	Pilih 25 pada fitur <i>limit show</i> .	Menampilkan 25 data pada tabel.	Tampil 25 data pada tabel.	Terpenuhi
3	Pilih 2 pada fitur <i>pagination</i> .	Menampilkan data tabel pada halaman ke 2.	Tampil data tabel yang ada pada halaman ke 2.	Terpenuhi
4	Melakukan sorting pada kolom nama.	Data yang ditampilkan akan terurut secara <i>ascending</i> pada klik yang pertama. Terurut secara <i>descending</i> pada klik yang kedua, dan <i>cancel sorting</i> pada klik yang ketiga.	Data yang ditampilkan akan terurut secara <i>ascending</i> pada klik yang pertama. Terurut secara <i>descending</i> pada klik yang kedua, dan <i>cancel sorting</i> pada klik yang ketiga.	Terpenuhi
5	Memilih bentuk grid pada data yang akan tampil.	Menampilkan data dalam bentuk grid.	Menampilkan data dalam bentuk grid.	Terpenuhi
6	Menekan tombol filter.	Menampilkan pop up form filter.	Menampilkan pop up form filter.	Terpenuhi
7	Memilih DIY pada input provinsi dan menekan tombol filter.	Menampilkan data yang beralamat DIY.	Menampilkan data yang beralamat DIY.	Terpenuhi
8	Menekan tombol close pada pop up filter.	Pop up filter akan menutup.	Pop up filter akan menutup.	Terpenuhi

9	Menekan tombol refresh.	Merefresh filter dan menampilkan semua data.	Merefresh filter dan menampilkan semua data.	Terpenuhi
10	Mengisi input search.	Menampilkan data sesuai dengan masukan pengguna.	Menampilkan data sesuai dengan masukan pengguna.	Terpenuhi

Tambah kader

Tabel 3.5 Pengujian tambah kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi NIK dengan benar.	Input tanggal lahir, provinsi, kabupaten, kecamatan otomatis terisi.	Input tanggal lahir, provinsi, kabupaten, kecamatan otomatis terisi.	Terpenuhi
2	Menekan tombol batal.	Kembali ke halaman data kader.	Kembali ke halaman data kader.	Terpenuhi
3	Menekan tombol sebelumnya.	Kembali ke halaman sebelumnya.	Kembali ke halaman sebelumnya.	Terpenuhi
4	Menekan tombol lanjut.	Berpindah ke halaman selanjutnya.	Berpindah ke halaman selanjutnya.	Terpenuhi
5	Upload foto.	Tampil preview foto.	Tampil preview foto.	Terpenuhi
6	Menekan tombol ikon sampah setelah preview foto tampil.	Foto akan terhapus.	Foto terhapus.	Terpenuhi
7	Menekan tombol simpan.	Menambah data baru pada tabel data kader.	Menambah data baru pada tabel data kader.	Terpenuhi

Tabel 3.6 Pengujian kegagalan tambah kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi NIK dengan asal.	Tampil pesan error bahwa nik tidak ditemukan.	Tampil toast dengan pesan error	Terpenuhi

			bahwa nik tidak ditemukan.	
2	Menekan tombol lanjut tanpa mengisi field yang mandatory.	Tampil pesan error bahwa input tidak boleh kosong.	Tampil pesan error pada input mandatory yang tidak diisi bahwa input tidak boleh kosong.	Terpenuhi
3	Menekan tombol simpan tetapi tidak dapat menyimpan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Ubah kader

Tabel 3.7 Pengujian ubah data kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol simpan.	Menyimpan perubahan data dan memperbarui data pada tabel.	Menyimpan perubahan data dan memperbarui data pada tabel.	Terpenuhi

Tabel 3.8 Pengujian kegagalan ubah data kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol simpan tetapi tidak dapat menyimpan perubahan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Lihat detail kader

Tabel 3.9 Pengujian lihat detail kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol ikon pensil.	Berpindah ke halaman ubah data kader.	Berpindah ke halaman ubah data kader.	Terpenuhi

2	Menekan tombol ikon <i>arrow left</i> .	Kembali ke halaman sebelumnya.	Kembali ke halaman sebelumnya.	Terpenuhi
3	Menekan tombol Buka Kode QR.	Membuka pop up dengan menampilkan QR Code.	Membuka pop up dengan menampilkan QR Code.	Terpenuhi

Hapus kader

Tabel 3.10 Pengujian hapus data kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol batal.	Data tidak terhapus dan pop up hapus data akan tertutup.	Data tidak terhapus dan pop up hapus data akan tertutup.	Terpenuhi
2	Menekan tombol hapus.	Menghapus data dan memperbarui data pada tabel data kader.	Menghapus data dan memperbarui data pada tabel data kader.	Terpenuhi

Tabel 3.11 Pengujian kegagalan hapus data kader

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol hapus tetapi data tidak terhapus.	Tampil pesan error bahwa data tidak dapat terhapus.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

3. Halaman Biografi

Tabel 3.12 Pengujian halaman biografi

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol Buka Kode QR.	Membuka pop up dengan menampilkan QR Code.	Membuka pop up dengan menampilkan QR Code.	Terpenuhi
2	Menekan tombol Ubah Biografi.	Berpindah ke halaman ubah biografi.	Berpindah ke halaman ubah biografi.	Terpenuhi

Ubah biografi

Tabel 3.13 Pengujian ubah biografi

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol Buka Kode QR.	Membuka pop up dengan menampilkan QR Code.	Membuka pop up dengan menampilkan QR Code.	Berhasil
2	Menekan tombol Ubah Biografi.	Berpindah ke halaman ubah biografi.	Berpindah ke halaman ubah biografi.	Berhasil
3	Menekan tombol simpan.	Menyimpan perubahan data dan data pada halaman biografi akan terupdate.	Menyimpan perubahan data dan data pada halaman biografi akan terupdate.	Berhasil

Tabel 3.14 Pengujian kegagalan ubah biografi

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol simpan tetapi tidak dapat menyimpan perubahan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

4. Halaman Jabatan

Tabel 3.15 Pengujian halaman jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengakses halaman jabatan.	Menampilkan data jabatan dalam bentuk tabel.	Menampilkan data jabatan dalam bentuk tabel.	Terpenuhi
2	Melakukan sorting pada kolom nama jabatan.	Data yang ditampilkan akan terurut secara <i>ascending</i> pada klik yang pertama. Terurut secara <i>descending</i> pada klik yang kedua, dan <i>cancel sorting</i> pada klik yang ketiga.	Data yang ditampilkan akan terurut secara <i>ascending</i> pada klik yang pertama. Terurut secara <i>descending</i> pada klik yang kedua,	Terpenuhi

			dan <i>cancel sorting</i> pada klik yang ketiga.	
3	Mengisi input search.	Menampilkan data sesuai dengan masukan pengguna.	Menampilkan data sesuai dengan masukan pengguna.	Terpenuhi
4	Menekan tombol Tambah Data.	Tampil pop up form tambah data.	Tampil pop up form tambah data.	Terpenuhi
5	Menekan tombol ikon plus berwarna hijau.	Tampil pop up form tambah anak jabatan.	Tampil pop up form tambah anak jabatan.	Terpenuhi
6	Menekan tombol ikon pensil.	Tampil pop up form ubah data jabatan.	Tampil pop up form ubah data jabatan.	Terpenuhi
7	Menekan tombol ikon sampah.	Tampil pop up form hapus data jabatan.	Tampil pop up form hapus data jabatan.	Terpenuhi
8	Menekan tombol ikon plus pada kolom pertama.	Menampilkan children data dari data yang dipilih.	Menampilkan children data dari data yang dipilih.	Terpenuhi
9	Menekan tab Badan Partai.	Menampilkan data jabatan pada Badan Partai.	Menampilkan data jabatan pada Badan Partai.	Terpenuhi

Tambah jabatan

Tabel 3.16 Pengujian tambah jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi seluruh input dan menekan tombol simpan.	Menyimpan data terbaru pada tabel data jabatan dan menampilkan pesan berhasil disimpan.	Menyimpan data terbaru pada tabel data jabatan dan menampilkan pesan berhasil disimpan.	Terpenuhi
2	Menekan tombol batal.	Pop up tambah data akan menutup.	Pop up tambah data menutup.	Terpenuhi
3	Menekan tombol ikon close (X).	Pop up tambah data akan menutup.	Pop up tambah data menutup.	Terpenuhi

Tabel 3.17 Pengujian kegagalan tambah jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol simpan tanpa mengisi input pada form.	Menampilkan pesan error bahwa input tidak boleh kosong.	Tampil pesan error bahwa input tidak boleh kosong.	Terpenuhi
2	Menekan tombol simpan tetapi tidak dapat menyimpan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Ubah jabatan

Tabel 3.18 Pengujian ubah jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi atau memperbarui masukan pada input dan menekan tombol simpan.	Menyimpan perubahan data pada tabel data jabatan dan menampilkan pesan bahwa berhasil melakukan edit data.	Menyimpan perubahan data pada tabel data jabatan dan menampilkan pesan "Berhasil edit data".	Terpenuhi
2	Menekan tombol batal.	Pop up ubah data akan menutup.	Pop up ubah data menutup.	Terpenuhi
3	Menekan tombol ikon plus.	Pop up ubah data akan menutup.	Pop up ubah data menutup.	Terpenuhi

Tabel 3.19 Pengujian kegagalan ubah jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menghapus masukan pada salah satu input dan menekan tombol simpan.	Menampilkan pesan error bahwa input tidak boleh kosong.	Tampil pesan error bahwa input tidak boleh kosong.	Terpenuhi
2	Menekan tombol simpan tetapi tidak dapat menyimpan perubahan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Hapus jabatan

Tabel 3.20 Pengujian hapus jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol batal.	Data tidak terhapus dan pop up hapus data akan tertutup.	Data tidak terhapus dan pop up hapus data akan tertutup.	Terpenuhi
2	Menekan tombol hapus.	Menghapus data dan memperbarui data pada tabel data jabatan.	Menghapus data dan memperbarui data pada tabel data jabatan.	Terpenuhi

Tabel 3.21 Pengujian kegagalan hapus jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol hapus tetapi data tidak terhapus.	Tampil pesan error bahwa data tidak dapat terhapus.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Tambah anak jabatan

Tabel 3.22 Pengujian tambah anak jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi input nama jabatan dan menekan tombol simpan.	Menyimpan data pada tabel data jabatan dan menampilkan pesan berhasil disimpan.	Menyimpan data pada tabel data jabatan dan menampilkan pesan berhasil disimpan.	Terpenuhi
2	Menekan tombol batal.	Data tidak tersimpan dan pop up tambah anak jabatan akan tertutup.	Data tidak tersimpan dan pop up tambah anak jabatan akan tertutup.	Terpenuhi
3	Menekan tombol ikon close (X).	Pop up tambah anak jabatan akan menutup.	Pop up tambah anak jabatan akan menutup.	Terpenuhi

Tabel 3.23 Pengujian kegagalan tambah anak jabatan

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol simpan tanpa mengisi input pada form.	Menampilkan pesan error bahwa input tidak boleh kosong.	Tampil pesan error bahwa input tidak boleh kosong.	Terpenuhi
2	Menekan tombol simpan tetapi tidak dapat menyimpan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

5. Halaman Pengguna

Tabel 3.24 Pengujian halaman pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengakses halaman pengguna.	Menampilkan data pengguna dalam bentuk tabel.	Menampilkan data pengguna dalam bentuk tabel.	Terpenuhi
2	Melakukan sorting pada kolom nama.	Data yang ditampilkan akan terurut secara <i>ascending</i> pada klik yang pertama. Terurut secara <i>descending</i> pada klik yang kedua, dan <i>cancel sorting</i> pada klik yang ketiga.	Data yang ditampilkan akan terurut secara <i>ascending</i> pada klik yang pertama. Terurut secara <i>descending</i> pada klik yang kedua, dan <i>cancel sorting</i> pada klik yang ketiga.	Terpenuhi
3	Mengisi input search.	Menampilkan data sesuai dengan masukan pengguna.	Menampilkan data sesuai dengan masukan pengguna.	Terpenuhi
4	Menekan tombol Tambah Data.	Tampil pop up form tambah data.	Tampil pop up form tambah data.	Terpenuhi
5	Menekan tombol ikon plus berwarna hijau.	Tampil pop up form tambah anak jabatan.	Tampil pop up form tambah anak jabatan.	Terpenuhi
6	Menekan tombol ikon pensil.	Tampil pop up form ubah data jabatan.	Tampil pop up form ubah data jabatan.	Terpenuhi

7	Menekan tombol ikon sampah.	Tampil pop up form hapus data jabatan.	Tampil pop up form hapus data jabatan.	Terpenuhi
---	-----------------------------	--	--	-----------

Tambah pengguna

Tabel 3.25 Pengujian tambah pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Hanya mengisi input nama.	Menyimpan data terbaru pada tabel data pengguna dan menampilkan pesan berhasil disimpan.	Menyimpan data terbaru pada tabel data pengguna dan menampilkan pesan berhasil disimpan.	Terpenuhi
2	Mengisi seluruh input dan menekan tombol simpan.	Menyimpan data terbaru pada tabel data pengguna dan menampilkan pesan berhasil disimpan.	Menyimpan data terbaru pada tabel data pengguna dan menampilkan pesan berhasil disimpan.	Terpenuhi
3	Menekan tombol batal.	Pop up tambah data akan menutup.	Pop up tambah data menutup.	Terpenuhi
4	Menekan tombol ikon close (X).	Pop up tambah data akan menutup.	Pop up tambah data menutup.	Terpenuhi

Tabel 3.26 Pengujian kegagalan tambah pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol simpan tanpa mengisi input nama, organisasi, dan jabatan.	Menampilkan pesan error bahwa nama tidak boleh kosong.	Menampilkan pesan error bahwa nama tidak boleh kosong.	Terpenuhi
2	Menekan tombol simpan tetapi tidak dapat menyimpan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Ubah pengguna

Tabel 3.27 Pengujian ubah pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Mengisi atau memperbarui masukan pada input dan menekan tombol simpan.	Menyimpan perubahan data pada tabel data pengguna dan menampilkan pesan bahwa berhasil melakukan edit data.	Menyimpan perubahan data pada tabel data pengguna dan menampilkan pesan “Berhasil edit data”.	Terpenuhi
2	Menekan tombol Reset Field.	Menghapus masukan pengguna.	Masukan pengguna pada input terhapus.	Terpenuhi
3	Menekan tombol batal.	Pop up ubah data akan menutup.	Pop up ubah data menutup.	Terpenuhi
4	Menekan tombol ikon plus.	Pop up ubah data akan menutup.	Pop up ubah data menutup.	Terpenuhi
5	Menekan tombol ikon close (X).	Pop up tambah data akan menutup.	Pop up tambah data menutup.	Terpenuhi

Tabel 3.28 Pengujian kegagalan ubah pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menghapus masukan pada salah satu input dan menekan tombol simpan.	Menampilkan pesan error bahwa input tidak boleh kosong.	Tampil pesan error bahwa input tidak boleh kosong.	Terpenuhi
2	Menekan tombol simpan tetapi tidak dapat menyimpan perubahan ke database.	Tampil pesan error bahwa data tidak dapat tersimpan.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Hapus pengguna

Tabel 3.29 Pengujian hapus pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol batal.	Data tidak terhapus dan pop up hapus data akan tertutup.	Data tidak terhapus dan pop up hapus data akan tertutup.	Berhasil

2	Menekan tombol hapus.	Menghapus data dan memperbarui data pada tabel data pengguna.	Menghapus data dan memperbarui data pada tabel data pengguna.	Berhasil
3	Menekan tombol ikon close (X).	Pop up hapus data akan menutup.	Pop up hapus data menutup.	Berhasil

Tabel 3.30 Pengujian kegagalan hapus pengguna

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol hapus tetapi data tidak terhapus.	Tampil pesan error bahwa data tidak dapat terhapus.	Tampil toast dengan pesan error yang meneruskan pesan dari backend.	Terpenuhi

Penentuan hak akses

Tabel 3.31 Pengujian penentuan hak akses

No	Input	Hasil yang diharapkan	Output	Hasil
1	Menekan tombol ikon plus.	Tampil data daftar menu.	Tampil data daftar menu.	Berhasil
2	Melakukan centang/checklist pada bagian create, read, update, dan delete.	Create, read, update, dan delete akan tercentang.	Create, read, update, dan delete akan tercentang.	Berhasil
3	Menekan tombol ikon close (X).	Pop up penentuan hak akses akan menutup.	Pop up penentuan hak akses akan menutup	Berhasil

Berdasarkan hasil pengujian yang dapat dilihat pada tabel diatas, dapat diketahui bahwa fungsionalitas sistem pada halaman login, kader, biografi, jabatan, dan pengguna dapat dijalankan dengan baik. Keluaran yang dihasilkan sudah sesuai dengan yang keluaran yang diharapkan.

3.3.5 Deployment

Tahap *deployment* dilakukan dengan memanfaatkan aplikasi GitLab. Terdapat dua *branch* berbeda untuk melakukan *deployment*, yaitu *branch developer* untuk server *staging/lingkungan developer*, dan *branch master* untuk server *production*. Setiap kali developer selesai

mengerjakan *task* atau fitur yang diberikan, maka mereka akan melakukan *push* ke *branch*-nya masing-masing. Setelah itu, tim DevOps akan melakukan *merge* ke *branch develop* untuk kemudian melakukan *deployment* pada server *staging*/lingkungan developer. Tujuannya adalah untuk menyebarkan *task* atau fitur yang telah diselesaikan tersebut agar dapat dicoba dan diuji oleh tim QA/QC. Nantinya, aplikasi yang sudah terselesaikan secara keseluruhan, akan dilakukan *merge* dari *branch develop* ke *branch master*.

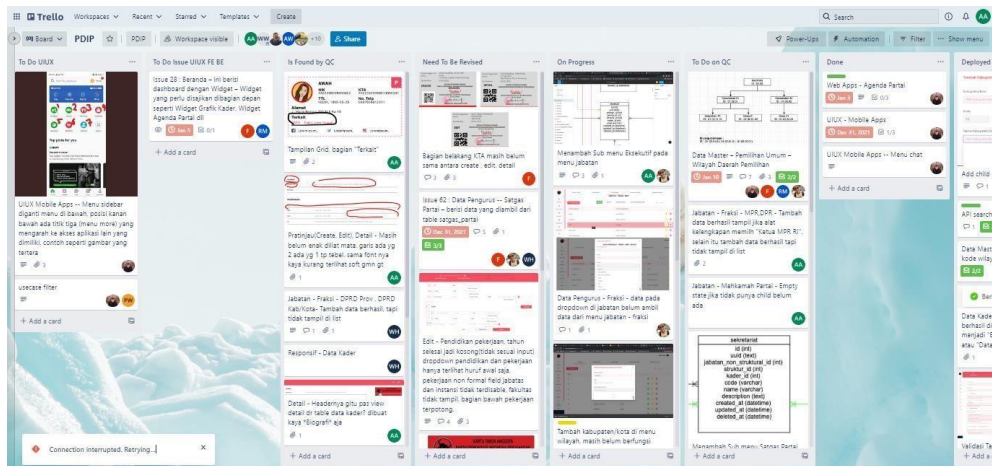
3.4 Pemantauan Proyek

Pemantauan atau pengendalian proyek dilakukan bersamaan dengan pelaksanaan proyek. Tujuannya adalah untuk mengontrol kemajuan proses pengembangan agar pengerjaan proyek berjalan sesuai dengan rencana yang telah ditetapkan. Proses pemantauan proyek pada pengembangan sistem informasi manajemen kader PDIP adalah dengan menggunakan aplikasi Trello dan GitLab.

a. Trello

Trello merupakan sebuah aplikasi berbasis web yang digunakan untuk memudahkan dalam mengelola atau melakukan manajemen tugas. Trello dapat membantu dalam pemantauan progress kerja setiap anggota tim yang terlibat. Dengan begitu, koordinasi tim menjadi lebih efektif. Dapat dilihat pada **Error! Reference source not found.** merupakan tampilan aplikasi Trello pada proyek SIMKADER PDIP. Bagian *To Do UIUX* dan *To Do FE BE* merupakan daftar *task* atau pekerjaan apa saja yang harus diselesaikan oleh tim UI/UX, FE, dan BE. Nantinya, masing-masing developer akan mendapatkan *task* yang harus dilakukan per harinya. Jika developer sedang mengerjakan *task*, maka *task* terkait akan dipindahkan ke bagian *On Progress*. Apabila aktivitas telah selesai dikerjakan, maka aktivitas berpindah ke bagian *To Do on QC*. Bagian ini merupakan pekerjaan yang dilakukan oleh tim QA/QC untuk mengecek dan menguji *task* yang telah diselesaikan oleh developer. Jika terdapat suatu kesalahan, maka *task* akan dipindahkan ke bagian *Need To Be Revised* untuk diperbaiki oleh developer. Sedangkan jika tidak terdapat kesalahan atau sesuatu yang tidak tepat, maka *task* akan dipindahkan ke bagian *Done*. *Task-task* yang berada di bagian *Done* akan dilanjutkan dengan melakukan *deployment* oleh tim DevOps. Apabila telah dilakukan *deployment*, maka *task* akan dipindahkan ke bagian *Deployed*. Sementara itu, bagian *Is Found by QC* merupakan sesuatu yang ditemukan oleh tim QA/QC diluar bagian *To Do on QC*. Bagian *Is Found by QC*

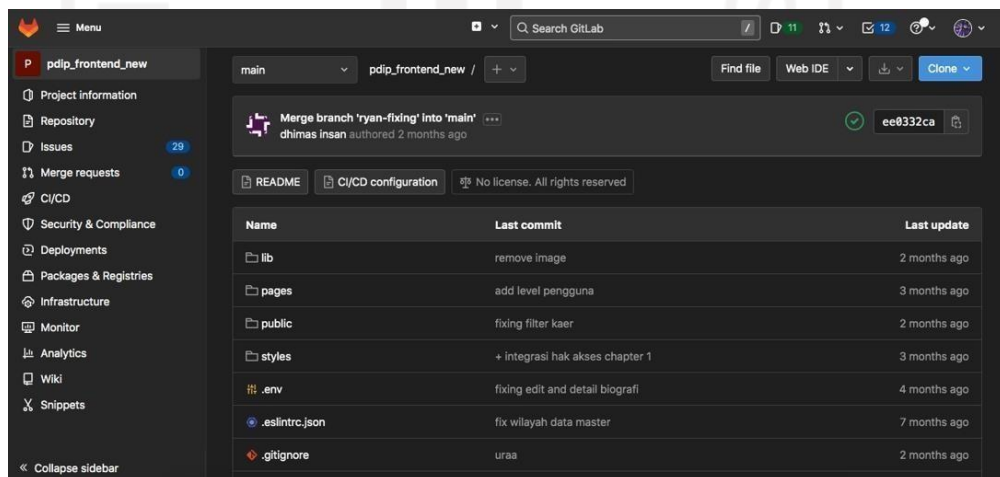
dilakukan untuk menjamin kualitas aplikasi tanpa memiliki error. Nantinya task pada bagian ini akan diperbaiki oleh developer.



Gambar 3.34 Tampilan Trello Proyek SIMKADER PDIP

b. GitLab

GitLab merupakan sebuah layanan git yang membantu dalam melakukan fungsi manajemen proyek, yaitu untuk pemantauan dan pengendalian proyek. Selain itu, GitLab digunakan untuk memudahkan dalam melakukan kolaborasi antar developer dengan menyimpan semua perubahan yang terjadi tanpa takut akan terjadinya bentrok. Gambar 3.35 merupakan tampilan GitLab untuk bagian front-end pada proyek SIMKADER PDIP.



Gambar 3.35 Tampilan GitLab Proyek SIMKADER PDIP

3.5 Penutupan Proyek

Penutupan proyek pengembangan sistem informasi manajemen kader partai PDIP belum dapat dilaksanakan dikarenakan masih adanya menu yang belum diselesaikan sementara masa

magang penulis telah berakhir. Menu yang belum terselesaikan tersebut yaitu menu Dashboard dan menu Pengaturan. Selain itu masih terdapat beberapa issue atau bug pada sejumlah menu. Menu dan bug yang belum terselesaikan tersebut akan dilanjutkan oleh salah satu karyawan front-end developer dari GeekGarden Software House.



BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

SIMKADER PDIP merupakan sebuah aplikasi sistem informasi manajemen berbasis website. Teori-teori yang diterapkan pada pengembangannya memiliki keterkaitan dengan beberapa mata kuliah, salah satunya yaitu mata kuliah Pengembangan Aplikasi Berbasis Website (PABW). Pada mata kuliah tersebut diajarkan beberapa penggunaan teknologi seperti HTML dan CSS, serta framework untuk membangun aplikasinya seperti Laravel dan Codeigniter. Pada penerapan di pengembangan aplikasi SIMKADER PDIP, hanya HTML dan CSS saja yang digunakan untuk membangun tampilan antarmukanya, sedangkan framework Laravel tidak digunakan. Hal tersebut karena aplikasi SIMKADER PDIP adalah aplikasi berbasis services dengan menggunakan API yang menghubungkan beberapa platform yang berbeda. Pengembangannya memisahkan antara frontend dan backend, serta dihubungkan dengan API. Frontend dengan menggunakan React.Js dan backend dengan menggunakan Express.Js. Sedangkan Laravel dan Codeigniter adalah framework dengan konsep MVC (Model-View-Controller) yang dapat membuat sebuah aplikasi dengan menggabungkan kode program frontend dan backend pada satu proyek yang sama sekaligus.

Terdapat pula mata kuliah Pengembangan Sistem Informasi (PSI) yang berkaitan dengan pembangunan aplikasi SIMKADER PDIP. Mata kuliah tersebut mengajarkan tentang logika query database, pembuatan BPMN, ERD, dan usecase diagram. Pada pelaksanaan pengembangan aplikasi SIMKADER PDIP, logika query database memang digunakan pada bagian backend untuk mengelola data pada database. Sedangkan BPMN, ERD, dan usecase diagram digunakan sebagai dasar dalam pembuatan desain database.

Selain dua mata kuliah tersebut, beberapa teori yang diberikan pada mata kuliah Manajemen Pengembangan Teknologi Informasi (MPTI) juga diterapkan pada proses manajemen pengembangan aplikasi SIMKADER PDIP. Sesuai dengan teori yang telah diberikan pada mata kuliah tersebut, bahwa tahapan manajemen proyek dimulai dari inisiasi proyek, perencanaan proyek, eksekusi proyek, pemantauan proyek, dan penutupan proyek. Tahapan-tahapan tersebut memang diterapkan dalam manajemen pengembangan proyek aplikasi SIMKADER PDIP seperti yang telah dijelaskan pada Bab III. Pada mata kuliah tersebut juga menjelaskan tentang pembuatan dokumen project charter dan dokumen project

planning management yang dimulai dari project integration management, project scope management, hingga project stakeholder management. Namun dalam praktiknya dokumen project charter dan dokumen project planning management tidak diterapkan, dokumen yang dibuat hanya berisi kebutuhan atau fitur-fitur yang diinginkan klien dan kebutuhan backend seperti usecase diagram, flowchart, atau ERD.

Metode pengembangan yang digunakan pada proyek aplikasi SIMKADER PDIP adalah metode agile, dimulai dari analisis, desain, implementasi, pengujian, hingga deploy. Pada pelaksanaannya, tahap desain, implementasi, pengujian, dan deploy dilakukan secara berulang. Setiap kali desain pada 1 fitur telah selesai dikerjakan dan sudah mendapat persetujuan dari klien, dilanjutkan ke tahap implementasi untuk menerjemahkan desain tersebut ke dalam bentuk kode oleh developer. Setiap developer mendapatkan task dari project manager yang dapat dilihat pada aplikasi Trello. Task/fitur yang telah selesai diimplementasikan oleh developer, kemudian akan dilakukan pengecekan dan pengujian oleh tim QA/QC. Agar dapat dilakukan pengujian oleh tim QA/QC, maka task/fitur tersebut harus berada di server staging atau server development dan bukan berada di local. Sehingga tahap deployment ke server staging akan dijalankan sebelum tahap pengujian dilakukan. Saat tahap pengujian dijalankan, tim QA/QC akan melakukan pengecekan untuk memastikan ada atau tidaknya bug. Apabila ditemukan bug, maka tim QA/QC akan memberikan bug tersebut pada aplikasi Trello dan memberikannya pada developer yang bersangkutan untuk kemudian diperbaiki. Dengan menggunakan metode agile, terkadang terjadi perubahan pada desain tampilan antarmuka, bahkan dapat terjadi perubahan alur aplikasi pada desain yang telah dieksekusi dalam pengkodean. Perubahan ini terjadi dikarenakan permintaan dari klien ataupun perubahan yang dilakukan oleh tim UI/UX sendiri untuk menyempurnakan aplikasi secara keseluruhan. Sehingga developer dituntut untuk dapat beradaptasi dan cepat tanggap terhadap segala bentuk perubahan.

Secara garis besar, tidak ada perbedaan antara teori metode agile yang sudah diuraikan pada bab 2 dengan pelaksanaannya. Mulai dari beberapa tahap yang dilakukan secara berulang hingga adanya perubahan-perubahan yang terjadi saat proses pengembangan. Namun, terdapat beberapa perbedaan yang penulis rasakan saat proses pelaksanaan dengan teori metode Agile, diantaranya sebagai berikut:

1. Secara teori, metode agile membutuhkan banyak keterlibatan dengan klien dalam proses pengembangan untuk meningkatkan kepuasan dan keberhasilan proyek. Namun, dalam pelaksanaannya, developer jarang melakukan interaksi dengan klien.

- Terkadang hanya project manager dan tim UI/UX yang melakukan diskusi dengan klien membahas kebutuhan sistem dan sebagian besar untuk mendapatkan persetujuan atas desain yang telah dibuat. Setelah desain mendapatkan persetujuan, kemudian desain tersebut akan langsung diimplementasikan oleh developer.
2. Jarangnya meeting harian atau mingguan yang membahas progress setiap developer ataupun permasalahan-permasalahan yang terjadi saat proses pengembangan. Proses pemantauan hanya melalui aplikasi Trello, apakah task/fitur yang diberikan sudah terselesaikan atau belum. Sedangkan menurut teori, metode agile membutuhkan evaluasi secara berkala agar pengembangan berjalan dengan efektif.
 3. Dalam metode agile, proses penentuan item atau task yang akan dikerjakan selama sprint sebaiknya dilakukan pembahasan bersama seluruh anggota tim. Namun, dalam praktiknya, penyusunan dan penentuan item/task yang akan dikerjakan selama sprint tidak dilakukan bersama developer, melainkan hanya dilakukan oleh project manager seorang.

Tabel 4.1 Perbandingan teori dengan pelaksanaan lapangan

No	Teori	Pelaksanaan lapangan
1	Metode agile merupakan metode pengembangan perangkat lunak yang dilakukan secara berulang-ulang dan cepat tanggap terhadap segala bentuk perubahan.	Dalam praktiknya, tahap desain, implementasi, pengujian, dan deployment dilakukan secara terus menerus setiap kali pengerjaan suatu task/fitur. Selain itu, terjadi beberapa perubahan pada desain karena permintaan klien, maka dari itu developer dituntut untuk beradaptasi terhadap segala perubahan yang diinginkan oleh klien.
2	Karena metode agile menekankan kepuasan dari klien, maka kerjasama antara tim klien dengan tim pengembang merupakan hal yang utama.	Hanya project manager dan tim UI/UX yang terlibat dengan klien untuk mendiskusikan tentang kebutuhan sistem dan untuk mendapatkan persetujuan atas desain yang telah dibuat. Sedangkan tim developer atau tim programmer jarang berinteraksi dengan klien.

3	Metode agile membutuhkan evaluasi secara berkala untuk memastikan keberhasilan proyek salah satunya dengan mengadakan meeting harian atau mingguan untuk membahas progress atau permasalahan yang terjadi.	Tidak adanya meeting harian atau mingguan yang membahas progress dan permasalahan-permasalahan para developer. Pemantauan hanya dilakukan dengan menggunakan aplikasi Trello.
4	Penentuan item yang akan dilakukan selama sprint sebaiknya dibahas bersama seluruh anggota tim developer.	Tidak adanya pembahasan bersama tim developer untuk menentukan item/task yang akan dilakukan selama sprint. Task diberikan oleh project manager melalui aplikasi Trello.

4.2 Pembelajaran Magang

Selama magang yang telah dilakukan selama 6 bulan dimulai dari 14 September 2021 hingga 14 Mei 2022, banyak pembelajaran yang diperoleh termasuk manfaat, kendala, hambatan, dan tantangan. Hal tersebut nantinya akan menjadi bekal yang sangat berguna ketika terjun kedalam dunia kerja. Adapun manfaat, kendala, hambatan, dan tantangan yang diperoleh selama kegiatan magang berjalan, antara lain:

Manfaat

a. Meningkatkan kemampuan dan keahlian terhadap teknologi

Dalam aktivitas magang yang telah dilakukan, banyak ilmu dan tools baru yang belum pernah penulis coba, seperti library react.js, JSX, Postman, dan Node.js. Pada pengembangan aplikasi SIMKADER PDIP, keempat tools tersebut merupakan tools utama yang dipakai pada sisi frontend, sehingga tim front-end dituntut untuk menguasai seluruh tools tersebut. Dengan terus mempelajari dan beradaptasi dengan tools tersebut, penulis yang berposisi sebagai front-end merasakan manfaat yang diperoleh mulai dari meningkatkan kemampuan dalam penulisan sintaks react.js dengan memanfaatkan JSX yang memungkinkan untuk mengkombinasikan HTML dengan JavaScript, mengetahui cara menggunakan komponen dan cara mengoper data antar komponen dengan menggunakan *props*, mengetahui *hooks* atau *function* apa saja yang disediakan react.js untuk mempermudah dalam pengembangan, hingga mengetahui *lifecycle method* atau siklus hidup yang terjadi pada react. Sedangkan pada tools postman, dapat mengetahui langkah-langkah penggunaan postman untuk

mendapatkan data seperti mengatur *environment* dan mengisi parameter apa saja yang diperlukan dalam melakukan *request*. Dalam pengkodean, tentunya tak lepas dari kesalahan/*error* dan *bug*, namun disitulah penulis dilatih untuk meningkatkan kemampuan dalam menyelesaikan masalah atau *problem solving*.

b. Mempelajari manajemen proyek dalam sebuah tim profesional

Pengembangan aplikasi tentu membutuhkan manajemen proyek yang baik agar aplikasi yang dihasilkan sesuai dengan kebutuhan, bermanfaat bagi klien maupun perusahaan, sesuai dengan anggaran, dan selesai tepat waktu. Selama proses pengembangan aplikasi SIMKADER PDIP, penulis menjadi mengerti tahap-tahap dalam melakukan manajemen proyek yang baik dan benar. Dimulai dari analisis terhadap semua kebutuhan klien yang pada akhirnya menghasilkan sebuah dokumen kebutuhan. Dari dokumen tersebut akan dilanjutkan dengan pembuatan desain dan implementasi. Pada tahap implementasi, task akan diberikan pada setiap programmer dengan memanfaatkan aplikasi Trello. Trello ini digunakan untuk membantu dalam mengatur dan mengawasi proses *development*. Setelah programmer memastikan bahwa *task* telah selesai, akan dilanjutkan dengan melakukan *push* ke GitLab yang kemudian tim DevOps akan melakukan *deploy* ke lingkungan *development*. Sesudah terdeploy, waktunya bagi tim QA/QC untuk menguji dan mengecek *task* yang sudah terdeploy tersebut. Jika dalam pengecekan terdapat hal-hal yang tidak sesuai, maka akan dilanjutkan dengan menulis *issue* pada Trello. *Issue* tersebut akan di *assign* ke developer yang mengerjakan *task* terkait untuk nantinya dilakukan penyelesaian *bug*.

c. Mengetahui aspek-aspek yang perlu diperhatikan dalam pengembangan aplikasi

Dalam mengembangkan sebuah aplikasi, tentu fungsionalitas dari aplikasi yang dihasilkan harus berjalan dengan baik. Namun tidak hanya itu saja, pada praktiknya banyak aspek yang perlu diperhatikan. Diantaranya:

- 1) Aspek keamanan terhadap ancaman yang mungkin datang. Aspek ini perlu diperhatikan dalam pengembangan aplikasi tak terkecuali pada pengembangan aplikasi SIMKADER PDIP. Pada aplikasi ini, penerapan aspek keamanan dilakukan saat proses penulisan kode oleh setiap developer baik itu front-end, back-end hingga devops. Contoh sederhana penerapan aspek keamanan pada sisi front-end yaitu adanya validasi pada setiap input atau masukan pengguna. Begitu pula pada sisi back-end juga melakukan

validasi saat data akan disimpan kedalam database. Sedangkan pada devops akan melakukan simulasi penyerangan DDoS pada aplikasi.

- 2) Aspek kualitas atau performa dari aplikasi yang dihasilkan. Dapat dilihat apabila load pada halaman lambat, yang artinya bahwa kualitas/performa aplikasi tersebut buruk. Begitu juga sebaliknya, apabila load pada halaman terasa cepat, maka kualitas/performa aplikasi tersebut dapat dikatakan baik. Pada praktik dalam aplikasi SIMKADER PDIP, pengecekan aspek ini dilakukan oleh tim QA/QC. Salah satu contoh tim QA/QC dalam menguji kualitas/performa aplikasi dengan melakukan stress test. Stress test yaitu pengujian untuk mengetahui kekuatan, daya tahan dan batas dari sebuah aplikasi. Sederhananya dilakukan dengan melakukan simulasi dengan sejumlah pengguna yang bersama-sama mengakses aplikasi. Dari *test* tersebut dapat diketahui batas performa dari sebuah aplikasi.
 - 3) Aspek kenyamanan pengguna. Tidak bisa dipungkiri bahwa tampilan antarmuka yang menarik dan mudah dipahami menjadi suatu hal yang penting dalam sebuah aplikasi. Selain itu, desain yang dibuat juga harus sesuai dengan kebutuhan pengguna. Desain yang jelas juga akan memudahkan tim front-end atau tim programmer lain dalam mewujudkan desain menjadisebuah kode. Maka dari itu, merupakan aspek yang penting untuk diperhatikan dari suatu pengembangan aplikasi.
 - 4) Aspek multi-device atau aplikasi yang tidak terbatas hanya bisa diakses di satu device saja, tetapi bisa diakses di beberapa device lainnya, seperti dari laptop maupun dari handphone. Hal ini tentu membuat pengguna menjadi lebih nyaman dalam menggunakannya karena memiliki pilihan untuk mengakses aplikasi tersebut.
- d. Mengembangkan kemampuan dalam bekerja sama

Pengembangan aplikasi sistem informasi manajemen kader PDIP tidak hanya dikerjakan oleh satu orang saja, namun juga melibatkan beberapa orang dengan posisi yang berbeda-beda, mulai dari project manager, front-end developer, back-end developer, UI/UX, dan QA/QC. Setiap anggota tim dituntut untuk saling bekerjasama, seperti membantu rekan dalam memecahkan suatu masalah dan berdiskusi dengan tim back-end terkait struktur data yang akan dikirim ke tim front-

end. Kerjasama yang baik akan membuat proses pengembangan menjadi lebih lancar dan hasil akhirnya pun akan sesuai dengan harapan.

Kendala, Hambatan, dan Tantangan

a. Beradaptasi dan mempelajari tools baru

Library react.js, JSX, dan postman merupakan suatu hal yang baru bagi penulis. Karena itulah, mempelajari, memahami, dan beradaptasi dengan beberapa tools tersebut merupakan tantangan tersendiri bagi penulis. Apalagi pada pengembangan aplikasi SIMKADER PDIP, tools yang telah disebutkan merupakan tools yang diterapkan untuk membangun tampilan antarmukanya, sehingga mau tidak mau penulis yang berposisi sebagai front-end developer dituntut untuk menguasai seluruh tools tersebut.

b. CORS

Cross Origin Resource Sharing atau disingkat menjadi CORS adalah suatu protokol untuk mengamankan sebuah request. Ketika tahap penulisan kode yang memiliki keterkaitan dengan aktivitas request, seringkali penulis menemui error yang disebabkan oleh CORS. Bahkan penulis pernah membutuhkan waktu 2 hari hanya untuk menyelesaikan CORS tersebut dengan berbagai macam cara, yang ternyata error tersebut berada pada sisi backend. Biasanya ini terjadi saat memiliki keterkaitan dengan melakukan request ke backend. Sudah dilakukan pengecekan dan pengujian di Postman pun hasilnya adalah sukses, tetapi ketika diimplementasikan terjadi error CORS. Sehingga penulis mengira ini terjadi pada sisi front-end. Maka dari itu CORS ini merupakan kendala dan hambatan yang menyebabkan saat penulis melakukan proses integrasi.

c. Integrasi dengan backend developer

Pengalaman bekerjasama dalam satu tim developer profesional dengan posisi yang berbeda-beda merupakan hal yang baru bagi penulis. Tentu saja, dalam proses implementasi atau pengkodean, tim front-end pasti akan berkomunikasi dan bekerjasama dengan tim back-end. Jembatan antara keduanya biasa disebut dengan API. Karena ini merupakan pengalaman pertama dalam melakukan integrasi dengan API tentu menjadi tantangan bagi penulis. Selain itu, penulis sebagai front-end developer seringkali menunggu API untuk mendapatkan data yang dibutuhkan, sehingga biasanya akan mengerjakan fitur yang lain sembari menunggu API dibuat. Baru setelah itu mengulang kembali ke fitur sebelumnya untuk mengerjakan

integrasi. Hal tersebut tidak efektif karena fitur yang sebelumnya masih belum selesai seratus persen dan menggantung. Karena pengembangan aplikasi yang baik yaitu menyelesaikan satu fitur sampai tuntas, sehingga tidak perlu untuk kembali mengulang menyelesaikan fitur tersebut.

d. Tidak adanya desain versi mobile

Website atau aplikasi yang dikembangkan memiliki tuntutan untuk memiliki kemampuan responsive atau bisa digunakan di desktop maupun mobile. Sementara desain yang dibuat hanya memiliki desain versi desktop. Hal ini membuat front-end developer tidak hanya menulis kode, namun juga memikirkan tampilan antarmuka yang nyaman dilihat pada versi mobile tanpa mengubah atau mempengaruhi tampilan versi desktop.

BAB V PENUTUP

5.1 Kesimpulan

Penerapan react.js pada sistem informasi manajemen kader PDIP telah berhasil dilakukan dengan menghasilkan sebuah aplikasi berbasis website yang bernama SIMKADER PDIP. Berdasarkan pengujian yang telah dilakukan menggunakan metode ATDD, dapat disimpulkan bahwa fungsionalitas dari aplikasi tersebut dapat berjalan dengan baik. Aplikasi tersebut dilatarbelakangi oleh kepercayaan partai PDIP terhadap GeekGarden untuk membangun sebuah aplikasi yang dapat membantu dalam mengelola kader PDIP. Pemilihan react.js dikarenakan react.js menyediakan berbagai fitur yang membantu dalam memberikan kemudahan bagi front-end developer untuk membuat tampilan antarmuka pengguna, salah satunya yaitu fitur *reusable component*. Fitur tersebut memungkinkan developer untuk menggunakan kembali kode yang telah dikembangkan sebelumnya dengan fungsi yang sama. Artinya, developer tidak perlu menulis kembali kode yang sama berulang kali. Metode yang digunakan dalam pengembangan aplikasi SIMKADER PDIP menggunakan metode agile yang terdiri dari beberapa tahapan, yaitu analisis, desain, implementasi, pengujian, dan deployment. Penggunaan metode Agile dikarenakan sifat fleksibilitasnya dalam setiap perubahan yang dapat terjadi selama proses pengembangan.

5.2 Saran

Adapun saran yang penulis berikan, guna memperbaiki dan meningkatkan kualitas pada sistem informasi manajemen kader PDIP adalah dengan menerapkan salah satu teknik dari *whitebox testing* yaitu unit testing agar memastikan bahwa setiap kode yang ditulis dapat berjalan dengan baik serta mengurangi resiko terjadinya error. Sehingga aplikasi yang dikembangkan memiliki kinerja yang lebih baik daripada aplikasi yang dalam pengujiannya hanya menerapkan pengujian secara manual. Selain itu, untuk lebih memperlancar proses pengembangan dalam suatu tim dengan posisi yang berbeda-beda khususnya antara front-end dengan back-end, ada baiknya untuk mendiskusikan di awal sebelum tahap implementasi terkait struktur data yang akan dikirimkan melalui API ke front-end, agar response data menjadi lebih rapi dan seragam. Sehingga hal tersebut akan memudahkan front-end developer dalam melakukan proses integrasi. Tidak hanya tentang struktur data, tetapi juga perlu untuk mendiskusikan tentang definisi dari kode error yang akan dikirimkan ke front-end apabila terjadi response error. Hal tersebut berfungsi untuk memudahkan ketika mencari kesalahan.

DAFTAR PUSTAKA

- Jimmy L. Gaol, Chr. (2008). *Sistem Informasi Manajemen Pemahaman dan Aplikasi*. Jakarta: PT Grasindo.
- Tung, K. (2018). *Developing a frontend application using ReactJS and Redux*. Laurea University of Applied Sciences.
- Satyal, A. (2020). *DESIGNING AND DEVELOPING A WEBSITE WITH REACTJS*. Oulu University of Applied Sciences.
- Bhalla, A., Garg, S., & Singh, P. (2020). *PRESENT DAY WEB-DEVELOPMENT USING REACTJS*. *International Research Journal of Engineering and Technology*, 07, 05.
- Bahtyar., Santoso, Nurudin., Amalia, Faizatul. (2019). *Implementasi Web Service Pada Sistem Rekam Medis Terpusat*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3, 1.
- Fajrin, Rachmat. (2017). *Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia*. *Jurnal Komputer Terapan*, 3, 1.
- O'Grady, Adam. (2018). *GitLab Quick Start Guide*. Birmingham: Packt Publishing Ltd.
- Nursaid, Faisal Fajar, Brata, Adam Hendra, Kharisma, Agi Putra. (2020). *Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri)*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(1), 46-55.

- Muhammad F. A. H. (2021). *PENGEMBANGAN FRONT END SISTEM INFORMASI MANAJEMEN DATA PATROLI PENCEGAHAN KARHUTLA DI WILAYAH SUMATERA*. Institut Pertanian Bogor.
- Alvin A (2019). *PENGEMBANGAN MODUL FRONTEND PADA SISTEM INFORMASI MANAJEMEN UNIT KANDANG HEWAN PERCOBAAN DI PUSAT STUDI BIOFARMAKA TROPIKA*. Institut Pertanian Bogor.
- Putra, I W A A., Kadyanan, I G A G A. (2019). *Pengembangan Front End Sistem Informasi Manajemen Absensi Pegawai Avatar Solutions*. *Jurnal Elektronik Ilmu Komputer Udayana*, 7(3).
- Putra, I W A A., Kadyanan, I G A G A. (2019). *Pengembangan Sistem Informasi Manajemen Tesis (Studi Kasus: Program Studi Magister Administrasi Bisnis Fakultas Ilmu Administrasi UB)*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(9), 3251-3257.
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). *Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap*. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(2), 206.
- Haryana, KM. (2019). *Penerapan Agile Development Methods Dengan Framework Scrum Pada Perancangan Perangkat Lunak Kehadiran Rapat Umum Berbasis Qr-Code*. *Jurnal Computech & Bisnis*, 13(2), 70–79.
- Pugh, Ken. (2010). *Lean-Agile Acceptance Test-Driven Development: Better Software Through Collaboration*. Addison-Wesley Professional.
- Sauvé, J. P., Abath Neto, O. L., & Cirne, W. (2006). *Easyaccept: a tool to easily create, run and drive development with automated acceptance tests*. In *Proceedings of the 2006 international workshop on Automation of software test* (pp. 111-117).
- Anggarwal, V., & Singh, M. (2014). *Acceptance Test Driven Development*. *Journal of Advanced Computing and Communication Technologies*. (ISSN: 2347 - 2804), 1-4.



LAMPIRAN

