BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan program kerja magang Intern sebagai *Fullstack Developer* dilaksanakan dengan tujuan memberikan pengalaman kerja nyata bagi mahasiswa magang. PT Generasi Muda Gigih mencari Intern yang akan membantu tim untuk mengembangkan *website* serta aplikasi maka dari itu diperlukan individu yang memiliki keahlian, minat serta rasa ingin belajar yang tinggi dalam bidang pengembangan *software* menggunakan *Laravel*. Dalam prosesnya, peserta magang lebih banyak belajar mandiri menggunakan AI (*Artificial Intelligence*) dan diperbolehkan bertanya apabila kebingungan.



Gambar 3.1 Kedudukan Mahasiswa Magang

Berdasarkan gambar 3.1 posisi *Intern* ditempatkan ditempatkan dibawah *Junior Programmer* dan *Frontend Developer* yang dimana tugas dari *Intern* sendiri adalah untuk membantu pekerjaan dua divisi tersebut seperti membangun halaman *web* sesuai dengan permintaan klien, memperbaiki perhitungan pemesanan, membangun fitur sesuai dengan kebutuhan klien dan melakukan testing pada *web* atau aplikasi yang sudah hampir selesai. Pekerjaan *Intern* biasanya langsung dari

arahan dua divisi tersebut dan jika merasa kesulitan diperbolehkan untuk bertanya kepada dua divisi tersebut atau ke *Senior Programmer*.

3.2 Tugas dan Uraian Kerja Magang

Full Stack Developer Intern PT Generasi Muda Gigih memiliki banyak tugas yang disesuaikan dengan kebutuhan masing masing proyek yang sedang berjalan. Saat ini, Generasi Muda Gigih memiliki beberapa proyek yang sedang berlangsung untuk pengembangan sistem kustomisasi untuk klien seperti "Farrasindo Group, Karada, Jaya Abadi, Ocha Bella, OE dan Vigor". Seluruh proyek ini sudah memasuki tahap akhir yakni proses "Finishing, Implementasi dan Maintenance" untuk memastikan sistem dapat berjalan sesuai dengan permintaan klien dan tidak terjadi error yang disebabkan oleh kesalahan internal seperti kesalahan coding atau kesalahpahaman flow kerja oleh klien dan Generasi Muda Gigih.

Intern Generasi Muda Gigih ditempatkan sebagai bagian dari tim pengembangan, dengan peran sebagai pendukung proses pengerjaan proyek baik dari sisi front end maupun back end. Tugas yang diberikan biasanya akan disesuaikan dengan arahan dari Junior Programmer, Frontend Developer maupun Project Manager, hal ini dilakukan agar semua divisi bekerja secara kolaboratif dan pekerjaan dapat diselesaikan tepat waktu. Uraian lebih rinci mengenai tugas dan kontribusi selama periode magang dapat dilihat pada tabel 3.1



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Tabel 3.1 Uraian Pelaksanaan Kerja Magang	
---	--

No	Nama Proyek	Pekerjaan Yang Dilakukan	Waktu	Hasil
1		Pengenalan lingkungan kerja	Minggu 3	
2		Pengenalan <i>framework</i> Laravel dan proyek	Minggu 4	
3	Farrasindo Group	Mengimport Database, memperbaiki HTML, membuat fitur dan memperbaiki error	Februari Minggu 1 -3	Membuka website, inputan pilihan <i>profit</i> dan button <i>export</i> <i>excel</i>
5	Ocha Bella	Import Database, membangun tampilan, membuat fitur, mengubah tampilan dan modifikasi database	Februari Minggu 4, Maret Minggu 1 dan 2	Membuka <i>website</i> , tampilan list ktp, Menambah inputan <i>Customer</i> , modifikasi tampilan <i>report</i> , membuat rumus perhitungan dan menambah tabel pada <i>database</i> .
6	Karada	Import Database, reporting dan membuat halaman	Maret Minggu 3 - 4	Membuka website, halaman report Room set available, Halaman detail kode voucher
7	Vigor	Import Database,	April	Dapat membuka

		membuat fitur baru	Minggu 1- 3	<i>website</i> dan membuat fitur <i>brodman</i>
8	OE	Import Database, membuat halaman baru, membuat fitur baru dan memodifikasi halaman	April minggu 4 dan Mei minggu 1	Dapat membuka website, halaman detail transaction, halaman check point, halaman transaksi void, membuat fitur filter transaction, merender status, modifikasi filter inventory dan modifikasi halaman customer
9	Sempana	<i>Import Database</i> , modifikasi halaman dan membuat fungsi	Mei minggu 2- 3	Dapat membuka website, menambahkan tooltip dan filter transaksi
10	Jaya Abadi Tapioka	<i>Import Database</i> dan membuat button <i>excel</i>	Mei minggu 4 dan Juni minggu 1	Dapat membuka website dan mendownload laporan dalam bentuk Xlsx

Alur pengerjaan pekerjaan biasanya berdasarkan arahan atau permintaan dari Junior Programmer, Frontend Developer atau Project Manager. Seperti, meminta merubah tampilan HTML sederhana, Mengubah perhitungan yang terdapat dalam web, membuat relasi antar tabel (Database), Membuat fitur yang diinginkan oleh klien, testing dan finding bug dari sebuah proyek yang sudah

hampir selesai. Tingkat kesusahan dari pekerjaan yang diberikan bervariasi sehingga mahasiswa magang diperbolehkan untuk menggunakan *AI* [6] dan dibolehkan bertanya apabila merasa sangat kesulitan. Gambar 3.2 adalah gambar alur pengerjaan tugas :



Gambar 3.2 Alur Pengerjaan Tugas

Sebelum mengerjakan tugas, pertama tama mahasiswa magang diharuskan untuk memiliki *folder* proyek yang ingin dikerjakan. Cara untuk mengambil proyek tersebut, mahasiswa magang harus memiliki akun "*GitHub*" yang nantinya akan di *invite* oleh divisi yang memberikan tugas lalu mahasiswa magang dapat melakukan "*Cloning folder*" melalui aplikasi "*Visual Studio code*" dan disimpan ke dalam penyimpanan *local* mahasiswa magang. Setelah mahasiswa magang mempunyai *file project*, mahasiswa akan melakukan *pull* pada aplikasi *Visual Studio kode* guna untuk mendapatkan kode *terupdate* sehingga dapat membantu melanjutkan pekerjaan proyek. Setelah mahasiswa magang sudah memiliki *file project* dan 26

codingan terupdate, langkah selanjutnya adalah mengimport Database ke dalam MySQL local mahasiswa magang yang akan diexport oleh divisi yang memberikan tugas melalui Whatsapp. Setelah sudah siap untuk mengerjakan tugas, divisi akan menjelaskan dulu proyek ini tentang apa dan sudah sejauh mana proyek ini sudah berjalan serta menjelaskan tugas apa yang nantinya akan dikerjakan oleh mahasiswa magang. Selama pengerjaannya, mahasiswa magang diperbolehkan untuk mencari bantuan dari mana saja termasuk menggunakan AI seperti "Chat GPT dan Deep Seek" atau diperbolehkan untuk bertanya jika merasa sangat kesulitan atau tidak paham. Jika dirasa tugas yang dikerjakan oleh mahasiswa magang sudah selesai, maka harus segera dikonsultasikan kepada divisi yang bersangkutan atau langsung kepada *project manager* apakah ini sudah benar atau masih ada yang harus diperbaiki, jika masih ada yang salah maka harus diperbaiki namun jika sudah benar maka dapat langsung melakukan push pada Visual Studio kode yang berfungsi agar tiap anggota yang mengerjakan proyek tersebut dapat melakukan pull untuk mendapatkan kode terupdate dan dapat melanjutkan proyek.

3.2.1 Pengenalan Lingkungan Kerja

Pada minggu pertama masuk mahasiswa magang melakukan perkenalan dengan lingkungan kerja termasuk berkenalan dengan anggota anggota karyawan dari tiap tiap divisi. Mahasiswa magang juga harus memahami bagaimana latar belakang perusahaan dapat berdiri, bagaimana budaya bekerja di PT Generasi Muda Gigih serta proyek apa saja yang sedang berjalan. Memahami bagaimana aturan kerja yang diberlakukan di PT Generasi Muda Gigih guna agar tercipta suasana kerja yang nyaman dan kondusif.

3.2.2 Pengenalan Framework dan Proyek

Pada minggu ke 1 mahasiswa magang akan dikenalkan dengan *Framework* atau aplikasi yang digunakan untuk mengerjakan proyek. Mahasiswa magang diwajibkan untuk mengerti terlebih dahulu apa yang biasanya dikerjakan oleh para karyawan tetap serta mengerti apa itu *Laravel* dan bagaimana cara kerjanya. Dalam masa pengenalan ini, mahasiswa magang disarankan untuk melakukan pembelajaran mandiri seperti menonton video *youtube* yang berkaitan dengan *Laravel* dalam pembuatan *website* serta diberikan bimbingan tentang proyek apa saja yang nantinya memerlukan bantuan mahasiswa magang.

Seiring dengan pembelajaran mandiri, mahasiswa magang juga akan menyiapkan beberapa *framework* yang akan diinstall ke dalam laptop mahasiswa magang.



Gambar 3.3. Laragon

Framework yang wajib dimiliki oleh mahasiswa magang adalah "*Laragon*". *Laragon* ini berfungsi untuk menjalankan website yang dibuat menggunakan *Laravel*, Laragon juga sangat membantu developer dalam membangun sebuah website karena sudah dilengkapi dengan *Database* yaitu "*HeidiSQL*" namun jika tidak ingin menggunakan "*HeidiSQL*" *Laragon* memungkinkan pengguna untuk menggunakan *Database* lain seperti "*phpMyAdmin*" yang digunakan oleh mahasiswa magang [7].

Selain Menggunakan "Laragon" mahasiswa magang juga harus menginstall "Visual Studio code"



Gambar 3.4 Visual Studio code

"Visual Studio code", merupakan aplikasi yang wajib dimiliki oleh mahasiswa magang maupun karyawan tetap. Framework ini berfungsi untuk menulis kode yang nantinya akan membangun tampilan website serta membuat fitur menggunakan bahasa pemrograman "PHP (Hypertext Preprocessor)" [8].

← → ♂ ≒ github.com				6 x D	I 🕶 🔞 E
🔠 🛛 M Gmail 👼 🤨 YouTube 🍳 Maps					All Bookmarks
Dashboard				 arbit1234 Set status 	≓ × Account switcher
Top repositories Item Find a repository Itelalanumijoja korada system Is varanežičo korab jolia sys Itelalanumijoja korada system Is varanežičo kab jolia sys Itelalanumijoja korada system Is repository Itelalanumijoja korada system Is varanežiče farasindo.group Itelalanumijoja korada system Is varanežiče farasindo.group Itelalanumijoja korada system Is varanežiče farasindo.group Itelalanumijoja korada system Is varanežiče farasindo.group.ali Itelalanumijoja korada system	Home Add Copilot Add Copilot C Explore Python ternary syntax D Demo Pyth C Explore Python ternary syntax D Demo Pyth C Explore Python ternary syntax D Demo Python C Explore Python C Explo		Latest ch 5 hou Upde custe 6 hou GitH Code 0 31 ho Depe frequ 0 12 ho	 A Your profile If Your repositories Your repositories Your projects Your projects Your dists Your organizations Your organizations Your enterprises Your enterprises 	
Sione more	Code with Coplet Develop with Al-powered code suggestions using GitHub Coplet. Codepares and VS Code See more tutorial projects O Start writing code	with GitHub Pages. Helio GitHub Actions Create a GitHub Action and use it in a workflow.	Remi APIs View Explore n Company Official Op- Magento C Source cod Security in	Iny Enterprise Frature preview Settings GitHub Website GitHub Docs A: GitHub Support GitHub Support GitHub Community	Free

Gambar 3.5 Github

Selanjutnya adalah "*Github*". *Github* ini bukan sebuah aplikasi yang harus *didownload* melainkan sebuah *platform* yang berfungsi untuk menyimpan, mengelola serta membangun proyek bersama [9]. Fungsi dari

platform ini adalah untuk membagikan folder project agar semua karyawan dan mahasiswa magang dapat mengerjakan bersama sama. "*Github*" ini juga sudah terhubung dengan "*Visual Studio code*" memiliki fitur "*Push* dan *Pull*". *Push* berfungsi untuk mengupload pekerjaan yang kita kerjakan sedangkan *Pull* berfungsi untuk mengambil pekerjaan rekan kerja yang sudah di *Push*, sehingga setiap individu memiliki kode yang paling baru.



Gambar 3.6 phpMyAdmin

Selanjutnya, yang terakhir adalah *phpMyAdmin*, sebuah aplikasi berbasis web yang digunakan sebagai alat bantu dalam pengelolaan *database* selama mahasiswa magang mengerjakan proyek di PT Generasi Muda Gigih. *phpMyAdmin* digunakan secara luas karena kemampuannya dalam mengelola *database MySQL* melalui antarmuka grafis yang intuitif, sehingga sangat memudahkan dalam pengaturan data tanpa perlu menulis perintah *SQL* secara manual. Dalam konteks magang, *phpMyAdmin* berfungsi untuk menyimpan dan mengelola seluruh data yang diinput oleh pengguna melalui aplikasi atau website yang dikembangkan.

Keunggulan penggunaan phpMyAdmin tidak hanya terletak pada fungsinya sebagai media penyimpanan data, tetapi juga pada kemampuannya dalam memvisualisasikan struktur data secara sistematis dalam bentuk tabel, relasi antar tabel, hingga *eksekusi query* yang kompleks. Data yang tersimpan dalam *phpMyAdmin* dapat dengan mudah diakses, dibaca, dimodifikasi, maupun dihapus, sesuai dengan prinsip *CRUD (Create, Read, Update, Delete)*. Dengan struktur data yang rapi dan terorganisir, *developer* dapat melakukan proses analisis, *debugging*, dan pengembangan fitur dengan lebih cepat dan efisien. Oleh karena itu, *phpMyAdmin* menjadi salah satu komponen penting dalam proses pengembangan sistem berbasis web, terutama untuk kebutuhan *backend* yang berhubungan langsung dengan pengelolaan data pengguna.

30

3.2.3 Farrasindo Group

3.2.3.1 Import Database

Sebelum memulai pekerjaan dalam proyek ini, mahasiswa magang diwajibkan untuk memiliki *database* yang sudah dibuat oleh *developer* lain. Dengan tidak adanya *database*, proyek tidak dapat dibuka dan pekerjaan tidak dapat dilanjutkan.

sales_invoice_headers	🚖 🔟 Browse	M Structure	Search 3	🖌 İnsert 💮 En	apty 🙆 Drop	18 InnoDB	utf8mb4_general_ci	96.0 KiB	
sales_order_detail2s	😭 📋 Browse	K Structure	* Search	🕯 Insert 🛛 🚍 En	npty 🥥 Drop	Ø InnoDB	utf8mb4_general_ci	16.0 KiB	
sales_order_details	🎪 🔟 Browse	M Structure	👒 Search 💈	🕯 Insert 🛛 🙀 En	npty 🤤 Drop	48 InnoDB	utf8mb4_general_ci	64.0 KIB	
sales_order_headers	🚖 📄 Browse	M Structure	Search 3	linsert 📻 En	opty 🥥 Drop	40 InnoDB	utf8mb4_general_ci	144.0 KiB	
sales_persons	🚖 🔟 Browse	M Structure	Search	🕯 Insert 💮 En	pty 🤤 Drop	2 InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
sales_quotation_details	🚖 🔄 Browse	M Structure	* Search	ie Insert 🚍 En	npty 🕥 Drop	@ InnoDB	utf8mb4_general_ci	16.0 KiB	-
sales_quotation_headers	🚖 🔟 Browse	M Structure	Rearch S	🗟 Insert 💮 En	pty 🥥 Drop	Ø InnoDB	utf8mb4_general_ci	32.0 KiB	
sales return details	Browse	> Structure	& Search	i Insert 📻 En	pty 🙆 Drop	Ø InnoDB	utf8mb4 general ci	48.0 KiB	
sales return headers	Browse	Structure	Search 3	🖌 İnsert 🚍 En	pty C Drop	Ø InnoDB	utf8mb4 general ci	64.0 KiB	
services	See III Browse	Te Structure	 Search 3 	la Insert 📾 En	noty @ Drop	2 InnoDB	utf8mb4 general ci	16.0 KiB	
ao timeline project	A III Province	De Structuro	A Coardh 2	i losort 💭 En	ntu 🔿 Dron	0 InnoDR	utf9mb2_general_cl	16 0 FIR	
So_unienne_project	Mr Drowse	Churchare	og Gearding	reinsen wert	ipty Obiop	o Innobb	utombo_general_ci	10.0 KID	
	S Drowse	M Structure	Search F	e insert me cr	ipty 🥥 Drop	a INNODP	utiomb3_unicode_ci	10.0 KID	
stock_opname_checkers	🚖 📖 Browse	M Structure	R Search	-i Insert 👾 En	npty 🥥 Drop	Ø InnoDB	utf8mb4_general_ci	16.0 KIB	-
stock_opname_details	🚖 🔲 Browse	M Structure	Search	🕯 Insert 🔗 En	npty 🥥 Drop	7 InnoDB	utf8mb4_general_ci	16.0 KiB	
stock_opname_headers	🊖 🔝 Browse	M Structure	🤹 Search 💈	🕯 Insert 💮 En	npty 🥥 Drop	1 InnoDB	utf8mb4_general_ci	16.0 KiB	-
 suppliers 	🚖 🔝 Browse	M Structure	Search	🖬 İnsert 📻 En	npty 🥥 Drop	3,165 InnoDB	utf8mb4_general_ci	704.0 KiB	
supplier_invoice_detail1s	🏫 🔲 Browse	M Structure	Search	🕯 Insert 🛛 🚍 En	npty 🥥 Drop	263 InnoDB	utf8mb4_general_ci	128.0 KiB	1
supplier_invoice_detail2s	Srowse	Structure	* Search	d Insert 📾 En	apty 🥥 Drop	Ø InnoDB	utf8mb4_general_ci	48.0 KiB	
supplier invoice headers	👉 🥅 Browse	Structure	Search 3	🖬 İnsert 📾 En	pty C Drop	199 InnoDB	utf8mb4 general ci	160.0 KIB	
unit materials	A Browse	Le Structure	A Search	losert 📾 En	noty A Drop	56 InnoDB	utf8mb4 general ci	48.0 KIB	
watebouses	A Browse	Re Structure	Search 3	Li Insert 🔗 Fr	anty Comp	5 InnoDB	utf8mb3_unicode_ci	48.0 KIB	
111 tables	Sum	1 ou ou ou ou	of ocuron 3	Pe moont My ch	hù Annh	65 569 JanoDB	utf8mb4_0000_al_cl	20 2 Mil	0.0
TTT tables	Juin					03,509 111000	udomb4_0500_al_ci	20.2 110	00
↑ Check all With se	lected:								
	sales_invoice_headers sales_order_detail2 sales_order_detail3 sales_order_details sales_order_headers sales_persons sales_quotation_headers sales_return_details sales_return_details sales_return_teaters sortices sortices stock_opname_heckers stock_opname_details supplier_invoice_detail1s supplier_invoice_detail2s supplier_invoice_detail1s supplier_invoice_detail2s supplier_invoice_detail1s supplier_invoice_detail1s supplier_invoice_detail1s supplier_invoice_detail1s supplier_invoice_detail2s supplier_invoice_detail2s supplier_invoice_detail2s supplier_invoice_detail2s supplier_invoice_detail1s supplier_invoice_detail1s supplier_invoice_detail1s supplier_invoice_detail2s supplier_invoice_detail2s supplier_invoice_detail3 supreprint_invoice_detail3 supplier_invoice_detail3 supplier	sales_invoice_headers sales_order_detail2 sales_order_details sales_order_details sales_order_headers sales_order_headers sales_order_headers sales_order_headers sales_order_headers sales_order_headers sales_order_headers sales_order_headers sales_return_headers sales_return_headers sales_return_headers sales_order_headers sales_return_headers sales_order_headers sales_return_headers so_timeline_project stock_opname_headers stock_opname_headers stock_opname_headers supplier_invoice_details supplier_invoice_details supplier_invoice_heatails supplier_invoice_heatails supplier_invoice_heatails supplier_invoice_heatails sales_returnetais warehouses supplier_invoice_heatails les_involce_headers sales_order_detaliz sales_order_detaliz sales_order_detalis sales_order_beaders sales_quotation_headers sales_quotation_headers sales_quotation_headers sales_quotation_headers sales_quotation_headers sales_return_teaters sales_return_teaters sales_return_teaters so timeline_project stock_opname_headers stock_opname_headers suppler_involce_detalis supple	sales_order_detall2x sales_order_detall2x sales_order_detall2x sales_order_detall2x sales_order_detall2x sales_order_detall2x sales_order_detall3 sales_return_beaders so_timeline_project so_timeline_project stock_opname_heckers so_tonse // Structure seards stock_opname_heckers so_tonse // Structure seards supplier_invoice_detall3 sales_order_detall3 supplier_invoice_detall3 so_tonse // Structure seards supplier_invoice_hedetall3 marehouses so_tonse // Structure seards supplier_invoice_hedetall3 morehouses so_tonse // Structure seards supplier_invoice_hedetall3 morehouses so_tonse // Structure seards supplier_invoice_hedetall3 morehouses so_tonse // Structure seards supplier_invoice_hedatall3 more	sales_invoice_headers Browse Kuuture Search Hinsett Errowse Kuuture Search Search Hinsett Errowse Kuuture Search	sales_involce_headers # Browse # Structure * Search # Insert Empty Drop sales_order_detail2 # Browse # Structure * Search # Insert Empty Drop sales_order_details # Browse # Structure * Search # Insert Empty Drop sales_order_headers # Browse # Structure * Search # Insert Empty Drop sales_groutation_details # Browse # Structure * Search # Insert Empty Drop sales_groutation_details # Browse # Structure * Search # Insert Empty Drop sales_groutation_headers # Browse # Structure * Search # Insert Empty Drop sales_return_teaders # Browse # Structure * Search # Insert Empty Drop satuses # Browse # Structure * Search # Insert Empty Drop statuses # Browse # Structure * Search # Insert	sales_invoice_headers # Browse Structure Search # Insert Empty Drop 18 InnoDB sales_order_details # Browse # Structure Search # Insert Empty Drop 48 InnoDB sales_order_details # Browse # Structure Search # Insert Empty Drop 48 InnoDB sales_order_headers # Browse # Structure Search # Insert Empty Drop 48 InnoDB sales_groutation_details # Browse # Structure Search # Insert Empty Drop 0 InnoDB sales_groutation_headers # Browse # Structure Search # Insert Empty Drop 0 InnoDB sales_return_teaders # Browse # Structure Search # Insert Empty Drop 0 InnoDB sales_return_teaders # Browse # Structure Search # Insert Empty Drop 0 InnoDB satures #	sales_invoice_headers	sales_invoice_headers	

Gambar 3.7 Table Database Farrasindo

Gambar 3.7 menunjukkan struktur *database* yang digunakan dalam proyek "Farrasindo Group". *Database* ini terdiri dari 111 tabel yang saling terhubung satu sama lain dan dirancang untuk mendukung proses bisnis secara menyeluruh. Dalam proyek ini, mahasiswa magang tidak terlibat langsung dalam tahap perancangan *database*. Oleh karena itu, *Project Manager* menyediakan *file database* yang telah dirancang dalam format "SQL file", yang kemudian dikirimkan melalui aplikasi *WhatsApp* sehingga mahasiswa magang dapat melakukan *import* ke dalam "*phpMyAdmin*" *local* mahasiswa magang.

3.2.3.2 Memperbaiki HTML

Proyek ini 80 persen sudah hampir selesai, semua tampilan dan fitur sudah jadi. Namun, proyek biasanya melakukan "*Meeting*" untuk melaporkan progres dan melihat "*Feedback*" dari klien apakah sudah puas atau masih ada yang ingin diperbaiki. Mahasiswa magang diharuskan untuk memperbaiki tampilan HTML pada bagian "*Balance Sheet*" dengan menambahkan pilihan "*Profit Center*"

ന	🔲 🥼 localhost / localhost / farrasindo 🔿 🦷 Laporan Balanc	SEMUA	A
		Farracon Sukabumi	
C	▲ Not secure farrasindo_group.test/report/general_le	Pabrikasi - Sentul	
		Proyek - Four Point Pontianak	
	🖉 Dashboard 🛛 📰 Master Data 👻 💼	Mitra - Emerald Bintaro	Produc
		Mitra - Sentul Borongan	
	💴 General Ledger 🝸 📑 Report 👻 F	Mitra - Four Point Pontianak	
		Proyek - Hotel Neo Sentul	
		Mitra - Hotel Neo Sentul	
		Mitra - Decorient	
	Laporan Balance Sneet	Proyek - Decorient SMB2	
		Proyek - Rumah Mewah Golf Pik	
		Mitra - Rumah Mewah Golf Pik	
		Proyek - Emerald Bintaro	
		Proyek - IKK	
	Isian dengan asterik (*) wajib diisi!	Proyek - Nuansa Cilangkap	
		Mitra - IKK	
	Profit Center *	SEMUA	
	Tanggal Mulai *	23-04-2025	
	Tanggal Berakhir *	24-04-2025	
	Buat Laporan		

Gambar 3.8 Tampilan Balance Sheet

Gambar diatas merupakan tampilan dari laporan *balance sheet* yang dikerjakan oleh mahasiswa magang dengan arahan dari *Senior Programmer*. Halaman ini berfungsi untuk menampilkan laporan neraca berdasarkan cabang mana yang akan di pilih (*Profit Center*) yang ditentukan oleh tanggal mulai dan tanggal berakhir. Halaman ini dibuat dalam *Blade* dan mengambil variabel yang sudah dibuat dalam *controller (\$ProfitCenters)* sehingga dapat memilih *profit center* yang diinginkan.



Gambar 3.9 Codingan Header

Gambar diatas merupakan codingan untuk membuat "*Header*" yang terdapat pada bagian atas tampilan, fungsi dari *\$type* adalah untuk mengambil pilihan laporan apa yang ingin dibuka. Laporan *Balance Sheet* terdapat dalam pilihan "*Report*" lalu "*General Ledger*" sehingga menggunakan *\$type* untuk membuat judul dari pilihan *report* apa yang di pilih oleh klien.



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara



Gambar 3.10 Letak laporan balance sheet

3.2.3.3 Membuat fitur button excel

Setelah membuat tampilan untuk memilih "Profit Center" dan memilih waktu mulai dan berakhir, maka langkah selanjutnya adalah masuk ke dalam halaman pembuatan laporan. Laporan aku terbuat sesuai dengan "Profit Center", waktu mulai dan waktu berakhir yang dipilih.

				Dist Date 2	
		Balance Sh	eet Reports	Print Date: 20	25-04-24 10:18:52
		Period: 2025-04-	24 To 2025-04-25		
		Profit Cen	ter: SEMUA		
ktiva			Pasiva		
1) AKTIVA	-920.982.453		(2) KEWAJIBAN	125.708.023.534	
(10110) Kas Farracon Precast		-4.697.900	(210201) HUTANG USAHA		961.800.448
1102102) BCA 287-0395852 DROPINGAN FBI SENTUL		-40.141.261	(210202) HUTANG USAHA UNINVOICE		124.205.099.325
1102117) FARRACON BCA 287-7771403		-649.083.292	(210311) Hutang Pajak - PPh 21 Karyawan		37.500
1102118) FARRACON BCA 287-7881403		-227.060.000	(210606) Hutang PT. Farrasindo perkasa		45.000.000
110295) Bank Garansi	125.907.248.570		(210609) Hutang Leasing		-29.277.739
(110513) Piutang Karyawan		800.008	(210611) Hutang PT. Fresh Beton Indonesia		525.364.000
(110518) Piutang PT. Farrasindo		142.000.000	TOTAL EQUITY	125.708.023.534	125.708.023.534
(10530) Piutang Kasbon Mandor		-10.502.103	TOTAL PASINA	125 708 023 534	125 708 023 534
(10531) Piutang PT Fresh Beton Indonesia		548.074.900		125.700.025.554	123.100.023.334
110612) PERSEDIAAN BAHAN BAKU		125.262.668.146			
110613) PERSEDIAAN BARANG JADI		8.458.127			
(10706) Uang Muka Pembelian		-66.138.500			
(1302) Aktiva Dalam Proses Pembangunan (Workshop Sentul)		21.888.000			
TOTAL ASSET TETAP	124.986.266.117	124.986.266.117			
FOTAL AKTIVA	124.986.266.117	124.986.266.117			

Gambar 3.11 Button Excel

Pada gambar 3.11 terdapat tiga *action button* yang dapat diklik. Mahasiswa magang mendapatkan tugas untuk membuat *button excel* berwarna hijau. Fungsi dari *button excel* ini adalah mencetak laporan yang

terdapat pada tampilan menjadi format "Xlsx". Berikut ini adalah kode yang membentuk fungsi untuk mendownload Xlsx.



Gambar 3.12 kode function export excel

Gambar diatas merupakan kode function dari export excel menggunakan "Javascript (JS)", fungsi ini yang memungkinkan laporan dapat didownload dalam bentuk "Xlsx". Ketika button excel di klik maka dia akan memunculkan notifikasi *"Loading" (#btn-excel-loading)* agar pengguna tau bahwa laporan sedang di proses. Lalu function akan mengambil nilai dari variabel *"typeReport"* yang diisi dari *blade directive* yaitu *{{\$type}}* yang dapat diartikan nilai tergantung dari server side Laravel saat halaman dirender. Variabel "url" dibentuk sebagai "endpoint" yang akan digunakan untuk mengekspor data ke route "admin.general ledger.export". Di dalam ditambahkan URL tersebut beberapa parameter seperti query "start date", "end date", "type" dan "profit center" yang diisi berdasarkan inputan dari pengguna. Selanjutnya, "window.location.href=url;" digunakan untuk mengarahkan browser ke url yang dituju sehingga file "Xlsx" dapat terdownload, Karena tidak menggunakan "AJAX" maka file akan langsung di download melalui browser. Setelah terdownload, terdapat "setTimeout()" selama 500 milidetik untuk mengembalikan tombol *export* ke keadaan awal.

3.2.3.4 Memperbaiki Error

Proyek "Farrasindo" saat ini sudah memasuki tahap "Finishing" dan proyek ini sudah mulai digunakan oleh klien sehingga memerlukan dukungan serta pengembangan yang lebih lanjut guna untuk penyempurnaan sistem. Dalam proyek ini, mahasiswa magang ditugaskan untuk melakukan "Testing" khususnya pada bagian "Sales" submenu "Rancangan Anggaran Biaya (RAB)" dan "Bill of Quantity (BQ)". Proses testing dilakukan dengan menjalankan beberapa skenario untuk melihat apakah ada terjadi error atau bug dalam fitur tersebut. Apabila terjadi error, mahasiswa magang diwajibkan untuk melakukan "Screenshot" pada bagian yang error dan wajib untuk melaporkan kepada "Project Manager" dalam bentuk dokumentasi tertulis seperti "Pdf" atau "Docs".



Gambar 3.13 Error pada bagian RAB

Error pertama yang ditemukan pada bagian "RAB" adalah tidak dapat melakukan "*Print Proforma*" yang disebabkan oleh "*item_category*" terbaca "*Null*" sehingga tidak dapat melakukan "*Print Proforma*". *Print proforma* adalah salah satu fitur yang berfungsi untuk mencetak "*Invoice*" dalam bentuk "*Pdf*" yang isinya sesuai dengan inputan klien.

"itemCategory" => \$itemCategory, "unitId" => \$dataDetail->unit_id, "unitName" => \$unitName,

Gambar 3.14 Perbaikan error pada bagian RAB

Gambar 3.14 adalah perbaikan yang dilakukan di dalam kode untuk mengatasi error pada gambar 3.13 Error yang terjadi disebabkan karena nilai dari "item category" tidak terbaca (Null), perbaikan yang dilakukan adalah menambahkan kondisi di dalam "foreach". "foreach" di sini berfungsi untuk mengambil satu per satu data dari relasi "budget plan details" milik "\$dataDB", dalam kasus ini berfokus untuk memanggil nilai dari "itemCategory" dan "unitName". Langkah pertama pembuatan variabel kosong "\$itemCategory=""," yang berfungsi untuk menyimpan nama kategori apabila ditemukan. Pada "If" yang pertama berfungsi untuk membedakan jenis data karena dalam kasus ini hanya mengambil jenis data "item", lalu mengecek apakah "\$dataDetail" memiliki relasi dengan "item", lalu yang terakhir memastikan bahwa "item" memiliki relasi dengan "item category". Lalu pada bagian "SunitName = " ";" dibuat untuk menampung nama unit, lalu pada "if" dibentuk untuk mengecek apakah data "unit material" ada atau tidak, jika ada maka "item" memiliki relasi dengan "unit material".



37 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Gambar 3.15 Error pada bagian BQ

Selanjutnya terdapat *error* pada bagian BQ yang terjadi akibat tidak ada daftar barang yang dipilih oleh klien lalu mengklik *button* "Simpan". BQ dan RAB saling berhubungan, data yang diambil dari BQ harus sudah di *approve* terlebih dahulu di bagian RAB, penambahan barang dan segala perhitungan harga akan dilakukan di RAB dan pada bagian BQ hanya filterisasi apakah ada barang yang akan dikurangi atau tidak. *Error* ini terjadi pada saat BQ tidak ada daftar barang sama sekali sehingga diperlukan beberapa perbaikan pada bagian "*Controller*" bagian "*Update*".



Gambar 3.16 Perbaikan error pada bagian BQ

Perbaikan yang dilakukan terletak pada bagian "BillOfQuantityController" pada "Function update" dengan menambahkan "if(\$itemIds==null)", kondisi ini membuat apabila "itemIds" memiliki nilai "null" atau tidak ada nilai maka akan memunculkan notifikasi "Barang tidak boleh kosong" yang ditulis dengan kode "return back ()".

3.2.4 Ocha Bella

3.2.4.1 Import Database

Sebelum mengerjakan proyek mahasiswa magang diwajibkan untuk melakukan *import database* kedalam "*Device*" pribadi agar dapat membuka *website*. Dalam pembuatan *database* proyek "Ocha Bella" mahasiswa magang tidak ikut dalam perancangan *database* karena proyek sudah berjalan sebelum mahasiswa magang melaksanakan magang.

phpMyAdmin	🗕 🧋 Server, localhost 33	06 s 📑 Database	karada									
650000C	🔀 Structure 🖉 SQ	L 🔍 Search	🗟 Qu	ery 🖶	Export 📻	Import	Operations	Privileges	Routines	S Events	Triggers	C Designer
lecent Equarites	stock_ins		*	Browse	M Structure	R Search	💱 Insert 👷 Em	ty 🤤 Drop	e InnoDB	utf8mb4_general_c	48.0 KiB	
	suppliers		*	Browse	K Structure	Rearch	💱 Insert 🗮 Em	ty 🤤 Drop	1 InnoDB	ut/8mb4_general_c	i 48.0 KiB	-
karada 🔺	supplier_invoice_de	tail1s	*	Browse	M Structure	R Search	👫 Insert 👳 Em	ty 😂 Drop	0 InnoDB	utf8mb4_general_c	i 48.0 KiB	
Type to filter these, Erth X	supplier_invoice_de	tail2s	*	Browse	V Structure	.e Search	Si Insert 🕀 Em	ty 😄 Drop	0 InnoDB	utt8mb4_general_c	48.0 K18	
1 7	supplier_invoice_he	aders	*	Browse	M Structure	R Search	Si Insert 🖶 Em	ty 😄 Drop	0 InnoDB	utf8mb4_general_c	64.0 KiB	
	temp bookings		-	Browse	M Structure	.e Search	Se Insert 😑 Em	ty 🥥 Drop	@ InnoDB	utt8mb4 general o	16.0 KiB	
Hald account receivables	temp invoices		-	Browse	M Structure	Search	Si Insert CE Em	ty C Drop	10 InnoDB	utf8mb4 general o	16.0 KiB	
Account receivable detai	transaction booking	1 slots	~	Browse	le Structure	. e Search	Si Insert 📾 Em	ty C Drop	@ InnoDB	uti8mb4_general_c	i 64.0 KiB	
t- admin_users	transaction details		-	Browse	Lei Structure	Search	Seinsert E Em	ty 🙆 Drop	78 InnoDB	ut/8mb4_general_c	і за.еків	
The app_birthday_coupons	transaction detail r	artners	-	Browse	le Structure	Search	Seinsert @ Em	ty 🔿 Drop	46 InnoDB	uti8mb4_general_c	i 48.0 KiB	
App_birthday_promo_usa	transaction headers		-	Browne	14 Structure	a Search	Salared B Em	hr 👄 Dron	46 InnoDB	uff8mb4_general_c	i 144 0 K18	
* M app_memberships	transaction_iteducit		THE I	- Drowne	1. Structure	a Search	Same and Same	ty 😸 Drop	24 InnoDB	utilizabil general c	i 32 8 K18	
+- Mapp_membership_advanc			- 14	III Browne	D. Churcher	G Search	Billored @ Com	ty 😸 Drop	24 InnoDB	uttemb4_general_c	1 64 0 F10	-
+- M app_membership_benefit	uansacuon_paymen	118	N I	Elowse	M Structure	og Search	Se insert me Eng	iy 😅 Diop	34 INNOUS	utomb4_general_c	04.0 K10	
The app_membership_bonus_	unit_materiais		(1)	Browse	M Structure	ing Search	are insert 👾 Em	ty 😅 Urop	6 INNOUR	uttsmb4_general_c	9 4876 KIR	
tric app_membership_details	users		宮	Browse	M Structure	Search	Se Insert 👾 Em	ty 🤤 Drop	8 InnoDB	utf8mb3_unicode_i	54.0 K18	
+ app_payment_types	user_addresses		賞	Browse	M Structure	্র Search	Se Insert 👾 Em	ty 🤤 Drop	@ InnoDB	utf8mb4_general_c	i 32.0 KiB	1
⊕ 🖟 ap_detail1s	user_app_spending	_messages	会	Browse	M Structure	R Search	💱 İnsert 👷 Em	ty 🥥 Drop	e InnoDB	utf8mb4_unicode_	3 16.0 KiB	
÷-₩ ap_detall2s	user_membership_l	nistories	会	Browse	M Structure	্র Search	👫 Insert 👷 Em	ty 🥥 Drop	e InnoDB	utf8mb4_unicode_u	;i 16.0 KiB	1
+ A ap_headers	user_notifications		- 🚖 - I	Browse	M Structure	R Search	🚰 Insert 👷 Em	ty 🤤 Drop	0 InnoDB	utf8mb4_unicode_	i 32.0 KiB	
+ Ar_detail1s	user_point_histories	•	2	Browse	M Structure	Rearch	💱 İnsert 👮 Em	ty 🤤 Drop	10 InnoDB	utf8mb4_unicode_	i 32.0 K18	-
+- r ar_detail2s	user_redeems		* 1	Browse	M Structure	R Search	👫 Insert 👮 Em	ty 😂 Drop	0 InnoDB	utf8mb4_general_c	i 48.0 K18	-
trig allendance confouration	user_reviews		- 🚖 - i	Browse	K Structure	R Search	👫 Insert 🛛 🚍 Em	ty 🤤 Drop	0 InnoDB	utf8mb4_unicode_u	i 48.0 K18	-
attendance_contiguration	user_review_images	•	*	Browse	M Structure	R Search	👫 Insert 👷 Em	ty 🤤 Drop	0 InnoDB	utf8mb4_unicode_	32.0 KiB	1.1
+- M authorization modules	user_vouchers		*	Browse	M Structure	👒 Search	💱 Insert 👷 Em	ty 🤤 Drop	12 InnoDB	utt8mb4_general_c	48.0 KiB	
+- authorization_positions	user_voucher_code	8	*	Browse	M Structure	Rearch	💱 Insert 👷 Em	ty 😄 Drop	25 InnoDB	utf8mb4_general_c	32.0 KiB	
H-M autonumbers	voucher_promo_com	les		Browse	M Structure	is Search	💱 Insert 🛛 📾 Em	ty 🥥 Drop	1 InnoDB	ut/8mb4_general_c	i S0.0 KiB	-
* auto_email_receivers	warehouses		*	Browse	M Structure	R Search	Si Insert 😁 Em	ty 😄 Drop	0 InnoDB	utf8mb3_unicode_	48.0 KiB	
+- k bank_cashin_details	167 tables		Sum						6,869 InnoDB	utf8mb4_0900_ai	ci 9.0 MiB	0 B
+ M bank_cashin_headers	+ Check all	With selected		~								
triv bank_cashout_details		with selected.		•								
+ / banners												
tooking_details	Bennt Ba Data dictionar	У										
+ M booking_detail_partners	Create new table											
+ kooking_headers												
+- M booking_notes	Table name	Number of c	olumns									
+ r k branches		4			Create							
triff carts												
category_coa_sales												
The coas	Console											

Gambar 3.17. Database Ocha Bella

Gambar 3.17 merupakan struktur *database* yang digunakan dalam proyek "Ocha Bella" yang terdiri dari 167 tabel yang saling berhubungan. Setiap tabel memiliki relasi yang dirancang untuk mendukung integrasi data serta kelancaran proses bisnis dalam sistem yang sedang dikembangkan.

3.2.4.2 Membangun tampilan

Mahasiswa magang ditugaskan untuk membuat sebuah halaman baru dengan nama "Laporan KTP Customer" yang berisi sebuah tabel yang berisikan tentang informasi "KTP Customer" yang di input dari bagian "*Master Customer*".

CV. Putra Nusa	C	List Ktp Customer				Print Export Excel	Dashboard List Ktp Custome
Ø Dashboard		DATA KTP CUSTOMER					
III Master Data	¢						
💼 Barang		NAMA CUSTOMER	NIK	NAMA КТР	LIMIT KTP	PENGGUNAAN LIMIT	
Purchasing		wif	7	Î	9,00	0,00	
Sales	¢	yuyuy	5	9	6,00	0,00	
Finance		kop	90	ро	96,00	0,00	
General Ledger	:5						
E Report	÷						
Purchasing							
Sales	v						
Laporan Sales Order							
Laporan Invoice Penji	jualan						
Laporan Invoice Issue	ed						
Laporan History Penji	jualan						
Laporan KTP Custo	omer						
Finance	¢						
General Ledger	4						Crafted by DT Cananai Muda Cialb
n Pennaturan	¢	CY FUTRA 1903A © 2023					Graned by Fill Generasi Muda Gigin

Gambar 3.18. Halaman List Customer

Gambar 3.18 menampilkan hasil dari pembuatan halaman "List Customer" yang dibuat dalam bentuk tabel dengan 5 kolom yaitu "Nama Customer", "NIK", "Nama KTP", "Limit KTP" dan "Penggunaan Limit". Data yang ditampilkan pada halaman ini tersimpan dalam database "Ocha Bella" yaitu tabel "Customer_ktps". Halaman ini juga memiliki 2 action button yang terletak pada bagian kanan atas yaitu "Print" dan "Excel". Kedua tombol ini berfungsi untuk mendownload data namun dalam format file yang berbeda yaitu "PDF" dan "Excel". Gambar 3.19 dan 3.20 dibawah adalah hasil dari action button ketika di klik.



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

List Customer KTP

Nama Customer	NIK	Nama KTP	Limit KTP	Penggunaan Limit
wif	7	j	9	9
yuyuy	5	g	6	6
kop	90	ро	96	96

Gambar 3.19. Hasil action button print

	А	В	С	D	E	F	G
1	Nama Cus	Nik	Nama Ktp	Limit KTP	Pengguna	an Limit	
2	wif	7	j	900	0		
3	yuyuy	5	g	600	0		
4	kop	90	ро	9600	0		
5							
6							
7							
8							
9				,			
10							
11							
12							
13							
14							

Gambar 3.20. Hasil action button excel (hapus)



Gambar 3.21 kode membangun tampilan List Ktp

Gambar diatas merupakan kode yang ditulis di dalam "Visual Studio code". kode ini hanya berfungsi untuk membuat visual yang dapat dilihat dari layar tidak memiliki function sama sekali, function akan ditulis dalam halaman lain yaitu "CustomerController". Bagian "Header Table (<thead>)" berfungsi untuk membuat kolom yaitu "Nama Customer", "NIK", "Nama Ktp", "Limit Ktp" dan "Pengguna Limit". Pada bagian "Body Table ()" berfungsi untuk mengisi data pada tiap kolom yang dimana data akan diambil dari "Controller" dengan variabel "\$customer", dikarenakan data akan terus bertambah maka digunakan "@foreach" untuk "Looping data" sehingga data yang baru diinput akan langsung masuk ke dalam halaman "List Customer Ktp". Bagian kolom "Limit Ktp" menggunakan format angka "number_format()" yang berfungsi untuk mengambil desimal 2 digit, memisahkan desimal menggunakan koma dan memisahkan ribuan menggunakan titik.

3.2.4.3 Membuat Fitur

Tugas selanjutnya yang dikerjakan oleh mahasiswa magang adalah membuat fitur berupa *button "Excel"* yang berfungsi untuk *mengexport* laporan *finance* kedalam *format "Xlsx"*. Semua laporan terdapat dalam menu *"Report"* submenu *"Finance"*, berikut ini adalah gambar dari menu *"Report"*

E	Report	~	
	Purchasing	<	
	Sales	<	
	Finance	~	
	Laporan AP Card		
	Laporan AP Aging		
	Laporan AP Invoice List		
	Laporan AR Card		
	Laporan AR Aging		
	Laporan AR Invoice List		

Gambar 3.22 Menu Report - Finance

Mahasiswa magang diharuskan untuk membuat fitur *export excel* dalam semua laporan yang ada dalam "*Finance*". Gambar 3.22 akan memperlihatkan hasil *button excel* yang dibuat oleh mahasiswa magang



Gambar 3.23 Button excel

Gambar 3.23 merupakan halaman laporan "AP CARD REPORT", terlihat pada bagian kanan atas terdapat 3 *action button* yang memiliki fungsi yang berbeda beda. Mahasiswa magang mengerjakan *button* yang berwarna hijau yang berfungsi untuk *mendownload* laporan dalam format "*Xlsx*".



Gambar 3.24 kode button excel

Gambar diatas adalah kode yang membentuk *button excel*, *button excel* menggunakan "*Bootstrap*" yaitu "*class*="*btn btn-success*"" yang membuat button menjadi warna hijau. Untuk mengatur tata letak menggunakan margin top "mt-lg-3" sehingga dapat terletak dibagian kanan atas, pada bagian "*onclick*="*exportToExcel()*"" akan menjalankan function "*exportExcel()*" yang terletak pada *javascript* ketika diklik. Untuk mengakses *javascript* dapat dipanggil melalui "*id=btn-excel*". Pada "**" pertama berfungsi untuk menampilkan teks "*Excel*", pada "**" kedua berfungsi untuk menampilkan teks "Tunggu Sebentar" pada saat proses *export* berlangsung, kode "*<i class="spinner-border spinner-border-sm"> berfungsi untuk menampilkan spinner kecil dari bootstrap* yang menjadi animasi *loading*.

<pre>function exportToExcel() { \$('#btn-excel').prop('disabled', true); \$('#btn-excel-label').hide(); \$('#btn-excel-loading').show(); }</pre>	
<pre>const typeReport = 'ap_card'; const url = '{{ route(name: 'admin.ap_report.export') }}' + '?start_date={{ \$start_date }}&end_date={{ \$end_date }}&type=' + typeReport;</pre>	
<pre>window.location.href = url;</pre>	
<pre>setTimeout(() => { \$('#btn-excel').prop('disabled', false); \$('#btn-excel-label').show(); \$('#btn-excel-loading').hide(); }, 500);</pre>	
}	

Gambar 3.25. Function export excel

Gambar diatas merupakan kode dari *function exportexcel* yang berfungsi untuk *mendownload* laporan ke dalam format "*Xlsx*", function ini

44

yang akan berjalan apabila *button excel* di klik sehingga laporan dapat didownload ke dalam "*Xlsx*".

3.2.4.4 Mengubah tampilan

Tugas mahasiswa magang selanjutnya adalah mengubah tampilan yang terdapat dalam menu "*Master Data*" submenu "*Customer*". Dalam halaman "*Customer*" terdapat 3 aktivitas yang dapat dilakukan yaitu *Create*, *Edit* dan *Show*. Mahasiswa magang di tugaskan untuk menambahkan beberapa inputan yang akan diisi oleh klien

Buat Customer		Dashboard 📎 Daftar Customer 📎 Buat Custome
lsian dengan asterik (*) waiih diisi		Simpan Batal
Nama *	Name	
Lokasi *	🔘 Dalam Kota 🔹 Luar Kota	
No Telepon	No Telepon	
No NPWP	No NPWP	
Plafon	Plafon	\$
Alamat 1	Alamat 1	
		Å
Alamat 2	Alamat 2	
		h

Gambar 3.26 Tampilan inputan customer

Gambar 3.26 adalah tampilan akhir yang sudah dimodifikasi oleh mahasiswa magang. Bagian yang dikerjakan meliputi penambahan fitur "Lokasi" yang dibuat menggunakan "*Radio Button*" dengan 2 pilihan yaitu "Dalam Kota" dan "Luar kota". Selain itu, mahasiswa magang juga membuat bagian "Alamat 2" yang dibentuk menggunakan "*Text area*" agar dapat memuat banyak "*String*".



Gambar 3.27 kode radio button lokasi

kode diatas berfungsi untuk membuat "*Radio button*" pada menu lokasi dengan dua pilihan yaitu "Dalam kota" dan "Luar kota". "*Label*" berfungsi untuk menampilkan "Lokasi" lalu pada bagian "*<div class="formcheck form-check-inline">*" berfungsi membuat *radio button* dengan tampilan horizontal, lalu pada bagian "*<input class="form-check-input" type="radio" name="location" id="dalam_kota" value="dalam_kota"*

{{ old('location') == 'dalam_kota' ? 'checked' : " }}>" berfungsi mengirimkan nilai yang dipilih ke dalam kolom *database* (*location*) dan menjaga *radio button* yang terpilih tetap tercentang setelah melakukan *submit* di bagian *edit* dan *show*.



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara



Gambar 3.28 Radio button pilihan bank

Gambar 3.28 merupakan tampilan "*Radio button*" yang berfungsi untuk memilih *bank* pembayaran beserta nomor rekening yang sudah tertera. Setiap opsi *radio button* memberikan informasi terkait nama *bank*, nomor rekening serta nama penerima dana secara jelas. Konsep pembuatan *radio button* ini hampir sama dengan pembuatan fitur "Lokasi" letak perbedaan hanya pada isi label saja.

Daftar KTP 🔳	
Nama	Name
No NIK KTP	No NIK KTP
Limit KTP	Limit KTP
Image	Choose File No file chosen

Gambar 3.29 Tampilan Daftar KTP

Gambar 3.29 adalah tampilan akhir dari "Daftar KTP" yang telah dimodifikasi oleh mahasiswa magang. 1 *customer* dapat menginput beberapa daftar ktp, tombol *action button* berwarna hijau berfungsi untuk menambah 47

daftar ktp. Mahasiswa magang ditugaskan untuk menambahkan bagian "Limit KTP" yang nantinya ketika sudah di submit data akan masuk kedalam *database* yang digunakan dalam proyek ini.



Gambar 3.30 kode membuat Limit KTP

Gambar 3.30 merupakan kode yang menghasilkan fitur "Limit KTP" dalam halaman "Daftar KTP". Fitur "Limit KTP" dibuat secara horizontal menggunakan kode "<div class="row">", pada bagian "<label class="colsm-2 col-form-label" for="limit_ktp">Limit KTP</label>" berfungsi memunculkan teks "Limit KTP" dalam kotak yang digunakan untuk memasukan nominal dan menggunakan "For" agar inputan dengan "id=limit_ktp" dapat aktif ketika diklik. Pada bagian <input> berfungsi agar input bertipe "text", atribut v-model="ktp.limit_ktp" menghubungkan antara input ini dengan data yang terdapat dalam "<u>Vue.js</u>" secara dua arah sehingga ketika mengetik sesuatu nilai data data Vue akan langsung berubah dan sebaliknya. Lalu "limit_ktps" berfungsi untuk menampung inputan yang lebih dari satu, lalu pada kelas "form-Control" berfungsi agar tampilan dalam input menjadi lebih rapi dan memenuhi lebar kolom secara otomatis. Menggunakan "maxLength=16" berfungsi untuk membatasi jumlah karakter yang diinput.

۵	Sales	~
	Sales Order	
	Delivery Order	
	Sales Invoice	
	Sales Return	

Gambar 3.31 Memperbaiki bagian print dalam semua sub menu sales Menu sales memiliki empat sub menu yaitu "Sales Order", "Delivery Order", "Sales Invoice" dan "Sales Return". Masing masing menu memiliki report yang dapat di print dalam format "PDF" yang sudah dapat berjalan dengan baik, namun klien memiliki beberapa revisi yang harus diperbaiki pada bagian isi report. Revisi dikirim menggunakan "Zip Folder" yang berisi gambar yang report yang sudah di print dalam bentuk fisik dan melingkari bagian yang perlu untuk direvisi.



49 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Gambar 3.32 Revisi report Sales order

Gambar 3.32 merupakan hasil revisi pada bagian "Sales Order". Klien ingin mengubah format tanggal dari "02-jan-2025" menjadi "2 January 2025". Gambar dibawah akan menjelaskan perubahan dalam kode



Gambar 3.33 Revisi format tanggal Sales order

Gambar 3.33 merupakan kode akhir yang sudah memenuhi keinginan klien. Letak perubahan terdapat dalam format yang awalnya menggunakan "format('d-M-Y)" menjadi "format('j F Y')".

Pengiri	nan pesanan	Pengiriman Pesanan							
: DO 2025 04 00011		ID	:	DO.2025.04.00011	Kepada Yth.,				
: 25-Apr-2025	Kepada Yth.,	Tgl. Transaksi	:	25 April 2025	GUDANG				
: Gudang F9A	GUDANG	Gudang	:	Gudang F9A	Jl. Jembatan Tiga Pluit Blok				
: BLA	Jakarta-Utara	Kategori	:	BLA	Jakarta-Utara				
DT DEDUAT VERM	Telp. 66696477	Deskripsi	:		Telp. 66696477				
 PT. DERNAI KERJA SELARAS, IAKAPTA 		Customer	:	PT. BERKAT KERJA					
N OU OWN				SELARAS, JAKARTA					
	Pengiri DO 2025.04.00011 25-Apr-2025/ Gudang FeA BLA PT. BERKAT KERJA SELARAS, JAKARTA	Kepada Yeh, 2020250400011 Kepada Yeh, 25-Apr-2025 GUDANG 2 Gudang FgA J. Jembatan Tiga Pluit Blok 3 BLA Jakarta-Utara 1 Derkat KERIA SELARAS, Telp. 66696477 2 PT. BERKAT KERIA SELARAS, JAKARTA	Pengiriman pesanan ID 1 D0 2025 04.00011 Kepada Yth, Tgl. Transaksi 2 Gudang Bak GUDANG Gudang 2 BLA Jakarta-Utara Gudang 3 H. Jembatan Tiga Pluit Blok Kategori 3 Jakarta-Utara Deskripsi 4 Telp. 66696477 Deskripsi 4 JAKARTA Customer	Pengiriman pesanan ID : 1 D0 2025 04.00011 Kepada Yth., Tgl. Transaksi : 2 Gudang Bak GUDMNG Gudang : : 2 Gudang Bak J. Jembatan Tiga Pluit Blok Gudang : 3 BLA Jakarta-Utara Deskripsi : 1 PT. BERKAT KERIA SELARAS. Telp. 66696477 Customer :	Pengiriman pesanan D0.2025.04.00011 Kapada Yuh, Gudang FaA BLA Jakarta-Utara Telp. 66696477 PT. BERKAT KERIA SELARAS: JAKARTA				

Gambar 3.34 Revisi format tanggal Delivery Order Print

Gambar 3.34 merupakan revisi dan hasil revisi yang diinginkan oleh klien pada bagian "Delivery Order Print". Klien ingin format tanggal sama seperti pada bagian "Sales Order" dan menggantikan huruf "p" kecil menjadi besar. Perubahan dalam kode tentu sama seperti pada "Sales Order" dengan mengganti format menjadi "format('j F Y')" dan mengganti huruf secara manual.

GUDANG 13 Januara Tiga Pisiti Baok e 13 April 1990 Nature 1990 Tellar Recenser?	SURAT PENGANT	AR	GUDANG JI.Jembatan Tiga Pluit Blok Jakarta Utara Telp. 66696477	F				
vor. DO.2023.04.00011 derwordt Pesanan Tuan Bersama ini kami romkan dengah Truck p. 821566NM		Secure VII. PT. BERKAT KERJA SELARAS, JAKANYA.	No. DO.2025.04.00011	SURAT PENGAN	Kepada Yth.			
Jumlah 625	Jenis Barang / Merk CN-BLA-01 - KC KURA CR44 2529	Ketarangan	Menurut Pesanan Tuan Bersa ini kami kirimkan dengan True No. B2156GNM	No. DO.2025.04.00011 Menurut Pesanan Tuan Bersama Ini kami kirimkan dengan Truck No. B2156GNM				
	\bigcirc	Jakana 26-Apr-2025	Jumlah	Jenis Barang / Merk	Keterangan			
Tanda Langan Penerima	Penimbang	Kepala Gudang	625	CN-BLA-01 - KC KUPAS CINA 2529				
					Jakarta, 8 May 2025			
			Tanda Tangan Penerima	Penimbang	Kepala Gudang			
tun Dire	Namajelas	Nama@las						
Ivanie Jeds			Nama Jelas	Nama Jelas	Nama Jelas			

Gambar 3.35 Revisi format tanggal Delivery Order Print surat jalan Gambar 3.35 merupakan bagian "*Delivery Order print* surat jalan" permasalahan sama seperti bagian bagian sebelumnya yaitu format tanggal dan huruf kecil. Perubahan yang dilakukan hanya mengubah format tanggal dan mengubah huruf kecil menjadi besar secara manual.

SL DU TRA NUSA	CV PUTRA NUSA Ekspor-Impor Hasiti J. KH. Wahid Hasiyin No. 70, 7 Thp	Pertanian Menteng, Jakarta Pusat 33.2 3028 / 024	- 2109/142						Stint Back
To: PT. BERKAT KERJA SELARAS, JAKARTA JI. Telepon no.4, Jakarta. Belakang Pos Hansip TH. Pengan PAX. Derga Homat.	TAGIHAN 51.2025.04.00007	Date 5 SO No. 5 Cust. Code 5	A April 2005) SC PARS SAN OPPORT	CV PUTRA I	NUSA	CV PUTRA NU Ekspor-Impoi JI. KH. Wahid Hasyi TLP : 021-2992-302	JSA r Hasil Pe m No. 70, Me 8 / 021-21392	r taniar nteng, Ja 262	1 karta Pusat
Melan groundwar naghtari Kan teil melan bargatari bargatari bargatari tean an	espada Sandra SBE DECORTON Face For an analysis of the second	QUANTITY ess to we'r conglos anglut serta mgymnun uang paling	0011 7452 (400 12 2 2 400) 70744 (40) 17 107 (500)	To: PT. BERKAT KE SELARAS, JAKJ JI. Telepon no. Jakarta. Belaka Hansip (0816 184 5120 Dengan Hormat, Melalui perusahaan	RJA ARTA 4, ang Pos D)	TAGIHAN \$1.2025.04.000	Date SO No. Cust. Co	: 25 : SO ode: CU	April 2025 2025.01.00014 \$0051
-RECEIVED BY	Yours . Yours Aithfully,	lours Sincerelu N. Putra Musa	1 / Ol Mathemaca -	DO NUMBER	mkan barang-ba	rang kepada Saudara SBE ESCRIPTION	QUANTITY	UNIT PRICE/Kg	AMOUNT
PT benunt verys S	laus.			DO.2025.04.00011	KC KUPAS CINA	2529	625 KG	27,500.00	17,187,500.00
				DESCRIPTION :	Tuiub Belar I	uta Seratus Delanan		TOTAL IDR	17,187,500.00
ENTING / IMPORTANT :				TENDEDING .	Puluh Tujuh F	Ribu Lima Ratus			

Demikianlah agar dapat diterima dengan baik dan dapat menyelesaikan pembayaran ongkos angkut serta kuli muat barang dengan pihak perusahaan angkutan. Mengenai pembayaran barang-barang tersebut di atas. kiranya dapat saudara atur pengiriman uang paling lambat 09 May 2025

PT. BERKAT KERJA SELARAS, JAKARTA Yours Sincerely Cv.Putra Nusa /Cv Baturasa

PENTING / IMPORTANT :

Gambar 3.36 Revisi Sales Invoice Print

Gambar 3.36 merupakan bagian "Sales Invoice" yang juga terdapat beberapa bagian yang harus direvisi dan penambahan sesuai dengan keinginan klien. Klien ingin ada penambahan informasi nomor telepon pada bagian "Header", penambahan nomor telepon pelanggan yang diambil dari database (customer->phone), penambahan (/Kg) pada tabel "Unit Price", mengganti "Received By" menjadi nama pelanggan yang ada di dalam database dan mengganti "YOUR FAITHFULLY" menjadi "Yours Sincerely Cv.Putra nusa / Cv Baturasa".

3.2.4.5 Modifikasi database

Mahasiswa magang diperbolehkan untuk memodifikasi *Database* apabila dibutuhkan, namun hal ini tentu harus dikonfirmasikan terlebih dahulu dengan "*Project Manager*". Mahasiswa magang menambahkan beberapa *column* di beberapa *table* dalam *database* yang digunakan dalam proyek ini

address2 province district city location zip_code top status_id plafon payment_information

Gambar 3.37 Column yang ditambah dalam table Customer

Pada *table "Customer*" mahasiswa magang menambahkan tiga *column* yaitu "*address2*", "*location*" dan "*payment_information*". Masing masing column memiliki fungsi untuk menampung *value* yang berbeda beda, *column "address2*" berfungsi untuk menampung inputan dari "Alamat 2" pada bagian "Buat customer", *column "location*" berfungsi untuk menampung inputan dari "Lokasi" dalam bentuk "*Radio button*" yang

terdapat dalam bagian "Buat *Customer*" dan *column "payment_information*" berfungsi untuk menampung hasil inputan dari *radio button* yang berisikan nama nama *bank* beserta dengan rekening yang sudah tertera.

limit_ktp

Gambar 3.38 Column yang ditambah dalam table customer_ktps

Pada *table "customer_ktps*" mahasiswa magang menambahkan satu *column* yaitu "*limit_ktps*" yang berfungsi untuk menampung inputan dari "Limit KTP" pada bagian "Daftar KTP".

3.2.4.6 Membuat rumus perhitungan

Pekerjaan selanjutnya yang dikerjakan mahasiswa magang pada proyek ini adalah membuat rumus perhitungan pada bagian "*Purchase Order*".

Buat Purchase Order					0	Dashboard	> Daftar Purchase	Order > Buat F	Purchase Or
Islan dengan asterik (*) wajib diisil Referensi Purchase Request *	PR.2025.05.0001	5				¥	Ambil Data	Simpan	Batal
Kode	AUTO								
Vendor	- Pilih Supplier -								
Currency	RP - Rupiah					*			
Currency Rate	RP 1,00								
Nomor Quotation (opsional)	Nomor Quotatio	m							
Tanggal Dibutuhkan	13-05-2025								
Metode Pengiriman	Metode Pengirir	nan							
Catatan	Catatan								
Paftar Barang									
NAMA BARANG	KATEGORI	UNIT	QTY PR	QTY	HARGA		TOTAL	DISC	
AF-GNK-06 - KC TANAH AFRIKA TANZANIA @25KG 🔻	GNK	KG	1,00	0,00	RP	0,00	RP	RP	0,00
AF-GNK-06 - KC TANAH AFRIKA TANZANIA @25KG 🔻	GNK	KG	10,00	0,00	RP	0,00	RP	RP	0,00
AF-GNK-01 - KC TANAH AFRIKA SUDAN	GNK	KG	2,00	0,00	RP	0,00	RP	RP	0.00

53 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Jangka Waktu Pembayaran	0							Hari
Subtotal	RP 0),00						
Diskon	0,00		%	RP	0,00			
Subtotal 2	RP 0),00						
Subtotal 2 (dalam Rupiah)	Rp 0),00						
Down Payment (DP)	0,00		%	RP	0,00			
Asuransi	RP 0),00				Rp	0,00	

Gambar 3.39 Tampilan Purchase Order

"Purchase order" berkaitan dengan "Purchase Request" apabila tidak ada "PR" maka "PO" tidak akan bisa dilanjutkan. Mahasiswa magang ditugaskan untuk membuat perhitungan untuk daftar barang dalam kurs rupiah maupun dollar. Klien akan mengisi dari bagian paling pertama yaitu "Referensi Purchase Request" yang lalu menentukan vendor serta "Currency" dengan pilihan "Rp" atau "\$", jika klien memilih "\$" maka klien harus mengisi bagian "Currency Rate" sesuai dengan harga yang beredar jika memilih "Rp" maka tidak perlu untuk mengisi bagian "Currency Rate". Selanjutnya klien perlu untuk menginput jumlah "QTY" dari barang yang sudah dipilih pada bagian terakhir, klien hanya perlu untuk menginput pada bagian "jangka waktu pembayaran", "diskon" dan "asuransi" selebihnya akan terisi otomatis dari rumus perhitungan harga yang dibuat oleh mahasiswa magang.

```
discPercentEdit() {
    if (this.disc1 > 0) {
        this.disc1Value = (this.subTotal * this.disc1) / 100;
    } else {
        this.disc1Value = 0;
    }
    if (this.disc2 > 0) {
        this.disc2Value = ((this.subTotal - this.disc1Value) * this.disc2) / 100;
    } else {
        this.disc2Value = 0;
    }
    this.disc2Value = 0;
    }
    this.grandTotal = this.subTotal - this.disc1Value - this.disc2Value + this.ppnValue;
}.
```

Gambar 3.40 Rumus perhitungan discount percent

Fungsi "discPercentEdit()" berfungsi untuk menghitung nilai diskon dan total keseluruhan berdasarkan persentase diskon yang diberikan. Fungsi pertama digunakan untuk memeriksa apakah nilai "dics1" lebih dari nol, jika iya maka nilai "dics1Value" akan dihitung dengan mengalikan "subTotal" dengan persentase "disc1" kemudian dibagi 100. Jika tidak ada diskon pertama, maka nilai "disc1Value" akan diatur menjadi 0 dan "disc2value" dihitung dari sisa "subTotal" setelah dikurangi "disc1Value" kemudian dikalikan dengan persentase "disc2" dan dibagi 100. Jika tidak ada diskon kedua, maka "disc2Value" diatur menjadi nol setelah kedua nilai diskon dihitung. Fungsi ini akan menghitung "grandTotal" yaitu total akhir yang harus dibayar dengan mengurangi "subTotal" dengan "disc1Value" dan "disc2Value", lalu ditambah dengan "ppnValue".





Gambar 3.41 Rumus perhitungan keseluruhan

Fungsi "calculatePrice" berfungsi untuk menghitung total harga dari "Daftar item" yang ada termasuk diskon dan PPN. Fungsi ini akan memeriksa apakah terdapat "item" dalam "itemList" jika ada maka "subTotal" akan diinisialisasi ke nol agar tidak menjumlah nilai sebelumnya. Kemudian, nilai dari "dropdown" dengan "ID ppn_type" diambil untuk mengetahui harga "item" sudah termasuk "ppn (include)", "ppn(exclude)" atau tidak menggunakan ppn. Selanjutnya, fungsi melakukan perulangan terhadap setiap "item" yang ada dalam "itemList" serta menghitung "subtotal" awal per "item" berdasarkan jumlah (QTY), jika "PPN" bertipe "include" maka subtotal akan disesuaikan agar nilai "PPN" dipisahkan dari harga total.

Setelah itu, fungsi akan mengurangi dua jenis diskon per "*item*" (disclValue dan disc2Value) dari "subtotal" yang hasilnya akan disimpan ke dalam "subtotal" masing masing "*item*". Total dari semua "*item*" akan ditambahkan ke "subTotal" keseluruhan. Jika jenis "PPN" adalah "none" maka persentase "PPN"di set menjadi nol. Setelah itu, fungsi akan menghitung "grandTotal" dengan mengurangi dua jenis diskon dari "subTotal", setelah itu nilai "PPN" dihitung dari "Grandtotal" berdasarkan persentase "PPN" yang hasilnya akan dibulatkan menggunakan "Math.floor".

Terakhir, nilai dari "PPN" ditambahkan kembali ke dalam "grandTotal" untuk mendapatkan nilai total akhir yang akan ditampilkan.

3.2.5 Karada

3.2.5.1 Import Database

Sebelum Mengerjakan mengerjakan proyek , mahasiswa magang harus memiliki *database* dari proyek yang akan dikerjakan. Maka dari itu, langkah awal dari pengerjaan proyek adalah melakukan "*Import Database*" ke dalam laptop pribadi mahasiswa magang. *Database* dikirim melalui "*Whatsapp*" dalam bentuk "SQL" *file* oleh "*Project Manager*" yang nantinya akan di import ke dalam "*phpMyAdmin*" mahasiswa magang.

karada	supplier_invoice_headers	窗	Browse	M Structure	R Search	are insert	🛒 Empty	😂 Drop	0 InnoDB	utt8mb4_general_ci	64.0 K18	
Type to litter these, Entr	temp_bookings	會	Browse	K Structure	A Search	Be Insert	👷 Empty	Drop	0 InnoDB	utf8mb4_general_ci	16.0 KiB	
1 🕶 >>>	temp_invoices	會	Browse	M Structure	Rearch	3 insert	层 Empty	Drop	10 InnoDB	utf8mb4_general_ci	16.0 KiB	-
- R New	transaction_booking_slots	*	Browse	€ Structure	.e Search	3-i Insert	层 Empty	😂 Drop	0 InnoDB	utf8mb4_general_ci	64.0 KiB	
- account_receivables	transaction_details	龠	Browse	K Structure	Rearch	3-i Insert	🚍 Empty	Drop	78 InnoDB	utf8mb4_general_ci	80.0 KiB	5
account_receivable_detai	transaction_detail_partners	*	Browse	K Structure	& Search	3 insert	🚍 Empty	Drop	46 InnoDB	utf8mb4_general_ci	48.0 KiB	
admin_users	transaction_headers	*	Browse	M Structure	Rearch	Se Insert	Empty	Drop	46 InnoDB	utf8mb4_general_ci	144.0 KiB	
app_birthday_coupons	transaction_invoices	合	Browse	F Structure	& Search	a insert	HE Empty	Drop	34 InnoDB	utf8mb4_general_ci	32.0 KiB	
- app_birthday_promo_usa	transaction_payments	*	Browse	Structure	Rearch	3-é Insert	Empty	Drop	34 InnoDB	utf8mb4_general_ci	64.0 KiB	
app_memberships	unit_materials	會	Browse	€ Structure	. Search	3 insert	Empty	Drop	0 InnoDB	utf8mb4_general_ci	48.0 KiB	
- app_membership_benefit	users .	會	Browse	M Structure	Rearch	3 insert	B Empty	Drop	8 InnoDB	utf8mb3_unicode_ci	64.0 KiB	
app_membership_bonus	user_addresses	命	Browse	Structure	* Search	Se Insert	Empty	Drop	0 InnoDB	utf8mb4_general_ci	32.0 KiB	
- kapp_membership_details	user_app_spending_messages	*	Browse	M Structure	Rearch	3-é Insert	Empty	Drop	0 InnoDB	utf8mb4_unicode_ci	16.0 KiB	14
app_membership_directs	user_membership_histories	-	Browse	> Structure	& Search	3-é Insert	Empty	C Drop	0 InnoDB	utf8mb4_unicode_ci	16.0 KiB	
- app_payment_types	user_notifications	*	Browse	M Structure	Rearch	- insert	Empty	C Drop	0 InnoDB	utf8mb4_unicode_ci	32.0 KiB	1
an detail?s	user_point_histories	-	Browse	M Structure	Search	3-i Insert	Empty	Drop	10 InnoDB	utf8mb4_unicode_ci	32.0 KiB	
- r ap headers	user_redeems	*	Browse	M Structure	Search	S-c Insert	Empty	Drop	e InnoDB	utf8mb4 general ci	48.0 KiB	
ar_detail1s	user reviews	4	Browse	Structure	. Search	S-é Insert	Empty	Drop	Ø InnoDB	utf8mb4 unicode ci	48.0 KiB	
- 🔀 ar_detail2s	user review images	*	Browse	Structure	· Search	Sé Insert	Empty	C Drop	0 InnoDB	utf8mb4 unicode ci	32.0 KiB	
- x ar_headers	user vouchers	-	Browse	Te Structure	. Search	Sé Insert	B Empty	O Drop	12 InnoDB	utf8mb4 general ci	48.0 KiB	
- mattendance_configuration	user youcher codes	-	Browse	Tel Structure	Search	Se Insert	Empty	C Drop	25 InnoDB	utf8mb4_general_ci	32.0 KiB	-
Attendance_shifts	voucher promo codes	-	Browse	Te Structure	· Search	Re Insert	Empty Empty	O Drop	1 InnoDB	utf8mb4_general_ci	80.0 KiB	
authorization_modules	warehouses	14 	Browse	Le Structure	Search	Se Insert	Empty Empty	Drop	0 InnoDB	utf8mb3_unicode_ci	48.0 KiB	
- Autonumbers	167 tables	Sum	E 0.04/30	are chactare	- source	ar moon	THE CHIPG	- Stop	6,869 InnoDB	utf8mb4 0900 ai ci	9.0 MiB	0 8

Gambar 3.42 Database Karada

Gambar 3.42 merupakan struktur dari *database* yang digunakan proyek "Karada", Terdiri dari 167 tabel yang saling berhubungan. Setiap tabel memiliki relasi yang dibuat untuk mendukung integrasi data serta kelancaran proses bisnis dalam sistem yang sedang dikembangkan.

3.2.5.2 Reporting

Mahasiswa magang ditugaskan untuk membuat halaman "Room Set Available". Halaman ini akan berbentuk tabel atau "report" berisikan tentang kamar mana saja yang siap dipakai. Report memiliki 4 kolom yaitu "Branch
Name", "Admin User", "Number Room", "Created At". Halaman ini berkaitan dengan menu "Bed" yang terdapat dalam "POS", di dalam "Bed" terdapat beberapa "room" yang bisa dibooking apabila ber indikator hijau (Available), ketika room sudah selesai maka indikator akan berubah menjadi warna pink (dibersihkan) yang dimana itu sudah dapat dibooking lagi dan laporan nya akan masuk kedalam halaman "Room Set Available".



Gambar 3.43. Membuat Route

Langkah pertama membuat halaman "room_set_available" adalah membuat "Route" yang akan merujuk pada halaman "room_set_available". "Route 'index'" berfungsi untuk menampilkan halaman utama dari fitur "room_set_available" dan ""Route 'datatable'" berfungsi untuk mengambil data dalam format "datatable", "route" ini nantinya akan dipanggil menggunakan "JavaScript" pada "blade room_set_available".



Gambar 3.44. Menampilkan menu Room Set Available dalam Navigation Gambar 3.44 berfungsi untuk menampilkan menu "*Room Set Available*" yang dapat diklik dan merujuk ke "*Route*" "*room_set_available*" di dalam bagian "*Report*".



Gambar 3.45. Membuat halaman Room Set Available

Gambar 3.45 merupakan halaman yang menampilkan tabel pada halaman "room_set_available". Halaman ini memiliki empat kolom yaitu "Branch Name", "Admin User", "Number Room" dan "Create At" yang dibuat menggunakan ">". Pada halaman ini juga nanti akan memanggil "datatable" menggunakan "JavaScript" dan "Json" untuk mengisi nilai dalam kolom.



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara



Gambar 3.46 Membuat halaman Room Set Available

Gambar 3.46 adalah kode yang berfungsi untuk mengambil nilai dalam "*datatable*" yang nantinya akan diisi dalam kolom yang sudah dibuat. Untuk menjalankan "*datatable*" membutuhkan beberapa *file "JavaScript*" seperti "*jQuery*" (untuk menjalankan *datatable*), "*plugin Datatables*", "<u>moment.js</u>" (untuk mengatur format tanggal) dan "*plugin notifikasi*". Saat halaman "*room_set_available*" dibuka maka fungsi pertama yang akan dijalankan adalah "*\$(document).ready()*", didalamnya terdapat tabel dengan id "*room-set-available-table*" yang akan diubah menjadi "*datatable*". Data untuk tabel tidak dapat diakses secara langsung harus diminta melalui "*route*" "*admin.log.room_set_available.datatable*". Setelah data berhasil didapatkan, maka data akan ditampilkan dan diletakan sesuai dengan kolom kolom yang sudah dibuat.

	KA·RA·DA カラダファクトリー		Room Set Available					Dashboard >	Room Set Available
TA	MAN		Show 10 🗸 entries					Search:	
A	NGGREK		BRANCH NAME	^	ADMIN USER \$	NUMBER ROOM	¢	REATED AT	\$
8	Dashboard		KARADA Taman Anggrek			A101	1	5-05-2025 09:30	
6	Report	~	KARADA Taman Anggrek			B101	1	2-03-2025 16:34	
		<	KARADA Taman Anggrek			D113	1	2-03-2025 16:41	
	Room Set Avaiable		KARADA Taman Anggrek			A002	1	2-03-2025 23:42	
		<	KARADA Taman Anggrek			D113	1	2-03-2025 23:45	
		<	KARADA Taman Anggrek		Superadmin	A002	1	2-03-2025 16:48	
		<	KARADA Taman Anggrek		Superadmin	B101	1	2-03-2025 16:50	
1		< .	KARADA Taman Anggrek		Superadmin	B101	1	2-03-2025 17:12	
0		<	Showing 1 to 8 of 8 entries					Pn	evious 1 Next
		<	4						

Gambar 3.47. Tampilan Room Set Available

3.2.5.3 Membuat halaman Detail Kode Voucher

Mahasiswa magang ditugaskan untuk menampilkan halaman yang dapat menampilkan "Kode Voucher" yang sudah dibuat oleh *admin*. Mahasiswa magang ditugaskan untuk membuat satu "action button" dengan tulisan "Show" yang ketika diklik akan masuk kedalam halaman "Detail Kode Voucher" yang berisi enam "Label" dengan judul "Pemilik", "Jenis Voucher", "Notes", "EXP Date", "Owner", "Quantity", "Status" dan "Tanggal Dibuat". Masing masing label nantinya akan memunculkan data yang akan diambil dari tabel "user_vouchers" yang berelasi dengan tabel "users".



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara



Gambar 3.48 Function show pada controller voucher

Gambar 3.48 adalah "*Controller*" yang digunakan untuk mengambil data data yang akan digunakan untuk menunjukan data pada halaman "Detail Kode Voucher". Terdapat delapan column yang diambil dari tabel "*user_voucher*" yang berelasi dengan "*user*", pada bagian akhir terdapat "return" yang artinya data akan dilempar ke "*route*" "*admin.voucher.show*".



62 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Gambar 3.49. Blade Detail Kode Voucher

Gambar 3.49 merupakan kode yang menampilkan tampilan pada halaman "Detail Kode Voucher", dalam kode ini terdapat delapan label yang tersusun empat di sebelah kanan dan sisanya di bagian kiri, setiap label akan memunculkan data yang berbeda beda dan hanya dapat dibaca tidak dapat di *edit*. Data dipanggil dengan kode "*value="{{ \$data['owner'] }}"*" pada setiap label namun mengganti variabel sesuai dengan judul label dan menggunakan "*readonly*" agar label hanya memunculkan data tidak dapat *diedit*. Gambar 3.50 merupakan hasil dari kode pada gambar 3.49

KA RA DA	Detail Kode Voucher	Dashboard > Detail Kode Voucher
TAMAN ANGGREK	Pemilik Band Ferdvante	Owner N/A
 Dashboard 	Jenis Voucher diskor20	Quantity 5
	Notes Pembelian/Hasil Redeem User	- Status Aktif
■ Report < ■ POS <	EXP Date 4	Tanggal Dibuat
Master Data ~	31 Mar 2025 20:28:06	01 Mar 2025 13:28:06
Jenis Voucher		
Kode Voucher		

Gambar 3.50 Tampilan halaman Detail Kode Voucher

3.2.6 Vigor

3.2.6.1 Import Database

Sebelum mengerjakan proyek , mahasiswa magang diwajibkan untuk melakukan import *database* kedalam "*Device*" pribadi agar dapat membuka *website*. Dalam pembuatan database proyek "*Vigor*" mahasiswa magang tidak ikut dalam perancangan *database* karena proyek sudah berjalan sebelum mahasiswa magang melaksanakan magang.

510 000	📝 Structure 🧾 SQL 🔍 Search 🗔 Que	y 🖶 E	xport 😸	Import 🥜	Operation	ai P	rivileges	Routine	s 🕑 Events	36 Triggers	4 Designer	
worites	transaction_checkout_headers	\$	Browse	K Structure	Search	34 Insert	💭 Empty	😄 Drop	40 InnoDB	utf8mb4_unicode_c	64.0 KiB	-
(= ee	transaction_details	會	Browse	M Structure	Rearch	3 i Insert	📻 Empty	Drop	574 InnoDB	utf8mb4_general_ci	208.0 KiB	-
	transaction_detail_samples	*	Browse	K Structure	& Search	3 i Insert	me Empty	Drop	0 InnoDB	utf8mb4_general_ci	96.0 KiB	-
filter thread, Ent. X	transaction_headers	会	Browse	M Structure	Rearch	3 i Insert	me Empty	😂 Drop	560 InnoDB	utf8mb4_general_ci	352.0 KiB	
1 🛩 >>>	transaction_histories		Browse	K Structure	& Search	3-i Insert	Empty	😂 Drop	@ InnoDB	utfSmb4_unicode_c	32.0 KiB	-
w	transaction_lab_narrations	会	Browse	M Structure	R Search	3 i Insert	me Empty	Orop	0 InnoDB	utf8mb4_general_ci	16.0 KiB	121
dress_cities	transaction_lab_results	倉	Browse	K Structure	Rearch	3-i Insert	Empty	Drop	@ InnoDB	utf8mb4_general_ci	16.0 KiB	-
Iress_countries	transaction_lab_result_details	会	Browse	M Structure	R Search	3-i Insert	me Empty	Drop	@ InnoDB	utf8mb4_general_ci	16.0 KiB	-
ress_districts	transaction_lab_thresholds	会	Browse	M Structure	. Search	3-i Insert	Empty	😂 Drop	@ InnoDB	utf8mb4_general_ci	16.0 KiB	-
iress_provinces iress_states	transaction_package_details	合	Browse	K Structure	Rearch	3-i Insert	Empty	Orop	1,569 InnoDB	latin1_swedish_ci	304.0 KiB	
logs	transaction_package_extra_items	会	Browse	K Structure	& Search	3-i Insert	me Empty	😂 Drop	6,880 InnoDB	utf8mb4_unicode_c	25.9 MiB	-
lio_items	transaction_package_narration_extra_items	合	Browse	M Structure	Search	3-i Insert	Empty	Orop	13,760 InnoDB	utf8mb4_general_ci	3.5 MiB	100
onumbers	transaction_payments	\$	Browse	M Structure	& Search	3+i Insert	Empty	Orop	7 InnoDB	utf8mb4_general_ci	64.0 KIB	-
3 8	transaction_shipping_details	*	Browse	K Structure	R Search	3-i Insert	m Empty	Orop	1 InnoDB	utf8mb4_unicode_c	32.0 KiB	-
_categories	transaction_shipping_items	숡	Browse	K Structure	- Search	3-i Insert	Empty	Orop	0 InnoDB	utf8mb4_general_ci	16.0 KiB	-
tag links	unit_materials	會	Browse	M Structure	R Search	3-i Insert	Empty	Orop	4 InnoDB	utf8mb4_general_ci	80.0 K18	-
nches	users .	合	Browse	K Structure	Search	3-i Insert	Emply	Orop	25 InnoDB	utf8mb4_unicode_c	64.0 KiB	-
adcasts	user_notifications	合	Browse	M Structure	R Search	3-i Insert	Empty	Orop	@ InnoDB	utf8mb4_general_ci	32.0 KiB	
imann_aspects	vehicles	会	Browse	M Structure	. Search	3 insert	Empty	😂 Drop	0 InnoDB	utf8mb4_general_ci	80.0 KiB	-
fmann_categories	vehicle_storages	余	Browse	M Structure	R Search	3-i Insert	me Empty	😂 Drop	@ InnoDB	utf8mb4_general_ci	112.0 KiB	-
imann_tests	video_items	合	Browse	K Structure	Search	34 Insert	🚍 Empty	😂 Drop	39 InnoDB	utf8mb4_general_ci	96.0 KiB	-
imann test details	vigor_options	会	Browse	M Structure	R Search	3-i Insert	m Empty	Orop	29 InnoDB	utf8mb4_general_ci	16.0 KiB	-
_details	vouchers	合	Browse	K Structure	Rearch	3-i Insert	📻 Empty	😂 Drop	12 InnoDB	utf8mb4_general_ci	96.0 KiB	-
_headers	voucher_cart_applies	會	Browse	M Structure	R Search	34 Insert	Empty	Orop	@ InnoDB	latin1_swedish_ci	64.0 KiB	-
5	voucher_usages	숣	Browse	M Structure	& Search	3-i Insert	m Empty	Orop	@ InnoDB	latin1_swedish_ci	48.0 KiB	-
igurations	voucher_usage_histories	合	Browse	K Structure	R Search	3-i Insert	Empty	Orop	@ InnoDB	latin1_swedish_ci	32.0 KiB	
orate_queues	warehouses	合	Browse	M Structure	& Search	3 i Insert	Empty	Drop	8 InnoDB	utf8mb4_general_ci	64.0 KiB	-
omers	152 tables	Sun	n			-		~	21,547 InnoDB	utf8mb4_0900_ai_0	ci 89.5 MiB	0 8
omer addresses	A D Charles II Mith anti-state											

Gambar 3.51 Struktur Database Vigor

Gambar 3.51 merupakan struktur *database* yang digunakan dalam proyek "Vigor" yang terdiri dari 152 tabel yang saling berhubungan. Setiap tabel memiliki relasi yang dirancang untuk mendukung integrasi data serta kelancaran proses bisnis dalam sistem yang sedang dikembangkan. Setelah melakukan "*Import Database*" mahasiswa magang ditugaskan untuk membuat sebuah fitur sesuai dengan permintaan klien.

3.2.6.2 Membuat Route

Mahasiswa magang ditugaskan untuk membuat fitur dengan nama "Brodman", fitur ini terdiri dari beberapa aksi yaitu CRUD (Create,Read,Update dan Delete) namun mahasiswa magang ditugaskan menyelesaikan sampai tahap "Read". Langkah pertama dari pembuatan fitur ini adalah membuat "Route" untuk masing masing aksi agar dapat diklik.

crpnp use App\Http\Controllers\BrodmanController; use Illuminate\Support\Facades\Route; use App\Http\Controllers\BranchController; Route::group(['prefix' => 'brodman', 'middleware' => ['auth']], function () { // Route untuk halaman index BrodMan Route::get('/', [BrodmanController::class, 'index'])->name('brodman.index'); // Route untuk halaman create BrodMan Route::get('/create', [BrodmanController::class, 'create'])->name('brodman.create'); Route::get('/datatable', [BrodmanController::class, 'store'])->name('brodman.store'); Route::get('/datatable', [BrodmanController::class, 'datatable'])->name('brodman.edit'); Route::get('/edit/{id}', [BrodmanController::class, 'edit'])->name('brodman.edit'); Route::get('/update', [BrodmanController::class, 'update'])->name('brodman.update'); Route::get('/show/{id}', [BrodmanController::class, 'show'])->name('brodman.show'); Route::get('/delete/{id}', [BrodmanController::class, 'delete'])->name('brodman.delete'); });

Gambar 3.52. Route fitur Brodman

Gambar 3.52 merupakan kode "*Route*" yang dikerjakan oleh mahasiswa magang untuk mengatur beberapa aksi terkait fitur "*Brodman*" seperti "*Create*, *Edit*, *Update*, *Show dan Delete*". Setiap "*Route*" dimasukan kedalam "grup" dengan "*prefix brodman*" dan "*middleware auth*" agar setiap pengguna yang sudah *login* dapat mengakses "*route*" ini. "*Route*" "*Create*, *Edit*, *Update*, *Show dan Delete*" masing masing akan merujuk ke halaman nya ketika diklik. "*Route Store*" akan merujuk pada "*function Store(*)" yang terletak dalam "*BrodmanController*" yang berfungsi untuk menyimpan data baru kedalam "*Database*". "*Route datatable*" berfungsi untuk mengambil serta menampilkan data dari "*database*" kedalam "*datatables*" yang akan ditampilkan di "*Frontend*".



Gambar 3.53 Fitur Brodman

Gambar 3.53 adalah kode yang membentuk tulisan "*Brodman*" di bagian "*Navigation*" yang dapat diklik. Ketika tulisan "*Brodman*" diklik maka akan langsung merujuk ke halaman "*Brodman.index*".

3.2.6.3 Membuat Halaman Index Brodman

Langkah selanjutnya setelah membuat "*Route*" dan menambahkan menu "*Brodman*" pada bagian "*Navigation*" adalah membuat tampilan halaman "Index Brodman". Halaman ini merupakan tampilan utama yang akan menampilkan seluruh data "*Brodman*" yang sudah terbuat dalam bentuk tabel dan halaman ini juga dapat melakukan aksi *CRUD* yang dapat di klik melalui "*Button*".

ኛ VNWT 💠				Superuser ~
බ Dashboard	Brodman			ඛ / Dashboard / Brodman
🕲 Menu Admin 🗸 🖌				
- Hierarki - Akses Menu	Daftar Brodman			+
Akun User				
🛞 Konfigurasi Konten App	10 🗸			Cari keyword
😫 Master Data 🛛 🗸 🖌				
- Cabang		ID	Montage	Created At
- Jabatan	Show Edit Delete	90		19 Mar 2025
Divisi	Show Edit Delete	8		19 Mar 2025
- Karyawan - Client		42		20 Mar 2025
- Client Relative	Edit Delete	15		ZU ININI ZUZO
Gudang	Show Edit Delete	99999		20 Mar 2025
+ Items ~	Show Edit Delete	1		20 Mar 2025

Gambar 3.54 Halaman Index Brodman

Gambar 3.54 adalah halaman "Index Brodman" terdapat tiga kolom yaitu "ID", "Montage" dan "Create At" kolom ini terbentuk ketika user membuat "Brodman" (Create) sehingga baris akan terus bertambah apabila user terus membuat "Brodman" baru. Halaman ini juga memiliki empat "Action Button" yaitu "Create" yang terletak di kanan atas dan "Show, Edit dan Delete" yang terletak di tiap baris.



Gambar 3.55. kode membentuk header dan tabel

kode pada gambar 3.55 adalah kode yang membentuk bagian "*Header*" dan membentuk tabel dengan empat kolom pertama untuk menampung 3 "Action *Button*", "*ID*", "*Montage*" dan "*Create At*".





Gambar 3.56 JavaScript dan Datatable

kode pada gambar 3.56 merupakan "JavaScript" yang berfungsi mengambil data untuk ditampilkan di dalam tabel. Pada bagian pertama, terdapat beberapa "script" yang bersifat penting seperti "Axios" (untuk komunikasi HTTP), "<u>Vue.js</u>"(untuk reactive interface), "JQuery DataTables" (untuk menampilkan tabel interaktif), "Flatpickr" (untuk mengatur tanggal) dan "<u>Moment.js</u>"(Untuk mengatur format tanggal). Setelah semua "library" yang penting sudah dimasukan, "Datatables" diinisialisasi pada elemen dengan "ID brodmanTable". Dilakukan pengaturan penting seperti "processing : true" dan "server side : true" untuk menunjukan bahwa data siap diproses di sisi server dan dikirim dalam format "JSON" dari "route brodman.datatable". Tabel ini juga hanya menampilkan 10 data per halaman yang diatur pada kode "pageLength : 10". Tampilan "Datatable" dibuat lebih "user friendly" dengan menambahkan tiga "Action button" tiap barisnya yang artinya tiap data dapat di edit, hapus dan lihat.

3.2.6.4 Create Brodman

Setelah membuat halaman "*Index*", mahasiswa magang bertugas untuk membuat halaman "*Create*" untuk menambah "*brodman*" baru.

					Batal Simpan			
Cabang *	VIGOR	Ŧ						
D*	ID							
lontage	Montage							
spek *								
Attention		Decision Making		Motivation				
Categorization		 Declarative Memor 	У	Procedural Memo	iry			
Sequential Memory		Event Sequencing		Spatial Sequence				
Facial Decoding								
Kategori	Delta	Theta	Alpha	Reta	Camma			

Gambar 3.57 Tampilan Create Brodman

Gambar 3.57 adalah tampilan ketika *user* mengklik *button* "+" yang terdapat dalam halaman "*Brodman Index*". Pada halaman ini terdapat "Cabang" yang dapat dipilih menggunakan "*Dropdown*", lalu terdapat "ID" yang dapat diisi di dalam "Label", terdapat "*Montage*" yang dapat diisi di dalam "Label", terdapat "*Montage*" yang dapat diisi di dalam "Label". Lalu terdapat "Aspek" yang terdapat banyak pilihan dan dapat memilih lebih dari satu "*Multiple*", "Aspek" dengan tabel yang terdapat dibawah aspek memiliki keterkaitan yang sangat kuat, ketika memilih salah satu "Aspek" maka kolom akan langsung terisi dengan kategori yang sudah dirancang dalam "*Database*"

Montage	Montage										
	montage										
Aspek *			Decision Maki	ng			Motivation				
Categorization			Declarative Me	emory	Procedural Memory						
Sequential Memory			Event Sequenc	ing .			Spatial Sequ	ience			
Eacial Decoding		_		-							
_ ruchin becoung											
Kategori	Delta		Theta		Alpha		Reta		Gamma		
Kategori	Delta		Theta		Alpha		Beta		Gamma		
Kategori Fz	Delta Up High		Theta Up		Alpha Regular		Beta Down		Gamma Down Low		
Kategori Fz Cz	Delta Up High None		Theta Up None		Alpha Regular None		Beta Down None		Gamma Down Low None		
Kategori Fz Cz P3	Delta Up High None None		Theta Up None None		Alpha Regular None None		Beta Down None None		Gamma Down Low None None		

69 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Gambar 3.58. Tampilan Tabel Aspek

Gambar 3.58 adalah tampilan ketika *user* mengklik salah satu "Aspek", tabel akan otomatis *mengenerate* kategori yang berbeda beda sesuai dengan "Aspek" yang dipilih. Setiap kategori memiliki empat pilihan yaitu "*Delta, Thera, Alpha, Beta* dan *Gamma*" masing masing pilihan dapat diisi dengan memilih "*Dropdown*" dan terdapat lima pilihan, masing masing pilihan ditandai dengan warna yang berbeda beda seperti pada gambar 3.58



Gambar 3.59 kode Cabang, ID dan Montage

Gambar 3.59 adalah kode yang membangun "Cabang", "ID" dan "" Montage. Pada bagian cabang dibangun menggunakan "dropdown", dalam "dropdown" memiliki beberapa pilihan yang datanya diambil dari "route('branch.select2')" dan "dropdown" ini menggunakan "required" yang artinya user harus memilih cabang. Bagian "ID dan Montage" dibuat menggunakan "label" dan dapat diisi oleh user.





Gambar 3.60 kode Checkbox

Gambar 3.60 adalah kode yang membentuk "*checkbox*" pada bagian "aspek", kode diatas memungkinkan *user* untuk memilih lebih dari 1 aspek. "*checkbox*" ini dibungkus dalam "*div class* = ""*bm-4>* " dan memiliki label "Aspek" dengan tanda bintang merah yang menandakan input ini bersifat wajib. Didalam "*div*" terdapat perulangan "*v-for*="*item in aspects*"" *dari* "*Vue,js*" yang berfungsi untuk secara dinamis menghasilkan "*checkbox*" berdasarkan data yang terdapat dalam "*array aspects*". Semua "*checkbox*" terikat ke dalam variabel "*pickedAspects*" melalui "*v-model*" sehingga setiap perubahan pilihan akan langsung masuk ke dalam "*data Vue*", maka dari itu *user* dapat bisa memilih beberapa aspek dan ata yang dipilih akan tersimpan secara otomatis dalam "*array pickedAspects*".



Nusantara

<div class="mb-4"></div>
<label class="form-label"></label>
<div class="table-responsive"></div>
<pre><thead></thead></pre>
<pre></pre>
Kategori Kolom 1: Lebar 20%
Delta Kolom 2: Lebar 16%
Theta Kolom 3: Lebar 16%
Alpha Kolom 4: Lebar 16%
Beta Kolom 5: Lebar 16%
Gamma Kolom 6: Lebar 16%
@{{ item.name }} Kolom 1: Kosong, akan diisi dari database
<
<input :name="'details['+index+'][id]'" type="hidden" v-model="item.id"/>
<pre><div class="d-flex align-items-center"></div></pre>
<pre><select :name="'details['+index+'][delta]'" <="" class="form-control dropdown-state" pre=""></select></pre>
style="width: 80%;" v-model="item.delta">
<pre><option data-color="red" value="0">None</option></pre>
<pre><option data-color="red" value="1">Up High</option></pre>
<option data-color="yellow" value="2">Up</option>
<pre><option data-color="green" value="3">Regular</option></pre>
<pre><option data-color="blue" value="4">Down</option></pre>
<pre><option data-color="darkblue" value="5">Down Low</option></pre>
<pre></pre> <
(dition class= color-indicator ms-2 :data-color= getcolor(item.oeita) > Kotak warna di</td

Gambar 3.61 kode tabel dan drop down warna

Pada bagian pertama terdapat kode yang membentuk tabel "Kategori, Delta, Theta, Alpha, Delta dan Gamma", tampilan tabel ini dibuat scroll secara horizontal jika melebihi layar dengan kode "<div class = "tableresponsive">", lebar tiap kolom diatur dalam "style =width". Pada bagian kedua merupakan kode yang membangun "dropdown" pilihan yang terdapat dalam tabel ketika "Aspek" dipilih seperti digambar 3.3.51. Pada kolom pertama "", nilai "item.name" ditampilkan menggunakan binding "Vue (@{{item.name}})" yang akan diisi dengan nama kategori dari "Database". Kolom kedua berisi input yang tersembunyi "(<input type = "hidden")" untuk menyimpan "id". Terdapat "< div class="d-flex align-items-center">" berfungsi untuk menyusun "dropdown (select box)" dan indikator warna secara "horizontal", terdapat enam kondisi dan masing masing memiliki indikator warna sendiri diantaranya "None (transparan)", "Up High (merah)", "Up (kuning)", "Regular (hijau)", "Down (biru terang)", "Down Low (biru gelap)". Nilai "dropdown" diikat dengan "v-model="item.delta"" berfungsi pada saat user memilih opsi, nilai akan langsung mengubah "item.delta", di sebelah kanan "dropdown" terdapat elemen "" dengan "class : color-indicator" yang menampilkan kotak kecil indikator warna. Warna ditentukan oleh hasil *"getColor(item.delta)"* yang berfungsi untuk mengembalikan warna berdasarkan nilai "Kategori" yang dipilih untuk memberikan indikator visual kepada *user* terhadap kondisi tersebut.

3.2.6.5 Brodman Controller

"Brodman Controller" dalam fitur ini merupakan halaman yang menjalan fungsi *"CRUD"*,tanpa adanya halaman ini fitur tidak dapat menjalankan fungsinya dan hanya berupa tampilan saja.



Gambar 3.62. Brodman Controller

Gambar 3.62 merupakan "Controller" yang digunakan dalam fitur "Brodman", terdapat tujuh function yang digunakan dalam fitur ini create(), diantaranya "Function *index()*, store(Request *\$request)*, datatable(Request \$request), show(\$id), edit(\$id), update(Request \$request, \$id), destroy(\$id)" dengan fungsinya masing masing. "Function index()" hanya berfungsi untuk menampilkan halaman "index brodman" dengan route "admin.brodman.index". "Function create()" berfungsi untuk menampilkan tampilan untuk membuat "brodman" baru, terdapat beberapa aspek di dalam "Function" ini seperti "Brodman aspects" untuk menampilkan semua "Aspek" dan "Brodman categories" yang hanya mengambil kategori dengan "status id =1". "Function store (Request \$request)" berfungsi untuk menyimpan data baru dari "form brodman" ke dalam "database", "Function datatable (Request \$request)" berfungsi untuk menampilkan data pada halaman "index brodman". "Function "show(\$id)" digunakan untuk

menampilkan detail pada saat menekan "action button show", "Function edit(\$id)" berfungsi agar "brodman" yang sudah dibuat dapat diedit ketika klien mengklik "action button edit" data dapat di edit berdasarkan "ID", "Function update (Request \$request, \$id)" berfungsi untuk memperbarui data "brodman", "Function" ini sangat berkaitan dengan "Function edit" ketika klien sudah melakukan edit dan mengklik button "simpan" maka data yang sudah masuk ke dalam database akan diperbarui dengan data yang sudah di edit, "Function destroy (ID)" berfungsi untuk menjalankan "action button delete" ketika di klik maka data akan terhapus di dalam "database" dan tampilan dalam "Index brodman".

3.2.6.6 Model Brodman

Model adalah sebuah jembatan yang digunakan antara "Database" dan "Visual Studio code" untuk menulis kode "PHP" tanpa harus menulis "query SQL" secara manual. Dalam fitur "brodman" terdapat lima model diantaranya "Brodman_aspects", "Brodman_categories", "Brodman_test_aspects", "Brodman_test_details" dan "Brodman_test" yang masing model memiliki relasi satu sama lain.

	php</th <th>1</th> <th>()nhn</th>	1	()nhn
			< pub
	<pre>namespace App\Models;</pre>		· · · · · · · · · · · · · · · · · · ·
			namespace App\Models;
	<pre>use Illuminate\Database\Eloquent\Model;</pre>		
			<pre>use Illuminate\Database\Eloquent\Model;</pre>
	5 references 0 implementations Windsurf: Refactor Explain		
	<pre>class Brodmann_aspects extends Model {</pre>		4 references 0 implementations Windsurf: Refactor Explain
			class Brodmann_categories extends Model {
	<pre>protected \$table = 'brodmann_aspects';</pre>		0 references
			<pre>protected \$table = 'brodmann_categories';</pre>
	0 references		
10	protected \$guarded =[0 references
11	'id'		protected \$guarded =[
12];	11	'id'
13		-12	1.
14		12	
	}	12	

`Gambar 3.63 Brodman Model

Gambar 3.63 adalah model yang digunakan dalam fitur "Brodman" terdiri dari lima model yang semuanya berada dalam "namespace App\Models" dan mewarisi "extends" kelas dasar "Model" dari Laravel yang dapat diartikan semua model ini dapat digunakan untuk berinteraksi langsung dengan tabel tabel yang ada di *database* sesuai dengan nama model untuk mengambil, menyimpan, mengedit serta menghapus data. "protected table" menyatakan bahwa model ini terhubung dengan tabel database dari masing masing model dan "protected \$guarded = ['id']" juga terdapat dalam semua model yang berfungsi untuk mengambil semua kolom tabel yang ada di "database" kecuali kolom "ID". Pada model "Brodman test aspects, Brodman test details dan Brodman test" memiliki relasi satu sama lain, "Brodman test aspects" memiliki relasi dengan "Brodman aspects" melalui "aspect id" yang ditunjukan dengan kode "return \$this->belongsTo(Brodmann aspects::class, 'aspect id');", lalu pada model "Brodman test details" memiliki relasi dengan model "Brodmann categories" melalui "category id" yang ditunjukan melalui *"return* kode Sthis->belongsTo(Brodmann categories::class,'category id');" dan pada model "Brodman test" memiliki kode Sthisreturn >hasMany(Brodmann test details::class,'header id');" dan "return \$this->hasMany(Brodmann test aspects::class,'header id');" yang memiliki arti "Brodman test" memiliki banyak "details" dan "aspects" ditunjukan dari kode "hasMany".



3.2.7 OE (Oemah Etnik)

3.2.7.1 Import Database

Sebelum mengerjakan proyek, mahasiswa magang diwajibkan untuk melakukan import "*Database*" kedalam "Device" pribadi agar dapat membuka *website*. Dalam pembuatan database proyek "Oe" mahasiswa magang tidak ikut dalam perancangan *database*, karena proyek sudah berjalan sebelum mahasiswa magang melaksanakan magang.

00	sales_invoice	e_details	- 10	Browse	K Structure	R Search	👫 Insert	🚍 Empty	Drop	0 InnoDB	utf8mb4_general_ci	48.0 KiB	-
Type to filter these, Ent.	sales_invoice	e_dps	\$2	Browse	K Structure	🤹 Search	🚰 Insert	👷 Empty	😂 Drop	0 InnoDB	utf8mb4_general_ci	96.0 KiB	
1 🕶 >>>	sales_invoice	e_headers	*	Browse	M Structure	R Search	Se Insert	🚍 Empty	Drop	0 InnoDB	utf8mb4_general_ci	S0.0 KiB	
New	sales_order_	detail2s	*	Browse	K Structure	R Search	Se Insert	🚍 Empty	😂 Drop	0 InnoDB	utf8mb4_general_ci	16.0 K18	
admin_users	sales_order_	details	*	Browse	K Structure	R Search	Sei Insert	👷 Empty	😂 Drop	0 InnoDB	utf8mb4_general_ci	64.0 KiB	-
+- Madvance_payments	sales_order_	headers	*	Browse	🔀 Structure	Rearch	Si Insert	🚍 Empty	Drop	e InnoDB	utf8mb4_general_ci	128.0 KiB	
-3/1 advance_receives	sales_quotat	ion_details	- 10	Browse	M Structure	R Search	👫 Insert	🚍 Empty	Drop	e InnoDB	utf8mb4_general_ci	16.0 KiB	-
eap_detail1s	sales_quotat	ion_headers	\$2	Browse	M Structure	@ Search	Se Insert	🚍 Empty	C Drop	0 InnoDB	utf8mb4_general_ci	16.0 KiB	
ap detail2s	sales return	details		Browse	M Structure	R Search	Sei Insert	Empty	C Drop	0 InnoDB	utf8mb4 general ci	48.0 K18	
- A ap_headers	sales return	headers	-	Browse	M Structure	@ Search	Sé Insert	B Empty	C Drop	0 InnoDB	utf8mb4 general ci	48.0 KiB	-
- ar detail?e	services			Browse	M Structure	Search	Si Insert	E Empty	C Drop	9 InnoDB	utf8mb4_general_ci	16.0 KiB	
- ar headers	eessions		alla.	Browse	Bé Structure	B Search	Se locart	Emply	O Drop	2 InnoDB	utt8mb4 unicode ci	48.0 KiB	
authorization_modules		project	24	- Drowne	Re Structure	A Coard	S- locart	Emoty .		A locoDP	utt?mb2_anosac_ci	16 0 19	
authorization_positions	C st_unienne_	project	- Mi	Diowse	To Characters	Contraction	2.1 Incont	Cimply	O Dese	23 Jacob D	utionib3_general_cr	10.0 100	
autonumbers	C statuses		- 24	E DIOWSE	M Siluciole	Search	ge msen	Sec. 1		55 IIII008	unsmus_uncode_ci	10.0 K10	
auto_email_receivers	U status_chan	nei_mappings	- 54	Browse	M Structure	search	3 insert	m Empty	G Drop	49 INNOUS	uttomb4_general_cl	10.0 K18	
bank_cashin_details	stock_opnan	ie_details	- 53	Browse	K Structure	Search	Se Insert	Here Empty	Drop	4 InnoDB	utf8mb3_general_ci	16.0 K18	
bank_cashin_headers	stock_opnan	ne_headers	- 🕸	Browse	K Structure	Rearch	si Insert	🚍 Empty	Drop	3 InnoDB	utf8mb3_general_ci	16.0 KiB	-
bank_cashout_details	suppliers		\$	Browse	K Structure	Search	📲 insert	🚍 Empty	Drop	2 InnoDB	utf8mb4_general_ci	48.0 KiB	
bank_cashout_headers	supplier_inve	pice_detail1s	- *	Browse	M Structure	R Search	3≓ Insert	👷 Empty	Drop	0 InnoDB	utf8mb4_general_ci	64.0 KiB	
ing pranches	supplier_inve	pice_detail2s	- 🚖	Browse	K Structure	া Search	§-€ Insert	🚍 Empty	😂 Drop	0 InnoDB	utf8mb4_general_ci	48.0 K18	
- cashiers	supplier_inve	pice_headers	*	Browse	M Structure	Rearch	si Insert	👷 Empty	Drop	0 InnoDB	utf8mb4_general_ci	80.0 KiB	-
cashier histories	unit_material	s	*	Browse	🔀 Structure	👒 Search	S-i Insert	🚍 Empty	Drop	53 InnoDB	utf8mb4_general_ci	48.0 KiB	
coas	users		*	Browse	K Structure	Rearch	3-i Insert	🚍 Empty	Drop	6 InnoDB	utf8mb3_unicode_ci	64.0 KiB	
- coa_types	variant_type	3	\$2	Browse	K Structure	Search	3ji insert	🚍 Empty	Drop	13 InnoDB	utf8mb4_general_ci	16.0 KiB	
- Credentials	warehouses		-	Browse	M Structure	Search	Sei Insert	C Empty	C Drop	4 InnoDB	utf8mb3 unicode ci	48.0 K18	
Currencies	139 tables		Sum	_			-			62,016 InnoDB	utf8mb4 0900 ai ci	20.3 MiB	0 B
+ - 2 customers													

Gambar 3.64. Struktur Database Oe

Gambar 3.64 merupakan struktur *database* yang digunakan dalam proyek "Oe" yang terdiri dari 139 tabel yang saling berhubungan. Setiap tabel memiliki relasi yang dirancang untuk mendukung integrasi data serta kelancaran proses bisnis dalam sistem yang sedang dikembangkan.

3.2.7.2 Membuat halaman Detail Transaction

Mahasiswa magang ditugaskan untuk membuat sebuah halaman untuk menampilkan *detail "Transaction*" namun hanya halamannya saja tanpa menarik data dari "*database*" melalui "*Controller*", halaman ini dibuat untuk dijadikan sebuah "*Template*" yang nantinya akan di kembangkan lagi oleh "*Senior programmer*".

Pesanan - SO-000	021981				Bata	ılkan Pesanan 🛛 Edit 🚥 🗙	
J	X		1	2		0	
Dibuat - 15 Feb	Diproses - 15 Feb		Dike	emas	Dikirim	Selesai	
2025 16:15	2025 17:22						
o. Pesanan: SO-000021981		Sumber: 🔵 🛛	NTERNAL				
elanggan: 📴		Toko: Toko D	efault				
o. Ref: -		Tanggal:					
lesman;;;;;;		Lokasi: FS Ba	ngka				
roduk (1)							
roduk	Harga	Qty Diskon %	Nilai Diskon	Total			
Prod Lilac Lingga Tenun Dress - S 5100-897-SM S-M	-M Rp 1.199.000,00	1 10	Rp 119.900,00	Rp 1.079.100,00			
enerima			Rincian				
Nama			Qty Total (1	Rp 1.199.000			
Alamat			Produk)				
No. Telepop			Diskon	Rp 119.000			
			Lainnya				
Produk (1)							
Produk	Harga	Qty Di	skon % Nilai D	iskon Total			
Prod Lilac Lingga Tenun Dre 5100-897-SM S-M	ss - S-M Rp 1.199.00	0,00 1 10	Rp 119	900,00 Rp 1.079.100,00			
Penerima					Rincian		
Nama					Qty Total (1 P	roduk) Rp 1.199	.000
Alamat					Diskon	Rp 119.	.000
No. Telepon					Diskon Lainny	/a F	Rp 0
					Ongkos Kirim		Rp 0
					Diskon Ongko	os Kirim F	
Pengiriman							Rp 0
Pengiriman _{Kurir}	Gosend	No. Re	esi: -			Lainnya 🗸	Rp 0
Pengiriman Kurir COD	Gosend Tidak	No. Re Total I	esi: - Berat (Kg):	0.90 Kg		Lainnya v	Rp 0

Gambar 3.65 Halaman Detail Transaction

Gambar 3.65 adalah halaman "Detail Transaction" yang isi dari halaman ini tidak di ambil dari database melainkan di ketik dari halaman blade. Halaman ini nantinya akan dimodifikasi oleh "Senior programmer" yang nantinya isi dari halaman ini akan diambil dari "database" melalui "controller" dan isinya tidak akan sama tergantung pada "transaction" yang dipilih pada halaman "index transaction".



Gambar 3.66 kode Halaman Detail Transaction

Gambar 3.66 adalah kode yang membentuk tampilan pada gambar 3.65, halaman ini dibangun menggunakan "Bootstrap versi 5.3.0" yang berfungsi untuk mempermudah proses "styling" dan pengaturan "layout grid". Dengan menggunakan "Bootstrap", elemen seperti "Button, form, table dan card" dapat dirancang dengan tampilan yang lebih modern dan konsisten tanpa harus menulis "CSS". Selain itu, halaman ini juga menggunakan "Font Awesome" sebagai pustaka ikon utama serta "Bootstrap Icons" sebagai alternatif ikon tambahan yang memperkaya tampilan visual halaman. Untuk mengaktifkan komponen interaktif dari "Bootstrap" di sertakan kode "<script

src="<u>https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.</u> <u>min.js</u>"></script>". Dalam halaman ini terdapat tombol yang dapat berubah warna apabila di klik yaitu pada bagian "COD", hal ini memanfaatkan fungsi dari "JavaScript bagian toggleCOD(button) dan toggleLunas(button)" yang ditulis dalam "tag <script>".

3.2.7.3 Membuat halaman cek point

Pekerjaan selanjutnya yang harus diselesaikan mahasiswa adalah membuat halaman untuk mengecek poin. Halaman tidak sepenuhnya dibangun oleh mahasiswa magang, mahasiswa magang di tugaskan untuk menghias halaman menggunakan "CSS" dan menambahkan elemen lain seperti gambar dan animasi lain.





Gambar 3.67 Halaman Cek point

Gambar 3.67 adalah hasil dari modifikasi halaman cek poin yang dikerjakan oleh mahasiswa magang. Mahasiswa magang mengubah warna "*Background*" menjadi warna coklat, menambahkan aksesoris bintang, awan dan planet yang dapat bergerak (*float*), menambahkan gambar di sisi kanan dan kiri, mengubah tampilan "*button*" "Minta Otp", mengubah label untuk menginput nomor telepon dan kode otp, memodifikasi tampilan ketika poin sudah keluar.



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

max-width: 100%; max-height: 100%; -radius: 10ox star1 { top: 10%: left: 5%: animation-duration: 4s: } con.star2 { top: 20%; right: 10%; animation-duration: 3.5s;] tar3 { bottom: 15%; left: 20%; animation-duration: 3s; } star5 { bottom: 10%; right: 20%; animation-duration: 2.5s; top: 10%; left: -15%; animation-duration: 35s; top: 50%; right: -15%; animation-duration: 40s;] top: 20%; left: 80%;] le1 { top: 40%; left: 30%; animation-duration: 1.5s; } top: 55%; right: 20%; animation-duration: 2s;

Gambar 3.68 kode Halaman Cek point

Gambar 3.68 adalah kode yang digunakan untuk membangun halaman checkpoint. Pada bagian "Head" terdapat "Meta Tags" yang berfungsi untuk mengatur "viewport" untuk responsif di berbagai "device", terdapat "Custom CSS" yang berfungsi untuk mengatur warna latar belakang, animasi untuk berbagai elemen (fadelm, bounceln dan pulse) dan terdapat "style" khusus untuk "Card" dengan "gradient warna pink ke kuning muda", "Button" dengan efek "glow", input "field" yang sudah di "styling" serta elemen dekoratif seperti bintang,awan dan planet. Pada bagian "Body" "Container utama" berisi "form" pengecekan poin dengan menginput nomor telepon dan button "Minta OTP", input "OTP" dan "Button" checkpoint, terdapat animasi yang digunakan dalam konten utama yaitu animasi "fadeln" pada judul "Cek Poin Anda", animasi "bounceln" pada "card" utama dan efek "pulse" pada bagian "Button". Halaman ini menggunakan banyak sekali "JavaScript" seperti "requestOtp()" untuk menampilkan "loading spinner" dan "alert" simulasi pengiriman otp dan "checkButton()" untuk menampilkan "loading spinner" dan hasil setelah "delay", fungsi "JavaScript" pada halaman ini tidak hanya untuk animasi namun juga terdapat fungsi untuk memastikan nomor telepon tidak kosong dan minimal 10 digit dan memastikan "OTP" tidak kosong. Penggunaan "CSS" pada halaman ini juga berkontribusi untuk memaksimalkan animasi elemen, dengan menggunakan "@keyframes" untuk efek animasi, "transform" untuk

pergerakan elemen dan "*transition*" agar efek elemen bisa berjalan dengan halus.

3.2.7.4 Filter Transaction

Mahasiswa magang ditugaskan untuk membuat fungsi filter untuk "Transaction POS". Filter ini berfungsi untuk mencari data dari tanggal mulai yang ditentukan dan tanggal akhir yang ditentukan pada halaman "*index*".

Transaction	n POS		Dashboard 🌖 Transaction POS
Tanggal Mulai	Tanggal Berakhir	Tipe Transaksi	
		SEMUA	
Filter Data Reset Filter			

Gambar 3.69 Tampilan filter

Gambar 3.69 adalah tampilan filter yang dapat ditentukan oleh klien, terdapat "tanggal mulai", "tanggal berakhir" yang harus ditentukan oleh klien agar filter dapat berfungsi dan terdapat pilihan "Tipe transaksi" yang dibuat menggunakan "*Dropdown*" dengan enak tipe transaksi yang dapat dipilih, ketika klien sudah menentukan pilihannya dan mengklik "*button* filter data" maka *index* akan menampilkan data sesuai dengan "tanggal mulai, tanggal berakhir dan tipe transaksi" yang dipilih oleh klien.

\$filterStartDate = Carbon::today()->startOfYear()->format('d-m-Y');
\$filterEndDate = Carbon::today()->format('d-m-Y');

if (\$request->start_date !=null && \$request->end_date != null){
 \$startDateDecoded = rawurldecode(\$request->start_date);
 \$endDateDecoded = rawurldecode(\$request->end_date);
 \$start = Carbon::createFromFormat('!d-m-Y', \$startDateDecoded, 'Asia/Jakarta');
 \$end = Carbon::createFromFormat('!d-m-Y', \$endDateDecoded, 'Asia/Jakarta');
 \$filterStartDate = \$startDateDecoded;
 \$filterStartDate = \$startDateDecoded;
 \$filterEndDate = \$endDateDecoded;
}

81 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara

Gambar 3.70 kode Tampilan filter

Gambar 3.70 merupakan kode yang digunakan untuk menjalankan fungsi filter pada halaman "*Transaction*", kode ini ditulis dibagian "*Controller*" didalam "*function getIndex()*". Pada bagian "*\$filterStartDate* dan *\$filterEndDate*" terdapat kode "*Carbon::today()*" berfungsi untuk menghasilkan objek tanggal hari ini , "*startOfYear*" berfungsi untuk mengubah tanggal menjadi awal tahun ini dan "format('d-m-Y)" berfungsi untuk mengubah tanggal ke format "*string*", terdapat kondisi untuk mengecek apakah terdapat inputan dari nilai "*start_date*" dan "*end_date*" yang ditunjukan dari kode "*if (\$request->start_date != null && \$request->end_date != null) { if (\$request->start_date != null && \$request->end_date != null) { "*. Selain pengecekan nilai, terdapat juga pengecekan apakah tanggal akhir lebih besar dari tanggal awal yang ditunjukkan pada kode "*if(\$end->greaterThanOrEqualTo(\$start)*)".

3.2.7.5 Render Status Stock Opname

Mahasiswa magang ditugaskan untuk mengubah halaman isi kolom "status" pada halaman "*stop_opname index*" untuk mengubah status apabila "aktif" maka "*background*" berwarna merah dan "*Approve*" maka "*background*" berwarna hijau.



STATUS	\$	
Aktif		
Approved		
Aktif		

Gambar 3.71 Tampilan status

Gambar 3.71 adalah tampilan yang sudah dimodifikasi oleh mahasiswa magang. Pada saat klien "<u>create</u>" "stock opname" maka status menjadi "aktif" dan status bisa berubah menjadi "Approved" apabila "stock opname" sudah di approve pada halaman "show_stock_opname".



Gambar 3.72 kode Render status

Gambar 3.72 adalah kode yang membuat filter data dapat berfungsi. Bagian "*render*" berfungsi untuk menentukan tampilan isi dari salah satu kolom yaitu "*Status*", bagian "*console.log*" berfungsi untuk menampilkan isi data status dan "*status_id*" ke "*console*" untuk "*debugging*" yang berfungsi

untuk memastikan data yang diterima benar sebelum membuat "badge". Pada bagian "const badgeStyle" berfungsi untuk memodifikasi dari "font, padding dan style huruf". Dalam kode ini juga memiliki kondisi jika "status_id==1" maka akan menampilkan "badge Aktif" dengan warna merah "bg-danger", jika "status_id==6" maka akan menampilkan "badge Approved" dengan warna hijau "bg-success".

3.2.7.6 Modifikasi Filter Inventory

Pekerjaan selanjutnya yang dikerjakan mahasiswa magang adalah memodifikasi filter gudang yang terdapat dalam menu "*Inventory*". Pada awalnya gudang tidak bisa dipilih lebih dari satu, mahasiswa magang ditugaskan untuk membuat tampilan agar kotak "*dropdown*" dapat memilih dan menampung gudang lebih dari 1.



Gambar 3.73 Tampilan Filter gudang

Gambar 3.73 adalah tampilah akhir yang sudah dimodifikasi oleh mahasiswa magang. Terdapat empat pilihan gudang yang dapat dipilih oleh

klien dan apabila gudang sudah terpilih maka latar belakang akan menjadi warna abu abu.



Gambar 3.74 kode Filter Gudang

Gambar 3.74 adalah kode yang ditambahkan mahasiswa magang agar filter gudang dapat dipilih lebih dari satu. Bagian penambahan yang pertama terletak pada "JavaScript" dan"CSS", penambahan "CSS" berfungsi untuk memodifikasi tampilan "dropdown" ketika gudang dipilih, kode yang "<link *rel="stylesheet"* digunakan adalah href="{{ asset('js/plugins/select2/css/select2.min.css') }}">" . "JavaScript" adalah komponen yang paling penting untuk menjalan fungsi "multiple" gudang *src="*{{ *"<script* penambahan kode dengan asset('js/plugins/select2/js/select2.min.js') }}"></script>". Setelah menambahkan "JavaScript dan CSS", pada bagian "Select" ditambahkan *"Multiple"* agar dapat memilih lebih dari satu. pada bagian "\$('filter warehouse')" adalah fungsi yang memungkinkan dapat memilih lebih dari satu gudang dan ".*select2({})*" berfungsi untuk mengaktifkan "*plugin Select2*" agar "*dropdown*" menjadi lebih iteratif. Lalu di bawah terdapat "*Function filterdata*" yang berfungsi agar data dapat terfilter berdasarkan gudang yang dipilih.

3.2.7.7 Halaman Transaksi Void

Mahasiswa magang ditugaskan untuk membuat halaman baru dengan nama "*Transaksi Void*" pada menu "Penjualan". Halaman ini berisi tabel dengan sepuluh kolom diantaranya "Tindakan, Kode, Kode Transaksi, Tanggal Penukaran, Nama *Customer*, Cabang Asal, Cabang Penukaran, *Note*, *Status* dan Dibuat". Halaman ini belum ada data dan belum terpakai masih dalam tahap perencanaan yang nantinya akan disambungkan dalam menu "*Finance*" dan nanti dilihat dalam halaman "*Transaksi Void*".

Fransaksi Voj	d					D	ashboard > 1	Transaksi
						-		
								_
							I	Approve
10 🗸							Search keywor	Approve
10 V TINDAKAN A	KODE 0 KODE TRANSAKSI	TANGGAL PENUKARAN	NAMA CUSTOMER 0	CABANG ASAL 0	CABANG PENUKARAN ©	NOTE 0	Search keywor	Approve d. DIBUAT
10 V TINDAKAN ^	KODE 0 KODE TRANSAKSI	• TANGGAL PENUKARAN •	NAMA CUSTOMER ©	CABANG ASAL ©	CABANG PENUKARAN ©	NOTE 0	Search keywor STATUS ©	Approve d DIBUAT

Gambar 3.75 Tampilan transaksi void

Gambar 3.75 adalah tampilan yang nantinya akan dipakai untuk menampilkan dan melakukan "*Approve*" per kolom atau perdata.



Gambar 3.76 Kode Route

Kode 3.76 adalah bagian "*Route*" yang masing masing memiliki fungsinya masing masing. "*Route Index*" berfungsi untuk mengantarkan klien ke halaman "Transaksi *Void Index*" pada saat meng klik "*Transaksi Void*" pada submenu "Penjualan", "*Route datatable*" nantinya akan berfungsi untuk menampilkan data dalam halaman "*Index*" dan "*Route Approve*" berfungsi agar *button "Approve*" dapat berfungsi.

3.2.7.8 Inputan Tier pada halaman Customer

Pekerjaan selanjutnya yang harus dikerjakan oleh mahasiswa magang menambahkan inputan baru dengan nama "*Tier*" yang dimana inputan ini berupa "*dropdown*" dengan lima pilihan yaitu "*Normal, VIP, Priority, Red Flag* dan *Blacklist*".



Buat Customer		Dashboard	Daftar Customer 📎 Buat Customer
			Simpan Batal
Isian dengan asterik (*) wajib diisi!			
Nama *	Name		
Tier	Pilih Tier	-	
No Telepon	Normal VIP		
Email	Priority		
Tgi Lahir	Red Flag Blacklist		
Alamat	Alamat 1		
			~

Gambar 3.77. Tampilan Tier

Gambar 3.77 adalah tampilan yang sudah dibuat oleh mahasiswa akhir dalam halaman "*Create Customer*". "*Tier*" dibuat menggunakan "*Select2*" dan "*Option*" untuk tiap opsi nya.

Gambar 3.78 Kode Controller

Gambar 3.78 adalah Kode yang ditulis dalam "Controller Customer" berfungsi untuk menerima input dari "tier" dalam halaman "Customer Create" dan masuk ke dalam "Table Customer" kolom "Tier". Kode ini ditulis dalam "Function store dan update" agar dapat pada halaman "Create" nilai dari pilihan "Tier" dapat masuk ke dalam table dan pada halaman "edit".

<pre><div class="row"></div></pre>
<label class="col-sm-3 col-form-label" for="tier_id">Tier</label>
<div class="col-sm-6"></div>
<pre><select class="form-select form-control-select2" id="tier_id" name="tier" required=""></select></pre>
<pre><option value=""></option></pre>
<pre><option value="Normal">Normal</option></pre>
<pre><option value="VIP">VIP</option></pre>
<pre><option value="Priority">Priority</option></pre>
<pre><option value="Red Flag">Red Flag</option></pre>
<pre><option value="Blacklist">Blacklist</option></pre>

Gambar 3.79 Kode Dropdown Tier

Kode 3.79 berfungsi untuk menampilkan pilihan yang terdapat dalam "*Tier*", terdapat lima pilihan yaitu "*Normal, VIP, Priority, Red Flag* dan *Blacklist*".

"Dropdown" ini dibuat menggunakan "Select2" nantinya akan diatur oleh "Function" yang ditulis pada bagian "JavaScript".

<div class="col-sm-6"></div>
<pre><select class="form-select form-control-select2" id="tier_id" name="tier" required=""></select></pre>
<pre><option value="">Pilih Tier</option></pre>
<pre><option \$customer-="" &&="" isset(\$customer)="" value="Normal" {{="">tier == 'normal' ? 'selected' : '' }}>Normal</option></pre>
<pre><option \$customer-="" &&="" isset(\$customer)="" value="VIP" {{="">tier == 'VIP' ? 'Selected' : '' }}>VIP</option></pre>
<pre><option \$customer-="" &&="" isset(\$customer)="" value="Priority" {{="">tier == 'Priority' ? 'selected' : '' }}>Priority</option></pre>
<pre><option \$customer-="" &&="" isset(\$customer)="" value="Red Flag" {{="">tier == 'Red Flag' ? 'selected' : '' }}>Red Flag</option></pre>
<pre><option \$customer-="" &&="" isset(\$customer)="" value="Blacklist" {{="">tier == 'Blacklist' ? 'selected' : '' }}>Blacklist</option></pre>

Gambar 3.80 Kode Dropdown Tier Edit

Kode 3.80 adalah yang digunakan pada bagian "*edit*", kode ini sama dengan yang digunakan pada "*create*" namun terdapat beberapa perbedaan. Pada bagian "*Edit*" "*option*" menggunakan "*isset*" agar pilihan yang pilih pada halaman "*Create*" akan masuk ke dalam halaman *edit*.

3.2.8 Sempana

3.2.8.1 Import Database

Sebelum mengerjakan proyek, mahasiswa magang diwajibkan untuk melakukan "*import database*" ke dalam "*Device*" pribadi agar dapat membuka *website*. Dalam pembuatan "*Database*" proyek "Sempana" mahasiswa magang tidak ikut dalam perancangan *database* karena proyek sudah berjalan sebelum mahasiswa magang melaksanakan magang.



sempana 🔺	0	sales_invoice_detail2s	*	Browse	M Structure	R Search	Hinsert	m Empty	C Drop	0 InnoD8	utt8mb4_general_ci	45.0 Ki8	
Type to Klass Press, Erit		sales_invoice_details	**	Browse	Structure	t Search	Se Insert	# Empty	Con Drop	0 InnoDB	uti8mb4_general_ci	48.0 Ki8	÷
1 🗸 >>>	0	sales_invoice_dps	*	Browse	Structure	R Search	S-i Insert	# Empty	C Drop	0 InnoD8	utt8mb4_general_ci	96.0 Ki8	
-Lo New	0	sales_invoice_headers	-	Browse	M Structure	& Search	Se Insert	# Empty	C Drop	0 InnoDB	uti8mb4_general_ci	50.0 K15	
€ k admin_users	0	sales_order_detail2s	*	Browse	M Structure	Rearch	34 Insert	# Empty	C Drop	e InnoDB	utt8mb4_general_ci	16.0 KiB	
advance_payments	0	sales_order_details	-	Browse	M Structure	e Search	Se Insert	# Empty	Orop	0 InnoDB	uti8mb4_general_ci	64.0 KiB	
advance_receives	0	sales order headers	-	Browse	Le Structure	+ Search	S-i Insert	B Empty	Drop	0 InnoDB	utt8mb4 general ci	128.0 KiB	
ap_detail2s	n	sales quotation details	-	Browse	Le Structure	+ Search	Linsert	E Fmaty	Oran	0 InnoDB	utBmb4 general ci	16.0 KiB	
ap headers		sales modation headers	-	Browne	Le Structure	a Search	S. Incert	Emety	B Dran	8 InnoDR	uttimba general ci	16.0 KIR	
ar detail1s	liñ	sales colum details	-	Brown	Te Chartero	A Carrel	E lacart	Emply	Drop	0 InnoDB	uttimba general ci	er o rin	
ar_detail2s		sales_return_deants	14	- Drowse	12 Churchure	- Constant	a moon	The County	O Dias	0 10000	dubino4_general_ci	40.0 KIR	
- r ar_headers	1	sales_return_neaders	H	erowse	N Soucture	a search	3- inperi	m Empty	O Drop	0 InnoL/B	ubsmb4_general_ci	40.0 K10	
- authorization_modules		services	TH.	Browse	IN Structure	* search	are insert	THE EMPTY	Urop	@ InnoDB	uttamo4_general_c	16.0 K18	
- authorization_positions	0	sessions	Ŵ	Browse	N Structure	R Search	Insert	W Empty	C Drop	1 InnoDB	utt8mb4_unicode_ci	48.0 KiB	
- Autonumbers		so_timeline_project		Browse	M Structure	& Search	Seinsert	# Empty	Orop	0 InnoD8	uti8mb3_general_ci	16.0 KiB	
auto_email_receivers	0	statuses	俞	Browse	M Structure	R Search	34 Insert	👷 Empty	Drop	30 InnoDB	utf8mb3_unicode_ci	16.0 Ki8	
bank_cashin_details		status_channel_mappings	14	Browse	K Structure	Search	S-i Insert	₩ Empty	😄 Drop	49 InnoDB	ut/8mb4_general_ci	16.0 KiB	
Dank_cashin_neaders		suppliers	*	Browse	M Structure	R Search	3 € Insert	me Empty	Drop	2 InnoDB	utt8mb4_general_ci	48.0 Ki8	
the bank cashout beariers		supplier_discount_details	*	Browse	M Structure	& Search	Se Insert	H Empty	C Drop	@ InnoDB	utt8mb4_general_ci	16.0 KiB	
branches	0	supplier_discount_headers	*	Erowse	Structure	R Search	H Insert	m Empty	C Drop	0 InnoDB	utf8mb4_general_ci	16.0 K18	
cancel_reason_order	0	supplier_invoice_detail1s	-	Erowse	M Structure	R Search	Se Insert	W Empty	Drop	3 InnoDB	utt8mb4_general_ci	64.0 Kim	
- cashiers	0	supplier invoice detail2s	-	Browse	M Structure	R Search	H Insert	# Empty	O Drop	0 InnoDB	utt8mb4 general ci	48.0 Ki8	
- cashier_histories	liñ	supplier invoice headers	-	Erowse	le Structure	* Search	S-i Insert	E Empty	O Drop	3 InnoDB	uttRmb4 general ci	sele kis	
coas	1 o	unit materials	-	Browse	Lei Structure	A Search	S. Incart	Emety	Drop	55 InnoDB	ut8mb4 neneral ci	48.0 Kin	
coa_types	liñ	union bunar	-	Browne	Le Churchure	a Caareb	Ed Incert	Emply Emply	Drop	1 InnoDD	ut@mbd_general_ci	16.0 158	
- configurations		warahouses	24	Drowne	P) outraine	A Castoli	E lacat	Tereti	Drop	< loneDD	utilizabili utilocda ci	49.0 210	
credentials	19	130 tables	Sum	Er Dwse	NJ annume	a search	are insert	M. Futbo	Chob	9.821 InnoDE	utfilmb4.0900 ai ci	6.4 Mill	0.8
Currences		too works	Sum							-, mmode	0900_al_cl	and have	

Gambar 3.81 Struktur Database Sempana

Gambar 3.81 merupakan struktur *database* yang digunakan dalam proyek "Sempana" yang terdiri dari 139 tabel yang saling berhubungan. Setiap tabel memiliki relasi yang dirancang untuk mendukung integrasi data serta kelancaran proses bisnis dalam sistem yang dikembangkan.

3.2.8.2 Menambahkan Tooltip

Mahasiswa magang tugaskan untuk memodifikasi icon yang sudah dibuat sebelumnya oleh "Senior programmer". Mahasiswa magang ditugaskan ketika kursor berada pada ikon atau melewati ikon akan memunculkan notifikasi mengambang yang berisikan informasi yang sudah diberikan oleh "Senior programmer" dengan menggunakan "tooltip".



"?" adalah untuk memberitahu "*button*" di sebelah kiri berfungsi untuk mencari barang yang diinginkan.



Gambar 3.83 kode Tooltip

kode diatas adalah yang dibuat oleh mahasiswa magang yang berfungsi untuk memunculkan informasi ketika kursor mengarah pada icon "?". Pada bagian "*<div>*" menggunakan "*class=col-auto*" berfungsi agar ukuran kolom disesuaikan secara otomatis sesuai dengan isi. Lalu bagian yang paling penting adalah "**" yang berfungsi agar icon"?" dapat memunculkan informasi yang diinginkan ketika kursor melewati icon tersebut. Pada bagian "*data-bs-placement="top"*" berfungsi untuk memunculkan notifikasi di bagian atas *icon*.

3.2.8.3 Filter Transaksi POS

Pekerjaan selanjutnya yang dikerjakan oleh mahasiswa magang adalah menjalankan fungsi filter transaksi pada bagian "POS". Pada bagian filter terdapat tiga bagian yang dapat difilter yaitu "Tanggal Mulai, Tanggal Berakhir dan Tipe Transaksi", dari tiga kolom tersebut filter "tanggal berakhir dan tipe transaksi" sudah berfungsi namun "tipe transaksi" belum bisa berfungsi.

Filter Data Reset Filter SEMUA POS PRE ORDER	
Filter Data Reset Filter SEMUA POS PRE ORDER	
POS PRE ORDER	
🗄 SEMUA 📀 BELUM BAYAR 🛛 BELUM PROSES 🎽 PICKING 🕌 PACKING 🐘 DALAM PENGIRIMAN 📀 SELESAI 📀 PROSES CANCEL 📀 DK	O DICANCEL
D PROSES RETUR	

Gambar 3.84 Tampilan Filter POS

Gambar 3.84 adalah tampilan pada bagian "Transaksi POS" yang terdapat tiga bagian filter, "Filter Tipe Transaksi" memiliki tiga pilihan yaitu "Semua", "POS" dan *"Pre Order*". Tiga filter ini tidak perlu untuk diisi semua, filter dapat berjalan jika hanya memilih salah satu dari tiga pilihan. Bagian filter "Tipe transaksi" akan masuk ke dalam kolom "*Source*" yang nantinya akan memunculkan data dengan "tipe transaksi" yang dipilih.

Gambar 3.85 Kode Filter POS

Gambar 3.85 adalah kode yang menjalan fungsi filter "Tipe Transaksi". Pada bagian pertama membuat kondisi untuk mengambil "Value dari type_transaction" yang bukan "999", "999" adalah id dari pilihan "semua" yang terdapat dalam "tipe transaksi". Pada bagian "*\$query*" berfungsi untuk menambahkan filter ke dalam "query" apabila kondisi "*if*" terpenuhi. Pada bagian "Select" ditulis dalam halaman "Blade" dan bagian "Option value" diisi dengan *id* sesuai dengan pilihan masing masing, id dapat diketahui dengan cara melakukan "dd" pada bagian "Controller". Kode "ajax" ditulis pada halaman yang sama dengan "option" yang berfungsi untuk

mengambil data dari server berdasarkan filter tanggal dan jenis transaksi yang dipilih.

3.2.9 Jaya Abadi Tapioka

3.2.9.1 Import Database

Sebelum mengerjakan proyek mahasiswa magang diwajibkan untuk melakukan import *database* kedalam "*Device*" pribadi agar dapat membuka *website*. Dalam pembuatan *database* proyek "Jaya Abadi Tapioka" mahasiswa magang tidak ikut dalam perancangan *database* karena proyek sudah berjalan sebelum mahasiswa magang melaksanakan magang.

(= w			· •			
ET alampung	sales_invoice_headers	👷 🔠 Browse 🔐	Structure 👒 Search 👫 Insert	🚍 Empty 🤤 Drop	2,816 InnoDB utf8mb4_0900_ai_ci	1.8 MiB -
Type to filter these, Entr	sales_invoice_item_details	👷 🔝 Browse 🥻	Structure 👒 Search 👫 Insert	🚍 Empty 🤤 Drop	2,833 InnoDB latin1_swedish_ci	448.0 KiB -
1 🕶 >>>	sales_order_details	👷 🔟 Browse 📝	Structure 👒 Search 👫 Insert	🚍 Empty 🤤 Drop	1,009 InnoDB utf8mb4_0900_ai_ci	352.0 КіВ -
- B New	sales_order_headers	👷 📑 Browse 🥻	Structure 👒 Search 👫 Insert	🚍 Empty 🥥 Drop	993 InnoDB utf8mb4_0900_ai_ci	432.0 KiB -
account_receivables	service_details	🌟 📄 Browse 🏼 🖌	Structure 👒 Search 👫 Insert	🚍 Empty 🔤 Drop	10 InnoDB utf8mb4_0900_ai_ci	64.0 KiB -
account_receivable_detai	service_headers	👷 🔄 Browse 📝 S	Structure 👒 Search 👫 Insert	🚍 Empty 🥥 Drop	15 InnoDB utf8mb4_0900_ai_ci	112.0 K18 -
+ admin_users	sick_leaves_employees	🚖 🗐 Browse 🛃	Structure 🤏 Search 👫 Insert	🚍 Empty 🥥 Drop	0 InnoDB latin1_swedish_ci	32.0 КІВ -
+ Advance_payments	so_schedules	🚖 🔲 Browse 🥻 S	Structure 🧟 Search 👫 Insert	🚍 Empty 🤤 Drop	3,188 InnoDB latin1_swedish_ci	1.0 MiB -
+- y ap_details	so_schedule_details	🚖 🔟 Browse 🏼 🕅	Structure 👒 Search 👫 Insert	🚍 Empty 🤤 Drop	3,203 InnoDB latin1_swedish_ci	448.0 Ків -
t z an headers	statuses	🚖 🔲 Browse 📝	Structure 👒 Search 📲 Insert	🚍 Empty 🥥 Drop	17 InnoDB utf8mb3_unicode_ci	16.0 K18 -
ar_detail1s	suppliers	🚖 📃 Browse 📝 S	Structure 🤏 Search 👫 Insert	🚍 Empty 🥥 Drop	98 InnoDB utf8mb4_0900_ai_ci	48.0 KIB -
	supplier_invoice_detail1s	🚖 🔟 Browse 🔐	Structure Search 👫 Insert	🚍 Empty 🥥 Drop	5,299 InnoDB utf8mb4_0900_ai_ci	2.0 MiB -
+ 🖈 ar_headers	supplier_invoice_detail2s	👷 🔟 Browse 🔰		🚍 Empty 🥥 Drop	Ø InnoDB utf8mb4_0900_ai_ci	48.0 KiB -
+ A assets	supplier invoice headers	🚖 🗐 Browse 🕼	Structure 👒 Search 😹 Insert	🚍 Empty 🥥 Drop	1,543 InnoDB utf8mb4 0900 ai ci	480.0 K18 -
+ A asset_budget_details	temp bookings	🔶 🗐 Browse 🖬	Structure 🗷 Search 👫 Insert	Empty O Drop	0 InnoDB utf8mb4 0900 ai ci	16.0 KiB -
*- M asset_budget_headers	timbangan headers	Strowse 14	Structure 👒 Search 🚟 Insert	Empty O Drop 4	2,582 InnoDB utf8mb4 0900 ai ci	14.5 MiB -
+ attendance shifts	timbangan temps	🖈 🗏 Browse 🕅	Structure 🕜 Search 👫 Insert	Empty Oppo	@ InnoDB_utf8mb4_0900_ai_ci	32.0 K18 -
authorization_modules	transaction details	🔶 🖂 Browse ile 3	Structure	Empty Oron	@ InnoDB_utf8mb4_0900_ai_ci	96.0 K18 -
authorization_positions	transaction detail partners	A E Browse 14	Structure in Search Seinsert	Empty Oron	0 InnoDB utf8mb4 0900 ai ci	48.0 KIR -
- autonumbers	fransaction beaders	A Browse lie 1	Structure le Search 34 Insert	Emply Drop	e InnoDB uff8mb4 0900 ai ci	112.0 KiB -
+ auto_email_receivers	transaction newments	Browse Let	Structure in Search S- Insert	Empty Orop	e InnoDB utf8mb4_0900 ai ci	54.8 KiB
+ / bank_cashin_details		A Rouse le 1	Structure in Search El Insert	Empty Drop	21 InnoDR utf9mb4_0000 ai ci	49 9 K18
back_cashin_headers		A Browne 144	Structure 😪 Search 📑 Insert	Empty Drop	3 InnoDD utfomb4_0000_al_ci	GA Q VID
tak cashoul headers		T Drowse (M a	Structure og Search Silverd	👷 Empty 🥥 Drop	5 InnobB unonb5_unicode_ci	04.0 KID -
tooking notes	U warenouses	Browse M	Structure og Search 34 Insert	empty 🔘 Drop	+ innous un8mb3_unicode_ci	40.0 NID -
+ m branches	□ warning_letters	😭 🖂 Browse 🕅	structure 👒 search 👫 Insert	: # Empty 😂 Drop	© Inno⊔B latin1_swedish_ci	32.0 K18 -
the cash_receipts	101 tables	Sum		~28	2,003 INNOUD ut/8mb4_0900_ai_ci	20.0 PLLD 0 B

Gambar 3.86 Struktur Tabel Jaya Abadi Tapioka

Gambar 3.86 merupakan struktur *database* yang digunakan dalam proyek "Jaya Abadi Tapioka" yang terdiri dari 151 tabel yang saling berhubungan. Setiap tabel memiliki relasi yang dirancang untuk mendukung integrasi data serta kelancaran proses bisnis dalam sistem yang sedang dikembangkan.

3.2.9.2 Export Excel

proyek Jaya Abadi Tapioka memiliki menu "*Finance*" dan didalam menu ini terdapat beberapa pilihan submenu yang dapat dipilih oleh klien. 93 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara
Mahasiswa magang ditugaskan untuk membuat sebuah "*Button*" yang dapat *mendownload* laporan kedalam file "*Xlsx*" di bagian "*Finance*" "Laporan AP" dan "Laporan AR". Pada bagian "Laporan AP" dan "Laporan AR" terdapat tiga pilihan yaitu "Laporan Ap / Ar *card*, *aging* dan *invoice list*".



Gambar 3.87 Tampilan Laporan

Gambar 3.87 adalah salah satu tampilan laporan dari "Laporan AR Invoice list". Sebelum mengeluarkan laporan, klien diharuskan untuk memilih "Tanggal Mulai" dan "Tanggal Berakhir" laporan dan terdapat "Button Buat laporan" yang nantinya akan menampilkan laporan seperti pada gambar 3.87 Pada halaman laporan terdapat tiga "Action Button" yang masing masing memiliki fungsi ketika diklik, mahasiswa magang membuat "Button" berwarna biru yang ketika diklik akan mendownload file "Xlsx" yang isinya sama seperti yang ada dalam tampilan.

```
function exportToExcel(tableID, filename = '') {
    let table = document.getElementById(tableID);
    let wb = XLSX.utils.book_new();
    let ws = XLSX.utils.table_to_sheet(table);
    XLSX.utils.book_append_sheet(wb, ws, "Sheet1");
    filename = filename ? filename + ".xlsx" : "export.xlsx";
    XLSX.writeFile(wb, filename);
}
```

Gambar 3.88 kode Export Excel

Gambar 3.88 adalah kode yang membentuk "Button Export Excel" dan "Function" yang memungkinkan file terdownload ke dalam format 94 Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara "Xlsx". Untuk mendownload file ke dalam format "Xlsx" diperlukan "JavaScript" khusus yang berfungsi untuk membuat file "excel", lalu membuat "Button" dengan judul "Export to Excel" dengan menggunakan "btn-primary" agar "Button" menjadi warna biru. Untuk dapat mendownload file diperlukan "Function" yang ditulis menggunakan "SheetJS" agar ketika "Button" diklik maka akan langsung mendownload file dalam format "Xlsx". kode ini nantinya akan ditambahkan pada laporan AR dan AP yang lain agar semua laporan dapat didownload ke dalam format "excel".

3.3 Kendala yang Ditemukan

Terdapat beberapa kendala yang ditemukan *Intern* pada saat proses magang berlangsung pada PT Generasi Muda Gigih adalah :

- Kurang terbiasa dengan bahasa pemrograman PHP, CSS dan JS.
 Bahasa pemrograman yang digunakan dalam PT GMG jarang digunakan di dalam dunia kuliah, sehingga membutuhkan waktu extra untuk memahami serta mengimplementasikan dengan baik dan benar.
- Kurang familiar dengan framework laravel. Laravel memiliki struktur dan konsep yang kompleks bagi pemula, sehingga memerlukan waktu yang lebih untuk mempelajari serta menggunakannya secara efektif dalam mengerjakan proyek.
- Perbedaan tingkat pengalaman dalam tim kerja. Sebagian besar karyawan yang bekerja di PT GMG sudah memiliki pengalaman kerja dalam bidang full-stack development sehingga bahasa yang digunakan terkadang tidak familiar oleh mahasiswa magang, dengan itu diperlukan adaptasi yang lebih cepat serta pemahaman teknis yang lebih dalam.

3.4 Solusi atas Kendala yang Ditemukan

Solusi atas kendala yang ditemukan selama proses magang berlangsung di PT Generasi Muda Gigih:

- Mempelajari tentang bahasa pemrograman dan framework yang kurang familiar diluar jam kerja sehingga membantu mahasiswa magang untuk dapat lebih cepat beradaptasi.
- Bertanya kepada rekan kerja yang lebih berpengalaman sehingga kesulitan yang dihadapi akan lebih cepat teratasi.
- Menggunakan bantuan AI (*ChatGPT* dan *Deepseek*) dalam pengerjaan proyek .



Implementasi dan Pengembangan Frontend, Mikhail Lim Mattagati, Universitas Multimedia Nusantara