

**PENGEMBANGAN FRONT END SISTEM INFORMASI
AKUNTANSI MENGGUNAKAN KERANGKA
KERJA VUE.JS**



Disusun Oleh:

N a m a : Andhika Rizky Aryasta

NIM : 18523073

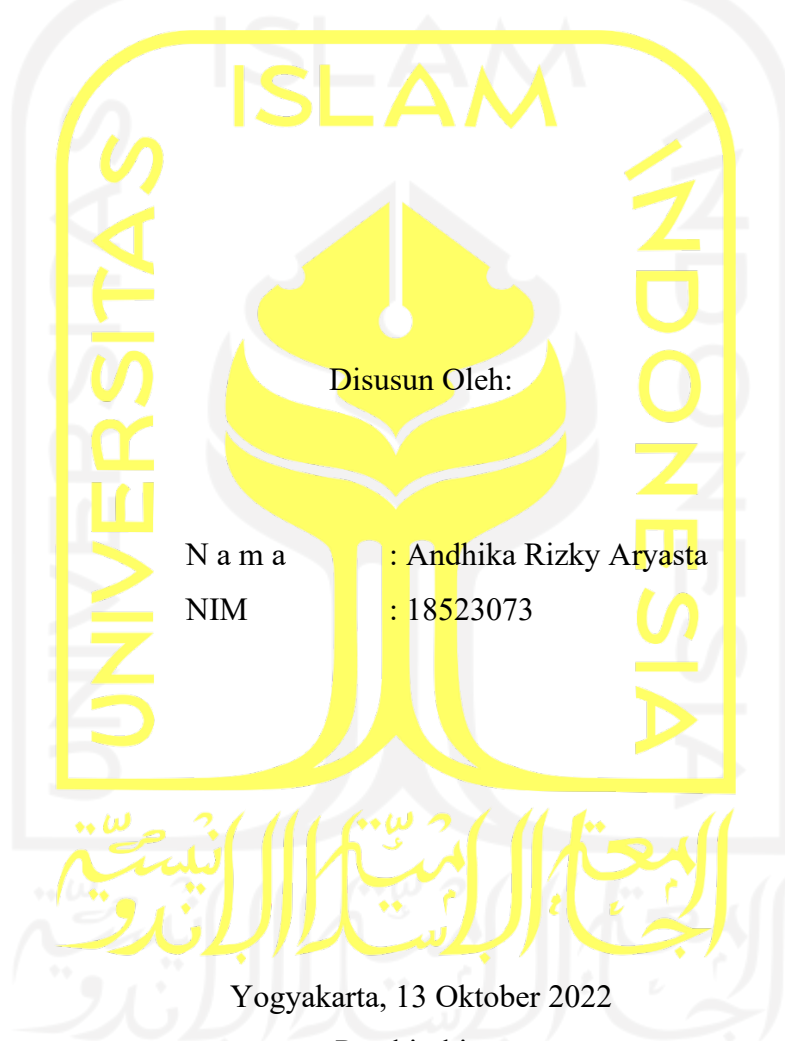
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2022

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN FRONT END SISTEM INFORMASI
AKUNTANSI MENGGUNAKAN KERANGKA
KERJA VUE.JS**

TUGAS AKHIR JALUR MAGANG



Disusun Oleh:

N a m a : Andhika Rizky Aryasta

NIM : 18523073

Yogyakarta, 13 Oktober 2022

Pembimbing,

(Andhik Budi Cahyono, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN FRONT END SISTEM INFORMASI
AKUNTANSI MENGGUNAKAN KERANGKA
KERJA VUE.JS**

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, 03 November 2022

Tim Penguji

Andhik Budi Cahyono, S.T., M.T.



Anggota 1

Rahadian Kurniawan, S.Kom., M.Kom.



Anggota 2

Sri Mulyati, S.Kom., M.Kom.



Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia




(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Andhika Rizky Aryasta

NIM : 18523073

Tugas akhir dengan judul:

PENGEMBANGAN FRONT END SISTEM INFORMASI AKUNTANSI MENGGUNAKAN KERANGKA KERJA VUE.JS

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, 13 Oktober 2022



(Andhika Rizky Aryasta)

HALAMAN PERSEMBAHAN

Alhamdulillah puji syukur kepada Allah SWT atas izin, rahmat, dan petunjuknya yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan perkuliahan hingga Tugas Akhir untuk mendapatkan gelar sarjana. Tugas akhir ini dipersembahkan untuk kedua orangtua dan almarhumah nenek penulis yang telah memberikan dukungan, semangat, motivasi, dan diiringi do'a selama masa perkuliahan. Penulis juga mengucapkan terima kepada teman-teman informatika UII dan semua pihak yang telah membantu dalam penulisan Tugas Akhir.



HALAMAN MOTO

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan”

QS. Al-Insyirah: 5

“Keberhasilan bukanlah milik orang yang pintar, tapi milik mereka yang senantiasa berusaha”

B.J. Habibie



KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufiq serta hidayat-Nya sehingga dapat menyelesaikan Tugas Akhir yang berjudul “Pengembangan *front end* Sistem Informasi Akuntansi menggunakan kerangka kerja Vue.js”. Laporan ini disusun sebagai persyaratan kelulusan jalur magang pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam penyusunan Tugas Akhir tidak terlepas dari bimbingan, dukungan, serta bantuan dari berbagai pihak. Maka dari itu, pada kesempatan ini diucapkan terima kasih kepada:

1. Kedua orang tua yang selalu memberikan do'a serta dukungan selama proses magang berlangsung hingga penyelesaian Tugas Akhir.
2. Bapak DThomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika.
3. Bapak Andhik Budi Cahyono, S.T., M.T., selaku dosen pembimbing yang telah membimbing dan memberikan arahan selama proses magang berlangsung hingga penyelesaian Tugas Akhir.
4. Bapak Dhimas Insan Saputra selaku pembimbing lapangan, seluruh staf dan karyawan Geekgarden Software House yang telah memberikan pengalaman, kepercayaan, serta arahnya selama magang berlangsung.
5. Seluruh dosen Program Studi Informatika Universitas Islam Indonesia yang memberikan ilmu pengetahuan selama perkuliahan.

Berdasarkan bantuan dari berbagai pihak dapat menyelesaikan Tugas Akhir dengan baik. Namun, selama proses penyusunan laporan masih terdapat banyak kekurangan dan jauh dari kata sempurna sehingga menerima segala saran dan masukan untuk menyempurnakan laporan ini. Semoga laporan ini dapat bermanfaat bagi pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 13 Oktober 2022



(Andhika Rizky Aryasta)

SARI

Sistem Informasi Akuntansi dikembangkan sebagai solusi untuk meningkatkan efisiensi aktivitas akuntansi pada perusahaan pelanggan Geekgarden, seperti: mengelola transaksi jurnal, menghasilkan sebuah laporan, dan memperhitungkan seluruh saldo pada setiap akun. Pengembangan Sistem Informasi Akuntansi dibagi menjadi dua proses utama, yaitu: pengembangan *front end* dan pengembangan *back end*. Kerangka kerja yang digunakan untuk pengembangan *front end* Sistem Informasi Akuntansi adalah Vue.js. Vue.js dipilih karena memiliki dukungan *library* tambahan untuk membangun tampilan antarmuka dan kemudahan dalam integrasi dengan *back end* menggunakan *Application Programming Interface* (API). Pada akhir magang telah diselesaikan *front end* dari Sistem Informasi Akuntansi yang telah berhasil diintegrasikan dengan API dan sudah lolos uji dengan metode *black box testing*. Vue.js sangat membantu proses pengembangan karena *library* yang disediakan mendukung kebutuhan pengembangan Sistem Informasi Akuntansi.

Kata kunci: Sistem Informasi Akuntansi, Vue.js, *front end*, *library*.

GLOSARIUM

<i>Back end</i>	pengembangan yang digunakan untuk mengolah basis data dan server.
<i>Bug</i>	kerusakan pada aplikasi yang tidak berjalan sesuai dengan harapan.
<i>Command prompt</i>	baris perintah penerjemah yang tersedia di sistem operasi Windows.
<i>Debug</i>	langkah untuk menelusuri kesalahan kode program.
<i>Front end</i>	tampilan antarmuka pada <i>website</i> yang berinteraksi langsung dengan pengguna.
<i>Library</i>	sekumpulan kode program yang digunakan untuk memenuhi kebutuhan sistem dan memudahkan pengembang dalam membangun aplikasi.
<i>Revamp</i>	perubahan yang bertujuan untuk memberikan pengalaman lebih baik kepada pengguna aplikasi dari sisi visual.
<i>Vue.js</i>	kerangka kerja yang progresif untuk membangun tampilan antarmuka pengguna.



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.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Ruang Lingkup	3
1.3 Tujuan.....	5
1.4 Manfaat.....	5
1.5 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA.....	7
2.1 Sistem Informasi Akuntansi	7
2.2 User Interface (UI)	7
2.3 User Experience (UX)	8
2.4 Vue.js.....	8
2.5 SPA (<i>Single Page Application</i>)	8
2.6 Web API.....	8
2.7 <i>Black Box Testing</i>	9
2.8 Tinjauan Pustaka	9
BAB III PELAKSANAAN MAGANG.....	12
3.1 Aktivitas Magang	12
3.1.1 Menganalisis studi kasus Digital Order	12
3.1.2 Mengembangkan <i>front end</i> modul <i>report</i> pada Aplikasi KlikPayroll.	13
3.1.3 Mengembangkan <i>front end</i> pada Aplikasi VONIX.	21
3.1.4 Mengembangkan <i>front end</i> pada Aplikasi Ayo BerAksi.....	27
3.1.5 Pengembangan <i>front end</i> pada Ticketing Sistem.....	33
3.1.6 Melakukan <i>bug fix</i> pada Sistem NES.....	35
3.1.7 Pengembangan <i>front end</i> pada Sistem Informasi Akuntansi	37
3.2 Manajemen Proyek.....	38
3.2.1 Pendefinisian Proyek.....	38
3.2.2 Perencanaan Proyek	38
3.2.3 Pelaksanaan Proyek.....	39
3.2.4 Pemantauan dan Pengendalian Proyek	73
3.2.5 Penutupan Proyek	75
BAB IV REFLEKSI PELAKSANAAN MAGANG.....	76
4.1 Relevansi Akademik.....	76
4.2 Pembelajaran Magang	77
4.2.1 Manfaat Magang	77
4.2.2 Kendala, Hambatan dan Tantangan selama magang	78

BAB V PENUTUP	80
5.1 Kesimpulan.....	80
5.2 Saran	80
5.2.1 Saran untuk Pengembangan Selanjutnya	80
5.2.2 Saran untuk Perusahaan	80
DAFTAR PUSTAKA	82
LAMPIRAN.....	83



DAFTAR TABEL

Tabel 2.1 Tabel perbandingan penelitian terkait.....	10
Tabel 3.1 Aktivitas Magang.....	12
Tabel 3.2 Aktivitas Sistem Informasi Akuntansi.....	39
Tabel 3.3 Pengujian pada halaman pengaturan Akun (COA).....	71
Tabel 3.4 Hasil pengujian setiap fitur.....	72



DAFTAR GAMBAR

Gambar 2.1 Halaman <i>dashboard</i> Sistem Informasi Akuntansi	7
Gambar 3.1 Dokumen Analisis Digital Order	13
Gambar 3.2 <i>Daily Scrum</i> proyek KlikPayroll.....	14
Gambar 3.3 Modul <i>Report</i> KlikPayroll	15
Gambar 3.4 Dokumen <i>report</i> Bank OCBC NISP	15
Gambar 3.5 Tampilan <i>front end report</i> Bank OCBC NISP.....	16
Gambar 3.6 Tampilan <i>front end</i> submenu BPJS <i>Report</i>	16
Gambar 3.7 Tampilan <i>front end</i> BPJS TK dan integrasi list NPP.....	17
Gambar 3.8 Tampilan <i>front end</i> BPJS TK dan <i>filter</i>	17
Gambar 3.9 Tampilan <i>front end</i> submenu <i>Tax Form Report</i>	18
Gambar 3.10 Tampilan <i>front end</i> dan <i>filter Tax Form</i> 1721-II	18
Gambar 3.11 Tampilan <i>front end</i> dan <i>filter Tax Form</i> 1721-T.....	19
Gambar 3.12 Tampilan <i>front end Tax Form</i> 1721-VI Tidak <i>Final</i>	19
Gambar 3.13 Tampilan <i>front end Tax Form</i> 1721-VI Tidak <i>Final</i> lanjutan	20
Gambar 3.14 Tampilan <i>front end Tax Form</i> 1721-VII <i>Final</i>	20
Gambar 3.15 Tampilan <i>front end Tax Form</i> 1721-VII <i>Final</i> lanjutan	21
Gambar 3.16 Tampilan <i>design</i> Figma fitur <i>Convert</i>	22
Gambar 3.17 Tampilan <i>front end home</i> VONIX	23
Gambar 3.18 Kode program JSON pada <i>content home</i> VONIX.....	23
Gambar 3.19 Tampilan <i>front end about</i> VONIX.....	24
Gambar 3.20 Kode program JSON pada <i>content about</i> VONIX.....	24
Gambar 3.21 Tampilan <i>front end sign in with email</i> VONIX	25
Gambar 3.22 Tampilan <i>front end sign in with phone number</i> VONIX	25
Gambar 3.23 Tampilan <i>front end</i> fitur OTP VONIX	26
Gambar 3.24 Kode OTP VONIX yang diterima melalui SMS	26
Gambar 3.25 Tampilan <i>front end convert responsive website</i>	27
Gambar 3.26 Tampilan <i>front end convert responsive mobile</i>	27
Gambar 3.27 Tampilan website Ayo BerAksi.....	28
Gambar 3.28 Tampilan <i>front end</i> Laporan Gratifikasi	28
Gambar 3.29 Tampilan <i>front end</i> Analisis Laporan Penyuaapan.....	29
Gambar 3.30 Tampilan <i>front end</i> Klasifikasi Laporan Pengaduan.....	29
Gambar 3.31 Tampilan <i>front end</i> Detail Laporan Pengaduan	30

Gambar 3.32 Tampilan <i>front end</i> ubah Klasifikasi Laporan Pengaduan.....	30
Gambar 3.33 Tampilan <i>front end</i> Analisis Laporan Gratifikasi	31
Gambar 3.34 Tampilan <i>front end</i> tambah Analisis Laporan Gratifikasi	31
Gambar 3.35 Tampilan <i>front end</i> kesimpulan Analisis Laporan Gratifikasi.....	31
Gambar 3.36 Tampilan <i>front end logbook</i> Laporan Penyusunan.....	32
Gambar 3.37 Tampilan <i>front end</i> ubah <i>logbook</i> Laporan Penyusunan	32
Gambar 3.38 Tampilan <i>front end logbook</i> Laporan Penyusunan	33
Gambar 3.39 Fitur <i>search</i> dan <i>filter</i> pada <i>Pool Ticket Level 1</i>	34
Gambar 3.40 Fitur <i>search</i> dan <i>filter</i> pada <i>Pool Ticket Level 2</i>	34
Gambar 3.41 Fitur <i>search</i> dan <i>filter</i> pada <i>Pool Ticket Level 3</i>	34
Gambar 3.42 Tampilan website Sistem NES.....	35
Gambar 3.43 Tampilan <i>front end</i> Area	36
Gambar 3.44 Tampilan <i>front end</i> Cabang.....	36
Gambar 3.45 Tampilan <i>front end</i> Harga Produk	37
Gambar 3.46 Desain <i>user interface</i> Akun (COA)	41
Gambar 3.47 Tampilan antarmuka Akun (COA)	41
Gambar 3.48 Tampilan antarmuka Akun (COA) lanjutan.....	42
Gambar 3.49 Desain <i>user interface</i> tambah Akun (COA).....	43
Gambar 3.50 Tampilan <i>pop up</i> tambah Akun (COA).....	43
Gambar 3.51 Tampilan Kategori Akun dinamis pada <i>pop up</i> tambah Akun (COA)	44
Gambar 3.52 Desain <i>user interface</i> ubah Akun (COA).....	45
Gambar 3.53 Tampilan <i>pop up</i> ubah Akun (COA).....	45
Gambar 3.54 Tampilan Kategori Akun dinamis pada <i>pop up</i> ubah Akun (COA)	46
Gambar 3.55 Desain <i>user interface</i> detail Akun (COA).....	47
Gambar 3.56 Tampilan <i>pop up</i> detail Akun (COA)	47
Gambar 3.57 Desain <i>user interface</i> halaman pengaturan tipe transaksi	48
Gambar 3.58 Tampilan antarmuka halaman pengaturan tipe transaksi	48
Gambar 3.59 Desain <i>user interface</i> tambah tipe transaksi	49
Gambar 3.60 Tampilan <i>pop up</i> tambah tipe transaksi	49
Gambar 3.61 Desain <i>user interface</i> ubah tipe transaksi	50
Gambar 3.62 Tampilan <i>pop up</i> ubah tipe transaksi	50
Gambar 3.63 Tampilan antarmuka Kode T	51
Gambar 3.64 Integrasi API Kode T	52
Gambar 3.65 Tampilan <i>pop up</i> tambah Kode T	52

Gambar 3.66 Tampilan <i>pop up</i> ubah Kode T	53
Gambar 3.67 Tampilan <i>pop up</i> detail Kode T	54
Gambar 3.68 Tampilan <i>pop up</i> delete.....	54
Gambar 3.69 Desain <i>user interface</i> Manajemen Periode	55
Gambar 3.70 Tampilan antarmuka pengaturan periode.....	56
Gambar 3.71 Tampilan antarmuka Manajemen Periode	57
Gambar 3.72 Desain <i>user interface</i> halaman transaksi jurnal	57
Gambar 3.73 Tampilan antarmuka halaman transaksi jurnal	58
Gambar 3.74 Tampilan antarmuka halaman transaksi jurnal lanjutan	58
Gambar 3.75 Integrasi API unggah jurnal	60
Gambar 3.76 Desain <i>user interface</i> tambah transaksi jurnal	61
Gambar 3.77 Implementasi <i>library</i> vue2-datepicker.....	62
Gambar 3.78 Implementasi <i>library</i> vue-numeric.....	63
Gambar 3.79 Tampilan antarmuka tambah transaksi jurnal	63
Gambar 3.80 Tampilan dropdown Kode T pada tambah transaksi jurnal	63
Gambar 3.81 Desain <i>user interface</i> ubah transaksi jurnal	64
Gambar 3.82 Tampilan antarmuka ubah transaksi jurnal	65
Gambar 3.83 Tampilan dropdown Kode T pada ubah transaksi jurnal	65
Gambar 3.84 Desain <i>user interface</i> detail transaksi jurnal	66
Gambar 3.85 Tampilan antarmuka detail transaksi jurnal	67
Gambar 3.86 Tampilan antarmuka <i>Account Listing</i>	68
Gambar 3.87 Tampilan <i>dropdown</i> Akun (COA).....	69
Gambar 3.88 Tampilan antarmuka <i>Journal Listing</i>	69
Gambar 3.89 Tampilan antarmuka <i>Trial Balance</i>	70
Gambar 3.90 Tampilan <i>repository</i> GitLab Sistem Informasi Akuntansi.....	74
Gambar 3.91 Tampilan grup telegram Sistem Informasi Akuntansi	74

BAB I PENDAHULUAN

1.1 Latar Belakang

Geekgarden *Software House* adalah suatu perusahaan teknologi informasi di Yogyakarta yang berfokus dalam jasa pengembangan perangkat lunak dan telah menangani berbagai macam proyek berbasis *website* maupun *mobile*. Saat ini, Geekgarden tengah mengembangkan beberapa proyek untuk membantu menangani permasalahan yang dimiliki pelanggan Geekgarden. Beberapa *software* yang sedang dikembangkan pada saat magang adalah Klik Payroll, VONIX, Ticketing System, Ayo BerAksi, Sistem Nes, dan Sistem Informasi Akuntansi.

Klik Payroll adalah suatu proyek yang digunakan untuk mengelola data karyawan yang terdapat pada perusahaan pelanggan. Pada proyek Klik Payroll belum memiliki modul *report* yang digunakan untuk melakukan perhitungan gaji karyawan dan menghasilkan sebuah laporan berdasarkan jenis laporan pada perusahaan pelanggan. Perusahaan pelanggan bekerja sama dengan Geekgarden untuk mengembangkan *front end* pada modul *report* proyek tersebut.

VONIX sebuah perusahaan yang bergerak di bidang transaksi *cryptocurrency* yang berlokasi di Jakarta. VONIX ingin mengembangkan sebuah *website* untuk memperkenalkan produk/layanan yang dimiliki dan melakukan berbagai aktivitas transaksi *cryptocurrency*. Perusahaan VONIX bekerja sama dengan Geekgarden untuk mengembangkan dari sisi *front end* dan *back end*.

Ayo BerAksi merupakan sebuah aplikasi yang tersedia pada aplikasi berbasis *website* dan aplikasi berbasis *mobile* untuk perangkat Android. Aplikasi ini dikembangkan sebagai membantu proses pelaporan penanganan kasus korupsi, berupa laporan penyuapan dan laporan gratifikasi yang terdapat di Balai Besar Karantina Pertanian Belawan.

Ticketing System merupakan sebuah aplikasi berbasis *website* yang akan memberikan layanan *support ticket* ketika terdapat *request* pada sistem. Aplikasi ini dikembangkan untuk membantu dalam memberikan pelayanan *support ticket* di perusahaan pelanggan sehingga seluruh *request* yang terdapat pada sistem dapat dilihat secara *real time*.

Sistem NES adalah suatu proyek berbasis *website* yang akan digunakan pada perusahaan makanan di Yogyakarta. Selama ini, proses manajemen data masih dilakukan menggunakan metode manajemen secara manual sehingga menjadi kurang efisien. Aplikasi ini

dikembangkan untuk memudahkan dalam manajemen master data di perusahaan pelanggan sehingga menjadi lebih terstruktur dan efisien.

Proyek Sistem Informasi Akuntansi memiliki dua proses utama, yaitu: pengembangan dari sisi *front end* dan pengembangan dari sisi *back end*. Pengembangan *front end* menggunakan kerangka kerja Vue.js dengan berbagai *library* tambahan untuk membangun tampilan antarmuka yang lebih menarik dan sesuai kebutuhan pengembangan (Herdinata Putra, Pramana, & Srinadi, 2019). Pengembangan *back end* untuk mengelola penyimpanan data dan merancang *Application Programming Interface* (API) dengan HTTP *method* GET, PUT, DELETE, POST. API digunakan untuk menjembatani proses transfer data antara *front end* dengan *back end*.

Proses pengembangan *front end* dilakukan berdasarkan hasil rancangan dari *user interface/user experience designer* (UI/UX Designer) dalam bentuk prototipe yang telah melalui tahap pengujian dengan pihak pelanggan. Rancangan desain kemudian dibagikan kepada pengembang *front end* yang dapat diakses melalui aplikasi Figma untuk diimplementasikan pada *website*. Kerangka kerja Vue.js memiliki keunggulan pada kinerja yang cepat, tersedia dokumentasi yang lengkap, dukungan berbagai *library*, kemudahan dalam proses integrasi dengan *back end* melalui API, serta mendukung teknologi *Single Page Application* (SPA). Penerapan SPA bertujuan untuk meningkatkan kinerja menjadi lebih baik, mengurangi beban pada server saat terjadi permintaan data dari pengguna, dan meminimalisir penggunaan sumber daya pada server (Belluano, 2018). Secara umum, SPA digunakan pada aplikasi *dashboard* yang menyediakan seluruh informasi mengenai segala aktivitas perusahaan. Proses perpindahan antar halaman di *handle* dengan *routing* sehingga *website* menampilkan data tanpa adanya *refresh/reload page*, tidak memakan banyak waktu, serta memberikan pengalaman pengguna menjadi lebih baik (Tampang, Kadang, & Arrang, 2020).

Pada laporan ini akan dijelaskan mengenai pengembangan *front end* Sistem Informasi Akuntansi berbasis *website* menggunakan kerangka kerja Vue.js. Penggunaan kerangka kerja Vue.js dikarenakan mendukung *Single Page Application* (SPA) yang menggunakan satu tampilan pada halaman *website* sehingga membuat kinerja menjadi lebih baik. Selain itu, Vue.js memiliki dukungan *library* tambahan yang mampu memberikan kemudahan dalam proses pengembangan sesuai kebutuhan sistem dan integrasi API dengan *back end* sehingga dapat menjalankan seluruh fungsionalitas sesuai dengan yang diharapkan.

1.2 Ruang Lingkup

Sebelumnya telah dilaksanakan magang di Geekgarden *Software House* yang berlangsung selama enam bulan dimulai dari September 2021 – Maret 2022 dan bergabung dengan divisi *front-end web developer*. Selama kegiatan magang berlangsung telah berkontribusi ke dalam enam pengembangan proyek berbasis *website* yang terdapat di Geekgarden yaitu, sebagai berikut:

- a. Mengembangkan *front end* modul report pada Aplikasi KlikPayroll.

KlikPayroll merupakan aplikasi berbasis *website* untuk memberikan layanan penggajian pegawai, mengelola data pegawai, perhitungan pajak pegawai, serta menghasilkan *report* sesuai dengan periode tertentu berdasarkan kebutuhan pengguna perusahaan yang berkepentingan. Pada proyek KlikPayroll telah berkontribusi selama 1,5 bulan menjadi *front-end web developer* yang dikembangkan dengan kerangka kerja Vue.js. Selain itu, memiliki batasan aktivitas dan tanggung jawab untuk mengembangkan *front end* pada modul *report*.

- b. Mengembangkan *front end* pada Aplikasi VONIX.

VONIX merupakan aplikasi berbasis *website* yang menyediakan berbagai informasi seputar perusahaan VONIX, melakukan transaksi *cryptocurrency*, dan *exchange cryptocurrency*. Pada proyek VONIX telah berkontribusi selama 1 bulan menjadi *front-end web developer* yang dikembangkan dengan kerangka kerja Vue.js. Selain itu, memiliki batasan aktivitas dan tanggung jawab untuk mengerjakan *bug fix*, melakukan integrasi API pada beberapa fitur, serta mengembangkan *front end* pada *Convert Cryptocurrency*.

- c. Mengembangkan *front end* pada Aplikasi Ayo BerAksi.

Ayo BerAksi merupakan sebuah aplikasi berbasis *website* untuk melakukan pelaporan terkait kasus penyuaipan, dan gratifikasi. Pada proyek Ayo BerAksi telah berkontribusi selama 2 pekan menjadi *front-end web developer* menggunakan kerangka kerja Laravel untuk pengembangannya. Selain itu, memiliki batasan aktivitas dan bertanggung jawab untuk melakukan penambahan fitur yang tidak terdapat pada *website* sesuai dengan dokumen sistem tetapi lebih fokus untuk melakukan *redesign user interface* dan membangun seluruh tampilan *front end* pada *website* Ayo BerAksi. Namun, pada pengerjaan proyek Ayo BerAksi dilakukan secara bersamaan dengan proyek sistem NES sehingga membuat proses pengerjaan membutuhkan waktu lebih lama.

d. Mengembangkan *front end* pada Ticketing System.

Ticketing System merupakan aplikasi berbasis *website* yang berfungsi sebagai *support ticket* ketika terdapat *request* pada sistem. Pada proyek Ticketing System telah berkontribusi selama 2 hari menjadi *front-end web developer* yang dikembangkan dengan kerangka kerja Vue.js. Selama bergabung dengan proyek Ticketing System terdapat batasan aktivitas dan tanggung jawab untuk menambahkan fitur pada *pool ticket* sekaligus melakukan integrasi API dengan *back end*. Pengerjaan proyek Ticketing System dilakukan bersamaan dengan proyek Ayo BerAksi.

e. Melakukan *bug fix* pada Sistem NES.

Sistem NES merupakan aplikasi berbasis *website* untuk melakukan pendataan sekaligus pencatatan yang diperlukan oleh perusahaan berbentuk halaman *dashboard* yang terdiri dari lima modul diantaranya, yaitu modul *core*, modul *inventory*, modul *back office*, modul *finance*, dan modul *human resource*. Pada proyek sistem NES telah berkontribusi selama 2 pekan menjadi *full-stack web developer* menggunakan kerangka kerja Laravel untuk pengembangannya. Selama bergabung dengan proyek sistem NES memiliki batasan aktivitas dan bertanggung jawab mengerjakan *bug fix* pada modul *core*, modul *back office*, serta melakukan pergantian beberapa tampilan antarmuka.

f. Mengembangkan *front end* pada Sistem Informasi Akuntansi.

Sistem Informasi Akuntansi merupakan aplikasi berbasis *website* untuk meningkatkan efektivitas dalam melakukan pembukuan transaksi akuntansi. Sistem tersebut memiliki 21 fitur didalamnya yang memberikan kemudahan pelanggan perusahaan Geekgarden untuk mengelola transaksi dan menghasilkan laporan transaksi secara *real time*. Sistem Informasi Akuntansi telah dikembangkan sebelum tetapi pengembang yang terlibat telah *resign* dari Geekgarden sehingga dilanjutkan oleh pemegang. Pada proyek Sistem Informasi Akuntansi telah berkontribusi selama 2 bulan menjadi *front-end web developer* menggunakan kerangka kerja Vue.js. Proyek Sistem Informasi Akuntansi dipilih berdasarkan diskusi yang dilakukan dengan *Project Manager (PM)* sehingga laporan ini nantinya akan fokus untuk membahas pekerjaan selama berkontribusi dalam proyek Sistem Informasi Akuntansi. Batasan aktivitas dan tanggung jawab selama bergabung di proyek Sistem Informasi Akuntansi telah selesai mengerjakan sembilan fitur dari 21 fitur diantaranya, sebagai berikut:

1. Melakukan *revamp front end* pada menu pengaturan Akun (COA).

2. Melakukan *revamp front end* pada menu pengaturan Tipe Transaksi.
3. Mengembangkan *front end* pada menu pengaturan Kode T.
4. Melakukan *revamp front end* pada menu pengaturan Periode
5. Melakukan *revamp front end* pada menu transaksi Manajemen Periode.
6. Melakukan *revamp front end* pada menu transaksi Jurnal.
7. Mengembangkan *front end* pada menu laporan *Account Listing*.
8. Mengembangkan *front end* pada menu laporan *Journal Listing*.
9. Mengembangkan *front end* pada menu laporan *Trial Balance*.

Pada proyek Sistem Informasi Akuntansi terdiri dari dua orang *front-end web developer*, satu orang *quality assurance*, dan satu orang *back-end developer*. Akan tetapi, selama kurang lebih tiga minggu tidak terdapat *back-end developer* sehingga menghambat proses integrasi API yang dilakukan pada *front end*.

1.3 Tujuan

Tujuan dari pelaksanaan magang untuk mengembangkan *front end* Sistem Informasi Akuntansi menggunakan kerangka kerja Vue.js dengan banyaknya dukungan *library* tambahan yang mempermudah proses pengembangan.

1.4 Manfaat

Manfaat yang diperoleh dari pengembangan *front end* Sistem Informasi Akuntansi menggunakan kerangka kerja Vue.js diantaranya, yaitu:

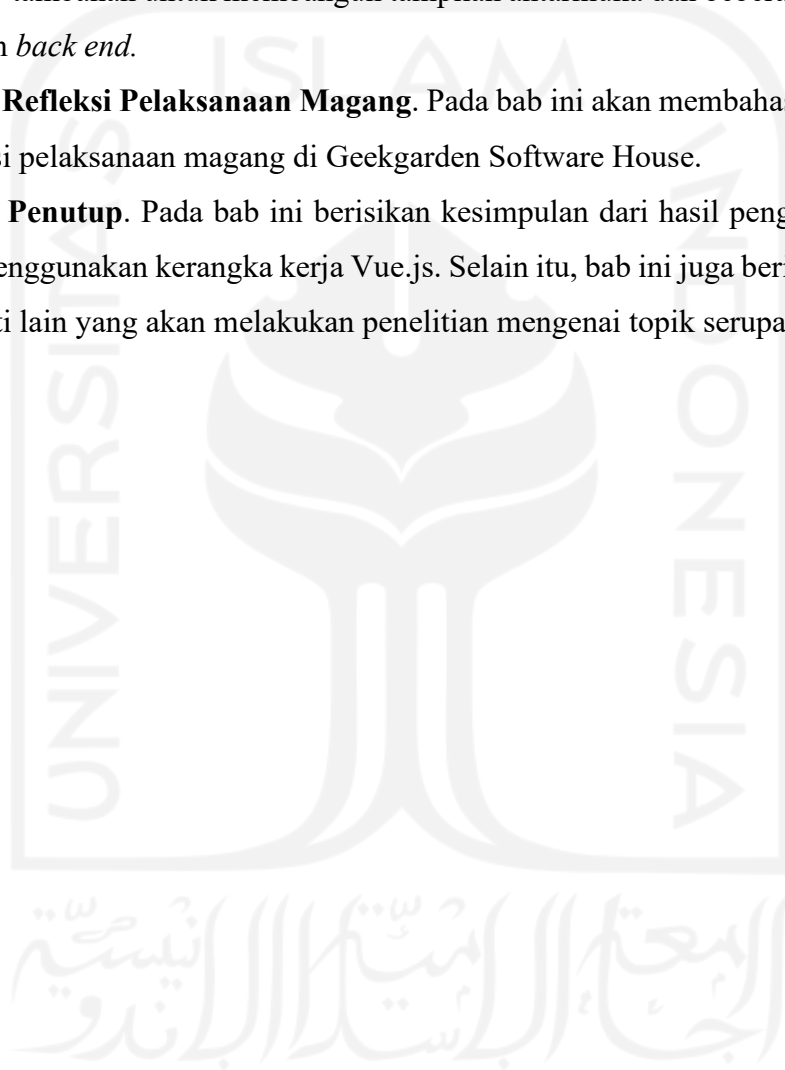
- a. Mengetahui proses pengambilan data atau integrasi API dengan *back end*.
- b. Mengetahui penggunaan *library* tambahan untuk membuat tampilan antarmuka pada *website*.
- c. Mengetahui penggunaan kerangka kerja Vue.js untuk pengembangan *front end* pada *website*.

1.5 Sistematika Penulisan

Sistematika penulisan disusun untuk memberikan gambaran umum terhadap keseluruhan isi laporan. Susunan sistematika penulisan pada laporan sebagai berikut:

- a. **Bab 1 Pendahuluan.** Pada bab ini akan memberikan pembahasan mengenai latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan.

- b. **Bab 2 Landasan Teori dan Tinjauan Pustaka.** Pada bab ini berisikan mengenai landasan teori yang berkaitan dengan topik laporan dan hasil ringkasan dari penelitian-penelitian sebelumnya.
- c. **Bab 3 Pelaksanaan Magang.** Pada bab ini memberikan gambaran aktivitas selama magang dan penjelasan mengenai pengembangan *front end* menggunakan kerangka kerja Vue.js pada *website*. Di dalam bab ini juga membahas mengenai penggunaan *library* tambahan untuk membangun tampilan antarmuka dan beberapa integrasi API dengan *back end*.
- d. **Bab 4 Refleksi Pelaksanaan Magang.** Pada bab ini akan membahas mengenai hasil refleksi pelaksanaan magang di Geekgarden Software House.
- e. **Bab 5 Penutup.** Pada bab ini berisikan kesimpulan dari hasil pengembangan *front end* menggunakan kerangka kerja Vue.js. Selain itu, bab ini juga berisikan saran bagi peneliti lain yang akan melakukan penelitian mengenai topik serupa.



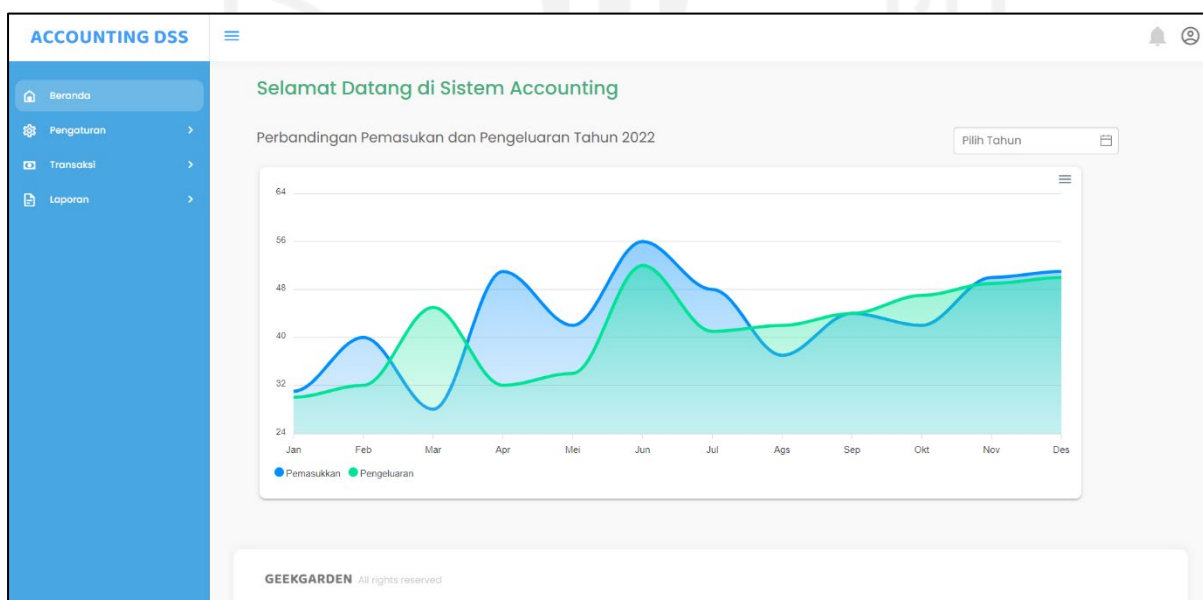
BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Sistem Informasi Akuntansi

Sistem Informasi Akuntansi merupakan sistem berbasis *website* sebagai solusi untuk meningkatkan efisiensi aktivitas akuntansi pada perusahaan pelanggan Geekgarden, seperti: mengelola transaksi jurnal, menghasilkan sebuah laporan, dan memperhitungkan seluruh saldo Akun (COA). Sistem ini digunakan untuk mengelola keuangan dana pensiun karyawan perusahaan pelanggan Geekgarden sehingga dapat terdistribusi dengan baik. Pengguna sistem terdiri dari beberapa divisi, yaitu: IT, Keuangan, Manager, Direktur dan jajarannya.

Pengembangan *front end* Sistem Informasi Akuntansi menggunakan kerangka kerja Vue.js. Sistem Informasi Akuntansi sendiri memiliki berbagai fitur yang mendukung kegiatan akuntansi, seperti: pengaturan yang digunakan sebagai dasar penggunaan sistem akuntansi, fitur transaksi yang mengolah proses akuntansi, dan fitur laporan untuk menampilkan laporan yang dibuat secara otomatis melalui sistem. Berikut merupakan tampilan halaman *dashboard* Sistem Informasi Akuntansi yang terdapat pada Gambar 2.1.



Gambar 2.1 Halaman *dashboard* Sistem Informasi Akuntansi

2.2 User Interface (UI)

User Interface adalah layout atau tata letak yang dirancang untuk memberikan gambaran secara visual oleh *end user* atau pengguna ketika berinteraksi secara langsung dengan website.

UI terdiri dari segi pemilihan warna, bentuk elemen tombol, jenis *font* yang digunakan, penggunaan gambar, ikon, *input field* dan karakteristik aplikasi (Muhyidin, Sulhan, & Sevtiana, 2020).

2.3 User Experience (UX)

User Experience adalah semua aspek yang berkaitan dengan pengalaman pengguna ketika menggunakan sebuah layanan atau produk yang interaktif. UX merupakan seluruh hal yang berkaitan dengan alur kerja produk mudah dipahami, dirasakan serta dapat mencapai tujuannya dalam menggunakan aplikasi interaktif (Razi, Mutiaz, & Setiawan, 2018).

2.4 Vue.js

Vue.js adalah sebuah kerangka kerja JavaScript progresif berbasis *open-source* untuk membangun *user interface* atau tampilan antarmuka pada aplikasi berbasis *website* dengan menerapkan konsep MVC (*model-view-controller*). Vue.js diciptakan oleh Evan You pada tahun 2014. Selain itu, Vue.js dibangun menggunakan HTML, CSS, JavaScript, dan komponen yang dapat digunakan kembali (Gani, 2018).

Pada konsep MVC, Vue.js hanya merepresentasikan layer *view* yang berfokus untuk membuat *user interface* tanpa memikirkan proses dalam mengelola dan memanipulasi data. Selain itu, Vue.js dipilih karena memiliki banyak dukungan dengan kontributor, memberikan kemudahan dalam proses integrasi dengan berbagai *library*, serta mendukung untuk pengembangan SPA (*Single Page Application*).

2.5 SPA (Single Page Application)

SPA merupakan suatu aplikasi yang berjalan pada *website* menggunakan satu halaman secara dinamis untuk menangani seluruh aktivitas tanpa perlu melakukan *refresh/reload page*. Adanya interaksi yang dilakukan pada sistem pengguna tetap berada pada halaman yang sama setiap mengirimkan *request* ke *server* (Belluano, 2018). Penerapan SPA bertujuan untuk meningkatkan kinerja proses memuat data dengan waktu yang singkat, meminimalisir penggunaan sumber daya *server*.

2.6 Web API

API atau *Application Programming Interface* merupakan sebuah sistem perangkat lunak untuk berkomunikasi dengan jenis perangkat lunak berbeda yang dioperasikan melalui koneksi

jaringan internet. Komunikasi antara *front end* dengan *back end* dilakukan menggunakan HTTP dengan serialisasi XML untuk melakukan pertukaran data dengan berbagai macam jenis *client* seperti aplikasi berbasis web, aplikasi mobile, dan aplikasi desktop (Rakhmah & Purwoko, 2021).

HTTP *methods* yang digunakan adalah POST, GET, PUT, dan DELETE. Pengembangan *front end* memanfaatkan endpoint API untuk memudahkan proses integrasi yang dilakukan pada seluruh fitur sehingga *front end* dapat berjalan sesuai dengan fungsionalitas yang diharapkan.

2.7 Black Box Testing

Black box testing adalah metode pengujian untuk memastikan seluruh fungsionalitas melalui *input* dan *output* yang dihasilkan pada sistem dapat berjalan sesuai dengan harapan. Proses pengujian menggunakan metode ini dilakukan tanpa memperhatikan kode program didalamnya untuk mengidentifikasi adanya kecacatan pada sistem ketika pengembangan. (Cholifah, Yulianingsih, & Sagita, 2018)

2.8 Tinjauan Pustaka

Pada beberapa penelitian yang telah dilakukan sebelumnya terkait pengembangan *front end* berbasis *website* menggunakan kerangka kerja Vue.js. Terdapat penelitian yang berfokus untuk membahas pengembangan SPA (*Single Page Application*) pada Sistem Informasi Akademik (Belluano, 2018). Hasil dari penelitian tersebut menjelaskan bahwa SPA membantu dalam mengaplikasikan Sistem Informasi Akademik dengan tampilan *user interface* yang mudah digunakan, *user friendly* sesuai dengan kebutuhan dan keinginan pengguna, menyajikan informasi berbentuk teks dan grafik sinkron, pengelolaan informasi secara *realtime* yang membuat komunikasi data lebih cepat, serta tidak memakan banyak sumber daya.

Dalam penelitian (Herdinata Putra et al., 2019), membahas mengenai membangun Sistem Manajemen Arsip berbasis *website* menggunakan kerangka kerja Laravel dan kerangka kerja Vue.js. Pada penelitian ini, disebutkan bahwa penggunaan kerangka kerja Laravel untuk memanipulasi, dan mengelola basis data sedangkan kerangka kerja Vue.js digunakan untuk membangun tampilan antarmuka menjadi interaktif karena memiliki kemampuan tanpa harus *refresh page* setiap berpindah halaman.

Penelitian lain membahas mengenai perbandingan dalam pengembangan *front end* menggunakan blade template dengan kerangka kerja Laravel dan Vue.js (Chastro &

Darmawan, 2020). Dalam kasus tersebut, peneliti melakukan perbandingan kedua kerangka kerja tersebut berdasarkan performa, kecepatan *script*, *routing*, dan tampilan. Hasil dari penelitian tersebut menyebutkan bahwa kerangka kerja Vue.js memiliki performa yang stabil untuk pengembangan aplikasi kompleks dibandingkan aplikasi sederhana sedangkan blade template dengan kerangka kerja Laravel lebih cocok digunakan untuk pengembangan aplikasi sederhana karena pada pengembangan aplikasi yang lebih kompleks terjadi penurunan performa secara drastis.

(Arhandi, 2016) melakukan penelitian terkait Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode Back End Dan Front End. Permasalahan pada penelitian tersebut proses pelayanan dalam pembuatan ijin praktek dan ijin kerja dilakukan secara manual sehingga membutuhkan lebih banyak waktu. Berdasarkan masalah tersebut, akan dikembangkan sebuah sistem untuk memudahkan dalam melayani pembuatan surat ijin kerja dan surat ijin praktek untuk tenaga kesehatan melalui sistem sehingga dapat lebih cepat dan efisien. Pengembangan *Front End* dilakukan menggunakan kerangka kerja Angular.js sedangkan *Back End* menggunakan kerangka kerja Laravel yang memanfaatkan REST API. Penelitian ini menerapkan metode System Development Life Cycle (SDLC) menggunakan Waterfall dengan beberapa tahapan yaitu: analisis sistem, perancangan sistem, implementasi perancangan, dan pengujian.

Adapun perbandingan dengan penelitian terdahulu apabila disajikan dalam bentuk tabel dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel perbandingan penelitian terkait

No	Peneliti (Tahun)	Judul	Teknologi	Pengujian
1.	Poetri Lestari Lokapitasari Belluano (2018)	Pengembangan Single Page Application pada Sistem Informasi Akademik	<i>Framework</i> Laravel dan Vue.js	<i>Black box testing</i>
2.	I Ketut Aditya Herdinata Putra, Dian Pramana, Ni Luh Putri Srinadi (2019)	Sistem Manajemen Arsip Menggunakan <i>Framework</i> Laravel dan Vue.js	<i>Framework</i> Laravel dan Vue.js	<i>Black box testing</i>

3.	Christian Chastro, Erico Darmawan (2020)	Perbandingan Pengembangan Front End Menggunakan Blade Template dan Vue Js	<i>Framework</i> Laravel dan Vue.js	Chrome DevTools
4.	Putra Prima Arhandi (2016)	Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode <i>Back End</i> dan <i>Front End</i>	<i>Framework</i> Laravel dan Angular.js	Tidak disebutkan secara spesifik

BAB III

PELAKSANAAN MAGANG

3.1 Aktivitas Magang

Kegiatan magang yang telah dilaksanakan selama enam bulan di Geekgarden *Software House* dimulai dari pertengahan September 2021 hingga pertengahan Maret 2022. Adapun aktivitas yang telah dilakukan selama magang di Geekgarden dapat dilihat pada Tabel 3.1.

Tabel 3.1 Aktivitas Magang

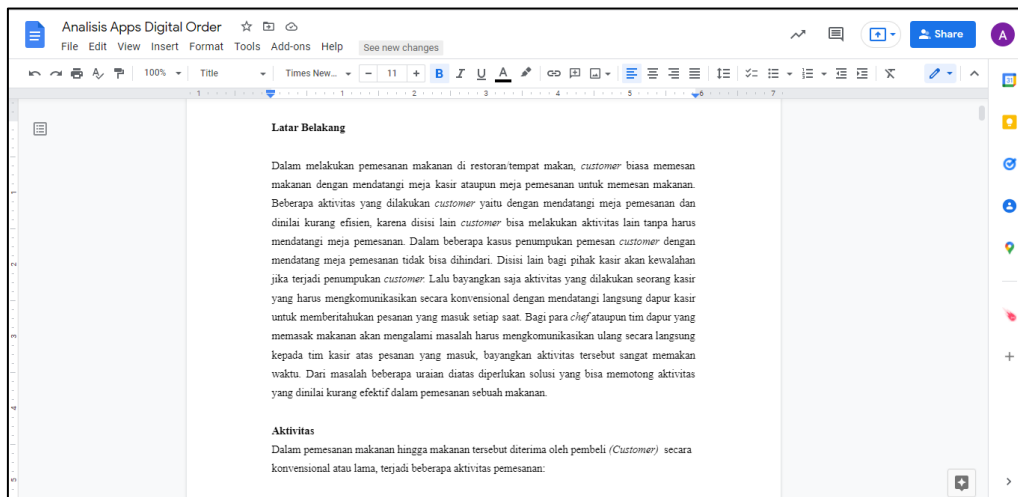
No	Aktivitas	Durasi Waktu
1.	Menganalisis studi kasus Digital Order.	13 – 17 September 2021
2.	Mengembangkan <i>front end</i> modul <i>report</i> pada Aplikasi KlikPayroll.	20 September 2021 – 05 November 2021
3.	Mengembangkan <i>front end</i> pada Aplikasi VONIX.	08 November 2021 – 15 November 2021
4.	Mengembangkan <i>front end</i> pada Aplikasi Ayo BerAksi.	16 Desember 2021 – 07 Januari 2022
5.	Mengembangkan <i>front end</i> pada Ticketing System.	20 – 22 Desember 2021
6.	Melakukan <i>bug fix</i> pada Sistem NES.	03 – 14 Januari 2022
7.	Mengembangkan <i>front end</i> pada Sistem Informasi Akuntansi.	17 Januari – 14 Maret 2022

3.1.1 Menganalisis studi kasus Digital Order

Aktivitas yang dilakukan pada minggu pertama saat melaksanakan magang, yaitu: berkenalan seluruh karyawan di Geekgarden, membentuk tim yang beranggota seluruh peserta magang di Geekgarden, dan *project manager* memberikan tugas kepada tim untuk menganalisis studi kasus Digital Order yang digunakan sebagai pembelajaran awal sebelum masuk ke dalam pengembangan proyek.

Digital Order merupakan sebuah aplikasi yang membantu proses pemesanan makanan ketika berada di suatu restoran dengan cara melakukan *scan barcode* yang terdapat pada meja makan masing – masing. Setelah melakukan *scan barcode* pengguna diarahkan ke website yang menampilkan seluruh menu yang tersedia, pada halaman tersebut pengguna dapat melakukan pemesanan tanpa perlu memanggil pelayan restoran. Proses untuk menentukan seluruh kebutuhan yang terdapat pada aplikasi digital order diantaranya, menentukan pengguna dan kebutuhannya sistem yang dirancang ke dalam Use Case Diagram, membuat Activity

Diagram, merancang Entity Relationship Diagram (ERD), dan membuat Business Process Model and Notation (BPMN) untuk memudahkan dalam proses pengembangan aplikasi. Hasil analisis disusun ke dalam sebuah dokumen yang dapat dilihat pada Gambar 3.1.

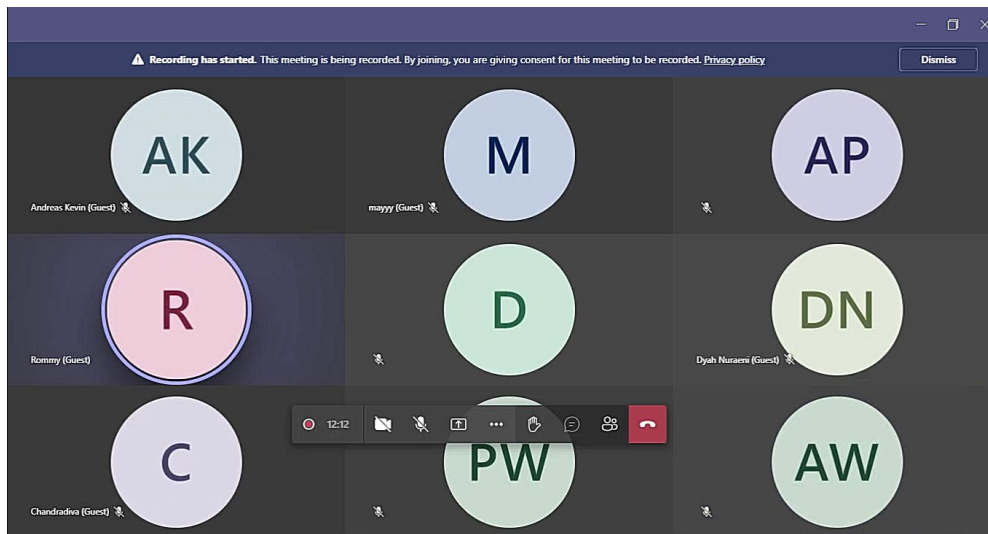


Gambar 3.1 Dokumen Analisis Digital Order

3.1.2 Mengembangkan *front end* modul *report* pada Aplikasi KlikPayroll.

Pemegang diminta *Project Manager* (PM) untuk bergabung ke dalam proyek KlikPayroll untuk mengembangkan beberapa *front end* pada modul *report*. Gambar 3.3 merupakan tampilan utama pada modul *report*. Sebelum mengerjakan proyek KlikPayroll diberikan waktu oleh PM untuk mempelajari terlebih dahulu struktur kode program, *clone repository* proyek ke dalam laptop, serta melakukan instalasi sesuai instruksi yang diberikan pada deskripsi *repository* tersebut. KlikPayroll merupakan sebuah website untuk memberikan layanan penggajian pegawai, mengelola data pegawai, perhitungan pajak pegawai, serta menampilkan data *report* berdasarkan periode tertentu.

Proyek ini menerapkan metode pengembangan menggunakan Scrum dan menerapkan *daily scrum* yang diadakan setiap hari senin hingga jum'at setelah jam kerja kantor atau tepatnya pada pukul 19.30 WIB menggunakan Aplikasi Microsoft Teams. *Daily Scrum* bertujuan untuk menyampaikan *update progress* yang telah dikerjakan, sebagai sarana berdiskusi dengan pihak pelanggan atau *client* terkait apabila ada kendala atau kesalahpahaman pada tugas yang dikerjakan, dan menyampaikan masalah selama pengembangan proyek berlangsung. Gambar 3.2 merupakan proses berlangsungnya kegiatan *daily scrum* yang diikuti oleh seluruh tim proyek dan beberapa pihak pelanggan yang terlibat.

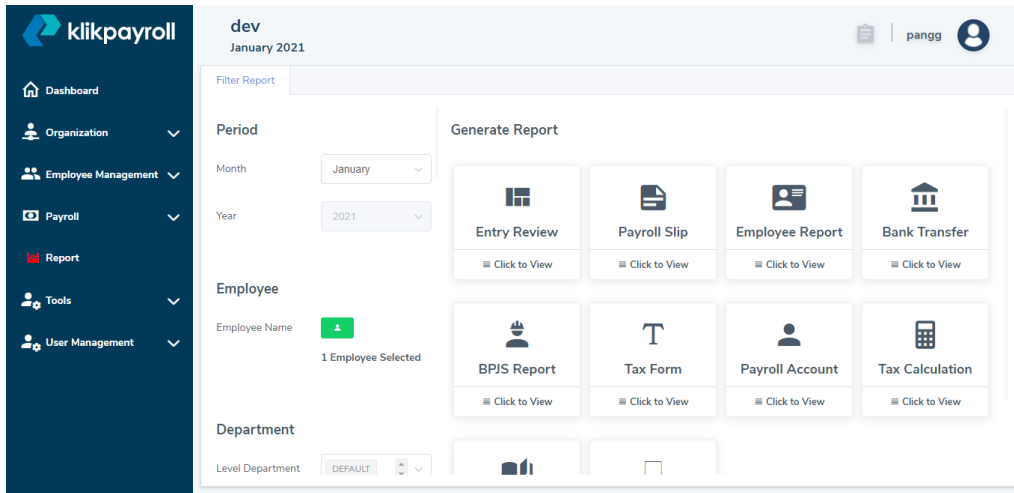


Gambar 3.2 *Daily Scrum* proyek KlikPayroll

Setelah selesai mempelajari struktur kode program, pemegang diberikan dokumen untuk memudahkan dalam penambahan *front end* pada halaman *report* bank transfer OCBC NISP sesuai dengan ketentuan yang terdapat pada dokumen. Awal penugasan dilakukan penambahan *filter* dan tampilan *user interface* pada bank OCBC NISP. *Filter* digunakan untuk menampilkan data ke dalam tabel. Tugas yang telah selesai dikerjakan kemudian dilaporkan kepada PM untuk mendapatkan umpan balik.

Penugasan berikutnya menambahkan tampilan *user interface* pada submenu BPJS *report* yang terdiri dari BPJS TK dan BPJS Kes. Pada submenu BPJS TK masih belum memiliki tampilan *user interface* sehingga dilakukan pengembangan *front end* sesuai dengan ketentuan yang terdapat pada dokumen. Tugas selanjutnya PM meminta untuk melanjutkan mengembangkan *front end* pada menu *Tax Form report*.

Payroll Admin memiliki akses untuk melihat seluruh *Tax Form report* sesuai dengan *filter* yang telah dipilih. Proses penambahan fitur dimulai dari pengerjaan seluruh tampilan *user interface* dan *filter* yang terdapat pada *Tax Form 1721-I*, *Tax Form 1721-II*, *Tax Form 1721-T*, *Tax Form 1721-VI Tidak Final*, dan *Tax Form 1721-VII Final* sesuai dengan dokumen. Setiap kali menyelesaikan tugas dilakukan pengujian tampilan antarmuka terlebih dahulu sebelum di *push ke repository* GitLab KlikPayroll sehingga website akan selalu mendapatkan pembaruan. Namun, memasuki awal bulan November terdapat berita buruk dikarenakan *client* melakukan pemutusan kontrak dengan Geekgarden sehingga terjadi *takeover* atau menghentikan pengembangan pada website KlikPayroll yang menyebabkan seluruh *developer* yang terlibat berhenti untuk mengembangkan proyek tersebut.

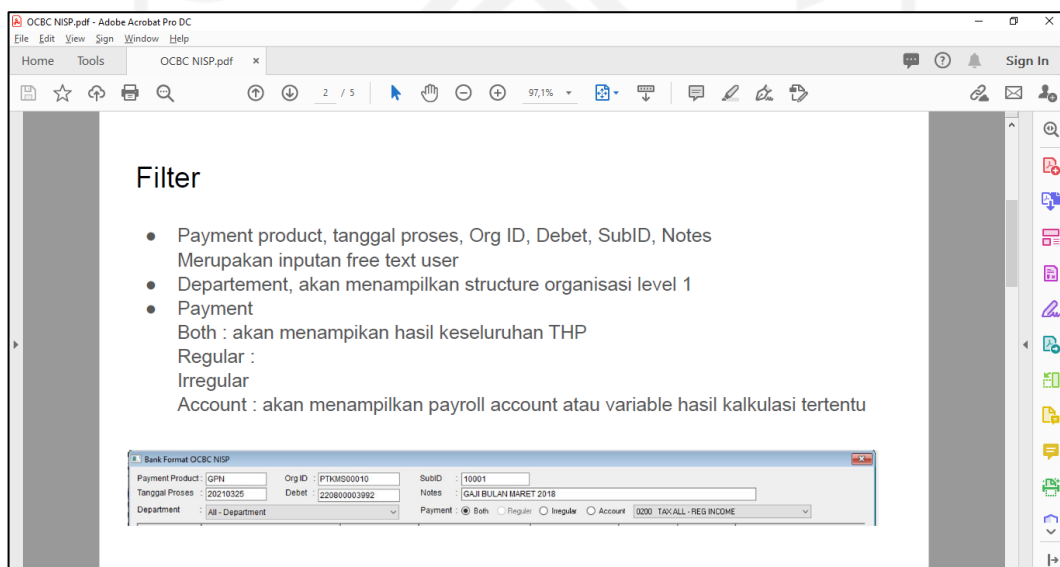


Gambar 3.3 Modul *Report* KlikPayroll

Aktivitas pekerjaan selama bergabung di proyek KlikPayroll, sebagai berikut:

3.1.2.1 Fitur Bank OCBC NISP *report*

Bank transfer OCBC NISP merupakan fitur untuk menampilkan *report* seluruh pegawai yang menggunakan bank OCBC NISP pada periode tertentu sesuai dengan *input filter* yang disediakan. Akan tetapi, pada bank OCBC NISP masih belum memiliki tampilan *user interface* dan *filter* untuk melakukan pencarian data atau *generate* data ke dalam bentuk tabel yang dapat diunduh ke dalam format excel atau pdf sehingga untuk memenuhi kebutuhan tersebut ditambahkan fitur untuk *report* pada bank transfer OCBC NISP berdasarkan ketentuan yang tercantum pada Gambar 3.4. Hasil dari penambahan fitur filter dan tampilan *user interface* pada Bank OCBC NISP *report* terdapat pada Gambar 3.5.



Gambar 3.4 Dokumen *report* Bank OCBC NISP

Gambar 3.5 Tampilan *front end report* Bank OCBC NISP

3.1.2.2 Fitur submenu BPJS Report

BPJS *Report* merupakan fitur untuk melaporkan perhitungan iuran BPJS yang dipotong berdasarkan jumlah gaji setiap karyawan. Perhitungan iuran berdasarkan jenis BPJS, yaitu: BPJS TK dan BPJS Kes sehingga dilakukan penambahan tampilan submenu yang dapat memudahkan dalam melakukan pencarian data sesuai dengan jenis BPJS yang terdaftar. Gambar 3.6 merupakan submenu BPJS *report* yang telah selesai dibuat.

Gambar 3.6 Tampilan *front end* submenu BPJS *Report*

3.1.2.3 Fitur BPJS TK report

BPJS TK digunakan untuk menampilkan *report* daftar pekerja untuk masing-masing NPP dengan bulan yang sama. Pada fitur BPJS pengguna dapat menampilkan data sesuai dengan *filter* yang dipilih. Sebelumnya menu BPJS TK tidak memiliki *user interface* dan

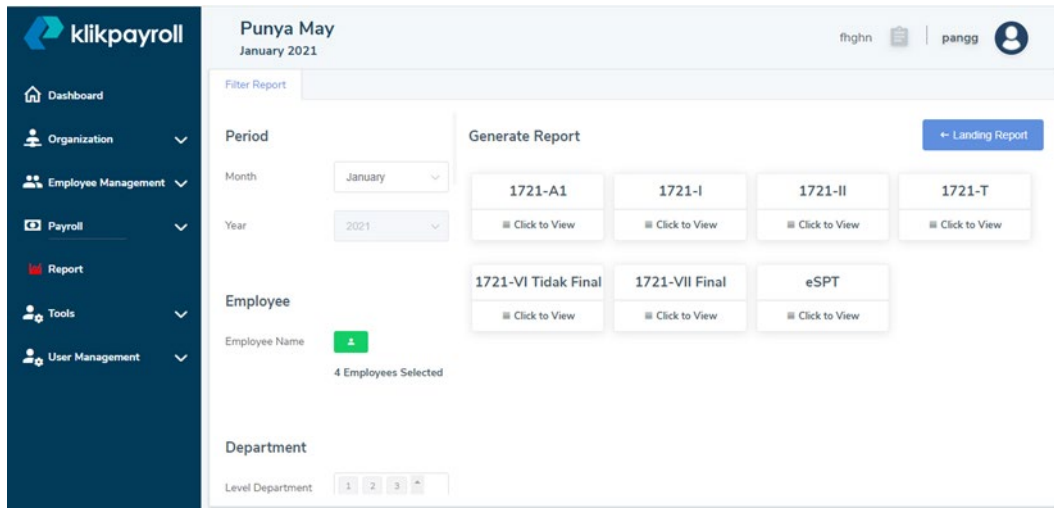
filter. Gambar 3.7 dan Gambar 3.8 merupakan hasil dari penambahan tampilan *user interface*, integrasi dengan *back end* untuk memanggil data NPP, dan *filter* untuk memudahkan dalam generate data.

Gambar 3.7 Tampilan *front end* BPJS TK dan integrasi list NPP

Gambar 3.8 Tampilan *front end* BPJS TK dan *filter*

3.1.2.4 Fitur *Tax Form report*

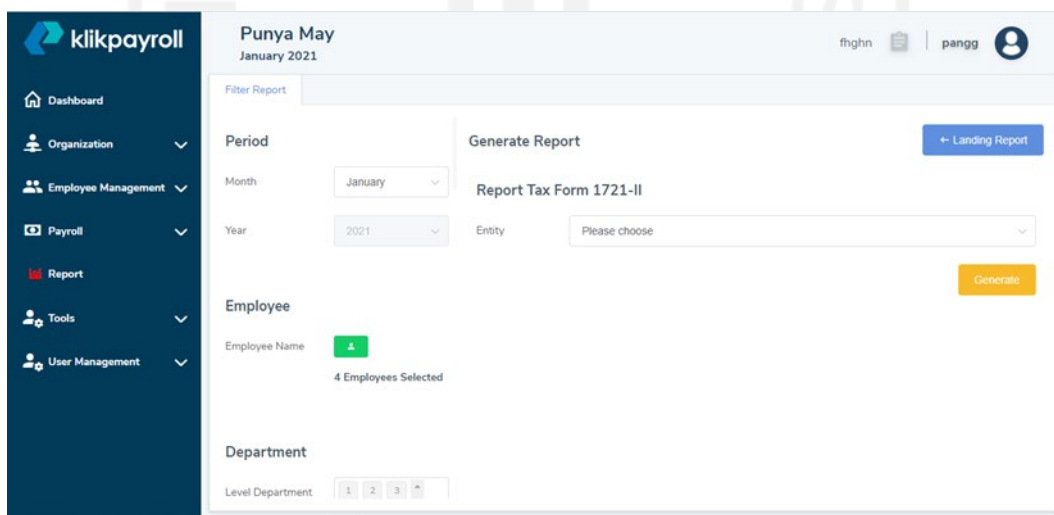
Tax Form report merupakan fitur untuk menampilkan pajak dan menghitung pajak masing - masing pegawai yang telah terdaftar pada sistem. Nantinya, terdapat berbagai macam jenis pajak pada halaman *Tax Form* dengan fungsionalitas berbeda – beda. Penambahan dimulai dari *user interface* pada submenu *Tax Form* agar memudahkan untuk melakukan *generate data* sesuai dengan jenis pajak yang telah terdaftar. Hasil dari pengembangan *user interface* submenu *tax form report* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Tampilan *front end* submenu *Tax Form Report*

3.1.2.5 Fitur *Tax Form report 1721-II*

Tax Form 1721-II merupakan hasil dari rekapitulasi daftar perubahan pegawai tetap yang terdapat pada perusahaan tersebut sehingga nantinya dapat menampilkan daftar pegawai yang keluar, daftar pegawai yang masuk, dan pegawai yang baru memiliki NPWP. Akan tetapi, tidak terdapat fitur untuk menampilkan *Tax Form 1721-II* kemudian ditambahkan tampilan *user interface* beserta *filter* untuk generate data. Hasil dari pengembangan *user interface* dan *filter Tax Form 1721-II* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Tampilan *front end* dan *filter Tax Form 1721-II*

3.1.2.6 Fitur *Tax Form report 1721-T*

Tax Form 1721-T digunakan untuk menampilkan report karyawan tetap yang memiliki jaminan pensiun. Setiap *tax parameter* dari *entity* yang dipilih akan menampilkan

report dengan data berbeda berdasarkan periode bulan dan tahun pada saat melakukan *generate*. Hasil dari pengembangan *user interface* dan *filter tax form 1721-T* berdasarkan *tax parameter* dapat dilihat pada Gambar 3.11.

Gambar 3.11 Tampilan *front end* dan *filter* Tax Form 1721-T

3.1.2.7 Fitur *Tax Form report 1721-VI Tidak Final*

Tax Form 1721-VI Tidak Final digunakan untuk mencetak *report* sebagai bukti pemotongan pajak pegawai atau hampir sama dengan pasal 21 Tidak Final seperti milik Kementerian Keuangan RI Direktorat Jenderal Pajak yang digunakan oleh pegawai tidak tetap. Hasil dari pengembangan *user interface* dan *filter Tax Form 1721-VI Tidak Final* dapat dilihat pada Gambar 3.12 dan Gambar 3.13.

Gambar 3.12 Tampilan *front end* Tax Form 1721-VI Tidak Final

Gambar 3.13 Tampilan *front end* Tax Form 1721-VI Tidak Final lanjutan

3.1.2.8 Fitur Tax Form report 1721-VII Final

Tax Form 1721-VII Final merupakan fitur untuk menampilkan pendapatan bruto dan pajak berdasarkan gaji pegawai. Report akan ditampilkan apabila semua filter telah sesuai dengan data yang tersimpan pada sistem. Untuk proses generate data dilakukan melalui *back end* dan lingkup pekerjaan pada saat itu hanya berfokus untuk mengembangkan *front end*.

Jika pengguna terjadi kesalahan *input* pada filter yang disediakan maka akan muncul *pop up* pemberitahuan bahwa data yang dicari tidak ditemukan. Hasil dari pengembangan *user interface* dan filter tax form 1721-VII Final dapat dilihat pada Gambar 3.14 dan Gambar 3.15.

Gambar 3.14 Tampilan *front end* Tax Form 1721-VII Final

Gambar 3.15 Tampilan *front end* Tax Form 1721-VII Final lanjutan

3.1.3 Mengembangkan *front end* pada Aplikasi VONIX.

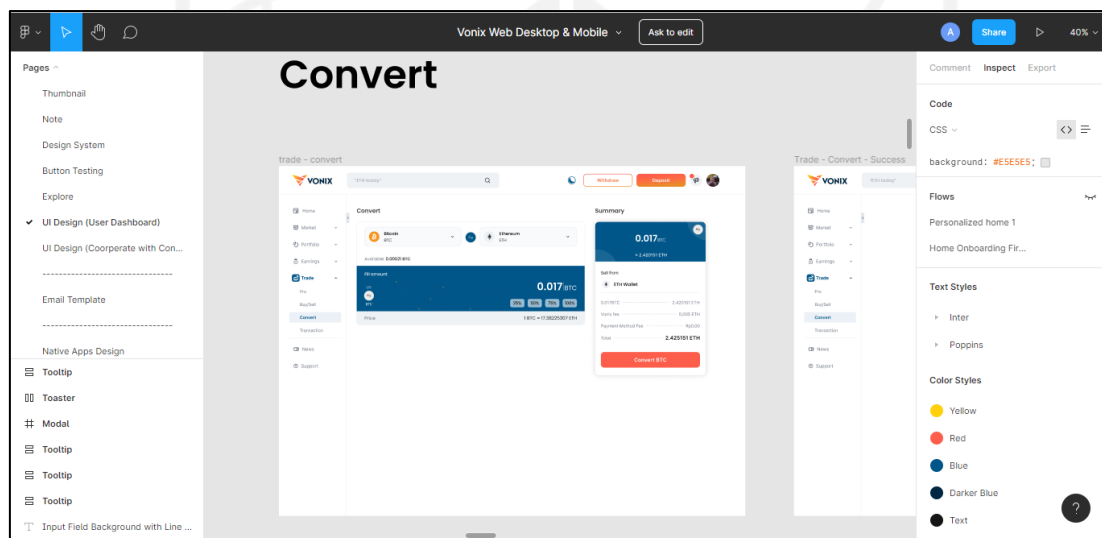
Pada proyek sebelumnya atau KlikPayroll, Geekgarden mengalami pemutusan kontrak kerja sama dengan pihak pelanggan atau *client* sehingga menyebabkan beberapa *developer* mengalami perpindahan proyek. *Project Manager* memberikan tugas untuk membantu dalam pengembangan website VONIX dikarenakan belum terdapat proyek *front end* lainnya. Sebelum memulai mengerjakan proyek diberikan waktu untuk memahami struktur kode program dan melakukan instalasi *tools* sesuai dengan kebutuhan proyek sehingga memenuhi standarisasi yang ditentukan VONIX.

Leader front end pada proyek VONIX memberikan tugas untuk melakukan konfigurasi *signed commit* pada GitHub sebelum memulai pengembangan dengan tujuan untuk memberikan kepercayaan *maintainer* karena *commit* yang dikirimkan memiliki label *verified*. Setelah seluruh *tools* yang digunakan pada proyek VONIX terinstal di laptop dilanjutkan mengerjakan tugas untuk menyimpan *content* ke dalam JSON dan melakukan integrasi pada halaman *Home* dan *about*.

Setelah menyelesaikan tugas tersebut dilanjutkan untuk memahami konsep dasar pembuatan *components* pada kerangka kerja Vue.js agar memudahkan dalam pembuatan *custom element*. Selama proses pengembangan berlangsung tim Quality Control melakukan pengujian pada *website* untuk mengetahui adanya *bug* pada sistem kemudian didokumentasikan pada menu *issue* di GitHub. *Issue* pada sistem memiliki berbagai tingkatan, yaitu: *minor issue*, *medium issue*, dan *major issue*. Apabila terdapat *issue* diprioritaskan untuk melakukan perbaikan pada tingkatan *major issue* terlebih dahulu dikarenakan *issue* tersebut paling berpengaruh terhadap fungsionalitas utama pada fitur yang sedang dikembangkan.

Pengembangan *front end* berikutnya dilakukan integrasi dengan *back end* pada fitur *login with phone number*, *send OTP*, dan *resend OTP*. Namun, terdapat kendala pada saat melakukan pengujian OTP karena jumlah *token* yang disediakan terbatas sehingga perlu berhati-hati dalam penggunaan *token* tersebut.

Pengembangan *front end* dilakukan dengan membuat tampilan *user interface* pada fitur *convert* yang dibuat *responsive* agar dapat digunakan pada *device mobile* dan *website* sesuai desain Figma pada Gambar 3.16. Namun, setelah menyelesaikan semua tugas kemudian meninggalkan proyek VONIX karena memiliki divisi yang sama dengan rekan magang dari UII sehingga disarankan oleh dosen mata kuliah manajemen diri untuk meminta kepada *Project Manager* agar dipindah ke proyek lainnya.



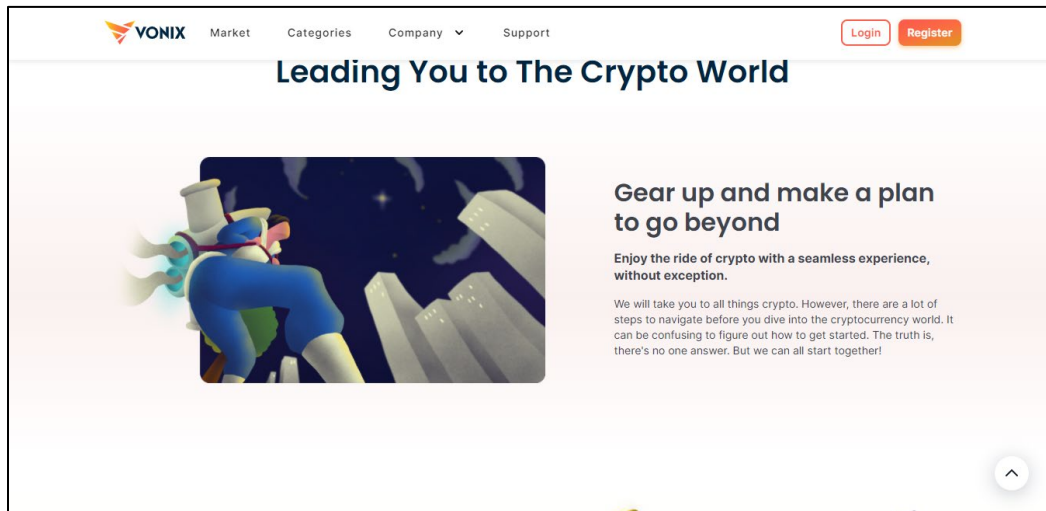
Gambar 3.16 Tampilan *design* Figma fitur *Convert*

Aktivitas pekerjaan selama bergabung di proyek VONIX, sebagai berikut:

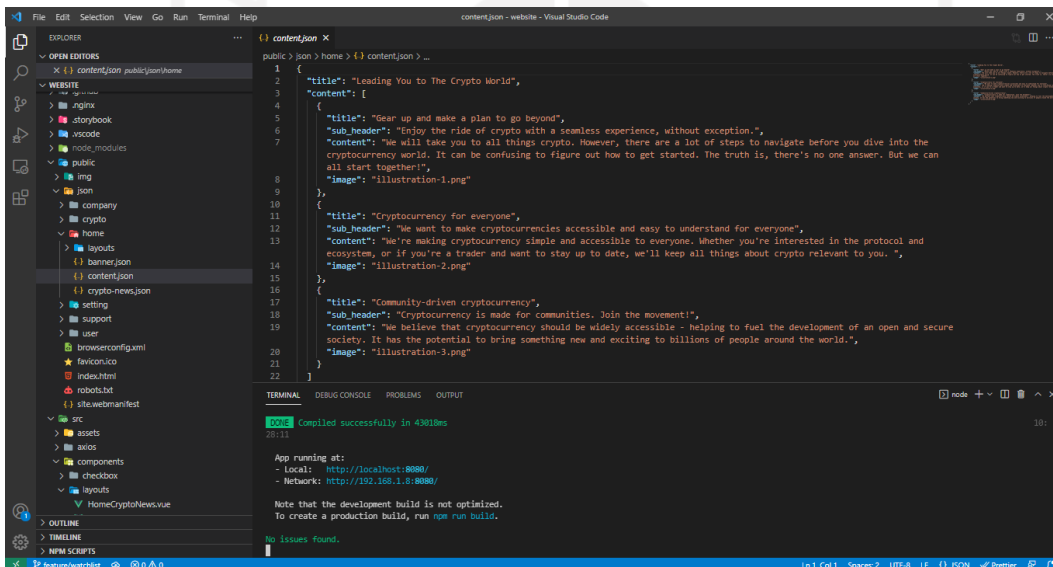
3.1.3.1 Menyimpan *content Home* ke JSON

Halaman *home* pada aplikasi VONIX digunakan untuk menyediakan berbagai informasi seputar cryptocurrency. Sebelumnya *content* disimpan secara statis tanpa menggunakan JSON sehingga ketika terjadi perubahan *content* pada halaman *home* harus melakukan perubahan pada kode program.

Perubahan penyimpanan ke dalam JSON memudahkan ketika integrasi API dengan *back end* dan memanggil parameter yang disediakan. Gambar 3.17 merupakan tampilan *user interface* pada halaman *home* VONIX. Gambar 3.18 merupakan kode program JSON untuk *content home*.



Gambar 3.17 Tampilan *front end home* VONIX

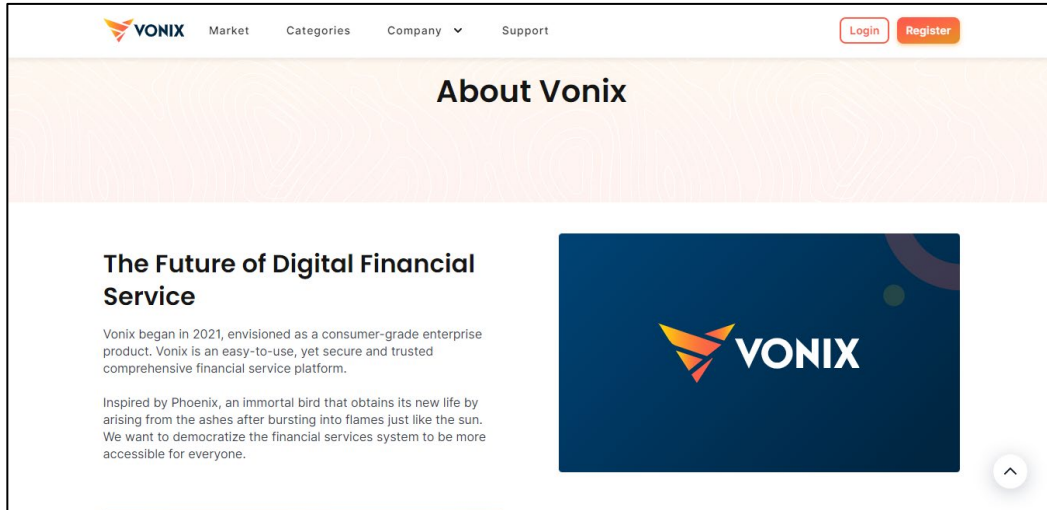


Gambar 3.18 Kode program JSON pada *content home* VONIX

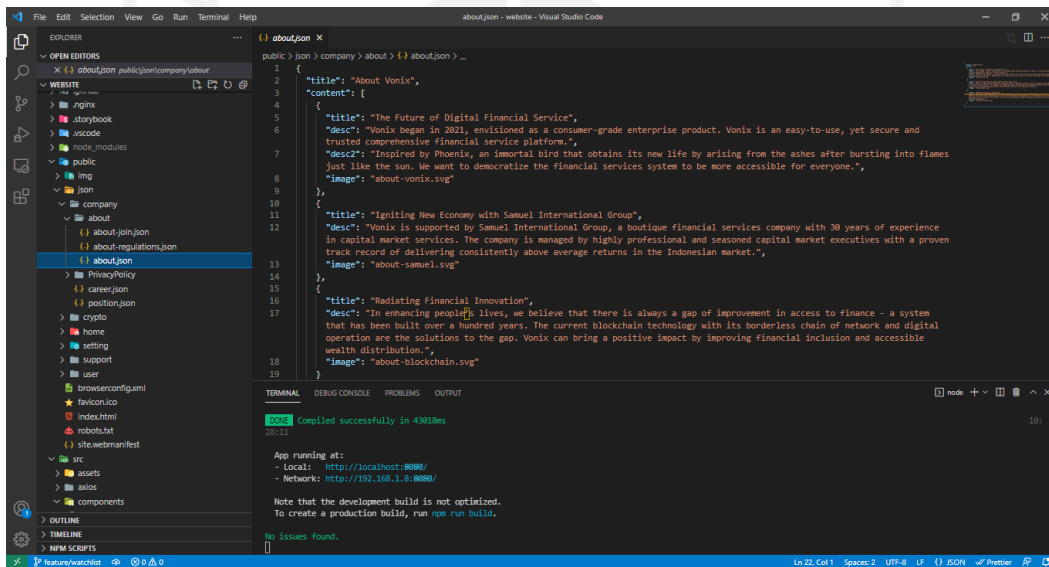
3.1.3.1 Menyimpan *content Home* ke JSON

Halaman *About* digunakan untuk menampilkan informasi seputar VONIX sehingga pengguna dapat mengetahui sejarah didirikannya VONIX, dan lokasi serta regulasi yang diterapkan oleh VONIX. Sebelumnya *content* disimpan secara statis tanpa menggunakan JSON.

Perubahan penyimpanan ke dalam JSON memudahkan ketika integrasi API dengan *back end* dan memanggil parameter yang disediakan. Gambar 3.19 merupakan tampilan *user interface* pada halaman *About*. Gambar 3.20 merupakan kode program JSON untuk *content about*.



Gambar 3.19 Tampilan *front end* about VONIX



Gambar 3.20 Kode program JSON pada *content* about VONIX

3.1.3.2 Melakukan perubahan *front end* login VONIX

Halaman *login* digunakan untuk mengakses seluruh fitur yang disediakan ketika pengguna telah memiliki akun dan terdapat dua cara login di antaranya, yaitu: *sign in with email* dan *sign in with phone number*. Proses *sign in with email* dilakukan dengan mengisi *email* dan *password* apabila benar akan diarahkan pada halaman *dashboard*. Namun, jika terjadi kesalahan dalam pengisian data akan muncul sebuah *error message*. Selanjutnya, untuk proses *sign in with phone number* pengguna diminta untuk mengisi *phone number* dan mengisi kode OTP yang telah dikirimkan melalui SMS (*Short Message Service*). Hasil dari pengembangan *user interface* login dapat dilihat pada Gambar 3.21 dan Gambar 3.22.

The screenshot shows the VONIX sign-in interface for email authentication. At the top left is the VONIX logo, and at the top right is a 'Register' button. The main content is a white card with the title 'Sign in'. It features an 'Email' field with a placeholder 'Type your email' and a link 'Sign in with phone'. Below it is a 'Password' field with a placeholder 'Type your password' and a toggle icon. There is a checkbox for 'Keep me signed in on this computer'. At the bottom of the card, there are links for 'Forgot password?' and 'Don't have an account? Register', and a prominent orange 'Login' button.

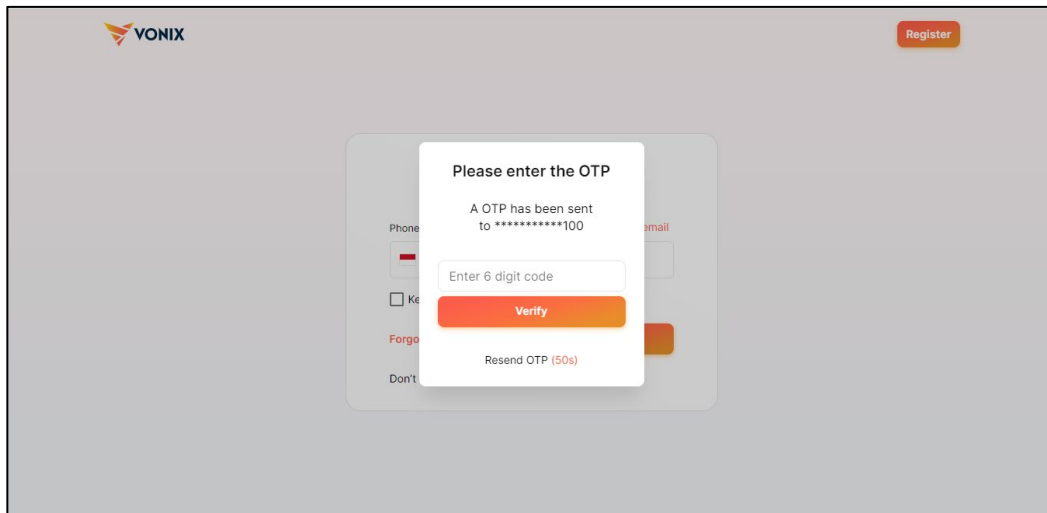
Gambar 3.21 Tampilan *front end sign in with email* VONIX

The screenshot shows the VONIX sign-in interface for phone number authentication. At the top left is the VONIX logo, and at the top right is a 'Register' button. The main content is a white card with the title 'Sign in'. It features a 'Phone Number' field with a dropdown menu showing '+62' and a placeholder. There is a link 'Sign in with email'. Below it is a checkbox for 'Keep me signed in on this computer'. At the bottom of the card, there are links for 'Forgot password?' and 'Don't have an account? Register', and a prominent orange 'Login' button.

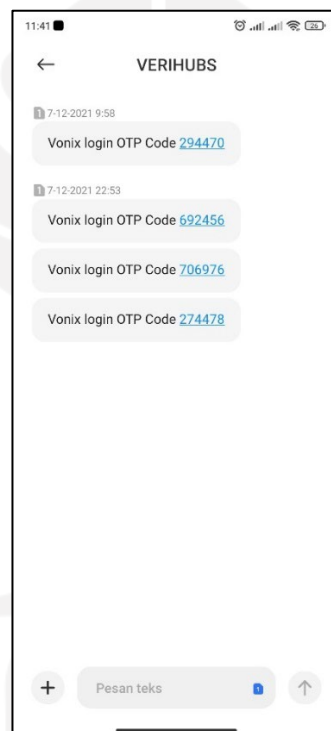
Gambar 3.22 Tampilan *front end sign in with phone number* VONIX

3.1.3.3 Fitur OTP dan *resend* OTP

OTP (*One Time Password*) merupakan fitur yang digunakan untuk melakukan *sign in with phone number* dan dilakukan *resend* OTP ketika kode OTP telah melebihi batas waktu atau *expired*. Penggunaan fitur OTP dilakukan dengan cara mengisi *phone number* yang telah terdaftar kemudian sistem akan mengirimkan pesan melalui SMS (*Short Message Service*) apabila kode OTP diisi dengan benar pengguna akan diarahkan ke halaman *dashboard*. Akan tetapi, jika kode OTP salah dipersilakan untuk melakukan *resend* OTP). Hasil dari pengembangan *user interface* dan integrasi OTP dapat dilihat pada Gambar 3.23, sedangkan Gambar 3.24 merupakan SMS (*Short Message Service*) yang berisikan kode OTP.



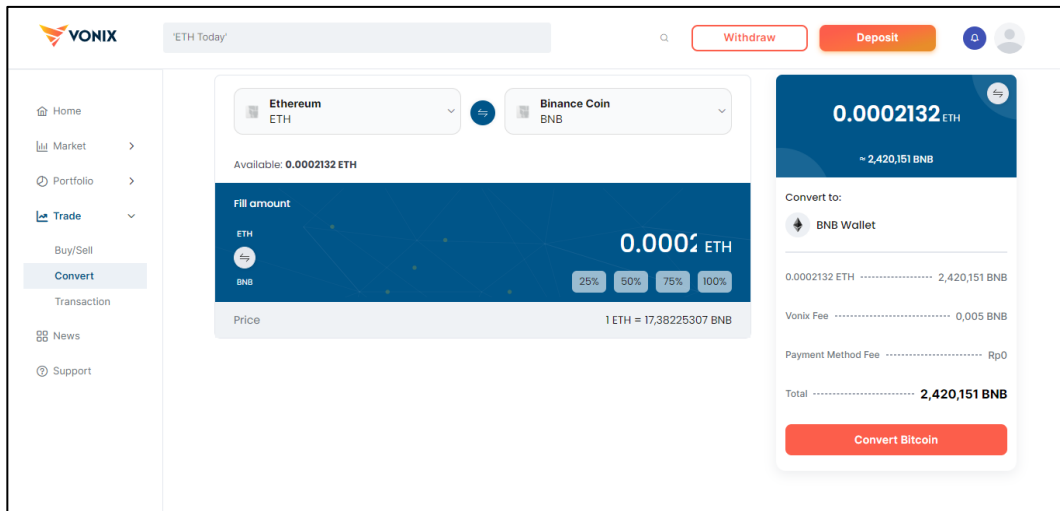
Gambar 3.23 Tampilan *front end* fitur OTP VONIX



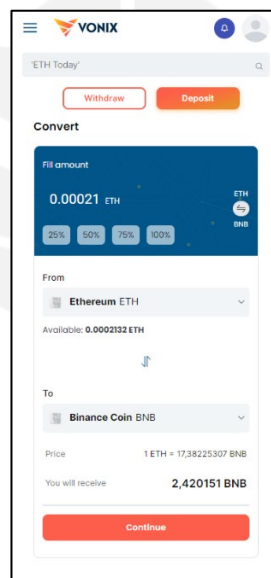
Gambar 3.24 Kode OTP VONIX yang diterima melalui SMS

3.1.3.4 Menambahkan *front end Convert*

Convert merupakan fitur untuk mengubah jenis *coin crypto* yang dimiliki ke jenis *coin crypto* lain sesuai dengan keinginan. Proses dilakukan dengan memasukkan jumlah *coin crypto* kemudian memilih jenis *coin crypto* yang ingin di *convert*. Hasil dari penambahan tampilan *user interface convert* dapat dilihat pada Gambar 3.25 dan Gambar 3.26.



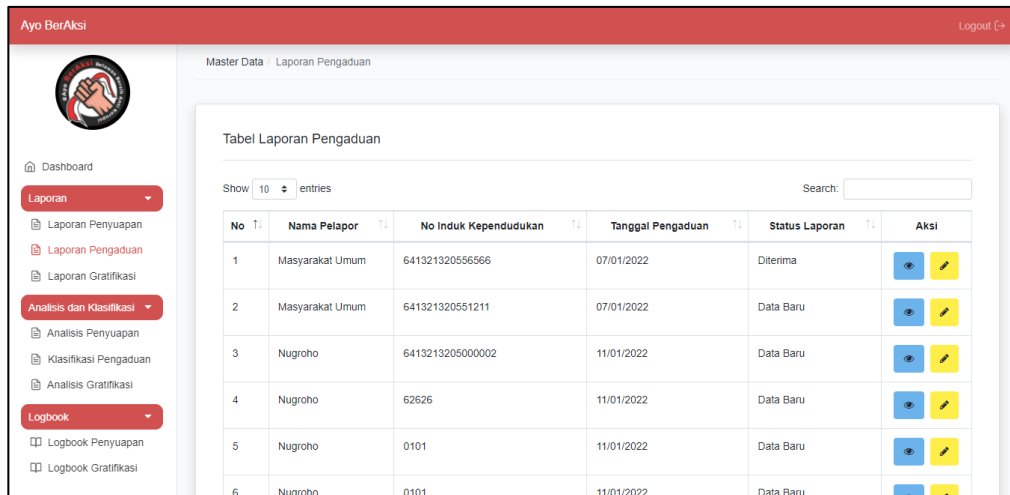
Gambar 3.25 Tampilan *front end convert responsive website*



Gambar 3.26 Tampilan *front end convert responsive mobile*

3.1.4 Mengembangkan *front end* pada Aplikasi Ayo BerAksi













Setelah berdiskusi dengan PM akhirnya dipindahkan ke dalam proyek AyoBeraksi. Proyek ini telah dikembangkan *back end* dan API tetapi belum memiliki tampilan *user interface* sehingga diprioritaskan untuk mengembangkan *front end*. Pada saat pengembangan *front end* pemangag menggunakan *library bootstrap* untuk membangun tampilan *user interface*. Setelah seluruh *front end* telah dikembangkan dilanjutkan untuk melakukan integrasi dengan *back end*. Seluruh fitur yang terdapat pada *website* digunakan untuk menampilkan data dan mengubah data, untuk penambahan data dilakukan melalui aplikasi *mobile*. Gambar 3.27 merupakan salah satu contoh tampilan antarmuka pada website Ayo BerAksi.



Master Data / Laporan Pengaduan

Tabel Laporan Pengaduan

Show 10 entries Search:

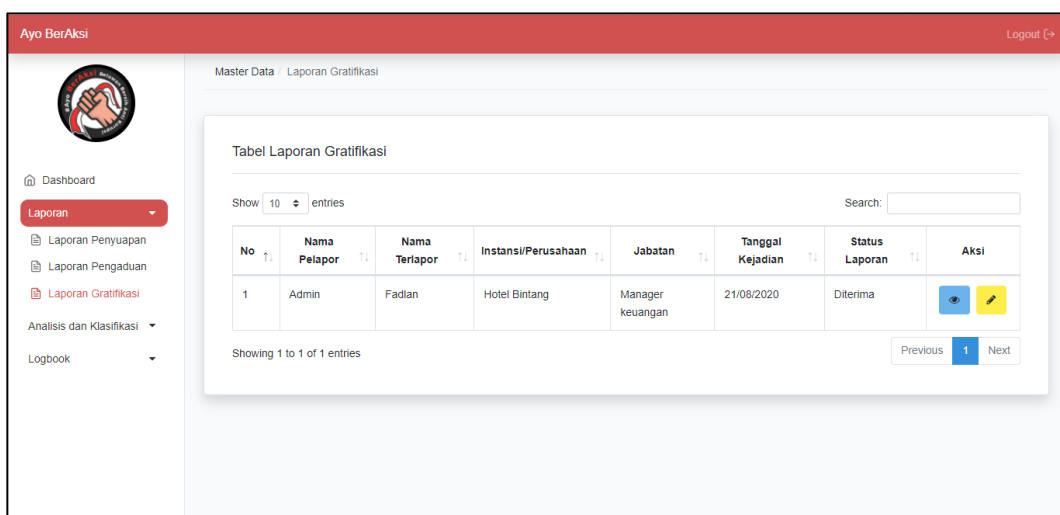
No	Nama Pelapor	No Induk Kependudukan	Tanggal Pengaduan	Status Laporan	Aksi
1	Masyarakat Umum	641321320556566	07/01/2022	Diterima	 
2	Masyarakat Umum	641321320551211	07/01/2022	Data Baru	 
3	Nugroho	6413213205000002	11/01/2022	Data Baru	 
4	Nugroho	62626	11/01/2022	Data Baru	 
5	Nugroho	0101	11/01/2022	Data Baru	 
6	Nugroho	0101	11/01/2022	Data Baru	 

Gambar 3.27 Tampilan website Ayo BerAksi

Aktivitas pekerjaan selama bergabung di proyek Ayo BerAksi, sebagai berikut:

3.1.4.1 Perubahan fitur Laporan HSKU menjadi Laporan Gratifikasi



Laporan HSKU (Hadiah, Sumbangan, dan Keuntungan Serupa serta Tawaran Penyuapan) mendapatkan perubahan nama laporan yang diminta oleh *client* menjadi Laporan Gratifikasi. Selain itu, laporan gratifikasi digunakan untuk mengirimkan laporan yang berkaitan dengan kasus pemberian hadiah, sumbangan, barang, dan keuntungan sehingga adanya Aplikasi Ayo BerAksi dapat lebih cepat untuk mengusut kejadian. Akan tetapi, laporan gratifikasi perlu untuk diperiksa terlebih dahulu oleh tim pemeriksa apakah laporan tersebut akan diterima atau ditolak. Selanjutnya, jika laporan diterima akan dipilih ketua, dan anggota untuk menindaklanjuti kasus tersebut. Hasil dari penambahan tampilan *user interface* fitur laporan HSKU dapat dilihat pada Gambar 3.28.



Master Data / Laporan Gratifikasi

Tabel Laporan Gratifikasi

Show 10 entries Search:

No	Nama Pelapor	Nama Terlapor	Instansi/Perusahaan	Jabatan	Tanggal Kejadian	Status Laporan	Aksi
1	Admin	Fadlan	Hotel Bintang	Manager keuangan	21/08/2020	Diterima	 

Showing 1 to 1 of 1 entries Previous 1 Next

Gambar 3.28 Tampilan *front end* Laporan Gratifikasi

3.1.4.2 Menambahkan *front end* fitur Analisis Laporan Penyuaan

Analisis laporan penyuaan merupakan fungsi untuk menindaklanjuti hasil laporan yang telah diterima oleh admin kemudian tim kepatuhan mengisi hasil investigasi dan kesimpulan investigasi yang telah dilakukan. Pengguna tim kepatuhan hanya memiliki akses untuk melakukan *edit* dan melihat detail analisis laporan penyuaan. Hasil dari pengembangan *user interface* analisis laporan penyuaan dapat dilihat pada Gambar 3.29.

No	Nama Terlapor	Jabatan	Kasus	Tanggal Kejadian	Tanggal Laporan	Aksi
1	Gilli	Sekretaris 3	suapan uang	13/09/2018	03/01/2022	
2	tisid	Sekretaris	suapan uang	13/09/2019	06/01/2022	

Gambar 3.29 Tampilan *front end* Analisis Laporan Penyuaan

3.1.4.3 Menambahkan *front end* fitur Klasifikasi Laporan Pengaduan

Klasifikasi laporan pengaduan merupakan fitur untuk dilakukan klasifikasi terhadap laporan pengaduan yang dikirimkan, selanjutnya tim kepatuhan akan melakukan pengecekan terkait detail klasifikasi laporan pengaduan berdasarkan uraian laporan, dan saran masukan. Hasil dari pengembangan fitur klasifikasi laporan pengaduan dapat dilihat pada Gambar 3.30, Gambar 3.31, dan Gambar 3.32.

No	Nama Pelapor	No Induk Kependudukan	Tanggal Pengaduan	Aksi
1	Admin	6413213205000002	03/01/2022	

Gambar 3.30 Tampilan *front end* Klasifikasi Laporan Pengaduan

Ayo BerAksi Logout [+]

Master Data / Klasifikasi Laporan Pengaduan / Ubah Klasifikasi Laporan Pengaduan

Detail Laporan Pengaduan

Kepada
Rangga

Nama Pelapor
Admin

Alamat Lengkap
Yogyakarta

No Induk Kependudukan
6413213205000002

Uraian Laporan
Laporana yang saya kirimkan kemaren tidak diproses secara baik

Saran dan Masukan
sebaiknya segera dikaji ulang pengambilan keputusan

Tanggal Pengaduan
03/01/2022

Gambar 3.31 Tampilan *front end* Detail Laporan Pengaduan

Ayo BerAksi Logout [+]

Tanggal Pengaduan
03/01/2022

Ubah Klasifikasi Laporan Pengaduan

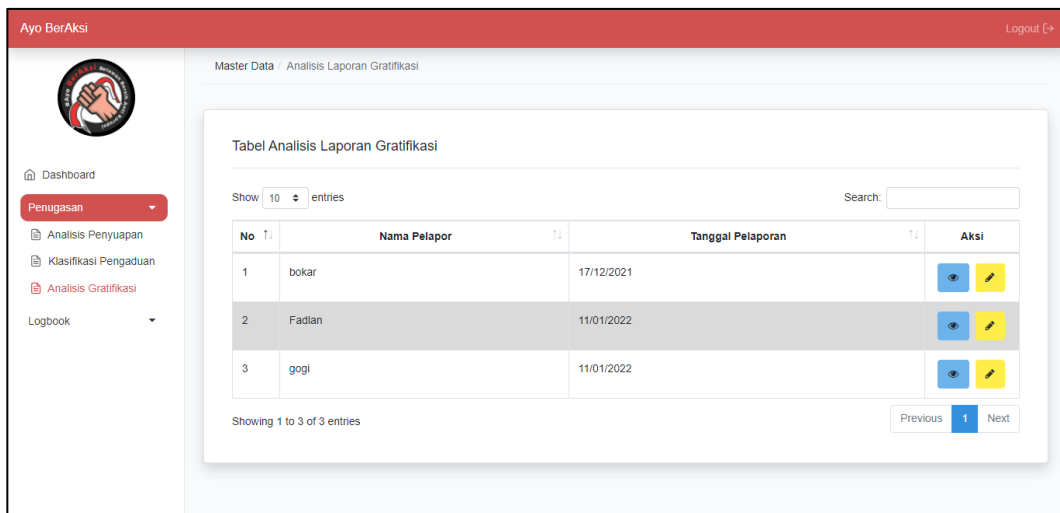
Klasifikasi
 A B

Keterangan Klasifikasi:
 Lingkari yang dipilih, ditulis dalam kolom yang tersedia
 A: Laporan tidak berkadar pengawasan
 B: Laporan berkadar pengawasan

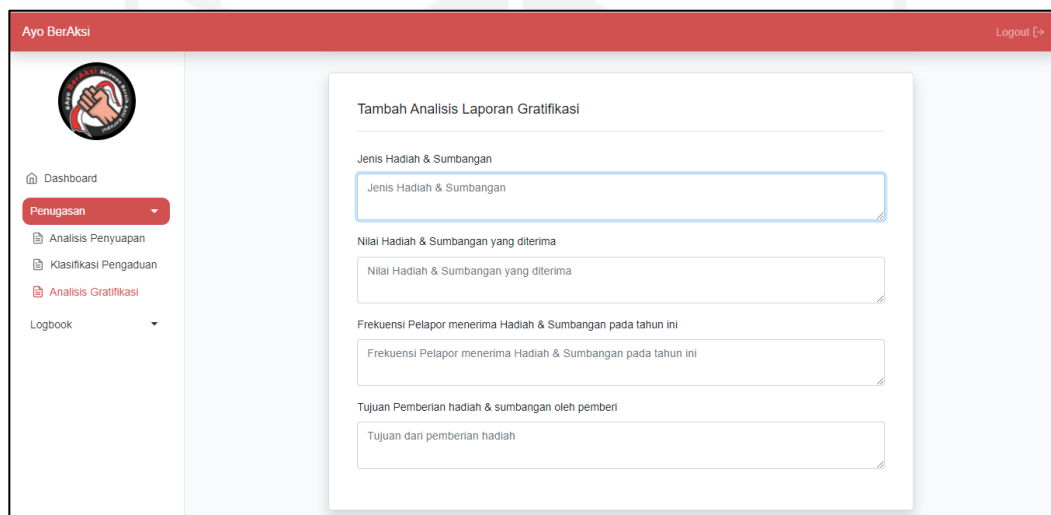
Gambar 3.32 Tampilan *front end* ubah Klasifikasi Laporan Pengaduan

3.1.4.4 Menambahkan *front end* fitur Analisis Laporan Gratifikasi

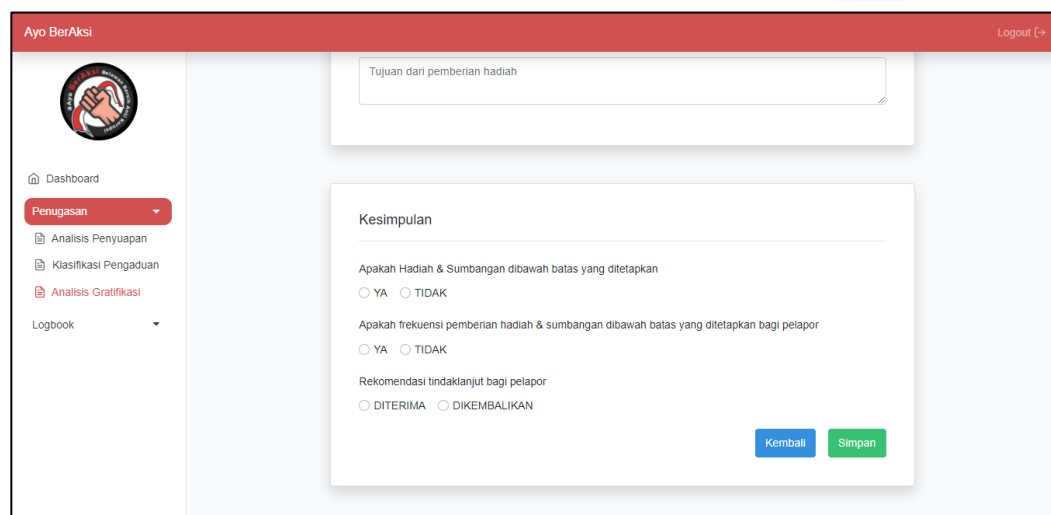
Analisis laporan gratifikasi digunakan setelah laporan gratifikasi diterima dan ditentukan tim kepatuhan sehingga dapat dilakukan proses analisis sesuai dengan formulir serta memberikan kesimpulan sesuai dengan analisis pada kasus yang ditangani. Hasil dari pengembangan *user interface* analisis laporan gratifikasi dapat dilihat pada Gambar 3.33, Gambar 3.34, dan Gambar 3.35.



Gambar 3.33 Tampilan *front end* Analisis Laporan Gratifikasi



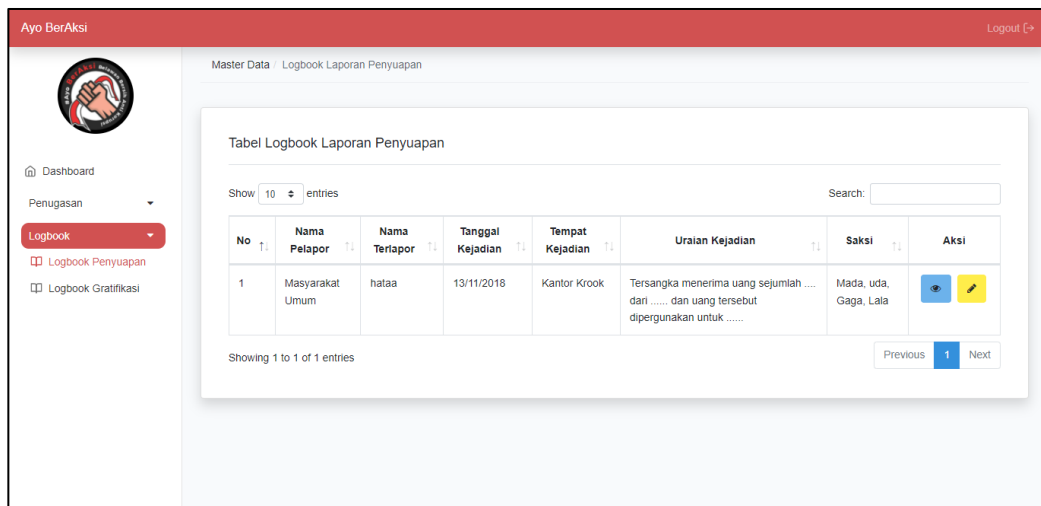
Gambar 3.34 Tampilan *front end* tambah Analisis Laporan Gratifikasi



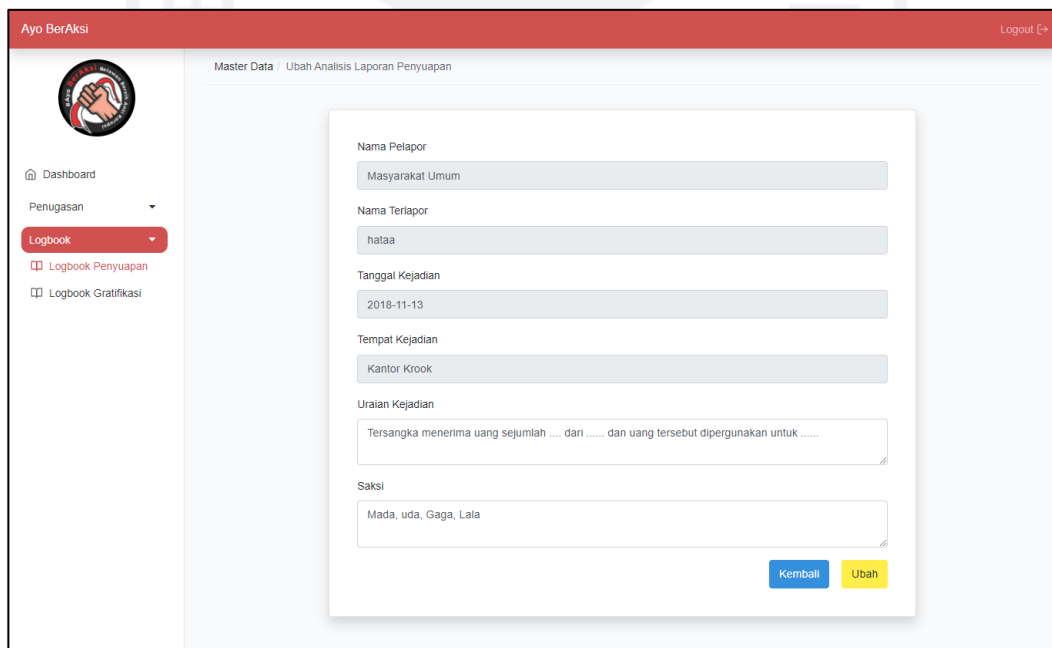
Gambar 3.35 Tampilan *front end* kesimpulan Analisis Laporan Gratifikasi

3.1.4.5 Menambahkan *front end* fitur *logbook* Laporan Penyuaan

Logbook laporan penyuaan merupakan kegiatan harian tim kepatuhan untuk menyelidiki kasus laporan penyuaan. Proses pengisian logbook dilakukan setiap melakukan sidak ke lokasi pada saat diindikasikan terjadinya transaksi penyuaan. Hasil dari pengembangan logbook dapat dilihat pada Gambar 3.36 dan Gambar 3.37.



Gambar 3.36 Tampilan *front end* logbook Laporan Penyuaan



Gambar 3.37 Tampilan *front end* ubah logbook Laporan Penyuaan

3.1.4.6 Menambahkan *front end* fitur *logbook* Laporan Gratifikasi

Logbook laporan gratifikasi merupakan catatan kegiatan yang dilakukan tim kepatuhan selama proses menyelidiki kasus berlangsung. Pengisian *logbook* berdasarkan

hasil analisis tim fungsi kepatuhan dan keterangan terkait laporan gratifikasi. Penambahan fitur *logbook* laporan gratifikasi dapat dilihat pada Gambar 3.38.

Gambar 3.38 Tampilan *front end* *logbook* Laporan Penyuapan

3.1.5 Pengembangan *front end* pada Ticketing Sistem.

Ticketing Sistem merupakan Aplikasi berbasis website yang berfungsi sebagai *support ticket* ketika terdapat *request* pada sistem. Sebelum mengerjakan proyek tersebut mempersiapkan kebutuhan yang diperlukan pada proyek Ticketing Sistem hanya dapat dijalankan menggunakan *server local* Ticketing sehingga diharuskan untuk melakukan instalasi VPN agar dapat mengerjakan proyek tersebut. Aktivitas pengembangan *front end* selama bergabung di proyek Ayo BerAksi, yaitu:

1. Fitur *search* dan *filter ticket* pada *Pool Ticket Level 1*.
2. Fitur *search* dan *filter ticket* pada *Pool Ticket Level 2*.
3. Fitur *search* dan *filter ticket* pada *Pool Ticket Level 3*.

Fitur *search* digunakan untuk mencari data berdasarkan *source* dan *hostname* sedangkan *filter ticket* untuk menampilkan data berdasarkan *priority* yang dipilih. Pengembangan dilakukan dengan membuat *component search* dan *filter* kemudian melakukan integrasi API dengan *back end* sehingga *component* dapat berjalan sesuai fungsionalitas yang diharapkan. Hasil pengembangan tampilan *user interface* yang telah ditambahkan fitur *search* dan *filter* dapat dilihat pada Gambar 3.39 untuk *pool ticket level 1*, Gambar 3.40 untuk *pool ticket level 2*, dan Gambar 3.41 untuk *pool ticket level 3*.

Priority

Filter search

Pool Ticket Level 1

No	Source	Hostname	Title	Desc	Event Received	Priority	Action
1	172.21.21.2	FW-1			December 17, 2021 9:29 AM	1	
2	172.16.8.6	worker5			December 17, 2021 9:29 AM	1	
3	172.16.8.6	worker5			December 17, 2021 9:29 AM	1	
4	172.16.8.6	worker5			December 17, 2021 9:29 AM	1	
5	172.21.21.2	FW-1			December 17, 2021 9:29 AM	1	
6	172.16.8.6	worker5			December 17, 2021 9:29 AM	1	
7	172.16.7.30	firewall-neculus-1			December 17, 2021 9:29 AM	1	
8	172.16.7.30	firewall-neculus-1			December 17, 2021 9:29 AM	1	

Gambar 3.39 Fitur *search* dan *filter* pada *Pool Ticket Level 1*.

Priority

Filter search

Pool Ticket Level 2

No	Source	Hostname	Title	Desc	Event Received	Priority	Action
----	--------	----------	-------	------	----------------	----------	--------

Gambar 3.40 Fitur *search* dan *filter* pada *Pool Ticket Level 2*.

Priority

Filter search

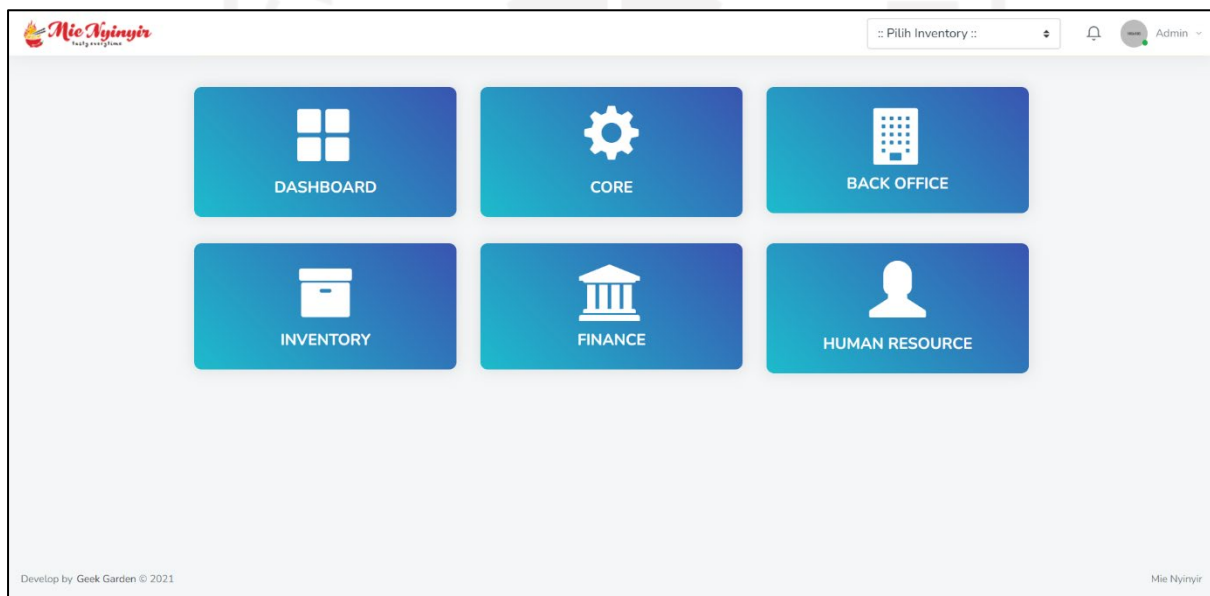
Pool Ticket Level 3

No	Source	Hostname	Title	Desc	Event Received	Priority	Action
----	--------	----------	-------	------	----------------	----------	--------

Gambar 3.41 Fitur *search* dan *filter* pada *Pool Ticket Level 3*.

3.1.6 Melakukan *bug fix* pada Sistem NES.

Proyek yang selanjutnya adalah Sistem NES. Proyek ini bertujuan untuk melakukan pendataan dan pencatatan produk pada perusahaan pelanggan Geekgarden berbentuk *dashboard* yang terdiri dari lima modul diantaranya, yaitu: modul *core*, modul *inventory*, modul *back office*, modul *finance*, dan modul *human resource*. Sistem NES sebelumnya telah dikembangkan tetapi *developer* yang mengembangkan proyek tersebut telah *resign* dari Geekgarden. Setelah dilakukan pengujian oleh *client* atau pihak pelanggan Geekgarden terdapat *bug* yang ditemukan pada modul *core*, modul *back office*. Aktivitas magang selanjutnya melakukan *bug fix* pada modul tersebut yang ditugaskan oleh *Project Manager* sebagai *Fullstack Developer* selama 2 minggu. Gambar 3.42 merupakan salah satu contoh tampilan antarmuka pada website Sistem NES.



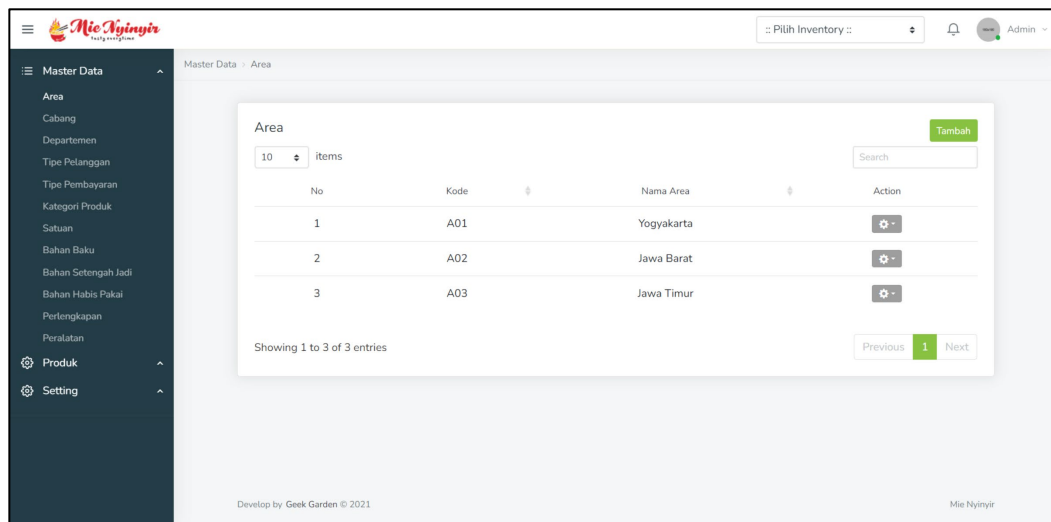
Gambar 3.42 Tampilan website Sistem NES.

Aktivitas pekerjaan selama bergabung di proyek Ayo BerAksi, sebagai berikut:

3.1.6.1 *Bug fix* modul *core* menu area

Menu Area yang terdapat di modul *core* merupakan lokasi area/wilayah rumah makan. Pengguna dapat menambahkan area berdasarkan lokasi rumah makan yang tersedia di daerah tertentu. Apabila suatu saat lokasi rumah makan pindah atau tidak ada dapat dilakukan ubah atau hapus pada area tersebut. Fitur ini sebelumnya telah dikembangkan tetapi memiliki *bug* data tidak tampil sehingga dilakukan pengecekan dari sisi *front end* dan *back end*. Pada kasus tersebut ditemukan salah satu data memiliki kode atau nama area

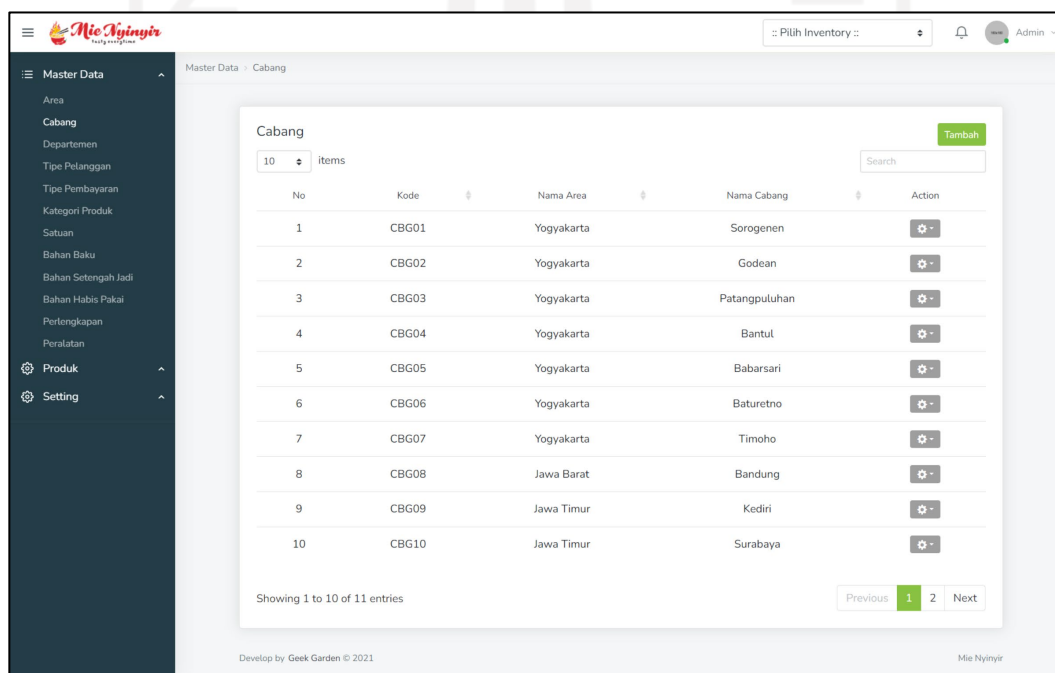
kosong (*null*) sehingga data tidak tampil. Hasil *bug fix* pada menu area dapat dilihat pada Gambar 3.43



Gambar 3.43 Tampilan *front end* Area

3.1.6.2 *Bug fix* modul core menu cabang

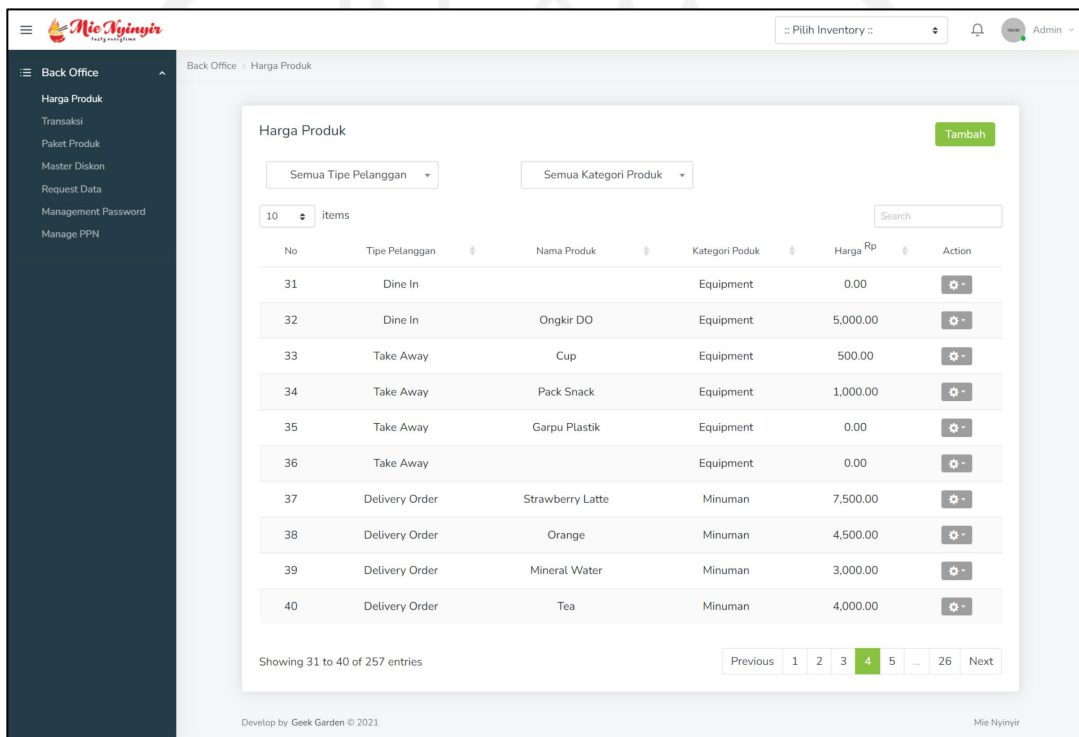
Menu cabang di modul *core* merupakan lokasi cabang rumah makan sesuai area yang dimiliki. Fitur ini sebelumnya telah dikembangkan tetapi memiliki *bug* data tidak tampil sehingga dilakukan pengecekan dari sisi *front end* dan *back end*. Pada kasus tersebut ditemukan pada salah satu data memiliki kode atau nama area kosong (*null*) sehingga data tidak tampil pada tabel. Hasil *bug fix* pada menu cabang dapat dilihat pada Gambar 3.44.



Gambar 3.44 Tampilan *front end* Cabang

3.1.6.3 Bug fix modul *Back Office* menu Harga Produk

Harga Produk merupakan fitur untuk menambahkan harga produk yang terdapat di rumah makan tersebut. Fitur ini sebelumnya telah dikembangkan tetapi memiliki *bug* ketika data nama produk kosong (*null*) tidak dapat menekan fitur *action* dan tampil pesan *error* yang terdapat pada data tersebut. Pengguna seharusnya dapat melakukan *action* walaupun data nama produk kosong. Hasil *bug fix* menu harga produk dapat dilihat pada Gambar 3.45.



Gambar 3.45 Tampilan *front end* Harga Produk

3.1.7 Pengembangan *front end* pada Sistem Informasi Akuntansi

Proyek terakhir yang dikerjakan ketika magang adalah Sistem Informasi Akuntansi. Pada proyek ini terdapat empat menu utama dengan jumlah 21 fitur di dalamnya yang terdiri dari: menu beranda, menu pengaturan memiliki sejumlah 15 fitur, menu transaksi memiliki sejumlah 2 fitur, dan menu laporan memiliki sejumlah 4 fitur. Pengguna Sistem Informasi Akuntansi digunakan oleh divisi IT dan Keuangan, divisi IT melakukan pemeliharaan pada sistem sedangkan divisi Keuangan untuk melakukan seluruh aktivitas akuntansi. Dari sejumlah 21 fitur yang telah disebutkan sebelumnya, pada beberapa fitur perlu dilakukan *revamp front end* dan pengembangan *front end*. Penjelasan secara detail mengenai Sistem Informasi Akuntansi terdapat pada bab 3.2.3.

3.2 Manajemen Proyek

Dalam pengembangan Sistem Informasi Akuntansi terdapat beberapa tahapan manajemen proyek yang dilakukan, yaitu: pendefinisian proyek, perencanaan proyek, pelaksanaan proyek, pemantauan dan pengendalian proyek, serta penutupan proyek.

3.2.1 Pendefinisian Proyek

Sistem Informasi Akuntansi merupakan sebuah sistem berbasis *website* yang digunakan untuk mengelola transaksi jurnal dan melakukan aktivitas akuntansi pada sistem. Sistem ini akan digunakan oleh divisi Keuangan dan IT pada perusahaan pihak pelanggan Geekgarden. Proses Akuntansi melalui *front end* dimulai dari penambahan Akun (COA), melakukan Transaksi Jurnal, hingga terbentuk menjadi sebuah laporan *Account Listing*, laporan *Journal Listing*, dan laporan *Trial Balance* untuk pengolahan data akan dilakukan dari *back end*. Sebelum adanya sistem ini seluruh aktivitas akuntansi dilakukan secara manual sehingga menjadi kurang efektif. Dengan harapan, adanya sistem ini dapat meningkatkan efisiensi dalam mengelola data keuangan dan menghasilkan laporan transaksi akuntansi secara *realtime*.

3.2.2 Perencanaan Proyek

Perencanaan proyek dilakukan untuk menentukan kebutuhan yang diperlukan dalam proses pengembangan *front end* dari awal hingga akhir pengembangan. Beberapa hal yang dilakukan, yaitu: *Deliverables/Scope* yang ingin dicapai, pembagian aktivitas dalam pengerjaan proyek, dan tim pengembang yang dibutuhkan ke dalam proyek. Adapun tim yang terlibat dalam proses pengembangan Sistem Informasi Akuntansi terbagi dalam sejumlah peran, yaitu:

- *Project Manager*, memiliki tanggung jawab untuk memimpin pelaksanaan proyek, melakukan koordinasi dengan pihak pelanggan, memantau pengembangan proyek, dan memberikan arahan serta *task* yang dikerjakan kepada tim. Selain itu, jika terdapat kendala tim pengembang dapat melakukan konsultasi kepada *project manager*.
- *UI/UX Designer*, bertanggung jawab membuat desain *user interface* berbentuk prototipe berdasarkan proses bisnis atau alur kerja sesuai dokumen kebutuhan sistem. Selain itu, memiliki peran untuk melakukan pengujian prototipe kepada pihak pelanggan untuk mendapatkan umpan balik.

- *Front-End Web Developer*, bertanggung jawab melakukan implementasi rancangan prototipe sesuai dengan kebutuhan sistem dalam bentuk kode program sehingga dapat berinteraksi secara langsung dengan pengguna.
- *Back-End Web Developer*, memiliki tanggung jawab mengelola penyimpanan basis data dan merancang API yang digunakan untuk menjembatani komunikasi antara *front end* dengan *back end* pada sistem.
- *Quality Assurance/Quality Control (QA/QC)*, bertanggung jawab untuk memastikan seluruh hasil implementasi yang dikembangkan *front-end web developer* dan *back-end web developer* berjalan sesuai dengan proses bisnis yang diharapkan. Selain itu, melakukan pengujian dengan beberapa kasus pengujian dan mengidentifikasi sekaligus mencatat ketika ditemukan *bug* atau *error* pada sistem.

Selama tahap pengembangan *front end* Sistem Informasi Akuntansi terdapat pembagian rincian aktivitas yang dilakukan seperti pada Tabel 3.2.

Tabel 3.2 Aktivitas Sistem Informasi Akuntansi

No	Aktivitas	Waktu
1.	Melakukan <i>revamp front end</i> pada menu pengaturan Akun (COA).	1 Minggu
2.	Melakukan <i>revamp front end</i> pada menu pengaturan Tipe Transaksi.	1 Minggu
3.	Mengembangkan <i>front end</i> pada menu pengaturan Kode T.	1 Minggu
4.	Melakukan <i>revamp front end</i> pada menu pengaturan Periode.	2 Hari
5.	Melakukan <i>revamp front end</i> pada menu transaksi Manajemen Periode.	3 Hari
6.	Melakukan <i>revamp front end</i> pada menu transaksi jurnal.	1 Minggu
7.	Mengembangkan <i>front end</i> pada menu laporan <i>Account Listing</i> .	3 Hari
8.	Mengembangkan <i>front end</i> pada menu laporan <i>Journal Listing</i> .	2 Hari
9.	Mengembangkan <i>front end</i> pada menu laporan <i>Trial Balance</i> .	2 Hari
10.	Melakukan <i>bug fix</i> yang terdapat pada <i>website</i> .	1 Minggu

3.2.3 Pelaksanaan Proyek

Proyek Sistem Informasi Akuntansi telah dikembangkan sebelumnya. Proses untuk mengembangkan *front end* secara alur dan elemen yang dibutuhkan mengikuti desain *user interface* pada Figma. Namun, berdasarkan hasil *meeting* antara PM Geekgarden dengan pihak pelanggan meminta dilakukan *revamp* pada *front end* Sistem Informasi Akuntansi yang disesuaikan dengan logo identitas perusahaan pelanggan sehingga beberapa elemen di *revamp*

menjadi warna biru. Artinya, secara alur dan proses bisnis masih sama dengan desain *user interface* tetapi hanya beberapa elemen yang perlu di *revamp*. Selain itu, pada beberapa fitur memiliki perubahan komponen yang perlu disesuaikan dengan *document requirements specification*. Proses pengembangan dan *revamp front end* pada Sistem Informasi Akuntansi sebagai berikut:

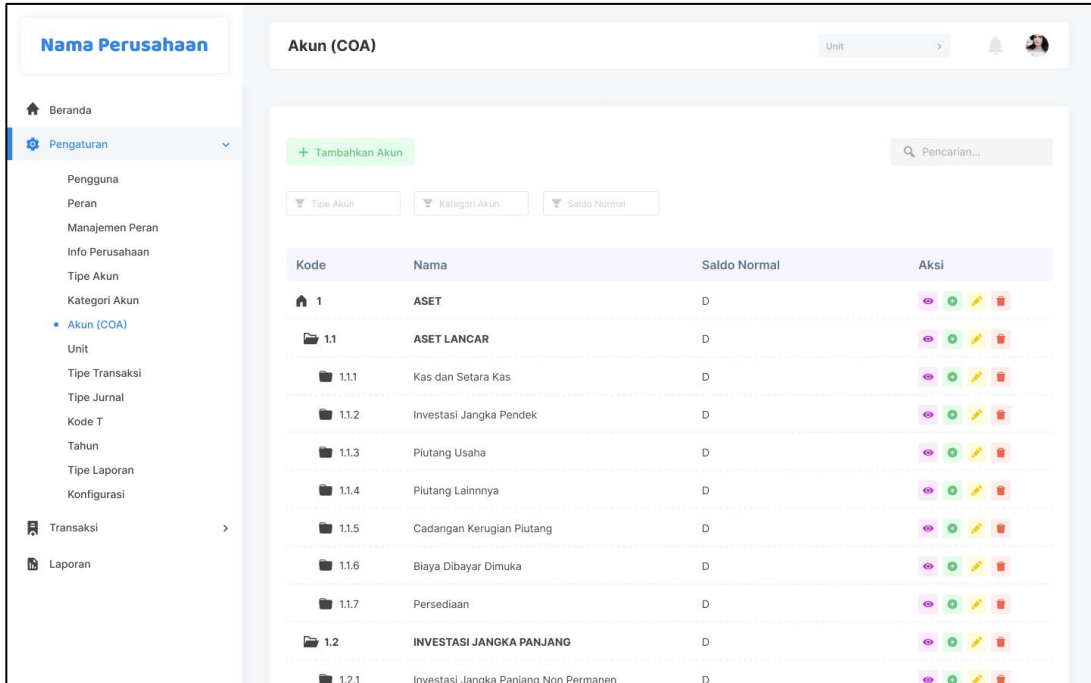
a. Melakukan *revamp front end* pada menu pengaturan Akun (COA)

Akun (COA) merupakan fitur pengaturan untuk mengelola akun. Data Akun (COA) memiliki relasi dengan menu transaksi dan laporan sehingga mendukung perubahan akun secara dinamis. Halaman pengaturan Akun (COA) digunakan untuk tambah akun, ubah akun, melihat detail akun, dan hapus akun. Seluruh data Akun (COA) dibuat dalam bentuk *input field* berjenis *dropdown* pada menu transaksi dan menu laporan untuk mengambil data Akun (COA) yang tersimpan pada sistem.

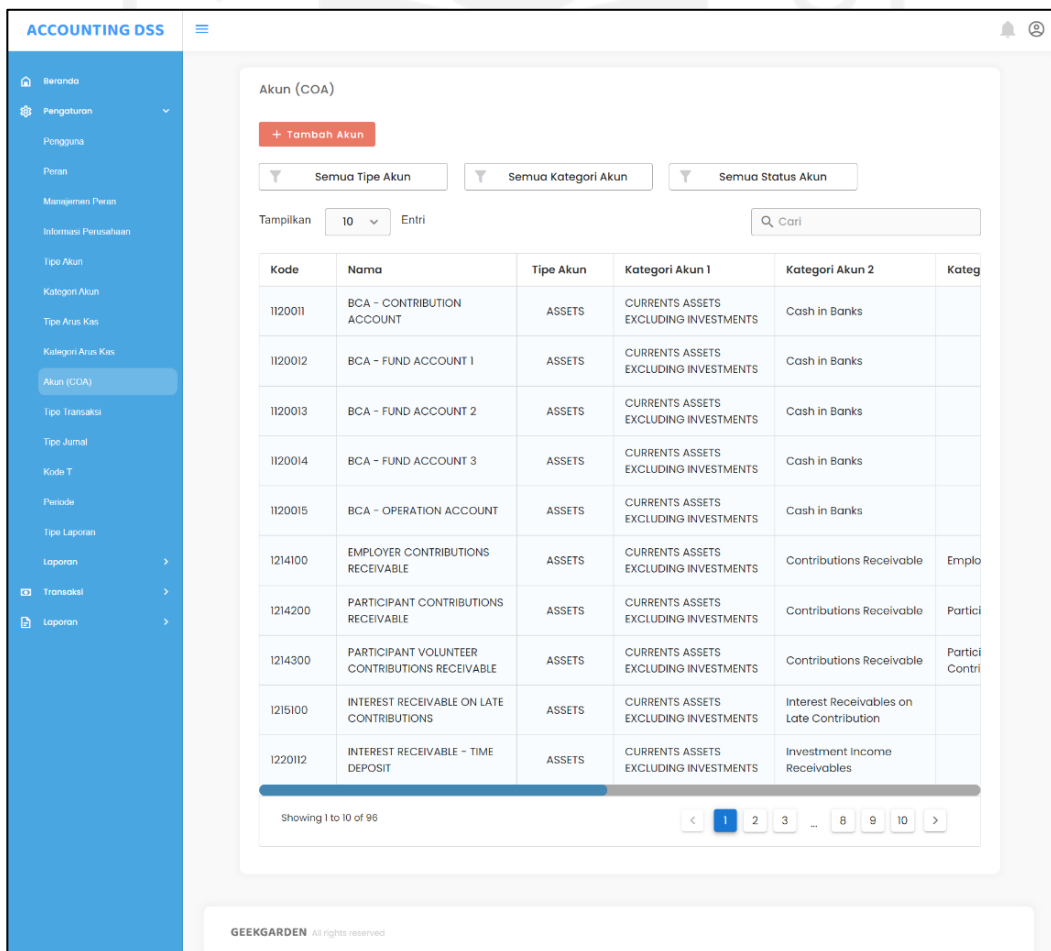
Fitur ini, sebelumnya telah dikembangkan sesuai dengan desain *user interface* yang terdapat pada Gambar 3.46. Seluruh tampilan antarmuka halaman Akun (COA) mendapatkan *revamp* diantaranya halaman utama atau *dashboard* Akun (COA), *pop up* tambah Akun (COA), *pop up* detail Akun (COA), dan *pop up* ubah Akun (COA).

Hasil tampilan antarmuka halaman Akun (COA) setelah dilakukan *revamp* dapat dilihat pada Gambar 3.47 dan Gambar 3.48. *Revamp* yang terdapat di halaman Akun (COA) diantaranya menambahkan, menghapus, dan mengganti *input field* sesuai dengan kebutuhan. Hasil *revamp* yang dilakukan pada halaman utama atau *dashboard* Akun (COA) diantaranya, yaitu:

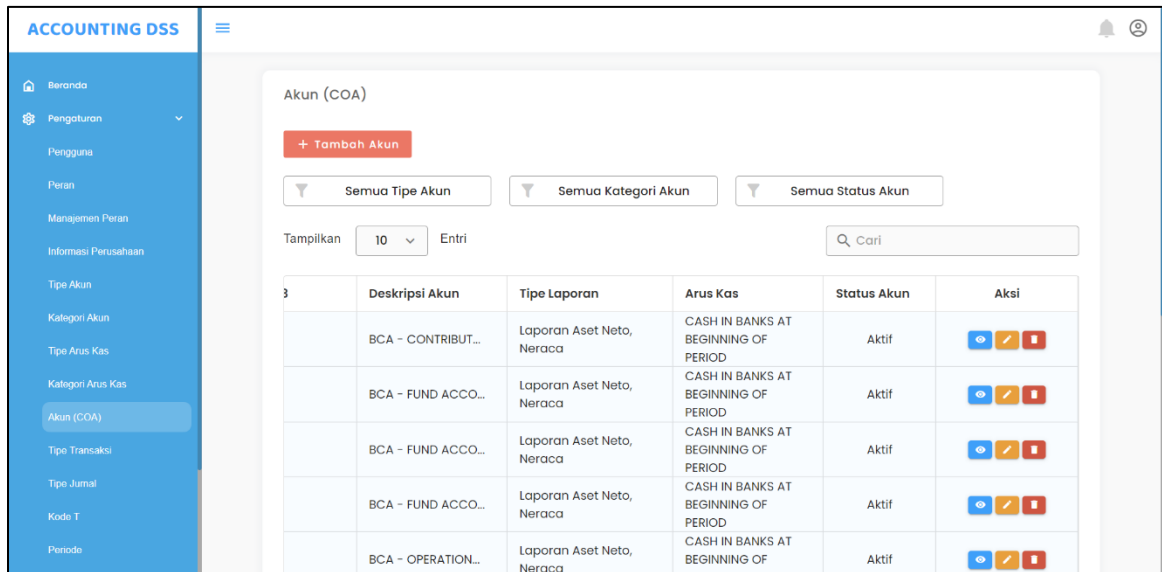
1. Menambahkan kolom Tipe Akun.
2. Mengubah kolom Kategori Akun dari statis (hanya memiliki satu kolom kategori akun) menjadi dinamis (memiliki lebih dari satu kolom kategori akun berdasarkan kategori akun yang ditambahkan atau diubah).
3. Menambahkan kolom Deskripsi Akun.
4. Menambahkan kolom Tipe Laporan.
5. Menambahkan kolom Arus Kas.
6. Menambahkan kolom Status Akun.
7. Menghapus kolom Saldo Normal.
8. Mengganti *filter* Saldo Normal menjadi *filter* Status Akun.



Gambar 3.46 Desain user interface Akun (COA)



Gambar 3.47 Tampilan antarmuka Akun (COA)



Gambar 3.48 Tampilan antarmuka Akun (COA) lanjutan

Tambah Akun (COA) adalah fitur yang digunakan pengguna untuk menambahkan Akun (COA) ke dalam sistem. Desain *user interface pop up* tambah Akun (COA) sebelum dilakukan *revamp* dapat dilihat pada Gambar 3.49. Terdapat beberapa *revamp* dan penambahan komponen yang dilakukan, sebagai berikut:

1. Menambahkan *input field* Tipe Laporan berjenis dropdown yang dapat dipilih lebih dari satu *value*. Data diambil dari halaman pengaturan Tipe Laporan sehingga data ditampilkan secara dinamis.
2. Menambahkan *input field* Status Akun berjenis dropdown dengan *value* Aktif dan Tidak Aktif.
3. Menambahkan *input field* Arus Kas berjenis dropdown yang dapat dipilih lebih dari satu *value*. Data diambil dari halaman pengaturan Kategori Arus Kas sehingga data ditampilkan secara dinamis.
4. Mengubah *input field* Kategori Akun yang sebelumnya statis (hanya terdapat satu dropdown Kategori Akun) menjadi dinamis (dapat menambahkan dan menghapus dropdown Kategori Akun) secara *default* memiliki satu dropdown Kategori Akun. *Input field* Kategori Akun ditambahkan fitur *search* untuk memudahkan dalam memilih data yang dapat dilihat pada Gambar 3.51. Data tersebut diambil secara dinamis dari halaman pengaturan Kategori Akun.
5. Menghapus *input field* Saldo Normal berjenis dropdown.

Hasil *revamp* tampilan *pop up* tambah Akun (COA) dapat dilihat pada Gambar 3.50 dan Gambar 3.51.

Tambahkan Akun

Kode

Nama

Tipe akun

Kategori akun

Saldo Normal

Deskripsi

Batal Simpan

Gambar 3.49 Desain *user interface* tambah Akun (COA)

Tambah Akun

Kode

Nama

Tipe Akun

Tipe Laporan

Status Akun

Arus Kas

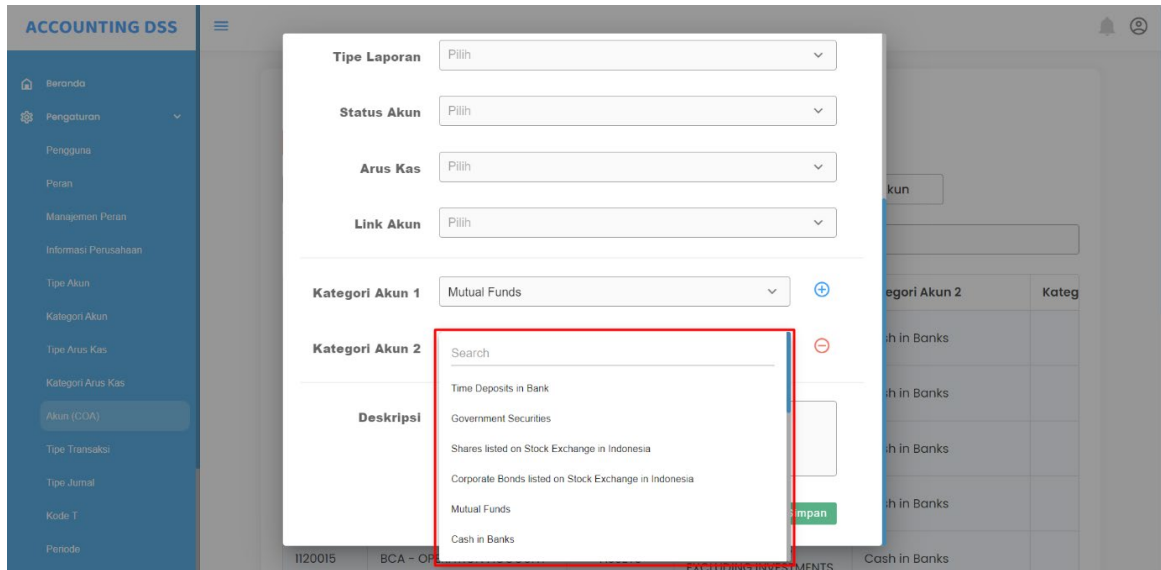
Link Akun

Kategori Akun 1 +

Deskripsi

Batal Simpan

Gambar 3.50 Tampilan *pop up* tambah Akun (COA)



Gambar 3.51 Tampilan Kategori Akun dinamis pada *pop up* tambah Akun (COA)

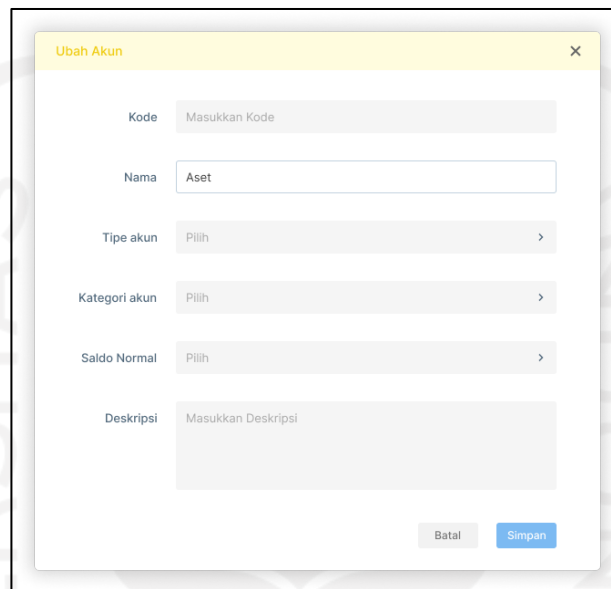
Ubah Akun (COA) adalah fitur yang digunakan pengguna ketika terdapat kesalahan dalam pengisian data atau perubahan data pada Akun (COA) yang telah disimpan ke dalam sistem. Tampilan *pop up* ubah Akun (COA) memiliki kesamaan dengan Gambar 3.50. Perbedaannya setiap *input field* pada *pop up* ubah Akun (COA) memiliki *value*, Gambar 3.50 secara *default* tidak memiliki *value* sedangkan pada *pop up* ubah Akun (COA) memiliki *value* berdasarkan data yang tersimpan sebelumnya. Gambar 3.52 merupakan desain *user interface* berbentuk prototipe *pop up* ubah Akun (COA). Terdapat beberapa *revamp* dan penambahan komponen yang dilakukan, yaitu:

1. Menambahkan *input field* Tipe Laporan berjenis dropdown yang dapat dipilih lebih dari satu *value*. Data diambil dari halaman pengaturan Tipe Laporan sehingga data ditampilkan secara dinamis.
2. Menambahkan *input field* Status Akun berjenis dropdown dengan *value* Aktif dan Tidak Aktif.
3. Menambahkan *input field* Arus Kas berjenis dropdown yang dapat dipilih lebih dari satu *value*. Data diambil dari halaman pengaturan Kategori Arus Kas sehingga data ditampilkan secara dinamis.
4. Mengubah *input field* Kategori Akun yang sebelumnya statis (hanya terdapat satu dropdown Kategori Akun) menjadi dinamis (dapat menambahkan dan menghapus dropdown Kategori Akun) secara *default* memiliki satu dropdown Kategori Akun. *Input field* Kategori Akun ditambahkan fitur *search* untuk memudahkan dalam

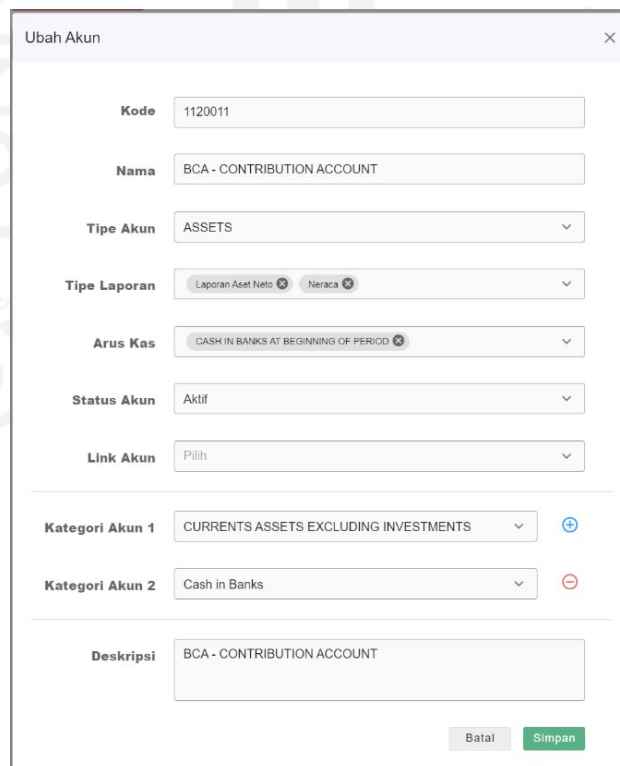
memilih data yang dapat dilihat pada Gambar 3.54. Data tersebut diambil secara dinamis dari halaman pengaturan Kategori Akun.

5. Menghapus *input field* Saldo Normal berjenis dropdown.

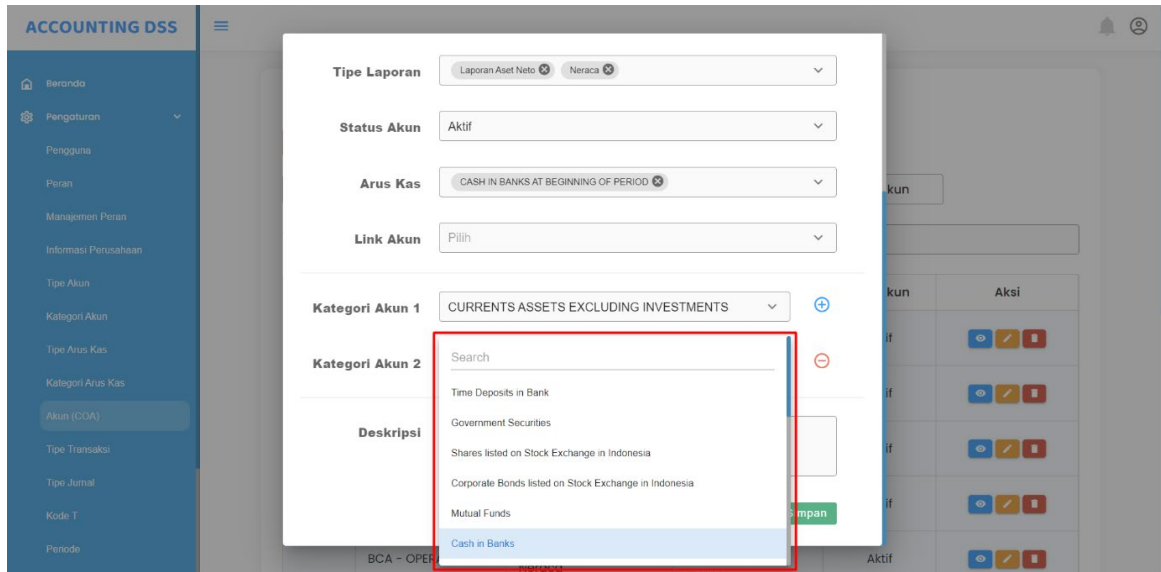
Hasil *revamp* tampilan *pop up* ubah akun (COA) dapat dilihat pada Gambar 3.53 dan Gambar 3.54.



Gambar 3.52 Desain *user interface* ubah Akun (COA)



Gambar 3.53 Tampilan *pop up* ubah Akun (COA)



Gambar 3.54 Tampilan Kategori Akun dinamis pada *pop up* ubah Akun (COA)

Detail Akun (COA) merupakan fitur yang digunakan untuk melihat atau menampilkan data Akun (COA) secara rinci. Pengguna dapat mengakses *pop up* detail Akun (COA) dapat menekan tombol berbentuk mata berwarna biru pada data yang dipilih. Tampilan *pop up* detail Akun (COA) memiliki kesamaan dengan tampilan *pop up* tambah Akun (COA) seperti pada Gambar 3.50 dan tampilan *pop up* ubah Akun (COA) pada Gambar 3.53.

Perbedaannya terletak pada *input field* detail Akun (COA). Seluruh *input field* dibuat *readonly* sehingga *value input field* tersebut tidak dapat di modifikasi oleh pengguna tetapi pengguna dapat menyalin *value* pada *input field* yang dipilih jika diperlukan. Gambar 3.55 merupakan desain *user interface* berbentuk prototipe *pop up* detail Akun (COA). *Revamp* yang dilakukan meliputi:

1. Menambahkan *input field* Tipe Laporan yang dapat menampilkan lebih dari satu *value*.
2. Menambahkan *input field* Status Akun.
3. Menambahkan *input field* Arus Kas yang dapat menampilkan lebih dari satu *value*.
4. Mengubah *input field* Kategori Akun yang sebelumnya statis (hanya terdapat satu Kategori Akun) menjadi dinamis (lebih dari satu Kategori Akun).
5. Menghapus *input field* Saldo Normal.

Tampilan *pop up* detail Akun (COA) setelah dilakukan *revamp* dan penambahan komponen dapat dilihat pada Gambar 3.56.

Gambar 3.55 Desain *user interface* detail Akun (COA)

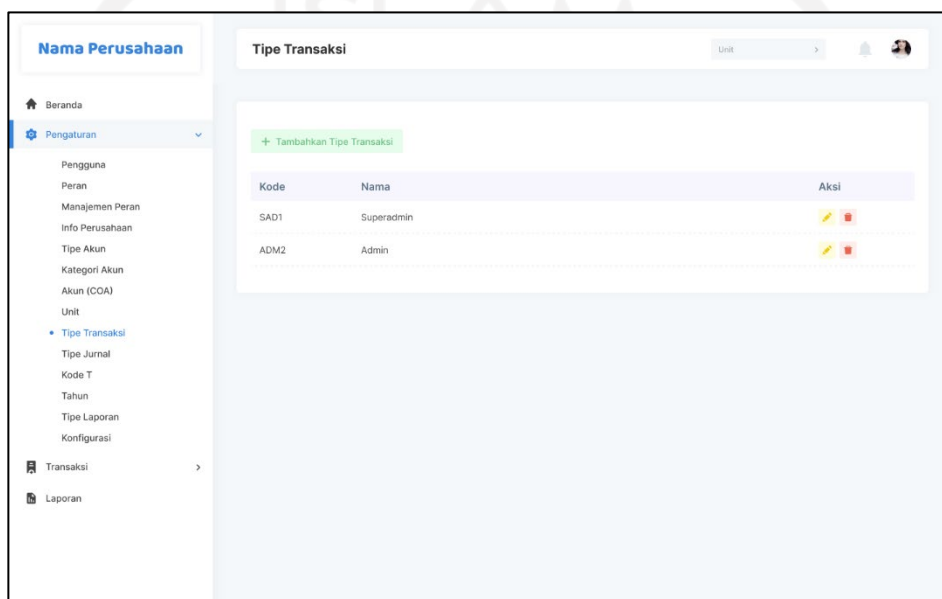
Gambar 3.56 Tampilan *pop up* detail Akun (COA)

- b. Melakukan *revamp front end* pada menu pengaturan Tipe Transaksi

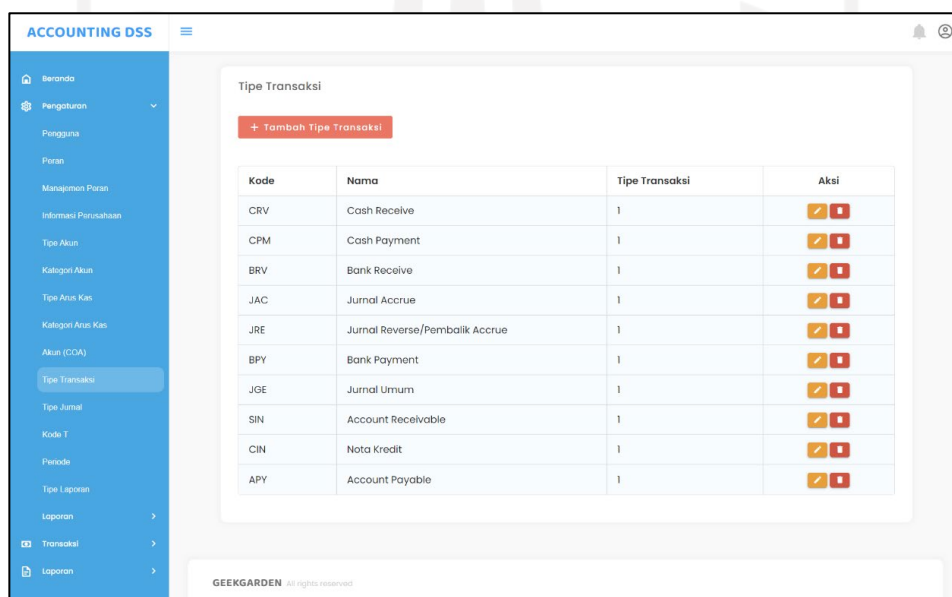
Tipe Transaksi merupakan pengaturan untuk mengelola tipe transaksi. Data Tipe Transaksi memiliki relasi dengan fitur transaksi jurnal dan fitur laporan *account listing* sehingga data ditampilkan secara dinamis. Halaman pengaturan Tipe Transaksi digunakan

untuk tambah, ubah dan hapus Tipe Transaksi. Fitur ini telah dikembangkan sebelumnya sesuai dengan desain *user interface* yang terdapat pada Gambar 3.57. Seluruh tampilan antarmuka halaman tipe transaksi mendapatkan *revamp* diantaranya halaman utama tipe transaksi, *pop up* tambah tipe transaksi dan *pop up* ubah tipe transaksi.

Gambar 3.58 merupakan hasil *revamp* yang dilakukan pada pengaturan tipe transaksi. *Revamp* yang terdapat di halaman utama tipe transaksi, yaitu: menambahkan kolom tipe transaksi pada tabel.



Gambar 3.57 Desain *user interface* halaman pengaturan tipe transaksi



Gambar 3.58 Tampilan antarmuka halaman pengaturan tipe transaksi

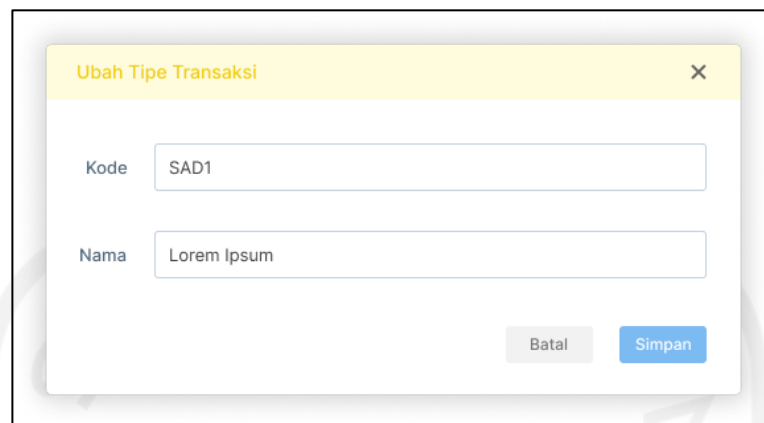
Tambah Tipe Transaksi adalah fitur untuk menambahkan tipe transaksi. Gambar 3.59 merupakan desain *user interface* tambah tipe transaksi sebelum dilakukan *revamp*. Proses *revamp* dilakukan dengan menambahkan *input field* bertipe *radio button* dengan *value* tipe transaksi 1 dan tipe transaksi 2. Hasil *revamp* pada tampilan antarmuka pengaturan tipe transaksi dapat dilihat pada Gambar 3.60.

Gambar 3.59 Desain *user interface* tambah tipe transaksi

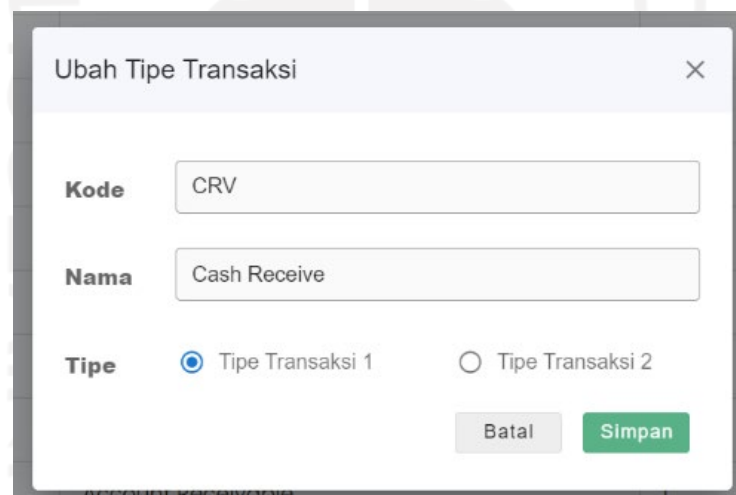
Gambar 3.60 Tampilan *pop up* tambah tipe transaksi

Ubah Tipe Transaksi adalah fitur yang digunakan ketika terjadi perubahan data pada tipe transaksi yang telah disimpan ke dalam sistem. Tampilan *pop up* ubah tipe transaksi memiliki kesamaan dengan *pop up* tambah tipe transaksi pada Gambar 3.60. Perbedaannya setiap *input field* memiliki *value*, Gambar 3.60 secara *default* tidak memiliki *value* sedangkan pada *pop up* ubah tipe transaksi memiliki *value* berdasarkan data yang tersimpan sebelumnya. *Revamp* dilakukan dengan menambahkan *input field* bertipe *radio button* dengan *value* tipe transaksi 1 dan tipe transaksi 2. Gambar 3.61 merupakan desain

user interface berbentuk prototipe *pop up* ubah tipe transaksi. Hasil *revamp* tampilan *pop up* ubah tipe transaksi dapat dilihat pada Gambar 3.62.



Gambar 3.61 Desain *user interface* ubah tipe transaksi



Gambar 3.62 Tampilan *pop up* ubah tipe transaksi

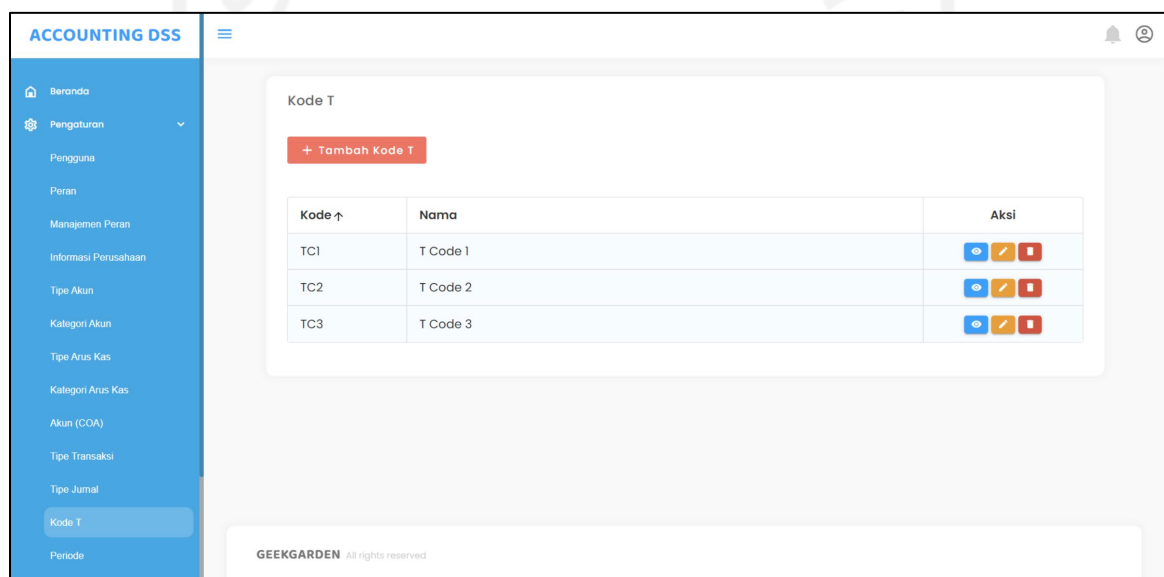
c. Mengembangkan *front end* pada menu pengaturan Kode T

Kode T adalah fitur untuk menambahkan, mengubah, menghapus, dan melihat detail data Kode T. Data Kode T dibuat dalam bentuk *input field* berjenis *dropdown* pada menu transaksi jurnal untuk mengambil data Kode T yang tersimpan pada sistem. Gambar 3.63 merupakan hasil pengembangan tampilan antarmuka halaman pengaturan Kode T untuk menampilkan seluruh Kode T ke dalam bentuk tabel dengan kolom kode dan nama. Berikut fungsi lain yang dapat digunakan pada halaman pengaturan Kode T:

1. Tombol Tambah Kode T.
2. Tombol Detail Kode T (berbentuk ikon mata).
3. Tombol Ubah Kode T (berbentuk ikon pensil).

4. Tombol Hapus Kode T (berbentuk ikon tempat sampah).
5. *Sorting* pada kolom Kode ditandai dengan tanda panah setelah tulisan Kode. *Sorting* dapat dilakukan secara *ascending* (pengurutan data dari terkecil ke terbesar) dan *descending* (pengurutan data dari terbesar ke terkecil).
6. *Sorting* pada kolom Nama ditandai dengan tanda panah setelah tulisan Nama. *Sorting* dapat dilakukan secara *ascending* (pengurutan data dari terkecil ke terbesar) dan *descending* (pengurutan data dari terbesar ke terkecil).

Selain itu, proses integrasi API untuk melakukan *request* dengan *back end* menggunakan *library* Axios method GET dapat dilihat pada Gambar 3.64.



Gambar 3.63 Tampilan antarmuka Kode T

```
import Vue from "vue";
import Vuex from "vuex";
import axios from "axios";

Vue.use(Vuex);

const tCodeModule = {
  state: {
    tCode: [],
  },
  mutations: {
    setTCode(state, payload) {
      state.tCode = payload;
    },
  },
  actions: {
    async getTCodeData({ commit }) {
      try {
        let res = await axios.get(process.env.VUE_APP_API_URL +
"/tCode");
```

```

    let data = res.data.data;
    commit("setTCode", data);
  } catch (error) {
    console.log(error);
  }
},
},
},
getters: {
  getTCode(state) {
    return state.tCode;
  },
},
},
};

export default tCodeModule;

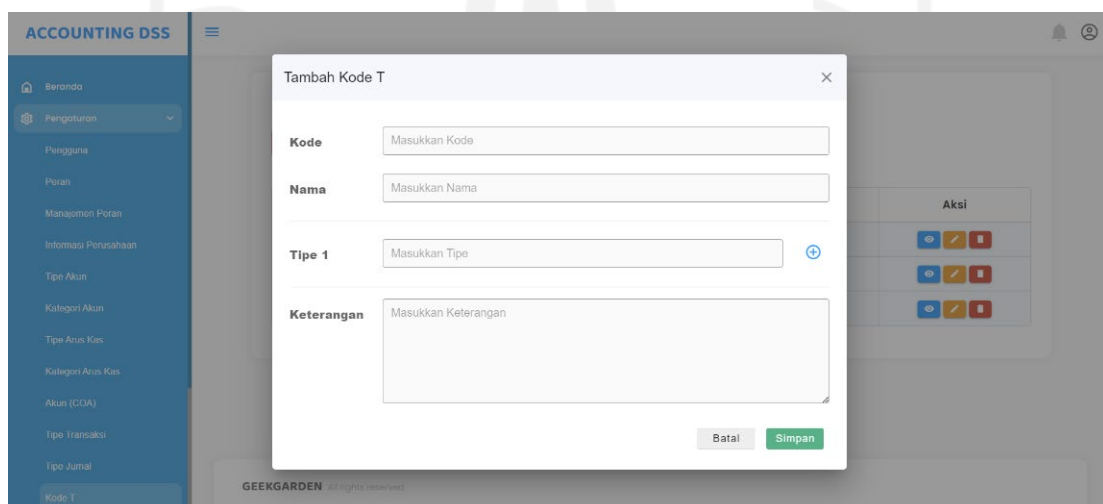
```

Gambar 3.64 Integrasi API Kode T

Tambah Kode T adalah fitur untuk menambahkan data Kode T ke dalam sistem. Fitur ini dibuat untuk menampilkan *pop up* Tambah Kode T yang memiliki beberapa *input field* diantaranya, yaitu:

1. *Input field* Kode berjenis text yang hanya bisa diisi maksimal dengan 3 huruf.
2. *Input field* Nama berjenis text.
3. *Input field* Tipe berjenis text secara dinamis (dapat menambahkan dan menghapus *input field* Tipe) secara *default* memiliki satu *input field* Tipe.
4. *Input field* Keterangan berjenis *text area*.

Gambar 3.65 merupakan tampilan *pop up* tambah Kode T. Seluruh data akan dikirimkan melalui API dengan *back end* menggunakan *library* Axios method POST.

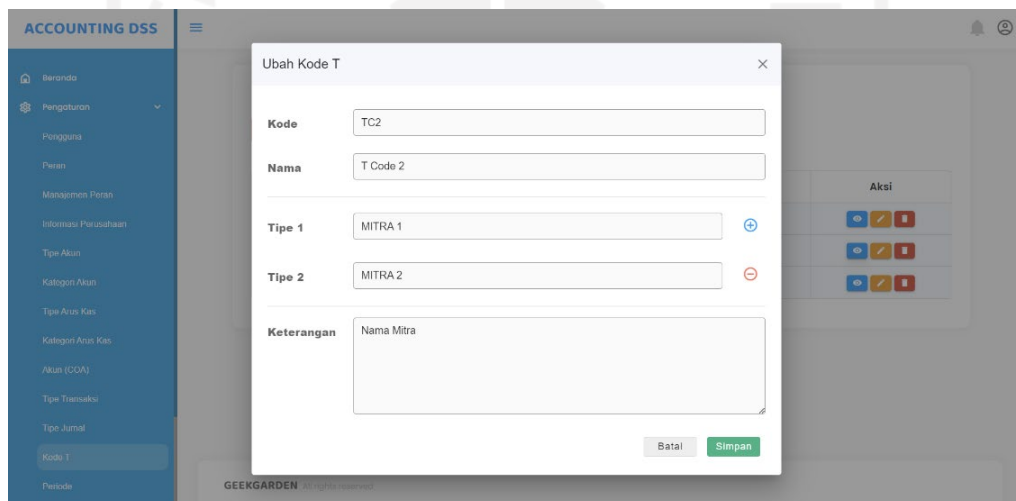
Gambar 3.65 Tampilan *pop up* tambah Kode T

Ubah Kode T adalah fitur yang digunakan ketika terjadi perubahan data Kode T pada sistem. Tampilan *pop up* ubah Kode T memiliki kesamaan dengan Gambar 3.65. Pada

Gambar 3.65 secara *default* tidak memiliki *value* sedangkan pada *pop up* ubah Kode T memiliki *value* berdasarkan data yang telah tersimpan sebelumnya. Fitur ubah Kode T memiliki beberapa *input field*, yaitu:

1. *Input field* Kode berjenis text yang hanya bisa diisi maksimal dengan 3 huruf.
2. *Input field* Nama berjenis text.
3. *Input field* Tipe berjenis text secara dinamis (dapat menambahkan dan menghapus *input field* Tipe) secara *default* memiliki satu *input field* Tipe.
4. *Input field* Keterangan berjenis *text area*.

Hasil pengembangan tampilan *pop up* ubah Kode T terdapat pada Gambar 3.66. Seluruh perubahan data dikirimkan melalui API dengan *back end* menggunakan *library* Axios method PUT.



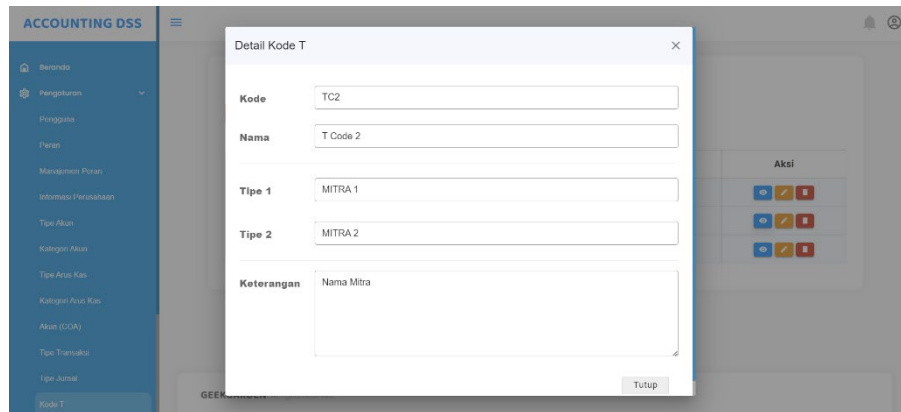
Gambar 3.66 Tampilan *pop up* ubah Kode T

Detail Kode T merupakan fitur untuk melihat data Kode T yang dipilih secara rinci. Tampilan *pop up* detail Kode T memiliki kesamaan dengan Gambar 3.65 dan Gambar 3.66. Perbedaannya terletak pada *input field* Detail Kode T. Seluruh *input field* dibuat *readonly* sehingga *value input field* tersebut tidak dapat di modifikasi oleh pengguna tetapi pengguna dapat menyalin *value* pada *input field* yang dipilih jika diperlukan. Fitur detail Kode T memiliki beberapa *input field*, yaitu:

1. *Input field* Kode.
2. *Input field* Nama.
3. *Input field* Tipe berjenis text secara dinamis (dapat menampilkan lebih dari satu Tipe) secara *default* memiliki satu *input field* Tipe.

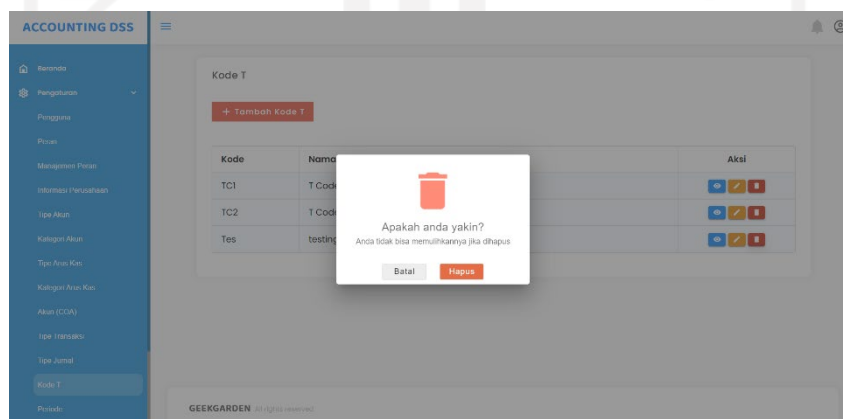
4. *Input field* Keterangan.

Gambar 3.67 merupakan hasil pengembangan tampilan *pop up* detail Kode T. Proses integrasi digunakan untuk melakukan *request* API dengan *back end* menggunakan *library* Axios method GET sehingga data ditampilkan ke dalam *pop up* detail Kode T.



Gambar 3.67 Tampilan *pop up* detail Kode T

Hapus Kode T merupakan fitur untuk menghapus data Kode T ketika pengguna menekan tombol ikon berbentuk tempat sampah berwarna merah pada tabel. Masing-masing tombol hapus memiliki id berdasarkan data di setiap barisnya sehingga data yang dipilih akan mengirimkan *request* ke *back end* menggunakan axios dengan method DELETE. Tampilan *pop up* hapus data dapat dilihat pada Gambar 3.68.



Gambar 3.68 Tampilan *pop up* delete

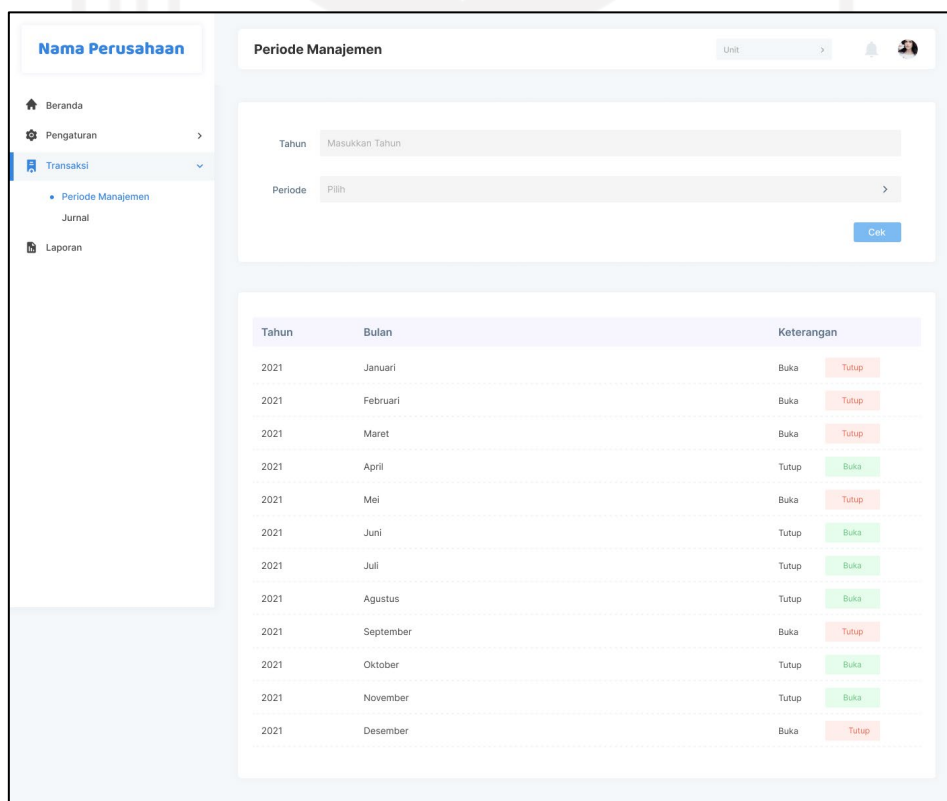
- d. Melakukan *revamp front end* pada menu pengaturan periode.

Pengaturan periode merupakan fitur untuk mendukung proses akuntansi yang terjadi pada sistem. Data periode berisikan bulan dan tahun memiliki relasi dengan menu transaksi manajemen periode, menu transaksi jurnal, menu laporan *account listing*, menu

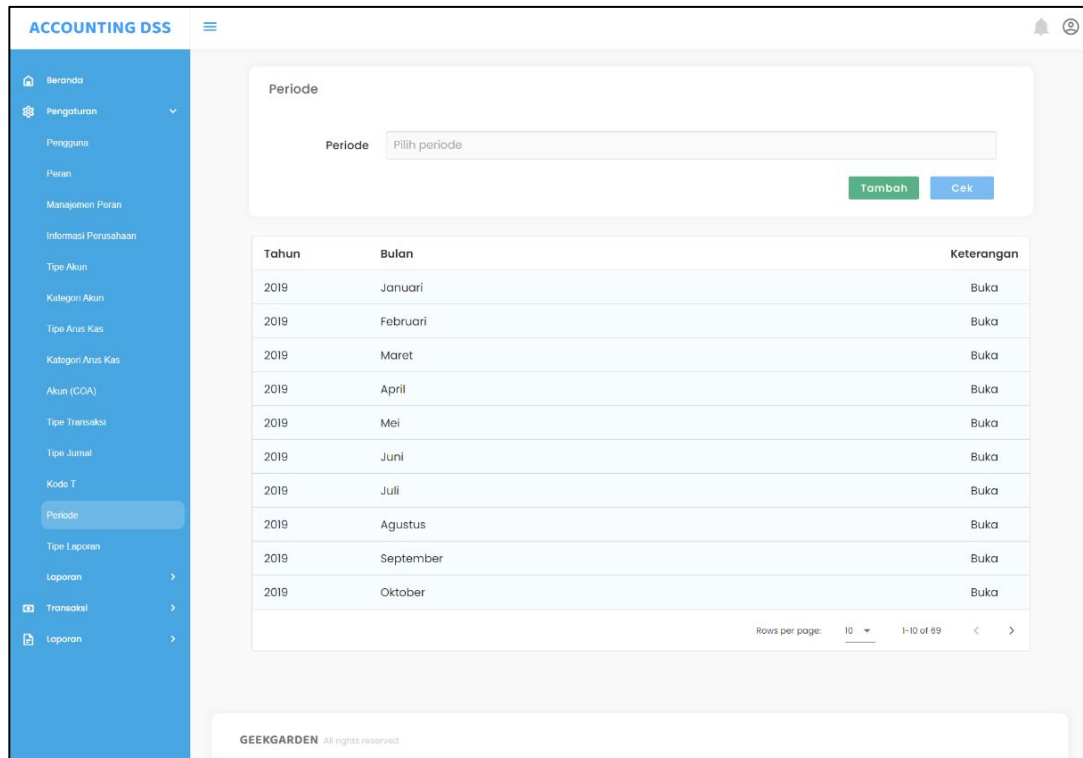
laporan *trial balance*. Pada menu transaksi jurnal, menu laporan *account listing*, dan menu laporan *trial balance* data periode dibuat dalam bentuk *dropdown* dengan *value* bulan, periode dapat ditampilkan ketika pengguna telah memilih tahun yang tersedia. Apabila pada menu lainnya pengguna memilih tahun yang tidak ada didalam sistem maka periode akan menampilkan pesan “Tidak ada data yang tersedia”.

Halaman ini sebelumnya memiliki dua *input field*, yaitu: tahun dan bulan periode kemudian dilakukan *revamp* pada *input field* yang awalnya berjumlah dua *input field* menjadi satu *input field* sehingga proses penambahan periode menjadi lebih efisien. Setelah dilakukan *revamp* halaman periode hanya memiliki satu *input field* periode dengan format YYYYMM, *input field* period menggunakan *library* vue2-datepicker untuk menambahkan periode dan menampilkan seluruh periode yang tersimpan di dalam sistem.

Periode yang sebelumnya telah ditambahkan datanya tidak akan tersimpan karena maksimal dalam satu periode hanya memiliki 12 bulan sesuai kalender dan menampilkan pemberitahuan bahwa periode tersebut telah tersedia di dalam sistem. Selain itu, periode yang telah ditambahkan tidak dapat dihapus. Gambar 3.70 merupakan tampilan antarmuka pada menu pengaturan periode.



Gambar 3.69 Desain *user interface* Manajemen Periode

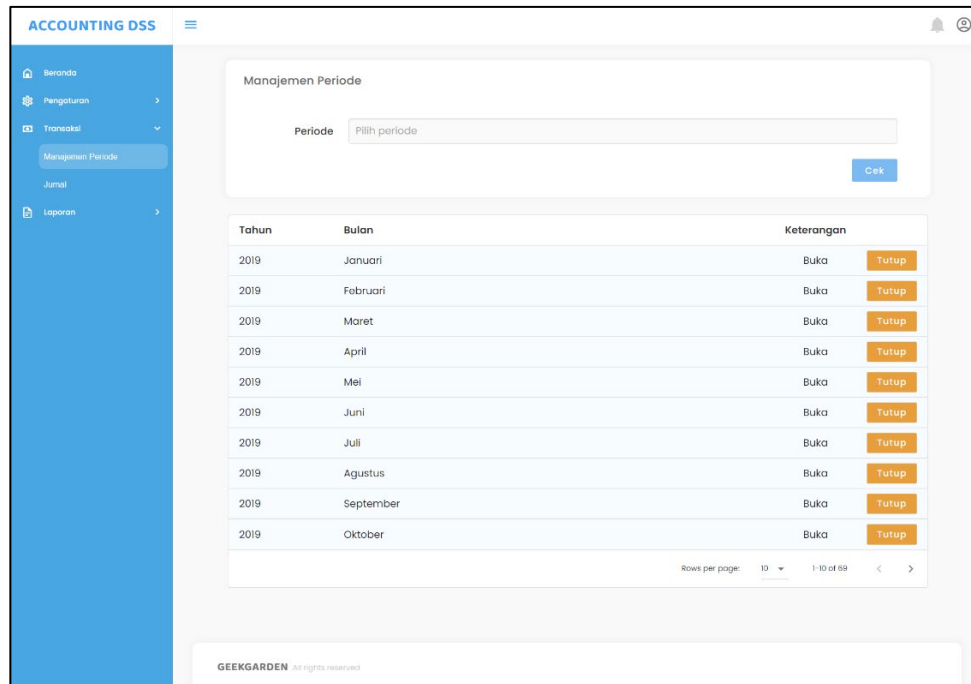


Gambar 3.70 Tampilan antarmuka pengaturan periode

e. Melakukan *revamp front end* pada menu transaksi halaman manajemen periode

Manajemen periode merupakan fitur untuk mengelola periode tertentu yang harus dibuka dan ditutup. Sebelumnya tampilan antarmuka manajemen periode telah dikembangkan tetapi mendapatkan *revamp* yang disesuaikan dengan Gambar 3.70. Desain *user interface* manajemen periode sebelum di *revamp* memiliki dua *input field*, yaitu: *input field* tahun dengan format YYYY dan *input field* periode dengan format MM seperti pada Gambar 3.69.

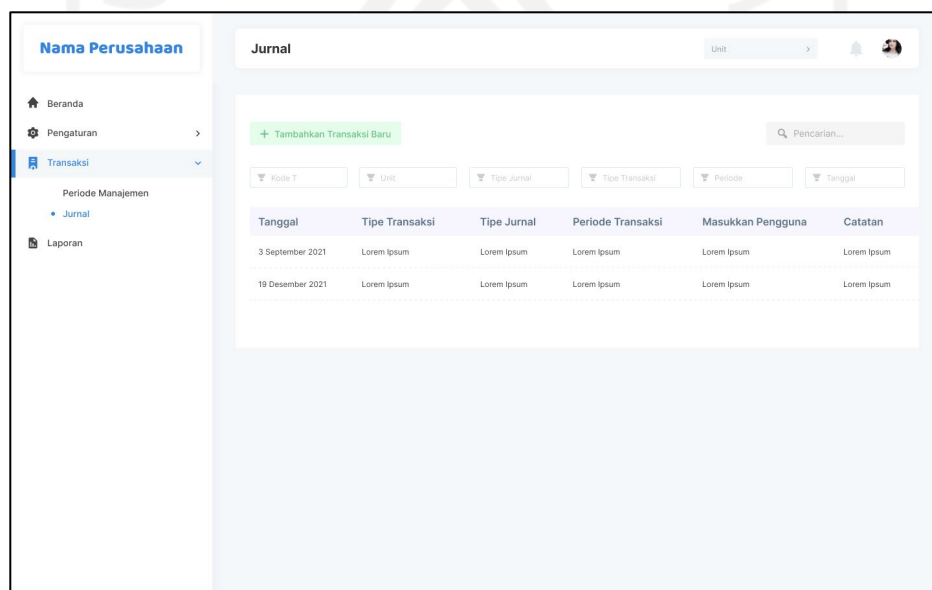
Gambar 3.71 merupakan tampilan antarmuka setelah dilakukan *revamp* dengan menggabungkan *input field* tahun dan *input field* periode menjadi satu *input field* bernama periode dengan format YYYYMM menggunakan *library vue2-datepicker*. Pada halaman manajemen periode memiliki tombol Cek untuk menampilkan seluruh data periode yang tersimpan di dalam sistem. Selain itu, pengaturan periode memiliki relasi dengan manajemen periode sehingga data yang ditampilkan sama dengan fungsionalitas berbeda. Gambar 3.70 digunakan untuk menambahkan periode dan menampilkan seluruh data periode tanpa bisa membuka tutup periode sedangkan Gambar 3.71 untuk menampilkan data periode dan membuka tutup periode sesuai kebutuhan.



Gambar 3.71 Tampilan antarmuka Manajemen Periode

f. Melakukan *revamp front end* pada menu transaksi jurnal.

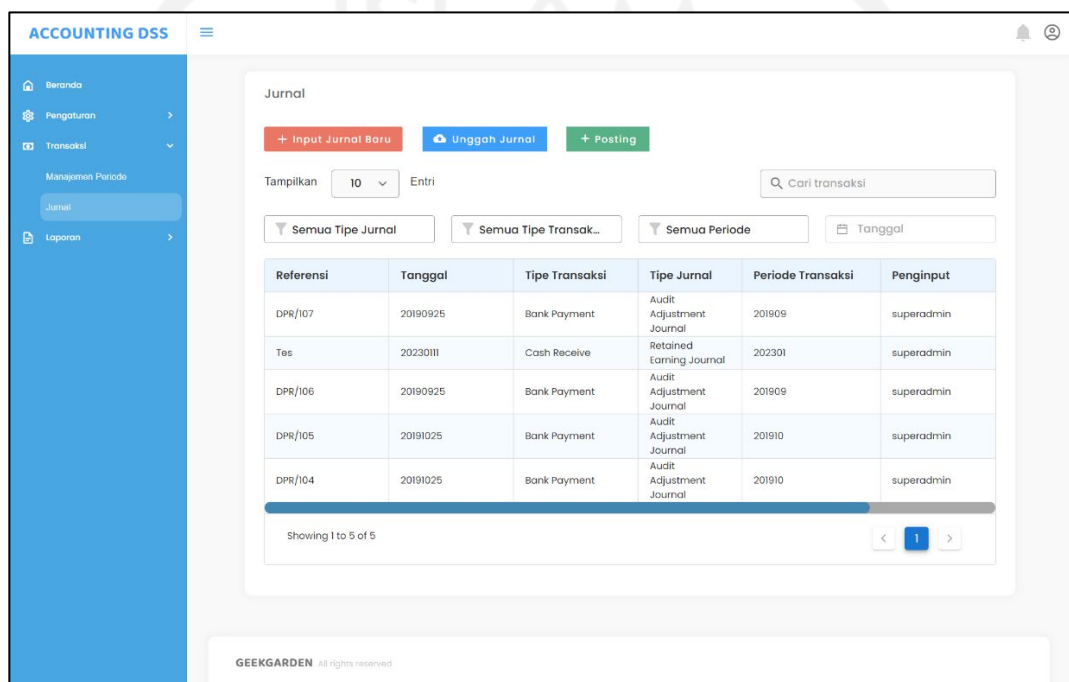
Transaksi jurnal merupakan fitur untuk melakukan seluruh aktivitas yang berkaitan dengan transaksi jurnal. Halaman transaksi jurnal digunakan untuk menambahkan, menampilkan, mengubah, dan menghapus data jurnal. Sebelumnya tampilan antarmuka transaksi jurnal telah dikembangkan sesuai desain prototipe pada Gambar 3.72 tetapi memiliki beberapa perubahan komponen sehingga perlu dilakukan *revamp*.



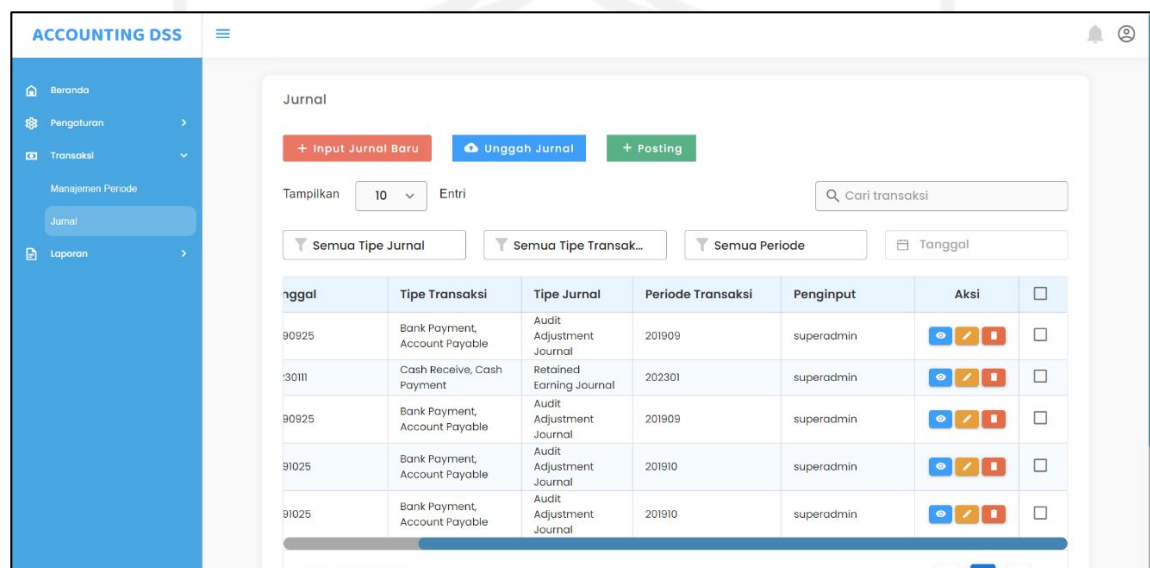
Gambar 3.72 Desain *user interface* halaman transaksi jurnal

Revamp dilakukan untuk memenuhi perubahan kebutuhan sistem yang terjadi sehingga dapat berjalan sesuai dengan harapan. Halaman utama transaksi jurnal memiliki beberapa *revamp* diantaranya, yaitu: menambahkan fitur unggah jurnal, menghapus *filter* Kode T, menghapus *filter* Unit, menambahkan kolom Referensi pada tabel, dan menghapus kolom Catatan pada tabel.

Gambar 3.73 dan Gambar 3.74 merupakan tampilan antarmuka yang telah dilakukan *revamp* sesuai dengan kebutuhan sistem.



Gambar 3.73 Tampilan antarmuka halaman transaksi jurnal



Gambar 3.74 Tampilan antarmuka halaman transaksi jurnal lanjutan

Unggah jurnal merupakan fitur untuk memudahkan staf akuntansi dalam menambahkan jurnal dengan cara mengunggah data jurnal yang disimpan dalam bentuk *files* ke dalam sistem sehingga tidak perlu mengisikan data secara manual. Pembuatan fitur unggah jurnal memanfaatkan komponen *button* yang tersedia di *library* Vuetify. Fitur ini hanya dapat menerima *files* dengan format *.txt*.

Pengguna yang mengunggah jurnal menyiapkan *files* berisi data sesuai dengan permintaan sistem kemudian menekan tombol unggah jurnal maka sistem akan mengarahkan untuk memilih *files* yang tersimpan di perangkat. Apabila tipe *files* yang diunggah memiliki *format* selain *.txt* maka sistem menampilkan *pop up error* dengan keterangan *error*. Namun ketika *format files* dan isi data *file* sesuai dengan yang diminta sistem maka muncul *pop up success*. Gambar 3.75 merupakan proses integrasi API untuk melakukan *request* yang dikirimkan ke *back end* agar fungsionalitas *button* unggah jurnal dapat berjalan sesuai yang diharapkan.

```

<script>
import BaseButton from "../base/BaseButton.vue";
import { mapActions, mapGetters } from "vuex";
import axios from "axios";

export default {
  name: "JournalList",
  components: { BaseButton },
  data: () => ({
    fileUpload: [],
  }),
  computed: {
    ...mapGetters([
      "getListTransaction",
    ]),
  },
  methods: {
    ...mapActions([
      "getListTransactionData",
    ]),
    actionUpload() {
      this.$refs.file.value = "";
      let formData = new FormData();
      formData.append("file", this.fileUpload[0]);
      axios
        .post(
          process.env.VUE_APP_API_URL + "/upload-api/upload-file",
          formData,
          {
            headers: { "Content-Type": "multipart/form-data" },
          }
        )
        .then((resp) => {
          console.log(resp);
          this.getSnackbarAdd = true;
        });
    }
  }
};

```

```

        this.$store.dispatch("getListTransactionData", formData);
    })
    .catch((err) => {
        console.log(err.response);
        this.fileUpload.splice(0, this.fileUpload.length);
        this.isShowError = true;
        const errorMessage = err.response.data;
        if (typeof errorMessage == "object") {
            if (typeof errorMessage.mesagge != "object") {
                this.textError = errorMessage.message;
            } else {
                this.textArray = errorMessage.mesagge;
            }
        } else {
            this.textError = "Something went wrong!";
        }
    });
    },
    },
</script>

```

Gambar 3.75 Integrasi API unggah jurnal

Tambah Jurnal Baru merupakan fitur untuk menambahkan transaksi jurnal baru. Salah satu proses akuntansi yang terdapat pada sistem adalah tambah jurnal baru. Tambah jurnal dilakukan ketika adanya pengeluaran atau pemasukan keuangan pada perusahaan sehingga seluruh aktivitas akuntansi dapat diketahui. Fitur ini telah dikembangkan sebelumnya tetapi memerlukan *revamp* pada tampilan dan komponen diantaranya, yaitu:

1. Menambahkan *input field* Referensi berjenis Text.
2. Menghapus *input field* Kode berjenis Text.
3. Menghapus *input field* Unit berjenis dropdown.
4. Perubahan *input field* Memo menjadi Deskripsi berjenis Text.
5. Menghapus fitur lampiran gambar.
6. Perubahan *input field* *Out of Balance* menjadi *Different*.
7. Mengubah *value input field* Kode T dari berjenis Text menjadi dropdown dengan fitur *search* untuk memudahkan dalam memilih Tipe yang dapat dilihat pada Gambar 3.80. Data diambil dari halaman pengaturan Kode T berdasarkan Tipe sehingga data ditampilkan secara dinamis.

Gambar 3.76 merupakan desain *user interface* berbentuk prototipe halaman tambah transaksi jurnal yang digunakan untuk menambahkan jurnal baru ke dalam sistem.

Gambar 3.76 Desain *user interface* tambah transaksi jurnal

Untuk memenuhi kebutuhan *revamp* pada halaman tambah transaksi jurnal baru perlu adanya *library* tambahan, yaitu: *vue2-datepicker*, dan *vue-numeric*. *Library* *vue2-datepicker* digunakan untuk memudahkan pengguna dalam memilih kalender secara lengkap. Selain itu, *vue2-datepicker* memiliki kemudahan bagi pengembang untuk mengkustomisasi *field* *datepicker* sesuai kebutuhan. Proses instalasi *library* *vue2-datepicker* dilakukan dengan mengetikkan: `npm i vue2-datepicker` ke terminal yang terdapat di Visual Studio Code. Gambar 3.77 merupakan implementasi *library* *vue2-datepicker* untuk membuat komponen *date* pada halaman tambah transaksi jurnal.

```
<template>
  <date-picker
    v-model="journal.date"
    type="date"
    format="YYYYMMDD"
    @change="onChangeDate"
    @clear="onClearDate"
    :disabled="journal.period == '' ? true : false"
    :popup-style="{
      top: '365px',
      left: '420px',
    }"
    placeholder="Masukkan tanggal"
```

```

    class="journal-datepicker mt-2"
  >
  </date-picker>
</template>
<script>
import DatePicker from "vue2-datepicker";
import "vue2-datepicker/index.css";

export default {
  name: "JournalAdd",
  components: { DatePicker },
  data: () => ({
    journal: {
      date: "",
    },
  }),
</script>

```

Gambar 3.77 Implementasi *library* vue2-datepicker

Library vue-numeric digunakan ketika pengguna mengetikkan angka pada *input field* maka secara otomatis *value* yang ditampilkan berubah ke dalam nilai mata uang. Kebutuhan sistem memerlukan nilai mata uang IDR atau rupiah sehingga *input field* akan dikustomisasi ke dalam mata uang tersebut. Proses instalasi *library* vue-numeric dilakukan dengan mengetikkan: `npm i vue-numeric` ke terminal yang terdapat di Visual Studio Code. Gambar 3.78 merupakan penggunaan *library* vue-numeric untuk membuat komponen *input field* pada halaman tambah transaksi sedangkan hasil *revamp* tampilan antarmuka tambah transaksi jurnal dapat dilihat pada Gambar 3.79 dan Gambar 3.80.

```

<template>
  <td class="text-center">
    <input class="rupiah-input" placeholder="Rp" disabled />
    <vue-numeric
      v-model="accountOne.debet"
      :empty-value="0"
      separator="."
      v-bind:precision="2"
      class="debit-input"
    >
  </vue-numeric>
  </td>
</template>
<script>
import VueNumeric from "vue-numeric";

export default {
  name: "JournalAdd",
  components: { VueNumeric },
  data: () => ({
    accountOne: {
      debet: "",
    },
  }),
</script>

```

</script>

Gambar 3.78 Implementasi *library* vue-numeric

ACCOUNTING DSS

Tambah Transaksi Baru

Referensi: DPR/107 Tipe Transaksi: Bank Payment, Account Payable

Periode: 201909 Tipe Jurnal: Audit Adjustment Journal

Tanggal: 20190925

Penginput: superadmin

Detail Transaksi

Kode Akun	Nama	Debit	Kredit	Deskripsi	T Code
1120012	BCA - FUND ACCOUNT 1	Rp. 20.000.000,00	Rp. 0,00	Transaction report from file	DF
1120013	BCA - FUND ACCOUNT 2	Rp. 30.000.000,00	Rp. 0,00	Transaction report from file	DF
Total		Rp. 50.000.000,00	Rp. 0		
Different		Rp. 50.000.000,00			

Batal Simpan

GEEKGARDEN All rights reserved

Gambar 3.79 Tampilan antarmuka tambah transaksi jurnal

ACCOUNTING DSS

Tambah Transaksi Baru

Referensi: 001/JU/I/2022 Tipe Transaksi: Pilih

Periode: Pilih Tipe Jurnal: Pilih

Tanggal: Masukkan tanggal

Penginput: superadmin

Detail Transaksi

Kredit	Deskripsi	T Code 1	T Code 2
Rp.		DPPK	
Rp.			
Rp. 0	Rp. 0		
Rp. 0			

Batal Simpan

GEEKGARDEN All rights reserved

Gambar 3.80 Tampilan dropdown Kode T pada tambah transaksi jurnal

Ubah Transaksi Jurnal merupakan fitur untuk mengubah data transaksi jurnal yang telah tersimpan ke dalam sistem. Tampilan antarmuka ubah jurnal memiliki kesamaan dengan Gambar 3.79. Perbedaannya terletak pada *value*, halaman ubah jurnal mengambil *value* berdasarkan data yang tersimpan sebelumnya.

Gambar 3.81 merupakan desain *user interface* ubah transaksi jurnal. Fitur ini telah dikembangkan sebelumnya tetapi memerlukan *revamp*, diantaranya:

1. Menambahkan *input field* Referensi berjenis Text.
2. Menghapus *input field* Kode berjenis Text.
3. Menghapus *input field* Unit berjenis dropdown.
4. Perubahan *input field* memo menjadi deskripsi berjenis Text.
5. Menghapus fitur lampiran gambar.
6. Perubahan *input field out of balance* menjadi *different*.
7. Mengubah *input field value* pada Kode T dari berjenis Text menjadi dropdown dengan fitur *search* untuk memudahkan dalam memilih data yang dapat dilihat pada Gambar 3.83. Data diambil dari halaman pengaturan Kode T berdasarkan Tipe sehingga data ditampilkan secara dinamis.

Hasil *revamp* pada halaman ubah transaksi jurnal dapat dilihat pada Gambar 3.82.

The screenshot shows a web form titled "Ubah Transaksi" with a close button (X) in the top right corner. The form is organized into several sections:

- Form Fields:**
 - Left Column:** Tanggal (Masukkan Tanggal), Periode (Pilih), Kode (Masukkan Kode), and Unit (Pilih).
 - Right Column:** Tipe Transaksi (Pilih), Tipe Jurnal (Pilih), Kode T (Pilih), and Uraian (Masukkan Uraian).
- Detail Transaksi Table:**

Kode Akun	Nama Akun	Debit	Kredit	Memo	Kode T
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Total		<input type="text"/>	<input type="text"/>		
Out of Balance		<input type="text"/>	<input type="text"/>		
- Gambar Section:** A section titled "Gambar" with a "Lampirkan Gambar" button and a placeholder image showing a person at a computer.
- Buttons:** "Batal" (Cancel) and "Simpan" (Save) buttons at the bottom right.

Gambar 3.81 Desain *user interface* ubah transaksi jurnal

ACCOUNTING DSS

Beranda Pengaturan Transaksi Manajemen Periode Jurnal Laporan

Ubah Transaksi

Referensi: DPR/107 Tipe Transaksi: Bank Payment Account Payable

Periode: 201909 Tipe Jurnal: Audit Adjustment Journal

Tanggal: 20190925 Penginput: superadmin

Detail Transaksi

Kode Akun	Nama	Debit	Kredit	Deskripsi	T Cc
1120012	BCA - FUND ACCOUNT 1	Rp. 20,000,000.00	Rp. 0.00	Transaction report from file	DF
1120013	BCA - FUND ACCOUNT 2	Rp. 30,000,000.00	Rp. 0.00	Transaction report from file	DF
Total		Rp. 50,000,000.00	Rp. 0.00		
Different		Rp. 50,000,000.00			

Batal Simpan

GEEKGARDEN All rights reserved

Gambar 3.82 Tampilan antarmuka ubah transaksi jurnal

ACCOUNTING DSS

Beranda Pengaturan Transaksi Manajemen Periode Jurnal Laporan

Ubah Transaksi

Referensi: DPR/107 Tipe Transaksi: Bank Payment Account Payable

Periode: 201909 Tipe Jurnal: Audit Adjustment Journal

Tanggal: 20190925 Penginput: superadmin

Detail Transaksi

	Debit	Kredit	Deskripsi	T Code 1
JND ACCOUNT 1	Rp. 20,000,000.00	Rp. 0.00	Transaction report from file	DPPK
JND ACCOUNT 2	Rp. 30,000,000.00	Rp. 0.00	Transaction report from file	
Total		Rp. 50,000,000.00	Rp. 0.00	
Different		Rp. 50,000,000.00		

Search
 DPPK
 DPU
 DPK
 DPA
 DPP
 DPO

GEEKGARDEN All rights reserved

Gambar 3.83 Tampilan dropdown Kode T pada ubah transaksi jurnal

Detail Transaksi Jurnal merupakan fitur untuk menampilkan data transaksi jurnal yang dipilih secara rinci. Tampilan antarmuka detail jurnal dibuat seperti Gambar 3.79 dan Gambar 3.82. Perbedaannya seluruh *input field* dibuat *readonly* sehingga *value* tidak dapat di modifikasi tetapi pengguna dapat menyalin *value* pada masing – masing *input field*.

Gambar 3.84 merupakan desain *user interface* detail transaksi jurnal. Fitur ini telah dikembangkan sebelumnya tetapi memerlukan *revamp*, diantaranya:

1. Menambahkan *input field* Referensi.
2. Menghapus *input field* Kode.
3. Menghapus *input field* Unit.
4. Perubahan *input field* memo menjadi deskripsi.
5. Menghapus fitur lampiran gambar.
6. Perubahan *input field out of balance* menjadi *different*.
7. Menampilkan *input field value* pada Kode T.

Hasil *revamp* pada tampilan antarmuka detail transaksi jurnal dapat dilihat pada Gambar 3.85.

The screenshot shows a 'Detail' window with the following fields:

- Tanggal: 3 September 2021
- Tipe Transaksi: Lorem Ipsum
- Periode: Lorem Ipsum
- Tipe Jurnal: Lorem Ipsum
- Kode: SAD1
- Kode T: Lorem Ipsum
- Unit: Lorem Ipsum
- Uraian: Investasi Jangka Panjang non Permanen

Detail Transaksi

Kode Akun	Nama Akun	Debit	Kredit	Memo	Kode T
		0	0		
		0	0		
Total		0	0		
Out of Balance		0	0		

Gambar

Lampirkan Gambar

Kembali

Gambar 3.84 Desain *user interface* detail transaksi jurnal

ACCOUNTING DSS

Detail Transaksi

Referensi: DFR/107 Tipe Transaksi: Bank Payment, Account Payable

Periode: 201909 Tipe Jurnal: Audit Adjustment Journal

Tanggal: 20190925

Penginput: superadmin

Detail Transaksi

Kode Akun	Nama	Debit	Kredit	Deskripsi	T Code I
1120012	BCA - FUND ACCOUNT 1	Rp. 20.000.000,00	Rp. 0,00	Transaction report from file	DPPK
1120013	BCA - FUND ACCOUNT 2	Rp. 30.000.000,00	Rp. 0,00	Transaction report from file	DPU
Total		Rp. 50.000.000,00	Rp. 0,00		
Different		Rp. 50.000.000,00			

Kembali

GEEKGARDEN All rights reserved

Gambar 3.85 Tampilan antarmuka detail transaksi jurnal

- g. Mengembangkan *front end* pada menu laporan *Account Listing*.

Account Listing adalah fitur untuk menghasilkan sebuah laporan berdasar transaksi jurnal yang telah di posting ke dalam sistem. Data laporan diambil dari Akun (COA) ketika menambahkan jurnal. Laporan ini digunakan oleh bagian keuangan perusahaan pelanggan Geekgarden untuk menampilkan dan mengekspor hasil laporan *account listing*. Fitur ini memiliki beberapa *input field*, yaitu:

1. *Input field* Akun Awal berjenis dropdown dengan fitur *search* karena data Akun (COA) berjumlah banyak. Data diambil dari halaman pengaturan Akun (COA) berdasarkan code akun – nama akun sehingga data ditampilkan secara dinamis. Fitur *search* dan tampilan data dapat dilihat pada Gambar 3.87.
2. *Input field* Akun Akhir berjenis dropdown dengan fitur *search* karena data Akun (COA) berjumlah banyak. Data diambil dari halaman pengaturan Akun (COA) berdasarkan code akun – nama akun sehingga data ditampilkan secara dinamis. Fitur *search* dan tampilan data dapat dilihat pada Gambar 3.87.
3. *Input field* Tahun berjenis datepicker dengan format YYYY (2022).
4. *Input field* Periode Awal berjenis dropdown dengan format MM (01) dan ditampilkan berdasarkan bulan (Januari) untuk memudahkan pengguna dalam

memilih Periode. Data diambil dari halaman pengaturan Periode berdasarkan bulan periode sehingga data ditampilkan secara dinamis.

5. *Input field* Periode Akhir berjenis dropdown dengan format MM (01) dan ditampilkan berdasarkan bulan (Januari) untuk memudahkan pengguna dalam memilih Periode. Data diambil dari halaman pengaturan Periode berdasarkan bulan periode sehingga data ditampilkan secara dinamis.
6. *Input field* Tipe Transaksi berjenis dropdown. Data diambil dari halaman pengaturan Tipe Transaksi berdasarkan code tipe transaksi – nama tipe transaksi sehingga data ditampilkan secara dinamis.
7. *Input field* Beginning Balance berjenis radio button dengan pilihan *value* Yes dan No.
8. *Input field* Ending Balance berjenis radio button pilihan *value* Yes dan No.

Seluruh *input field dropdown* merupakan *filter* untuk menampilkan laporan sesuai dengan kebutuhan. Tahapan untuk menampilkan dan mengekspor laporan *account listing* dimulai ketika seluruh *input field dropdown* yang tersedia pada Gambar 3.86 telah dipilih. Selanjutnya, menekan tombol *preview* ketika ingin menampilkan data sesuai *filter* yang dipilih sedangkan tombol Cetak Excel digunakan untuk mengekspor data berformat excel.

Gambar 3.86 Tampilan antarmuka *Account Listing*

The screenshot shows the 'Account Listing' form in the ACCOUNTING DSS application. The form is titled 'Account Listing' and contains the following fields and options:

- Akun Awal:** 1120011 - BCA - CONTRIBUTION ACCOUNT
- Akun Akhir:** Search
- Tahun:** 1120011 - BCA - CONTRIBUTION ACCOUNT
- Periode Awal:** 1120012 - BCA - FUND ACCOUNT 1
- Periode Akhir:** 1120013 - BCA - FUND ACCOUNT 2
- Tipe Transaksi:** 1120014 - BCA - FUND ACCOUNT 3
- Beginning Balance:** Yes No
- Ending Balance:** Yes No

Buttons for 'Preview' and 'Cetak Excel' are located at the bottom of the form. The footer of the application reads 'GEEKGARDEN All rights reserved'.

Gambar 3.87 Tampilan *dropdown* Akun (COA)

h. Mengembangkan *front end* pada menu laporan *Journal Listing*

Journal Listing adalah fitur untuk menghasilkan laporan transaksi jurnal. Nomor posting ditentukan secara otomatis dari *back end* ketika pengguna melakukan posting jurnal pada halaman transaksi jurnal. Data yang terdapat pada laporan merupakan jurnal yang telah diposting ke dalam sistem dan menampilkan keseluruhan jumlah saldo pada masing – masing jurnal. Gambar 3.88 merupakan tampilan antarmuka laporan *journal listing* yang memiliki dua *input field dropdown* Nomor Posting Awal dan Nomor Posting Akhir. Kedua *input field* tersebut merupakan *filter* yang digunakan untuk menampilkan dan mengekspor laporan *journal listing* ke dalam bentuk file dengan format excel.

The screenshot shows the 'Journal Listing' form in the ACCOUNTING DSS application. The form is titled 'Journal Listing' and contains the following fields and options:

- Nomor Posting Awal:** Pilih
- Nomor Posting Akhir:** Pilih

Buttons for 'Preview' and 'Cetak Excel' are located at the bottom of the form. The footer of the application reads 'GEEKGARDEN All rights reserved'.

Gambar 3.88 Tampilan antarmuka *Journal Listing*

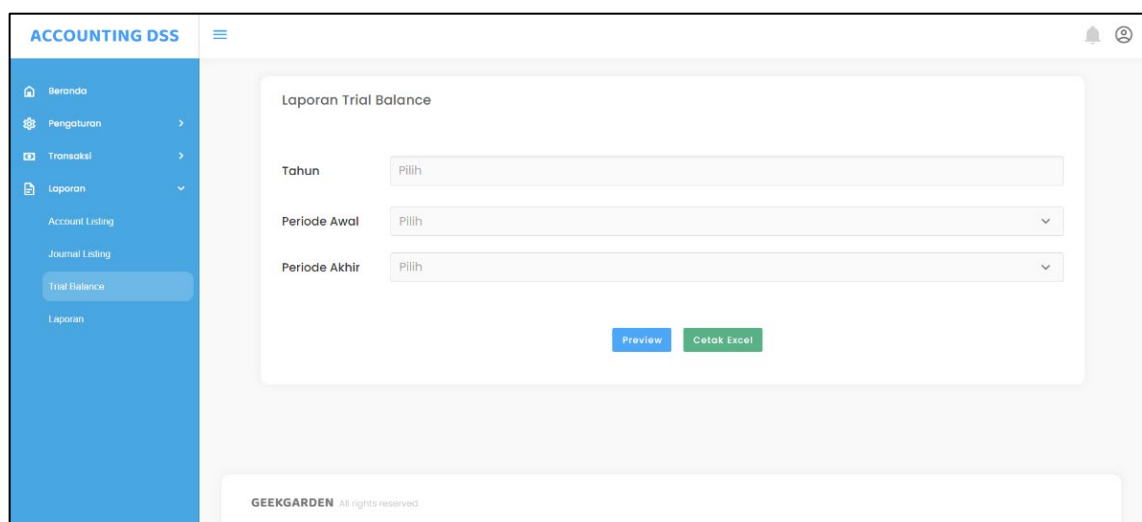
i. Mengembangkan *front end* pada menu laporan *Trial Balance*

Trial balance merupakan fitur untuk melakukan seluruh perhitungan saldo pada masing – masing Akun (COA) yang terdapat di transaksi jurnal setelah diposting. Perhitungan saldo dilakukan berdasarkan seluruh jumlah saldo awal, debit, kredit, dan saldo akhir yang terdapat pada setiap Akun (COA). Saldo awal didapatkan ketika Akun (COA) digunakan dari periode pertama yang terdapat pada sistem.

Tahapan untuk menampilkan atau mengekspor laporan *trial balance* dengan memilih *filter* yang terdapat pada Gambar 3.89, diantaranya:

1. *Input field* Tahun berjenis datepicker dengan format YYYY (2022).
2. *Input field* Periode Awal berjenis dropdown dengan format MM (01) dan ditampilkan berdasarkan bulan (Januari) untuk memudahkan pengguna dalam memilih Periode. Data diambil dari halaman pengaturan Periode berdasarkan bulan periode sehingga data ditampilkan secara dinamis.
3. *Input field* Periode Akhir berjenis dropdown dengan format MM (01) dan ditampilkan berdasarkan bulan (Januari) untuk memudahkan pengguna dalam memilih Periode. Data diambil dari halaman pengaturan Periode berdasarkan bulan periode sehingga data ditampilkan secara dinamis.

Data periode pada *input field dropdown* dari integrasi dengan menu transaksi manajemen periode yang tersimpan pada sistem sehingga *value* ditampilkan secara dinamis. Artinya, untuk menampilkan atau mengekspor file berformat excel pada laporan *trial balance* memiliki batasan sesuai periode yang terdapat pada sistem. Namun, apabila tahun periode yang dipilih tidak ada maka periode tersebut tidak ditampilkan.



Gambar 3.89 Tampilan antarmuka *Trial Balance*

3.2.3.1 Pengujian

Pengujian dilakukan oleh tim QA/QC setelah pengembangan setiap fitur selesai dikerjakan menggunakan metode *black box testing*. Pengujian bertujuan untuk memastikan seluruh fungsionalitas yang dikembangkan pada sistem dapat berjalan sesuai dengan proses bisnis dan kebutuhan pelanggan. Tahapan – tahapan pengujian dilakukan dengan menambahkan data dan menguji setiap fitur yang terdapat di masing – masing halaman tersebut menggunakan beberapa kasus pengujian dilakukan secara manual.

Berdasarkan hasil keluaran yang ditampilkan pada sistem, tim QA/QC dapat menilai dan menentukan apakah hasil tersebut telah berjalan sesuai dengan alur proses bisnis yang diharapkan. Namun, ketika keluaran yang dihasilkan masih ditemukan *issue* maka akan dicatat ke dalam dokumen QC kemudian diserahkan kepada pengembang. Selanjutnya, pengembang segera memperbaiki *issue* tersebut sehingga keluaran yang dihasilkan tidak terdapat kecacatan atau *bug*. Adapun salah satu contoh hasil pengujian yang dilakukan pada halaman pengaturan Akun (COA) dapat dilihat pada Tabel 3.3.

Tabel 3.3 Pengujian pada halaman pengaturan Akun (COA)

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses halaman pengaturan Akun (COA).	Menampilkan seluruh data Akun (COA) yang tersimpan ke dalam bentuk tabel.	Menampilkan seluruh data Akun (COA) yang tersimpan ke dalam bentuk tabel.	<i>Pass</i>
2.	Menekan tombol tambah akun.	Menampilkan <i>pop up form</i> tambah akun.	Menampilkan <i>pop up form</i> tambah akun.	<i>Pass</i>
3.	Memilih <i>filter</i> tipe akun tertentu.	Menampilkan data Akun (COA) berdasarkan tipe akun yang dipilih pengguna.	Menampilkan data Akun (COA) berdasarkan tipe akun yang dipilih pengguna.	<i>Pass</i>
4.	Memilih <i>filter</i> kategori akun tertentu.	Menampilkan data Akun (COA) berdasarkan kategori akun yang dipilih pengguna.	Menampilkan data Akun (COA) berdasarkan kategori akun yang dipilih pengguna.	<i>Pass</i>
5.	Memilih <i>filter</i> status akun tertentu.	Menampilkan data Akun (COA) berdasarkan status akun yang dipilih pengguna.	Menampilkan data Akun (COA) berdasarkan status akun yang dipilih pengguna.	<i>Pass</i>
6.	Memilih tampilkan 25 entri.	Menampilkan 25 data pada tabel Akun (COA).	Menampilkan 25 data pada tabel Akun (COA).	<i>Pass</i>
7.	Mencari data menggunakan <i>input</i> cari.	Menampilkan data berdasarkan kode dan nama akun yang dimasukkan oleh pengguna.	Menampilkan data berdasarkan kode dan nama akun yang dimasukkan oleh pengguna.	<i>Pass</i>

8.	Melakukan <i>sorting</i> pada masing-masing kolom.	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	<i>Pass</i>
9.	Menekan tombol detail berwarna biru dengan ikon berbentuk mata.	Menampilkan <i>pop up form</i> detail Akun (COA) berdasarkan data yang dipilih.	Menampilkan <i>pop up form</i> detail Akun (COA) berdasarkan data yang dipilih.	<i>Pass</i>
10.	Menekan tombol <i>edit</i> berwarna kuning dengan ikon berbentuk pensil.	Menampilkan <i>pop up form edit</i> Akun (COA) berdasarkan data yang ingin diubah.	Menampilkan <i>pop up form edit</i> Akun (COA) berdasarkan data yang ingin diubah.	<i>Pass</i>
11.	Menekan tombol <i>delete</i> berwarna merah dengan ikon berbentuk tempat sampah.	Menampilkan <i>pop up</i> konfirmasi hapus data.	Menampilkan <i>pop up</i> konfirmasi hapus data.	<i>Pass</i>

Rangkuman hasil pengujian yang dilakukan pada setiap fitur dapat dilihat pada Tabel 3.4. Hasil pengujian menunjukkan bahwa setiap fitur yang dikembangkan lolos pengujian. Selanjutnya, detail pengujian fitur akan disampaikan di lampiran.

Tabel 3.4 Hasil pengujian setiap fitur

No	Fitur	Penjelasan Fitur	Hasil pengujian berhasil	Hasil pengujian gagal
1.	Pengaturan Akun (COA)	Akun (COA) merupakan pengaturan yang digunakan untuk mengelola akun pada sistem	23	0
2.	Pengaturan Tipe Transaksi	Tipe transaksi merupakan fitur untuk mengelola tipe transaksi dan mendukung pemilihan tipe transaksi yang digunakan menu lainnya.	12	0
3.	Pengaturan Kode T	Kode T merupakan fitur untuk mengelola Kode T dan mendukung pemilihan Kode T ketika menambahkan transaksi jurnal.	17	0
4.	Pengaturan Periode	Periode merupakan fitur untuk menambahkan periode yang digunakan	5	0

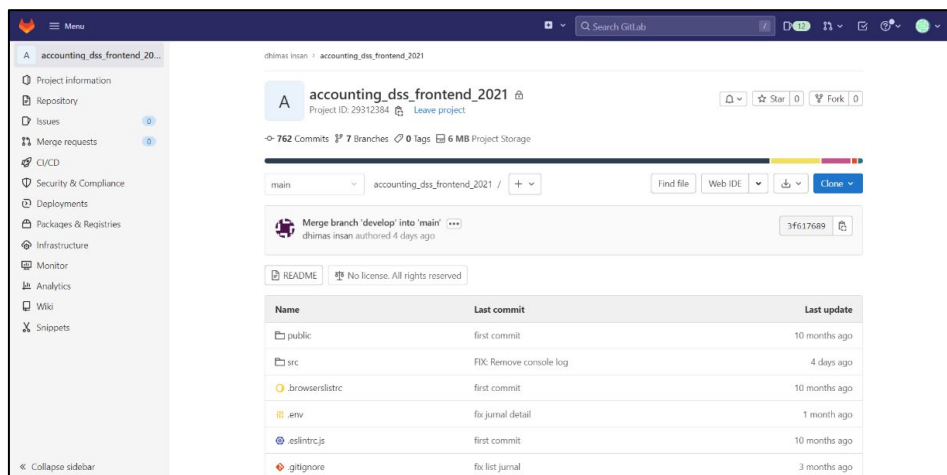
		dalam proses <i>input</i> transaksi jurnal.		
5.	Transaksi Manajemen Periode	Manajemen periode merupakan fitur untuk mengelola periode yang harus dibuka dan ditutup.	4	0
6.	Transaksi Jurnal	Transaksi jurnal merupakan halaman untuk melakukan seluruh aktivitas transaksi jurnal.	30	0
7.	Laporan <i>Account Listing</i>	<i>Account Listing</i> merupakan fitur untuk menghasilkan sebuah laporan berdasar transaksi jurnal yang telah di posting ke dalam sistem. Data laporan diambil dari Akun (COA) ketika menambahkan jurnal.	2	0
8.	Laporan <i>Journal Listing</i>	<i>Journal Listing</i> merupakan fitur untuk menghasilkan laporan transaksi jurnal. Nomor posting ditentukan secara otomatis dari <i>back end</i> ketika pengguna melakukan posting jurnal pada halaman transaksi jurnal.	2	0
9.	Laporan Trial Balance	<i>Trial balance</i> merupakan fitur untuk melakukan seluruh perhitungan saldo pada masing – masing Akun (COA) yang terdapat di transaksi jurnal setelah diposting. Perhitungan saldo dilakukan berdasarkan seluruh jumlah saldo awal, debit, kredit, dan saldo akhir yang terdapat pada setiap Akun (COA).	2	0

3.2.4 Pemantauan dan Pengendalian Proyek

Pemantauan dan pengendalian proyek Sistem Informasi Akuntansi yang dilakukan menggunakan aplikasi GitLab dan Telegram selama proses pengembangan.

a. GitLab

GitLab merupakan sebuah layanan *cloud* berbasis website sebagai tempat kolaborasi antar pengembang untuk menyimpan seluruh perubahan kode program yang dilakukan secara *real time*. Selain itu, GitLab digunakan untuk memudahkan dalam mengakses *repository* secara *remote*. Gambar 3.90 merupakan tampilan GitLab untuk pengembangan *front end* pada proyek Sistem Informasi Akuntansi.



Gambar 3.90 Tampilan *repository* GitLab Sistem Informasi Akuntansi

b. Telegram

Telegram merupakan aplikasi untuk komunikasi antar tim pengembang yang terlibat pada proyek Sistem Informasi Akuntansi. Di dalam grup yang terdapat pada Gambar 3.91 *front-end developer*, *back-end developer*, QA/QC, DevOps, dan *Project Manager* (PM). PM akan memberikan arahan, tugas, dan *update* kepada tim berdasarkan hasil *meeting* dengan pihak pelanggan Geekgarden. Selain itu, progress harian dan kendala dalam proses pengerjaan pengembang dapat disampaikan ke dalam grup tersebut.



Gambar 3.91 Tampilan grup telegram Sistem Informasi Akuntansi

3.2.5 Penutupan Proyek

Penutupan pengembangan proyek Sistem Informasi Akuntansi belum dilakukan karena masa magang penulis di Geekgarden telah berakhir. Selain itu, apabila masih terdapat beberapa tugas yang belum dikerjakan akan dialihkan oleh karyawan Geekgarden dengan posisi *front-end web developer* di Geekgarden Software House.



BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Sistem Informasi Akuntansi merupakan aplikasi berbasis *website* yang menerapkan kerangka kerja Vue.js dan API dalam pengembangannya. Pada pengembangan aplikasi tersebut memiliki keterkaitan dengan teori mata kuliah Pengembangan Aplikasi Berbasis Website (PABW). Mata kuliah tersebut memberikan teori dan praktek mengenai penggunaan HTML, PHP, CSS, JavaScript, serta *framework* Laravel untuk mengembangkan aplikasi. Penerapan teknologi yang didapatkan saat kuliah hanya memanfaatkan CSS dan JavaScript untuk membangun *front end* pada Sistem Informasi Akuntansi. Pengembangan aplikasi ini dibagi menjadi 2 bagian utama, yaitu: *layer view* dan *layer model controller*. *Front end* hanya mengimplementasikan *layer view* untuk membangun tampilan antarmuka menggunakan kerangka kerja Vue.js. Dari sisi *back end* menggunakan *layer model* dan *controller* diimplementasikan ke dalam API untuk mengelola penyimpanan data dan melakukan *request* pada server. Dimana API merupakan HTTP yang memanfaatkan method GET, POST, PUT, dan DELETE.

Proses manajemen proyek yang terdapat di Geekgarden tidak ditemukan perbedaan antara teori mata kuliah Manajemen Pengembangan Teknologi Informasi (MPTI) dengan pelaksanaan di lapangan. Berdasarkan teori tahapan manajemen proyek dimulai dari inisiasi proyek, perencanaan proyek, eksekusi proyek, pemantauan proyek, dan penutupan proyek. Namun, pada tahap pemantauan proyek Geekgarden tidak menerapkan teknologi seperti Trello atau Jira yang menyebabkan pemantauan proyek masih dilakukan secara manual menggunakan grup Telegram sehingga kurang terkontrol dengan baik. Hal tersebut membuat To Do yang diberikan *Project Manager* menjadi tidak terstruktur dan PM harus memantau secara berkala *update progress* antar divisi seharusnya masalah tersebut dapat diatasi dengan menerapkan teknologi manajemen proyek.

Metode pengembangan proyek atau metode System Development Life Cycle (SDLC) tidak diterapkan pada sebagian besar proyek yang terdapat di Geekgarden. Secara teori yang pada mata kuliah Pengembangan Sistem Informasi (PSI) terdapat aktivitas utama dalam pengembangan proyek dimulai dari tahap System Analysis, System Design, Programming, Testing, Conversation, Production and Maintenance. Dari enam proyek yang terlibat selama

magang hanya satu proyek bernama KlikPayroll yang menerapkan metode SDLC menggunakan scrum. Scrum sendiri memiliki satu kegiatan bernama *daily scrum* untuk menyampaikan *update progress*, kendala, dan masalah yang dialami satu tim kepada *Project Manager* (PM) sehingga dapat masalah tersebut dapat ditemukan solusi. Tentunya penggunaan metode pengembangan proyek sangat diperlukan agar tercapai *deliverable* atau *scope* sesuai dengan kebutuhan proyek dan meminimalisir terjadinya pembekakan waktu pengerjaan proyek.

Pada pengujian menggunakan metode *black box testing* tidak ditemukan perbedaan antara teori dengan praktik dilapangan. Secara teori *black box testing* merupakan metode pengujian tanpa mengetahui kode program didalamnya untuk memastikan seluruh *input* dan *output* berjalan sesuai proses bisnis yang diharapkan. Pengujian dilakukan oleh tim QA/QC melalui *front end* website untuk mengetahui hasil *input* dan *output* tetapi praktik dilapangan masih melibatkan pengembang untuk melakukan pengujian. Hal tersebut, dikarenakan kurangnya tim QA/QC di tempat magang, pengujian *black box testing* membutuhkan waktu yang lama dan harus membuat laporan ketika ditemukan *bug / issue*. Pada saat bersamaan terdapat proyek lain di tempat magang yang harus melaporkan *update progress* setiap harinya sehingga tim QA/QC terfokus melakukan pengujian dalam proyek tersebut. Untuk meminimalisir terjadi hal serupa tempat magang mulai menerapkan pengujian menggunakan metode *automation testing*. Selain itu, *automation testing* memiliki keunggulan dalam melakukan pengujian dalam waktu yang singkat dan menghasilkan laporan *bug / issue* secara otomatis setelah pengujian selesai dilakukan sehingga seluruh proyek dapat diuji oleh tim QA/QC.

4.2 Pembelajaran Magang

4.2.1 Manfaat Magang

- a. Mendapatkan pengetahuan baru dan *tools*.

Selama melaksanakan magang mendapatkan pengetahuan baru mengenai *tools* dan berkesempatan untuk mengembangkan *front end* menggunakan kerangka kerja Vue.js yang belum pernah didapatkan selama kuliah. *Tools* yang digunakan ketika magang diantaranya, yaitu: GitLab digunakan untuk berkolaborasi antar pengembang dan Postman digunakan untuk menyediakan endpoint API yang disediakan *back end* sehingga memudahkan dalam integrasi dengan *front end*.

b. Manajemen diri

Manajemen diri yang baik sangat diperlukan ketika menjalankan kegiatan magang sekaligus kuliah karena harus pintar dalam membagi waktu antara tugas kuliah dengan pekerjaan saat magang. Adanya tuntutan dan tanggung jawab yang diberikan oleh perusahaan membuat kesulitan untuk memanajemen waktu. Oleh karena itu, pentingnya manajemen diri yang baik dapat menyelesaikan seluruh tugas kuliah dan pekerjaan dengan tepat waktu.

c. Bekerja sama dengan tim

Pada pengembangan aplikasi terdapat beberapa peran yang terlibat ke dalam suatu proyek tersebut. Adanya pembagian peran membuat pekerjaan menjadi lebih terstruktur dan teratur karena tugas yang diberikan sesuai dengan ruang lingkup masing-masing anggota tim. Selain itu, pemegang membiasakan diri untuk mempelajari hal baru ketika dihadapkan dengan masalah yang belum pernah ditemukan sebelumnya.

4.2.2 Kendala, Hambatan dan Tantangan selama magang

Pada saat berlangsungnya kegiatan magang, mengalami beberapa hambatan dan tantangan karena kegiatan kuliah yang bersamaan dengan aktivitas magang menyebabkan sulit untuk membagi waktu antara kuliah dan pekerjaan ketika magang. Hal tersebut menyebabkan pemegang menjadi kurang fokus untuk mengikuti kedua kegiatan tersebut. Kesulitan dalam berkomunikasi dengan pembimbing lapangan dikarenakan keterbatasan waktu yang dimiliki sehingga menghambat pekerjaan terutama saat pekerjaan dilakukan secara *work from home*.

Kendala lain ketika bekerja dalam satu project dengan karyawan *freelance*, membuat pemegang kesulitan dalam melakukan komunikasi karena mereka bekerja pada saat malam hari. Sering kali dipindah proyek oleh pembimbing lapangan membutuhkan waktu lebih untuk memahami alur dan mempelajari kode program pada sistem sehingga setiap kali berpindah proyek memerlukan penyesuaian *tools* yang dibutuhkan. Tidak stabilnya koneksi internet dikantor dan laptop sering mengalami *hank* membuat pekerjaan menjadi lambat, spesifikasi laptop yang kurang mendukung juga mempengaruhi ketika sedang menjalankan *tools* pada proses pengembangan.

Tantangan yang dihadapi ketika awal magang yaitu saat di tempatkan pada divisi *front-end web developer* menggunakan kerangka kerja Vue.js. Hal tersebut dikarenakan sebelumnya tidak mengetahui dalam mengembangkan *front end* dengan kerangka kerja tersebut sehingga

dibutuhkan waktu lebih untuk mempelajari konsep dan dasar penggunaannya. Selain itu, adanya tuntutan untuk menyelesaikan proyek sesuai dengan waktu yang ditentukan.



BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pelaksanaan magang, pengembangan *front end* Sistem Informasi Akuntansi menggunakan kerangka kerja Vue.js didapatkan kesimpulan sebagai berikut:

- a. Pada magang ini telah berhasil menyelesaikan tugas mengembangkan *front end* dari sembilan fitur yang terdapat pada Sistem Informasi Akuntansi dengan menggunakan kerangka kerja Vue.js. Kesembilan fitur tersebut telah berhasil diintegrasikan dengan *back end* Sistem Informasi Akuntansi dan telah diuji menggunakan pendekatan *black box testing*. Hasil pengujian menunjukkan bahwa semua skenario yang dilakukan pada kesembilan fitur telah lolos uji.
- b. Berdasarkan pengalaman ketika magang, adanya kerangka kerja Vue.js mempermudah proses pengembangan *front end* karena memiliki banyaknya dukungan *library* dan sesuai dengan kebutuhan pengembangan.

5.2 Saran

5.2.1 Saran untuk Pengembangan Selanjutnya

Terdapat beberapa saran yang diberikan untuk pengembangan *front end* selanjutnya, sebagai berikut:

- a. Sebaiknya desain UI/UX dirancang hingga *final* sebelum diimplementasikan ke dalam kode program sehingga meminimalisir terjadinya *change request* atau *revamp* pada saat pengembangan.
- b. Perlu diterapkan *generate password* secara otomatis dan dikirimkan melalui email setelah admin menambahkan pengguna ke dalam sistem sehingga hanya pemilik email yang mengetahui kata sandi agar tidak disalahgunakan.

5.2.2 Saran untuk Perusahaan

Terdapat beberapa saran yang diberikan untuk perusahaan tempat magang, sebagai berikut:

- a. Sebaiknya setiap proyek yang ada di perusahaan menerapkan metode *System Development Life Cycle* (SDLC) agar lebih terstruktur.

- b. Sebaiknya diadakan *weekly meeting* dengan tim *internal* perusahaan sehingga permasalahan yang ada pada proyek dapat segera terselesaikan.
- c. Sebaiknya *developer* yang terlibat pada sebuah proyek tidak dipindah ke proyek lain sebelum penutupan proyek.



DAFTAR PUSTAKA

- Arhandi, P. P. (2016). Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode Back End Dan Front End. *Teknologi Informasi*, 7(1), 39–48.
- Belluano, P. L. L. (2018). Pengembangan Single page Application Pada Sistem Informasi Akademik. *ILKOM Jurnal Ilmiah*, 10(1), 38–43.
- Chastro, C., & Darmawan, E. (2020). Perbandingan Pengembangan Front End Menggunakan Blade Template dan Vue Js. *Jurnal STRATEGI-Jurnal Maranatha*, 2(2), 302–313.
- 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. <https://doi.org/10.30998/string.v3i2.3048>
- Gani, L. (2018). *Panduan Praktis Menguasai Vue.js*. Yogyakarta: Lokomedia.
- Herdinata Putra, I. K. A., Pramana, D., & Srinadi, N. L. P. (2019). Sistem Manajemen Arsip Menggunakan Framework Laravel dan Vue.Js (Studi Kasus : BPKAD Provinsi Bali). *Jurnal Sistem Daninformatika*, 13(2), 97–104.
- Muhyidin, M. A., Sulhan, M. A., & Sevtiana, A. (2020). Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma. *Jurnal Digit*, 10(2), 208. <https://doi.org/10.51920/jd.v10i2.171>
- Rakhmah, A. H., & Purwoko, H. (2021). Efektivitas Web Api Dalam Integrasi Bahasa Pemrograman Multi Platform. *Semnas Ristek (Seminar Nasional Riset Dan Inovasi Teknologi)*, 5(1), 18–22. Retrieved from <http://www.proceeding.unindra.ac.id/index.php/semnasristek/article/view/4790>
- Razi, A. A., Mutiaz, I. R., & Setiawan, P. (2018). Penerapan Metode Design Thinking Pada Model Perancangan Ui/Ux Aplikasi Penanganan Laporan Kehilangan Dan Temuan Barang Tercecer. *Desain Komunikasi Visual, Manajemen Desain Dan Periklanan (Demandia)*, 3(02), 219. <https://doi.org/10.25124/demandia.v3i02.1549>
- Tampang, R., Kadang, M. O., & Arrang, H. (2020). Pengiriman Barang Berbasis Web Pada Ekspedisi Anugrah Indah Dengan Menggunakan Teknologi Single Page Application (Spa). *Paulus Informatics Journal*, 1(2), 21–26. Retrieved from www.degananda.com.

LAMPIRAN

Lampiran 1. Sertifikat Magang



SERTIFIKAT KULIAH KERJA LAPANGAN

Diberikan Kepada :

ANDHIKA RIZKY ARYASTA

No : 001/SM/PTKI/03/22

Bahwa Yang Bersangkutan Telah Melaksanakan **Kuliah Kerja Lapangan** (KKL) Di GeekGarden Software House. KKL Tersebut Telah Dilaksanakan Selama 6 (Enam) Bulan, Yaitu Mulai Tanggal 13 September 2021 S/D 14 Maret 2022. Selama Menjalani KKL Di GeekGarden Software House, Yang Bersangkutan Telah Belajar Sebagai **Frontend Developer**

A DINYATAKAN LULUS

Yogyakarta, 25 Maret 2022



Chief Executive Officer
GeekGarden Software House



Firmansyah, M.B.A.

Lampiran 2. Halaman pengaturan Akun (COA)

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses halaman pengaturan Akun (COA).	Menampilkan seluruh data Akun (COA) yang tersimpan ke dalam bentuk tabel.	Menampilkan seluruh data Akun (COA) yang tersimpan ke dalam bentuk tabel.	Pass
2.	Menekan tombol tambah akun.	Menampilkan <i>pop up form</i> tambah akun.	Menampilkan <i>pop up form</i> tambah akun.	Pass
3.	Memilih <i>filter</i> tipe akun tertentu.	Menampilkan data Akun (COA) berdasarkan tipe akun yang dipilih pengguna.	Menampilkan data Akun (COA) berdasarkan tipe akun yang dipilih pengguna.	Pass
4.	Memilih <i>filter</i> kategori akun tertentu.	Menampilkan data Akun (COA) berdasarkan kategori akun yang dipilih pengguna.	Menampilkan data Akun (COA) berdasarkan kategori akun yang dipilih pengguna.	Pass

5.	Memilih <i>filter</i> status akun tertentu.	Menampilkan data Akun (COA) berdasarkan status akun yang dipilih pengguna.	Menampilkan data Akun (COA) berdasarkan status akun yang dipilih pengguna.	<i>Pass</i>
6.	Memilih tampilkan 25 entri.	Menampilkan 25 data pada tabel Akun (COA).	Menampilkan 25 data pada tabel Akun (COA).	<i>Pass</i>
7.	Mencari data menggunakan <i>input</i> cari.	Menampilkan data berdasarkan kode dan nama akun yang dimasukkan oleh pengguna.	Menampilkan data berdasarkan kode dan nama akun yang dimasukkan oleh pengguna.	<i>Pass</i>
8.	Melakukan <i>sorting</i> pada masing-masing kolom.	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	<i>Pass</i>
9.	Menekan tombol detail berwarna biru dengan ikon berbentuk mata.	Menampilkan <i>pop up form</i> detail Akun (COA) berdasarkan data yang dipilih.	Menampilkan <i>pop up form</i> detail Akun (COA) berdasarkan data yang dipilih.	<i>Pass</i>
10.	Menekan tombol <i>edit</i> berwarna kuning dengan ikon berbentuk pensil.	Menampilkan <i>pop up form edit</i> Akun (COA) berdasarkan data yang ingin diubah.	Menampilkan <i>pop up form edit</i> Akun (COA) berdasarkan data yang ingin diubah.	<i>Pass</i>
11.	Menekan tombol <i>delete</i> berwarna merah dengan ikon berbentuk tempat sampah.	Menampilkan <i>pop up</i> konfirmasi hapus data.	Menampilkan <i>pop up</i> konfirmasi hapus data.	<i>Pass</i>

Lampiran 2A. Tambah Akun (COA)

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol simpan tanpa mengisi data pada seluruh <i>field</i> yang tersedia.	Menampilkan <i>error messages</i> pada <i>field</i> kode akun, nama akun, tipe akun, dan deskripsi dengan keterangan wajib diisi serta tombol simpan berubah menjadi <i>disabled</i> .	Menampilkan <i>error messages</i> pada <i>field</i> kode akun, nama akun, tipe akun, dan deskripsi dengan keterangan wajib diisi serta tombol simpan berubah menjadi <i>disabled</i> .	<i>Pass</i>
2.	Menekan tombol simpan hanya mengisi data pada <i>field mandatory</i> .	Menampilkan <i>snackbar</i> bahwa akun berhasil ditambahkan selama 2,5 detik dan menambah data akun baru pada tabel Akun (COA).	Menampilkan <i>snackbar</i> bahwa akun berhasil ditambahkan selama 2,5 detik dan menambah data akun baru pada tabel Akun (COA).	<i>Pass</i>

3.	Menekan tombol simpan dengan mengisi data pada seluruh <i>filed</i> yang tersedia.	Menampilkan <i>snackbar</i> bahwa akun berhasil ditambahkan selama 2,5 detik dan menambah data akun baru pada tabel Akun (COA).	Menampilkan <i>snackbar</i> bahwa akun berhasil ditambahkan selama 2,5 detik dan menambah data akun baru pada tabel Akun (COA).	<i>Pass</i>
4.	Menekan tombol batal atau ikon x.	Menutup <i>pop up</i> tambah akun.	Menutup <i>pop up</i> tambah akun.	<i>Pass</i>
5.	Menekan ikon + pada kategori akun.	Menambahkan <i>field dropdown</i> kategori akun secara dinamis.	Menambahkan <i>dropdown</i> kategori akun secara dinamis.	<i>Pass</i>
6.	Menekan ikon - pada kategori akun.	Menghapus <i>field dropdown</i> kategori akun secara dinamis.	Menghapus <i>field dropdown</i> kategori akun secara dinamis.	<i>Pass</i>

Lampiran 2B. Ubah Akun (COA)

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol batal atau ikon X.	Menutup <i>pop up</i> ubah akun.	Menutup <i>pop up</i> ubah akun.	<i>Pass</i>
2.	Mengisi atau memperbarui data dan menekan tombol simpan.	Menampilkan <i>snackbar</i> bahwa akun berhasil diubah selama 2,5 detik dan menyimpan perubahan data serta memperbarui data pada tabel Akun (COA).	Menampilkan <i>snackbar</i> bahwa akun berhasil diubah selama 2,5 detik dan menyimpan perubahan data serta memperbarui data pada tabel Akun (COA)	<i>Pass</i>

Lampiran 2C. Detail Akun (COA)

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses detail akun.	Menampilkan data akun secara rinci.	Menampilkan data akun secara rinci.	<i>Pass</i>
2.	Menekan tombol tutup atau ikon X.	Menutup <i>pop up</i> detail akun.	Menutup <i>pop up</i> detail akun.	<i>Pass</i>

Lampiran 2D. Hapus Akun (COA)

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol batal.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	<i>Pass</i>
2.	Menekan tombol hapus.	Data terhapus dan menampilkan <i>snackbar</i> bahwa data berhasil dihapus selama 2,5 detik.	Data terhapus dan menampilkan <i>snackbar</i> bahwa data berhasil dihapus selama 2,5 detik.	<i>Pass</i>

Lampiran 3. Halaman pengaturan Tipe Transaksi

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses halaman pengaturan tipe transaksi.	Menampilkan seluruh data tipe transaksi yang tersimpan ke dalam bentuk tabel.	Menampilkan seluruh data tipe transaksi yang tersimpan ke dalam bentuk tabel.	Pass
2.	Melakukan <i>sorting</i> pada masing-masing kolom.	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Pass
3.	Menekan tombol tambah tipe transaksi.	Menampilkan <i>pop up form</i> tambah tipe transaksi.	Menampilkan <i>pop up form</i> tambah tipe transaksi.	Pass
4.	Menekan tombol <i>edit</i> berwarna kuning dengan ikon berbentuk pensil.	Menampilkan <i>pop up form</i> ubah tipe transaksi.	Menampilkan <i>pop up form</i> ubah tipe transaksi.	Pass
5.	Menekan tombol <i>delete</i> berwarna merah dengan ikon berbentuk tempat sampah.	Menampilkan <i>pop up</i> konfirmasi hapus data.	Menampilkan <i>pop up</i> konfirmasi hapus data.	Pass

Lampiran 3A. Tambah Tipe Transaksi

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol simpan tanpa mengisikan data.	Menampilkan <i>error messages</i> pada seluruh <i>input</i> dengan keterangan wajib diisi serta tombol simpan berubah menjadi <i>disabled</i> .	Menampilkan <i>error messages</i> pada seluruh <i>input</i> dengan keterangan wajib diisi serta tombol simpan berubah menjadi <i>disabled</i> .	Pass
2.	Menekan tombol simpan dengan mengisikan seluruh <i>field</i> .	Menampilkan <i>snackbar</i> bahwa data berhasil ditambahkan selama 2,5 detik dan menambah data pada tabel tipe transaksi.	Menampilkan <i>snackbar</i> bahwa data berhasil ditambahkan selama 2,5 detik dan menambah data pada tabel tipe transaksi.	Pass
3.	Menekan tombol batal atau ikon X.	Menutup <i>pop up</i> tambah tipe transaksi.	Menutup <i>pop up</i> tambah tipe transaksi.	Pass

Lampiran 3B. Ubah Tipe Transaksi

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengisi atau memperbarui data dan menekan tombol simpan.	Menampilkan <i>snackbar</i> bahwa data berhasil diubah selama 2,5 detik dan memperbarui data pada tabel tipe transaksi.	Menampilkan <i>snackbar</i> bahwa data berhasil diubah selama 2,5 detik dan memperbarui data pada tabel tipe transaksi.	Pass
2.	Menekan tombol batal atau ikon X.	Menutup <i>pop up</i> ubah tipe transaksi.	Menutup <i>pop up</i> ubah tipe transaksi.	Pass

Lampiran 3C. Hapus Tipe Transaksi

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol batal.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	Pass
2.	Menekan tombol hapus.	Data terhapus dan menampilkan <i>snackbar</i> bahwa data berhasil dihapus selama 2,5 detik.	Data terhapus dan menampilkan <i>snackbar</i> bahwa data berhasil dihapus selama 2,5 detik.	Pass

Lampiran 4. Halaman pengaturan Kode T

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses halaman pengaturan Kode T.	Menampilkan seluruh data Kode T yang tersimpan ke dalam bentuk tabel.	Menampilkan seluruh data Kode T yang tersimpan ke dalam bentuk tabel.	Pass
2.	Menekan tombol tambah Kode T.	Menampilkan <i>pop up form</i> tambah Kode T.	Menampilkan <i>pop up form</i> tambah Kode T.	Pass
3.	Melakukan <i>sorting</i> pada masing-masing kolom.	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Pass
4.	Menekan tombol detail berwarna biru dengan ikon berbentuk mata.	Menampilkan <i>pop up form</i> detail Kode T berdasarkan data yang dipilih.	Menampilkan <i>pop up form</i> detail Kode T berdasarkan data yang dipilih.	Pass
5.	Menekan tombol <i>edit</i> berwarna kuning dengan ikon berbentuk pensil.	Menampilkan <i>pop up form</i> ubah Kode T berdasarkan data yang dipilih.	Menampilkan <i>pop up form</i> ubah Kode T berdasarkan data yang dipilih.	Pass

6.	Menekan tombol <i>delete</i> berwarna merah dengan ikon berbentuk tempat sampah.	Menampilkan <i>pop up</i> konfirmasi hapus data.	Menampilkan <i>pop up</i> konfirmasi hapus data.	<i>Pass</i>
----	--	--	--	-------------

Lampiran 4A. Tambah Kode T

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol simpan tanpa mengisikan data.	Menampilkan <i>error messages</i> pada seluruh <i>input</i> dengan keterangan wajib diisi serta tombol simpan berubah menjadi <i>disabled</i> .	Menampilkan <i>error messages</i> pada seluruh <i>input</i> dengan keterangan wajib diisi serta tombol simpan berubah menjadi <i>disabled</i> .	<i>Pass</i>
2.	Menekan tombol simpan dengan mengisikan seluruh <i>field</i> .	Menampilkan <i>snackbar</i> bahwa data berhasil ditambahkan selama 2,5 detik dan menambah data pada tabel Kode T.	Menampilkan <i>snackbar</i> bahwa data berhasil ditambahkan selama 2,5 detik dan menambah data pada tabel Kode T.	<i>Pass</i>
3.	Menekan ikon + pada tipe.	Menambahkan <i>input</i> tipe secara dinamis.	Menambahkan <i>input</i> tipe secara dinamis.	<i>Pass</i>
4.	Menekan ikon – pada tipe.	Menghapus <i>input</i> tipe secara dinamis.	Menghapus <i>input</i> tipe secara dinamis.	<i>Pass</i>
5.	Menekan tombol batal atau ikon X.	Menutup <i>pop up</i> tambah tipe transaksi.	Menutup <i>pop up</i> tambah tipe transaksi.	<i>Pass</i>

Lampiran 4B. Ubah Kode T

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengisi atau memperbarui data dan menekan tombol simpan.	Menampilkan <i>snackbar</i> bahwa data berhasil diubah selama 2,5 detik dan memperbarui data pada tabel Kode T.	Menampilkan <i>snackbar</i> bahwa data berhasil diubah selama 2,5 detik dan memperbarui data pada tabel Kode T.	<i>Pass</i>
2.	Menekan tombol batal atau ikon X.	Menutup <i>pop up</i> ubah tipe transaksi.	Menutup <i>pop up</i> ubah tipe transaksi.	<i>Pass</i>

Lampiran 4C. Detail Kode T

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses detail Kode T.	Menampilkan data Kode T secara rinci.	Menampilkan data Kode T secara rinci.	<i>Pass</i>
2.	Menekan tombol batal atau ikon X.	Menutup <i>pop up</i> detail Kode T.	Menutup <i>pop up</i> detail Kode T.	<i>Pass</i>

Lampiran 4D. Hapus Kode T

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol batal.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	<i>Pass</i>
2.	Menekan tombol hapus.	Data terhapus dan menampilkan <i>snackbar</i> bahwa data berhasil dihapus selama 2,5 detik.	Data terhapus dan menampilkan <i>snackbar</i> bahwa data berhasil dihapus selama 2,5 detik.	<i>Pass</i>

Lampiran 5. Halaman pengaturan Periode

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol tambah tanpa memilih periode.	Menampilkan <i>error messages</i> pada <i>input</i> bahwa periode harus dipilih.	Menampilkan <i>error messages</i> pada <i>input</i> bahwa periode harus dipilih.	<i>Pass</i>
2.	Menekan tombol tambah dengan memilih periode yang belum ada.	Menampilkan <i>snackbar</i> bahwa data berhasil ditambahkan selama 2 detik dan menambah data pada tabel periode.	Menampilkan <i>snackbar</i> bahwa data berhasil ditambahkan selama 2 detik dan menambah data pada tabel periode.	<i>Pass</i>
3.	Menekan tombol tambah dengan memilih periode yang sudah ada.	Menampilkan <i>snackbar</i> bahwa data sudah ada selama 2 detik.	Menampilkan <i>snackbar</i> bahwa data sudah ada selama 2 detik.	<i>Pass</i>
4.	Menekan tombol cek tanpa memilih periode.	Menampilkan seluruh data periode dalam bentuk tabel.	Menampilkan seluruh data periode dalam bentuk tabel.	<i>Pass</i>
5.	Menekan tombol cek dengan memilih periode	Menampilkan hanya 1 data periode yang dipilih oleh pengguna.	Menampilkan hanya 1 data periode yang dipilih oleh pengguna.	<i>Pass</i>

Lampiran 6. Halaman Transaksi Manajemen Periode

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol cek tanpa memilih periode.	Menampilkan seluruh data periode dalam bentuk tabel.	Menampilkan seluruh data periode dalam bentuk tabel.	<i>Pass</i>
2.	Menekan tombol cek dengan memilih periode	Menampilkan hanya 1 data periode yang dipilih oleh pengguna.	Menampilkan hanya 1 data periode yang dipilih oleh pengguna.	<i>Pass</i>
3.	Menekan tombol buka pada periode tertentu.	Menampilkan <i>snackbar</i> bahwa periode berhasil dibuka selama 2 detik	Menampilkan <i>snackbar</i> bahwa periode berhasil dibuka selama 2 detik dan tombol berubah menjadi tutup.	<i>Pass</i>

		dan tombol berubah menjadi tutup.		
4.	Menekan tombol tutup pada periode tertentu.	Menampilkan <i>snackbar</i> bahwa periode berhasil ditutup selama 2 detik dan tombol berubah menjadi buka.	Menampilkan <i>snackbar</i> bahwa periode berhasil ditutup selama 2 detik dan tombol berubah menjadi buka.	<i>Pass</i>

Lampiran 7. Halaman Transaksi Jurnal

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Mengakses halaman transaksi jurnal.	Menampilkan seluruh data transaksi jurnal yang tersimpan pada basis data ke dalam bentuk tabel.	Menampilkan seluruh data transaksi jurnal yang tersimpan pada basis data ke dalam bentuk tabel.	<i>Pass</i>
2.	Menekan tombol input jurnal baru.	Berpindah ke halaman tambah transaksi baru dan menampilkan <i>form</i> tambah transaksi jurnal.	Berpindah ke halaman tambah transaksi baru dan menampilkan <i>form</i> tambah transaksi jurnal	<i>Pass</i>
3.	Menekan tombol unggah jurnal.	Pengguna diarahkan untuk memilih <i>files</i> dengan <i>format .txt</i> pada perangkat yang digunakan.	Pengguna diarahkan untuk memilih <i>files</i> dengan <i>format .txt</i> pada perangkat yang digunakan.	<i>Pass</i>
4.	Menekan tombol unggah jurnal dan melakukan unggah <i>files</i> berisikan data yang sesuai dengan format <i>.txt</i>	Data ditampilkan pada tabel transaksi jurnal dan menampilkan <i>pop up success</i> .	Data ditampilkan pada tabel transaksi jurnal dan menampilkan <i>pop up success</i> .	<i>Pass</i>
5.	Menekan tombol unggah jurnal dan melakukan unggah <i>files</i> dengan format selain <i>.txt</i>	Sistem menampilkan <i>pop up error</i> bahwa <i>files</i> harus berformat <i>.txt</i> .	Sistem menampilkan <i>pop up error</i> bahwa <i>files</i> harus berformat <i>.txt</i> .	<i>Pass</i>
6.	Menekan tombol unggah jurnal dan melakukan unggah <i>files</i> berisikan data yang tidak sesuai dengan format <i>.txt</i>	Sistem menampilkan <i>pop up error</i> .	Sistem menampilkan <i>pop up error</i> .	<i>Pass</i>
7.	Menekan tombol posting tanpa memilih jurnal.	Menampilkan <i>pop up</i> peringatan bahwa belum ada jurnal yang dipilih.	Menampilkan <i>pop up</i> peringatan bahwa belum ada jurnal yang dipilih.	<i>Pass</i>

8.	Menekan tombol posting dan memilih jurnal dengan periode yang sedang ditutup.	Sistem tidak dapat memproses posting dan menampilkan <i>pop up</i> bahwa periode sedang ditutup.	Sistem tidak dapat memproses posting dan menampilkan <i>pop up</i> bahwa periode sedang ditutup.	<i>Pass</i>
9.	Menekan tombol posting dan memilih jurnal dengan periode terbuka.	Sistem memproses posting dan data yang dipilih akan hilang dari tabel serta menampilkan <i>pop up</i> bahwa jurnal berhasil diposting.	Sistem memproses posting dan data yang dipilih akan hilang dari tabel serta menampilkan <i>pop up</i> bahwa jurnal berhasil diposting.	<i>Pass</i>
10.	Mencari data menggunakan <i>input</i> cari.	Menampilkan data berdasarkan referensi yang dimasukkan oleh pengguna.	Menampilkan data berdasarkan referensi yang dimasukkan oleh pengguna.	<i>Pass</i>
11.	Memilih <i>filter</i> tipe jurnal tertentu.	Menampilkan data transaksi berdasarkan tipe jurnal yang dipilih pengguna.	Menampilkan data transaksi berdasarkan tipe jurnal yang dipilih pengguna.	<i>Pass</i>
12.	Memilih <i>filter</i> tipe transaksi tertentu.	Menampilkan data transaksi berdasarkan tipe transaksi yang dipilih pengguna.	Menampilkan data transaksi berdasarkan tipe transaksi yang dipilih pengguna.	<i>Pass</i>
13.	Memilih <i>filter</i> periode tertentu.	Menampilkan data transaksi berdasarkan periode yang dipilih pengguna.	Menampilkan data transaksi berdasarkan periode yang dipilih pengguna.	<i>Pass</i>
14.	Melakukan <i>sorting</i> pada masing-masing kolom.	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	Klik pertama data ditampilkan secara <i>ascending</i> , klik kedua data ditampilkan secara <i>descending</i> , dan pada klik ketiga <i>cancel sorting</i> .	<i>Pass</i>
15.	Menekan tombol detail berwarna biru dengan ikon berbentuk mata.	Berpindah ke halaman detail transaksi dan menampilkan data transaksi secara rinci berdasarkan data yang dipilih.	Berpindah ke halaman detail transaksi dan menampilkan data transaksi secara rinci berdasarkan data yang dipilih.	<i>Pass</i>
16.	Menekan tombol <i>edit</i> berwarna kuning dengan ikon berbentuk pensil.	Berpindah ke halaman ubah transaksi dan menampilkan data transaksi berdasarkan data yang dipilih.	Berpindah ke halaman ubah transaksi dan menampilkan data transaksi berdasarkan data yang dipilih.	<i>Pass</i>
17.	Menekan tombol <i>delete</i> berwarna merah dengan ikon berbentuk tempat sampah.	Menampilkan <i>pop up</i> konfirmasi hapus data.	Menampilkan <i>pop up</i> konfirmasi hapus data.	<i>Pass</i>

Lampiran 7A. Tambah Jurnal

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Memilih tanggal yang tidak sesuai periode.	Menampilkan <i>error messages</i> bahwa bulan dan tahun harus sesuai dengan periode.	Menampilkan <i>error messages</i> bahwa bulan dan tahun harus sesuai dengan periode.	<i>Pass</i>
2.	Menekan tombol simpan tanpa mengisi seluruh <i>field</i> yang tersedia.	Menampilkan <i>error messages</i> pada <i>mandatory field</i> .	Menampilkan <i>error messages</i> pada <i>mandatory field</i> .	<i>Pass</i>
3.	Memilih tipe transaksi lebih dari 2 tipe.	Menampilkan <i>error messages</i> bahwa maksimal pilih hanya 2 tipe transaksi.	Menampilkan <i>error messages</i> bahwa maksimal pilih hanya 2 tipe transaksi.	<i>Pass</i>
4.	Mengisikan nominal 0 pada <i>field</i> debit dan <i>field</i> kredit.	Tombol simpan berubah menjadi <i>disable</i> .	Tombol simpan berubah menjadi <i>disable</i> .	<i>Pass</i>
5.	Menekan tombol batal.	Berpindah ke halaman transaksi jurnal.	Berpindah ke halaman transaksi jurnal.	<i>Pass</i>
6.	Menekan tombol + pada kolom kode akun.	Menampilkan seluruh data Akun (COA) yang tersimpan di pengaturan Akun (COA) dengan status akun aktif.	Menampilkan seluruh data Akun (COA) yang tersimpan di pengaturan Akun (COA) dengan status akun aktif.	<i>Pass</i>
7.	Menekan tombol simpan setelah seluruh <i>field</i> diisikan.	Menyimpan data dan berpindah ke halaman transaksi jurnal dengan <i>pop up</i> data berhasil disimpan serta menampilkan data ke dalam tabel jurnal.	Menyimpan data dan berpindah ke halaman transaksi jurnal dengan <i>pop up</i> data berhasil disimpan serta menampilkan data ke dalam tabel jurnal.	<i>Pass</i>

Lampiran 7B. Ubah Jurnal

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol batal.	Berpindah ke halaman transaksi jurnal.	Berpindah ke halaman transaksi jurnal.	<i>Pass</i>
2.	Menekan tombol simpan.	Menyimpan perubahan data dan berpindah ke halaman transaksi jurnal dengan <i>pop up</i> data berhasil diubah serta memperbarui data yang terdapat pada tabel jurnal.	Menyimpan perubahan data dan berpindah ke halaman transaksi jurnal dengan <i>pop up</i> data berhasil diubah serta memperbarui data yang terdapat pada tabel jurnal.	<i>Pass</i>

Lampiran 7C. Detail Jurnal

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol kembali.	Berpindah ke halaman transaksi jurnal.	Berpindah ke halaman transaksi jurnal.	Pass
2.	Memodifikasi data pada <i>field</i> yang tersedia.	Seluruh <i>field</i> pada detail jurnal berjenis <i>disable</i> sehingga pengguna tidak dapat memodifikasi data.	Seluruh <i>field</i> pada detail jurnal berjenis <i>disable</i> sehingga pengguna tidak dapat memodifikasi data.	Pass

Lampiran 7D. Hapus Jurnal

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol batal.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	<i>Pop up</i> hapus tertutup dan data tidak terhapus.	Pass
2.	Menekan tombol hapus.	Data terhapus dan muncul <i>pop up</i> data berhasil dihapus selama 3 detik.	Data terhapus dan muncul <i>pop up</i> data berhasil dihapus selama 3 detik.	Pass

Lampiran 8. Halaman Laporan *Account Listing*

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol cetak excel tanpa memilih <i>field</i> yang tersedia.	Menampilkan <i>error messages</i> pada Akun awal, Akun akhir, Tahun, Periode, dan Tipe Transaksi dengan keterangan harus dipilih serta tombol cetak excel menjadi <i>disabled</i> .	Menampilkan <i>error messages</i> pada Akun awal, Akun akhir, Tahun, Periode, dan Tipe Transaksi dengan keterangan harus dipilih serta tombol cetak excel menjadi <i>disabled</i> .	Pass
2.	Mengisikan seluruh <i>field</i> yang tersedia dan menekan tombol cetak excel.	Tombol cetak excel berubah menjadi <i>loading</i> dan sistem akan menghasilkan laporan <i>Account Listing</i> .	Tombol cetak excel berubah menjadi <i>loading</i> dan sistem akan menghasilkan laporan <i>Account Listing</i> .	Pass

Lampiran 9. Halaman Laporan *Journal Listing*

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol cetak excel tanpa memilih <i>field</i> yang tersedia.	Menampilkan <i>error messages</i> pada Nomor posting awal dan Nomor posting akhir dengan keterangan	Menampilkan <i>error messages</i> pada Nomor posting awal dan Nomor posting akhir dengan keterangan harus dipilih	Pass

		harus dipilih serta tombol cetak excel menjadi <i>disabled</i> .	serta tombol cetak excel menjadi <i>disabled</i> .	
2.	Mengisikan seluruh <i>field</i> yang tersedia dan menekan tombol cetak excel.	Tombol cetak excel berubah menjadi <i>loading</i> dan sistem akan menghasilkan laporan <i>Journal Listing</i> .	Tombol cetak excel berubah menjadi <i>loading</i> dan sistem akan menghasilkan laporan <i>Journal Listing</i> .	<i>Pass</i>

Lampiran 10. Halaman Laporan *Trial Balance*

No	Kasus Pengujian	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
1.	Menekan tombol cetak excel tanpa memilih <i>field</i> yang tersedia.	Menampilkan <i>error messages</i> pada Tahun dan Periode dengan keterangan harus dipilih serta tombol cetak excel menjadi <i>disabled</i> .	Menampilkan <i>error messages</i> pada Tahun dan Periode dengan keterangan harus dipilih serta tombol cetak excel menjadi <i>disabled</i> .	<i>Pass</i>
2.	Mengisikan seluruh <i>field</i> yang tersedia dan menekan tombol cetak excel.	Tombol cetak excel berubah menjadi <i>loading</i> dan sistem akan menghasilkan laporan <i>Trial Balance</i> .	Tombol cetak excel berubah menjadi <i>loading</i> dan sistem akan menghasilkan laporan <i>Trial Balance</i> .	<i>Pass</i>