BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Saat menjalani kegiatan magang di PT Cranium Royal Aditama, tugas yang dilakukan adalah sebagai *Fullstack Developer* yang memfokuskan pada *frontend* dan *backend*. Selama magang, Bapak Sugito selaku *VP Engineer* bertindak sebagai *Supervisor* langsung yang membimbing dan juga sebagai *Team Leader*. Untuk komunikasi antar supervisi dan anggota menggunakan aplikasi *Whatsapp* dan *Discord*. Platform yang digunakan untuk sharing *project* menggunakan *Github* sebagai *repository*.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama kegiatan magang di PT Cranium Royal Aditama adalah membangun sistem *Enterprise Resource Planning* menggunakan Java Springboot sebagai *back-end* dan *Next.Js* sebagai *framework* untuk *frontend*. Pengerjaan kali ini dimulai dengan pembuatan tampilan awal sebagai templat yang akan digunakan seterusnya berdasarkan desain pada *Figma*, lalu jika tampilan awal tersebut sudah selesai maka akan dilanjutkan pembuatan *CRUD* modul-modul berdasarkan tampilan awal yang sudah dibuat.

Peserta magang diberikan tugas untuk membuat templat halaman *Create* dan *Update*. Komponen global yang banyak digunakan seperti komponen *Modal, Breadcrumb* dan *Button* dibuat dan disesuaikan dengan desain *Figma*. Setelah template halaman untuk *CRUD* telah dibuat, pengerjaan selanjutnya adalah membuat halaman *CRUD* untuk tiap-tiap modul. Berikut modul dan submodul yang dikerjakan dalam pembuatan halaman *CRUD*

- 1. Modul Master
 - (a) Submodul Item Supplier
 - (b) Submodul Salesman
- 2. Modul Pembelian yaitu submodul Purchasing Order Item Receipt
- 3. Modul Produksi yaitu submodul Production Planned Order

3.3 Uraian Pelaksanaan Magang

Berikut adalah uraian pelaksanaan magang yang dilakukan di PT Cranium Royal Aditama.

3.3.1 Pelaksanaan Kerja Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Minggu ke-	Aktivitas yang dikerjakan								
1	Memepelajari dasar dari React dan Next.Js dan mempelajari								
	template yang diberikan								
2	Mencoba menambahkan fitur CRUD untuk menghubungkan ke								
	backend								
3	Membuat halaman create dan update sesuai dengan Figma								
4	Melanjutkan pembuatan halaman create dan update agar sesuai								
	dengan Figma dan membuat komponen Modal								
5	Melanjutkan pembuatan komponen modal agar dapat menjadi								
	Global atau digunakan kembali dan merapihkan styling, mencoba								
	membuat Unit Test								
6	Mulai membuat halaman submodul master Item Supplier dimulai								
	dengan perbaikan pada backend Item Supplier								
7	Melanjutkan pembuatan halaman Item Supplier yaitu halaman								
	create dan update, lalu penambahan fitur searching								
8	Melanjutkan pembuatan halaman dashboard view dan list paging								
	dan mulai membuat unit test untuk halaman Item Supplier								
9	Melanjutkan ke pembuatan submodul master lainnya yaitu								
	Salesman dan perbaikan unit test pada Item Supplier								
10	Melanjutkan pembuatan halaman submodul master Salesman								
11	Mulai membuat halaman modul Purchasing submodul Order								
N	Item Receipt dimulai dengan perbaikan pada backend Order Item								
	Receipt								
	Dilanjutkan di halaman selanjutnya								

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu ke-	Aktivitas yang dikerjakan								
12	Melanjutkan pembuatan halaman CRUD dan implementasi fitur								
	searching pada halaman Order Item Receipt								
13	Melanjutkan pembuatan halaman CRUD Order Item Receipt dan								
	revisi pada komponen global breadcrumb								
14	Melakukan perbaikan pada backend Master Warehouse								
15	Mulai membuat halaman modul Production submodul Planned								
	Order dimulai dengan perbaikan pada backend Planned Order								
16	Melanjutkan pembuatan halaman CRUD Planned Order								
17	Melanjutkan pembuatan halaman CRUD Planned Order								
18	Melakukan penambahan breadcrumb pada submodul yang belum								
	diterapkan								
19	Mulai membuat halaman modul Warehouse submodul Item Receipt								
	dimulai dengan perbaikan pada backend Item Receipt								

Tabel 3.1 – dilanjutkan dari tabel sebelumnya

3.3.2 Sitemap

Sitemap adalah alat pengembangan web yang dapat membantu pengunjung melakukan navigasi situs web dengan lebih mudah. Sistem ERP yang sedang dikembangkan memiliki *sitemap* yang mempunyai halaman *login* merupakan halaman awal yang diakses pertama kali, lalu halaman berikutnya merupakan turunan yang hanya dapat diakses dari halaman tertentu.



Gambar 3.1. Sitemap sistem ERP

A. Sitemap Master Item Supplier

Submodul *Master Item Supplier* memiliki *sitemap* yang diawali dengan halaman *list paged data*. Halaman selanjutnya yang diakses yaitu *dashboard view* data berdasarkan id dan *create form*. Pada halaman *dashboard view* data, halaman yang dapat diakses yaitu *update form*. Berikut merupakan *sitemap* submodul *Master Item Supplier* yang dapat dilihat pada Gambar 3.2.



Gambar 3.2. Sitemap halaman Item Supplier

B. Sitemap Master Salesman

Submodul *Master Salesman* memiliki *sitemap* yang diawali dengan halaman *list paged data*. Halaman selanjutnya yang diakses yaitu *dashboard view* data berdasarkan id dan *create form*. Pada halaman *dashboard view* data, halaman yang dapat diakses yaitu *update form*. Berikut merupakan *sitemap* submodul *Master Salesman* yang dapat dilihat pada Gambar 3.3.

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.3. Sitemap halaman Salesman

C. Sitemap Purchasing Order Item Receipt

Submodul Purchasing Order Item Receipt memiliki *sitemap* yang diawali dengan halaman *list paged data*. Halaman selanjutnya yang diakses yaitu *dashboard view* data berdasarkan id dan *list Purchase Order data*. Pada halaman *dashboard view* data, halaman yang dapat diakses yaitu *update form*. Pada *list Purchase Order data*, halaman yang dapat diakses selanjutnya yaitu *create form*. Berikut merupakan *sitemap* submodul Purchasing Order Item Receipt yang dapat dilihat pada Gambar 3.4.



Gambar 3.4. Sitemap halaman Purchasing Order Item Receipt

D. Sitemap Production Planned Order

Submodul Production Planned Order memiliki *sitemap* yang diawali dengan halaman *list paged data*. Halaman selanjutnya yang diakses yaitu *dashboard view* data berdasarkan id dan *create form*. Pada halaman *dashboard view* data, halaman yang dapat diakses yaitu *update form*. Berikut merupakan *sitemap* submodul Production Planned Order yang dapat dilihat pada Gambar 3.5.



Gambar 3.5. Sitemap halaman Production Planned Order



3.3.3 Flowchart

Flowchart digunakan untuk mengetahui alur kerja dari suatu sistem. Pada sistem ERP yang dibagun, *flowchart* digunakan untuk menjelaskan alur secara garis besar CRUD (*Create, Read, Update, Delete*) modul.

A. Create

Pada alur *create*, dimulai dari halaman *list paging* dari tiap-tiap submodul lalu *user* akan menekan tombol *create*. Sebelum masuk ke dalam halaman *create form*, user akan dicek apakah memiliki otoritas, jika tidak punya maka notifikasi *error* akan keluar. Jika sudah memasuki *create form*, user akan melakukan pengisian data pada *form* tersebut. Setelah data sudah diisi, *user* akan menekan tombol *create* yang memunculkan *modal* konfirmasi. Jika data tidak *valid* seperti ada *textfield* yang perlu diisi namun user tidak mengisinya, maka data tidak bisa di *submit*. Apabila data sudah *valid*, setelah menekan tombol konfirmasi pada modal, *frontend* akan melakukan *hit API*. Data yang berhasil masuk yang berarti *hit API* berhasil, sedangkan jika gagal maka akan mengeluarkan notifikasi *error*. Setelah data berhasil dibuat, halaman akan kembali ke *list paging* data submodul tersebut.







Gambar 3.6 menunjukan bagaimana alur *create* dapat berkerja dari awal hingga akhir. Untuk sebagian kasus, pembuatan *create form* dapat berdasarkan dokumen referensi atau membuat tanpa dokumen referensi. Jika *user* memilih berdasarkan dokumen referensi, halaman akan memunculkan *list paging* dari dokumen referensi. Selanjutnya *user* dapat memilih data yang akan dijadikan referensi dokumen saat membuat data baru

B. Update

Alur *update* cukup mirip dengan alur *create*, dimulai dari halaman *dashboard info* yang berdasarkan *id* dari data submodul lalu *user* akan menekan tombol *update*. Sebelum masuk ke dalam halaman *update form*, user akan dicek apakah memiliki otoritas, jika tidak punya maka notifikasi *error* akan keluar. Jika sudah memasuki *update form*, data akan muncul pada *field* masing-masing dan *user* hanya perlu mengubah data. Setelah data sudah diubah, *user* akan menekan tombol *update* yang memunculkan *modal* konfirmasi. Jika data tidak valid seperti *textfield* yang perlu diisi namun user tidak mengisinya, maka data tidak bisa di *submit*. Apabila data sudah valid, setelah menekan tombol konfirmasi pada modal, *frontend* akan melakukan *hit API*. Data yang berhasil masuk yang berarti *hit API* berhasil, sedangkan jika gagal maka akan mengeluarkan notifikasi *error*. Setelah data berhasil dibuat, halaman akan kembali ke *dashboard* dari submodul tersebut berdasarkan *id* data.





Berikut merupakan alur *update* yang dapat dilihat pada Gambar 3.7 yang menunjukan bagaimana proses alur *update* dari awal hingga akhir.

C. Read

Pada alur *read*, dimulai dari halaman *dasboard* dari tiap-tiap modul lalu *user* akan menekan singkatan dari modul pada *sidebar* selain *dashboard*. Sebelum masuk ke dalam halaman *list paging*, user akan dicek apakah memiliki otoritas, jika tidak punya maka notifikasi *error* akan keluar. Jika sudah memasuki *list paging*, user akan menekan *icon opening file* untuk melihat data berdasarkan *id* yang ada pada *list paging*. *User* akan dicek otoritasnya kembali. Setelah klik berhasil, maka akan muncul halaman *dashboard* berdasarkan *id* data.



Gambar 3.8. Alur read

Berikut merupakan alur *read* yang dapat dilihat pada Gambar 3.8 yang menunjukan bagaimana proses alur *read* dari awal hingga akhir.

D. Delete

Pada alur *delete*, dimulai dari halaman *dashboard info* yang berdasarkan *id* dari data submodul lalu *user* akan menekan tombol *delete*. Selanjutnya akan muncul *modal* sebagai konfirmasi penghapusan data. Jika data berhasil dihapus, maka halaman akan berpindah ke *list paging* dari submodul tersebut. Untuk otoritas delete, tombol *delete* tidak akan ditampilkan jika tidak memiliki otoritas.



Berikut merupakan alur *delete* yang dapat dilihat pada Gambar 3.9 yang menunjukan bagaimana proses alur *delete* dari awal hingga akhir.

3.3.4 Implementasi Halaman

A. Halaman Master Item Supplier

Master Item Supplier Memiliki halaman list paged data, dashboard view berdasarkan id, create form, dan update form.

A.1 Halaman List Paged Data

Halaman *list paged* data *Item Supplier* memiliki komponen *searching*, tombol *create* untuk menuju halaman *create form*, dan tabel berisi data yang memiliki tombol setiap baris untuk menuju halaman *dashboard view* berdasarkan id dari data. Berikut merupakan hasil implementasi dari halaman *list paged data Item Supplier* pada Gambar 3.10.



Gambar 3.10. Tampilan List Paged Data Item Supplier

A.2 Halaman Create

Halaman *create* data *Item Supplier* menampilkan sebuah *form* yang akan diisi oleh *user* untuk membuat data baru. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan dibuat dan akan berpindah ke halaman *list paged* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *list paged* data. Berikut merupakan hasil implementasi dari halaman *create* data *Item Supplier* pada Gambar 3.11.

M U L T I M E D I A N U S A N T A R A

٤	cranium	¢ 2
DSB	Item Supplier Create	
PRD	Supplier* Item*	
STK		
COA	Cancel Save	
РТ		
VHC		
AS		
UOM		
PLT		
AREA		
PRO		
DRV		
	IN DEVELOPMENT	

Gambar 3.11. Tampilan Create Form Item Supplier

A.3 Halaman Dashboard View Data

Halaman *dashboard view* data *Item Supplier* menampilkan informasi lebih lengkap dari data setiap Item Supplier berdasarkan id. Komponen tombol *edit* untuk menuju halaman *update* dari data dan tombol *delete* untuk menghapus data tersebut. Berikut merupakan hasil implementasi dari halaman *dashboard view* data *Item Supplier* pada Gambar 3.12.

\$	cranium	¢ 2
IB	Item Supplier + View	
IP	SKU Suppler Name	
	50002 PT Ayam Keren	
CST	Status	
10		
13	Created At Created By	
ІТМ	2023-11-09 14:18:59 Cranium	
	Updated At Updated By	
IRS	2023°11'03 14,23,22 Galiluit	
AT		
	Edit Cedit Delete)
AT		
ST		
IG		
EFN		
PRT		
	IN DEVELOPMENT	

Gambar 3.12. Tampilan Dashbaord View Data Item Supplier

A.4 Halaman Update

Halaman *update* data *Item Supplier* menampilkan sebuah *form* yang tiap *field* terisi dengan data dari *Item Supplier* berdasarkan id. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan diperbarui dan akan berpindah ke halaman *dashboard view* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *dashboard view* data, sedangkan *view* data berdasarkan id. Berikut merupakan hasil implementasi dari halaman *update* data *Item Supplier* pada Gambar 3.13.

٤	cranium	 ب ٤
IB	Item Supplier → Update	
IP	Supplier* VND0000002 PT Ayam Keren v	•
CST	_ Status	
IS	Active ~	
ΙТΜ		Cancel Save
IRS		
AT		
AT		
ST		
IG		
EFN		
PRT		
вотти	IN DEVELOPMENT	

Gambar 3.13. Tampilan Update Form Item Supplier

B. Halaman Master Salesman

Master Salesman Memiliki halaman list paged data, dashboard view berdasarkan id, create form, dan update form.

B.1 Halaman List Paged Data E R S T A S

Halaman *list paged* data *Salesman* memiliki komponen *searching*, tombol *create* untuk menuju halaman *create form*, dan tabel berisi data yang memiliki tombol setiap baris untuk menuju halaman *dashboard view* berdasarkan id dari data. Berikut merupakan hasil implementasi dari halaman *list paged* data *Salesman* pada Gambar 3.14.



Gambar 3.14. Tampilan List Paged Data Salesman

B.2 Halaman Create

Halaman *create* data *Salesman* menampilkan sebuah *form* yang akan diisi oleh *user* untuk membuat data baru. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan dibuat dan akan berpindah ke halaman *list paged* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *list paged* data. Berikut merupakan hasil implementasi dari halaman *create* data *Salesman* pada Gambar 3.15.

٩					cran	IUM					¢	0°		
DSB	Salesman → Create	Salesman > Create												
PRD	Sales	sman Code *				Plant *				*				
STK	Supe	nicor				Warehouse*				_				
COA	Supe	111301				watehouse				•				
PT	Sales	sman Name *				Province *				*				
VHC	KTP	Number *				CANVAS				-				
UOM	Sales	sman Address *				Salesman Title *				•				
PLT	Phon	ne Number *				Status * Active				-				
AREA														
PRO									Cancel	Save				
DRV														
					IN DEVELO	PMENT								

Gambar 3.15. Tampilan Create Form Salesman

B.3 Halaman Dashboard View Data

Halaman *dashboard view* data *Salesman* menampilkan informasi lebih lengkap dari data setiap *Salesman* berdasarkan id. Komponen tombol *edit* untuk menuju halaman *update* dari data dan tombol *delete* untuk menghapus data tersebut. Berikut merupakan hasil implementasi dari halaman *dashboard view* data *Salesman* pada Gambar 3.16.

¢3		cranium	Ų
Sales	man → View		
PRD	Salesman Code	Chart of Account	
	SLS1231331	1	
тк	Salesman Name	Sales Chart of Account	
	Mamat	1	
AC	KTP Number	Sales Return Chart of Account	
	00038497123	1	
т	Salesman Address	Price Tag	
	JI babat 3	1	
IC	Phone Number	Warehouse	
	0812344123112	Warehouse Bonang 1	
AS	Plant	Salesman Title	
OM	Plant 1	SUPERVISOR	
om .	Province		
PLT	Nusantara	Supervisor	
	Salesman Type	Status	
REA	CANVAS	ACTIVE	
PRO	Created At	Created By	
	2023-11-18 17:27:34	Cranium	
DRV	Updated At	Updated By	
	2023-11-18 17:27:34	Cranium	

Gambar 3.16. Tampilan Dashboard View Data Salesman

B.4 Halaman Update

Halaman *update* data *Salesman* menampilkan sebuah *form* yang tiap *field* terisi dengan data dari *Salesman* berdasarkan id. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan diperbarui dan akan berpindah ke halaman *dashboard view* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *view* data berdasarkan id. Berikut merupakan hasil implementasi dari halaman *update* data *Salesman* pada Gambar 3.17.

M U L T I M E D I A N U S A N T A R A

ŵ	Cr	anium	<u>۾</u> ڪ
DSB	Salesman → Update		
PRD	C Salesman Code *	Plant*	
0714	SLS1231331	Plant 1	·
SIK		Warehouse *	
COA	Supervisor	 Warehouse Bonang 1 	-
	- Salesman Name *	Province *	
PT	Mamat	Nusantara	*
VHC	KTP Number *	Salesman Type *	
	00038497123	CANVAS	*
AS	Salesman Address *	Salesman Title *	
UOM	JI babat 3	Supervisor	Ŧ
	Phone Number *	Status*	
PLT	0812344123112	Active	*
AREA			
PRO		Cancel	Save
PRO			
DRV			
~			
	IN DE	EVELOPMENT	

Gambar 3.17. Tampilan Update Form Salesman

C. Halaman Purchasing Order Item Receipt

Purchasing Order Item Receipt Memiliki halaman list paged data, list purchase order reference, dashboard view berdasarkan id, create form, dan update form.

C.1 Halaman List Paged Data

Halaman *list paged* data *Order Item Receipt* memiliki komponen *searching*, tombol *create* untuk menuju halaman *create form*, dan tabel berisi data yang memiliki tombol setiap baris untuk menuju halaman *dashboard view* berdasarkan id dari data. Berikut merupakan hasil implementasi dari halaman *list paged* data *Order Item Receipt* pada Gambar 3.18.

UNIVERSITAS MULTIMEDIA NUSANTARA

۵						cra	nium		¢	8
DSB	Purchasing	• Order	Item Receipt							
PO								() + Create		
POR										
POIR	Docum No	^{ent} ≎	Creation Otate	Supplier Name	Plant Name	Open	Status 🗘			
РВ		101	2023-11-07	PT Ayam Keren	Plant 1	true	NEW			
PRB		101	2023-10-27	PT Ayam Keren	Plant 1	true	APPROVED_CA			
PR		Rows p	er Page: 10 💌					Page: 1 (PREV NEXT)		
						IN DEV	ELOPMENT			

Gambar 3.18. Tampilan List Paged Data Order Item Receipt

C.2 Halaman List Create by Order Paged Data

Halaman *list create by purchase order paged* data memiliki komponen *searching*, tombol *cancel* untuk menuju kembali ke halaman *list paged*, dan tabel berisi data yang memiliki tombol setiap baris untuk menuju halaman *create* berdasarkan id dari data *Purchase Order*. Berikut merupakan hasil implementasi dari halaman *list create by purchase order paged* data *Order Item Receipt* pada Gambar 3.19.

۵						cranium			ļ s
DSB	Purch	asing → Order Item	n Receipt 🔸 Crea	te By Order					
РО								٩	× Cancel
POR	_								
POIR		Document No 🗘	Creation Date	Plant Name	Supplier Document Reference No	Supplier Code	Supplier Name	Receipt Quantity Quantity	9 Status 🗘
РВ	D,	P0D000002	2023-10-27	Plant 1	testing	VND0000002	PT Ayam Keren		APPROVED
PRB	C,	P0D000003	2023-10-27	Plant 1	testing	VND0000002	PT Ayam Keren		APPROVED
PR	C,	POD000001	2023-10-27	Plant 1	testing	VND0000002	PT Ayam Keren		APPROVED
		Rows per Pa	age: 10 💌					Page: 1	NEXT >
~									
						IN DEVELOPMENT			

Gambar 3.19. Tampilan List Purchase Order Paged Data

C.3 Halaman Create

Halaman *create* data *Order Item Receipt* menampilkan sebuah *form* yang beberapa *field* terisi dengan data dari Purchase Order berdasarkan id. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan dibuat dan akan berpindah ke halaman *list paged* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *list paged* data. Berikut merupakan hasil implementasi dari halaman *create* data *Order Item Receipt* pada Gambar 3.20.

۵	cranium	¢ ٤										
DSB	Purchasing											
PO	Purchase Order Document No * Plant Name * POD0000001 Plant 1 v											
POR	Supplier Name *											
POIR	Document No * VND0000002 PT Ayam Keren v											
РВ	Creation Date * Supplier Address * dd/mm/yyyy D Supplier 1 Address											
PRB												
PR												
	Details											
	Sku Item Name Quantity Uom Name Warehouse Remarks											
	50001 Item Name 001 I0 UOM 1 ▼ test											
	50002 Item Name 002 10 UOM 1 ▼ test2											
	50004 Item Name 004 I0 UOM 3 ▼ test3											
*	+ Add new item Delete selected item											

Gambar 3.20. Tampilan Create Form Order Item Receipt

C.4 Halaman Dashboard View Data

Halaman *dashboard view* data *Order Item Receipt* menampilkan informasi lebih lengkap dari data setiap *Order Item Receipt* berdasarkan id. Komponen tombol *edit* untuk menuju halaman *update* dari data dan tombol *delete* untuk menghapus data tersebut. Lalu komponen *field* status untuk mengubah status dari dokumen tersebut. Berikut merupakan hasil implementasi dari halaman *dashboard view* data *Order Item Receipt* pada Gambar 3.21.

USANTAR

۵						crar	NUN	N		Ļ	. 6		
DSB	Purchasing →	Purchasing Order Item Receipt											
PO		Document No POIR11101					Plant Na Plant 1	me					
POR		Purchase Order Docu POD000001	iment No				Supplier	Name 000002 PT Ayam Keren					
POIR		Creation Date 2023-11-07					Supplier Supplie	Address er 1 Address					
PB PRB		Created By Cranium					Created	At 1-07 14:18:08					
PR		Updated By Cranium					Updated 2023-1	At 1-07 14:18:08					
		Approved By					Approv	ved At					
		Approval Canceled B	y				Approv	val Canceled At					
		Closed By					Closed	At					
		Opened By					Opene	d At					
A		Void By					Void A	t					
		DETAILS	<u>}</u>										
		Item Name	Quantity	Uom Name	Warehouse	Remarks		Batch No					
		Item Name 001	10	UOM 1	Warehouse Bonang 4	test		1					
		Item Name 002	10	UOM 1	Warehouse Bonang 3	test2		1					
		Item Name 004	10	UOM 3	Warehouse Bonang 1	test3		1					
		Status NEW							Delete Edit Print				
						IN DEVEL	OPMEN	т					

Gambar 3.21. Tampilan Dashboard View Data Order Item Receipt

C.5 Halaman Update

Halaman *update* data Order Item Receipt menampilkan sebuah *form* yang tiap *field* terisi dengan data dari Salesman berdasarkan id. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan diperbarui dan akan berpindah ke halaman *dashboard view* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *dashboard view* data, sedangkan *view* data berdasarkan id. Berikut merupakan hasil implementasi dari halaman *update* data *Order Item Receipt* pada Gambar 3.22.

M U L T I M E D I A N U S A N T A R A

۵		cranium								
DSB	Purchasing + Order Item Receipt + Update Order Item Receipt									
РО		Purchase Order Document I POD000001	lo *		F	Plant Name *				
POR		- Document No *				Supplier Name *				
POIR		POIR11101				VND0000002 PT Ayam Keren *				
РВ		07/11/2023		<u></u>		Supplier 1 Address				
PRB										
PR										
	Deta	ails								
	Sku Sku	Item Name	Quantity Uom Na	me Warehouse		Remarks				
	50001	Item Name 001	10 UOM	1 • Warehous	*	test				
	50002	Item Name 002	10 UOM	1 • Warehous	Ŧ	test2				
	50004	Item Name 004	10 UOM	3 💌 Warehous	*	test3				
	Delete selected Item									
						Cancel Save				
	IN DEVELOPMENT									

Gambar 3.22. Tampilan Update Form Order Item Receipt

D. Halaman Production Planned Order

Production Planned Order memiliki halaman list paged data, dashboard view berdasarkan id, create form, dan update form.

D.1 Halaman List Paged Data

Halaman *list paged* data *Planned Order* memiliki komponen *searching*, tombol *create* untuk menuju halaman *create form*, dan tabel berisi data yang memiliki tombol setiap baris untuk menuju halaman *dashboard view* berdasarkan id dari data. Berikut merupakan hasil implementasi dari halaman *list paged* data *Planned Order* pada Gambar 3.23.

UNIVERSITAS MULTIMEDIA NUSANTARA

ę)						cranium		¢ 2			
DSB	Production Planned Order										
PO							Q + Create				
POIR											
РРО		Document Number	Creation Otate	Plant Name	Status 🗘						
	D,	PR0123113	2023-11-06	Plant 2	NEW						
	D,	123qaad	2023-11-06	Plant 2	NEW						
	D,	PP01212	2023-11-03	Plant 2	NEW						
	D,	роророр	1211-12-12	Plant 3	APPROVED						
	D,	pspskpsk	1111-11-11	Plant 4	NEW						
	D,	PPP0101	2023-11-01	Plant 2	NEW						
	D,	ITR0000001	2023-03-28	Plant 3	APPROVED						
~		Rows p	er Page: 10 💌				Page: 1				
~						IN DEVELOPMENT					

Gambar 3.23. Tampilan List Paged Data Planned Order

D.2 Halaman Create

Halaman *create* data *Planned Order* menampilkan sebuah *form* yang akan diisi oleh *user* untuk membuat data baru. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan dibuat dan akan berpindah ke halaman *list paged* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *list paged* data. Berikut merupakan hasil implementasi dari halaman *create* data *Planned Order* pada Gambar 3.24.

ę	cranium									
DSB	Production Planned Order									
PO	Document Number*									
POIR	Creation Date *									
PPO	aa/mm/yyyy									
	Details									
	SKU Item Name Quantity Unit									
	+ Add new item Delete selected item									
	Cancel Save									
~										
	IN DEVELOPMENT									

Gambar 3.24. Tampilan Create Form *Planned Order*

D.3 Halaman Dashboard View Data

Halaman *dashboard view* data *Planned Order* menampilkan informasi lebih lengkap dari data setiap *Planned Order* berdasarkan id. Komponen tombol *edit* untuk menuju halaman *update* dari data dan tombol *delete* untuk menghapus data tersebut. Lalu komponen *field* status untuk mengubah status dari dokumen tersebut. Berikut merupakan hasil implementasi dari halaman *dashboard view* data *Planned Order* pada Gambar 3.25.

ę					crar	num		Ļ	ß
DSB	Production	→ Planned Order → Vi	ew						
РО		Document Number ITR0000001				Creation Date 2023-03-28			
POIR		Plant Name Plant 3							
PPO		Created By Cranium				Created At 2023-11-01 13:49:09			
		Updated By Cranium				Updated At 2023-11-02 14:03:34			
		Approved By							
						Approved At			
		Approval Canceled By				Approval Canceled At			
		Closed By							
		Opened By				Closed At			
		-,				Opened At			
		Void By				Void At			
	DE	TAILS							
	SKU	Item Name Q	uantity Uni	t					
	50001	Item Name 001 1	00 UO	м 1					
	50002	Item Name 002 5	0 UO	M 2					
		Status APPROVE	•				Delete Edit Print	- 	
					IN DEVE	LOPMENT			

Gambar 3.25. Tampilan Dashboard View Data *Planned Order*

D.4 Halaman Update

Halaman *update* data *Planned Order* menampilkan sebuah *form* yang tiap *field* terisi dengan data dari Salesman berdasarkan id. Jika tombol *save* ditekan maka komponen *modal* akan muncul sebagai konfirmasi kepada *user* apakah data akan diperbarui dan akan berpindah ke halaman *dashboard view* data, sedangkan jika tombol *cancel* ditekan maka halaman akan berpindah ke halaman *dashboard view* data berdasarkan id. Berikut merupakan hasil implementasi dari halaman *update* data *Planned Order* pada Gambar 3.26.

÷.	cranium									
DSB	Production Planned Order Update									
PO	Document Number* Plant Name* ITR0000001 Plant 3									
POIR	Creation Date* 28/03/2023									
	Details									
	SKU Item Name Quantity Unit									
	50001 • Item Name 001 100 U0M 1 •									
	50002 • Item Name 002 50 UOM 2 •									
	+ Add new item Delete selected item									
	Cancel Save									
	IN DEVELOPMENT									

Gambar 3.26. Tampilan Update Form *Planned Order*

3.3.5 Implementasi Unit Testing

Pembuatan *unit testing* dilakukan dengan tujuan bahwa program dapat berjalan sesuai yang diharapkan. Pengujian unit biasanya dilakukan secara otomatis dan fokus pada bagian-bagian terkecil dari kode, seperti fungsi atau metode, untuk memverifikasi kebenarannya. *Framework* yang digunakan pada *unit testing* adalah *Jest*. Pada sistem ERP ini, *unit testing* dibuat dengan menyesuaikan alur kerja berdasarkan *CRUD* yang sudah dibuat. Setiap submodul yang sudah selesai dilakukan implementasi, *unit test* akan dibuat untuk memastikan apakah setiap submodul sudah berjalan sesuai dengan alur kerja sistem.

Pada *list paging* dan *dashboard view* berdasarkan id, *unit testing* dibuat untuk memastikan data dapat tampil pada halaman. Bagian *create* dan *update* memastikan bahwa data dapat dikirim dengan *API* yang sesuai dan berhasil memunculkan notifikasi bahwa data berhasil dibuat. Pada bagian *delete*, *unit test* diawali dengan menampilkan *dashboard view* yang memiliki komponen tombol *delete*. Setelah tombol delete ditekan, *API* akan dipanggil dan akan menampilkan notifikasi jika data berhasil dihapus.

3.4 Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan pada saat pelaksanaan magang yaitu:

 Beberapa komponen yang membutuhkan waktu yang cukup lama pengerjaan seperti *field autocomplete* dikarenakan kompleksitas yang cukup tinggi.

- Mempelajari *framework Next.Js* seperti *file-system based router* yang dimiliki oleh *Next.js* untuk mendukung pengembangan sistem ERP ini karena belum pernah digunakan sebelumnya.
- Banyaknya perubahan pada desain *figma* dikarenakan *project* yang dikerjakan masih dalam tahap awal pengembangan.

Solusi yang ditemukan pada saat pelaksanaan magang yaitu:

- Mencari referensi terkait komponen yang sulit diimplementasikan.
- Mencari tutorial dan membaca dokumentasi dari Next.Js
- Rutin melaksanakan rapat dengan supervisi, desainer sehingga proses pekerjaan dapat terkontrol dengan baik.

