

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Kedudukan saat pelaksanaan kerja magang di PT Central Proteina Tbk adalah sebagai *Frontend Developer Intern* yang tergabung dalam divisi *ICT (Information and Communication Technology)*. Pada saat pelaksanaan magang, seluruh tugas dan kegiatan diawasi oleh *Lead Developer* sebagai supervisor. Supervisor memberikan tugas serta arahan dan juga solusi terkait masalah yang dialami.

3.2 Tugas yang Dilakukan

Sebagai *Frontend Developer Intern*, tugas yang diberikan yaitu melakukan pengembangan fitur dan revisi dari permintaan *System Analyst*. Tugas yang dikerjakan adalah sebagai berikut.

1. Memahami semua informasi terkait jasa dan produk yang ada di perusahaan.
2. Mempelajari *Framework Vue.js*, dan *Laravel*.
3. Melakukan pengembangan fitur sesuai dengan permintaan *user*.
4. Membuat aplikasi sesuai dengan desain antarmuka yang diberikan.
5. Melakukan revisi sistem yang sedang dikembangkan sesuai dengan permintaan *System Analyst*.
6. Melakukan diskusi bersama dengan *Backend Developer Intern* terkait permasalahan dan solusi.

3.3 Uraian Pelaksanaan Magang

Tugas dan tanggung jawab kegiatan magang yang dilakukan akan dijelaskan tugas yang harus diselesaikan setiap minggunya pada Tabel 3.1.

Tabel 3.1. Uraian tugas yang dikerjakan

Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan perusahaan dan pemberian tugas awal melakukan revisi pada sistem PPBJS.
2	Melaksanakan penambahan fitur kecil sesuai dengan yang diberikan oleh supervisor dan melakukan intergrasi dengan backend.
3	Menunggu task selanjutnya dan mendapatkan <i>challenge</i> membuat <i>canvas form</i> .
4	Melakukan cloning project Rhea dan melakukan UAT (<i>User Acceptance Testing</i>) pada Snap BI.
5	<i>Explore</i> aplikasi Rhea Sales Activity dan melanjutkan UAT pada Snap BI.
6	Mempelajari <i>router</i> di <i>Vue.js</i> dan menghubungkan antara PPBJS dan LP3ET, serta menunggu tim <i>UI(User Interface)</i> .
7	Melakukan <i>slicing</i> dari figma dan mengerjakan revisi dari <i>Analyst</i> terkait sistem PPBJS.
8	Mengerjakan perubahan sistem kerja PPBJS dan <i>update</i> di <i>Open Project</i> .
9	Mengerjakan revisi tampilan PPBJS dan <i>update</i> di mendapatkan tugas <i>redirect token</i> untuk sistem <i>login</i> .
10	Mengerjakan perubahan sistem kerja kondisi pembuatan LP3ET dan mulai mengerjakan <i>redirect token</i> untuk sistem <i>login</i> .
11	Menambahkan validasi pada LP3ET dan melanjutkan mengerjakan <i>redirect token</i> .
12	Mulai mengerjakan aplikasi Rhea Sales Activity modul Sales Call Plan.
Lanjut pada halaman berikutnya	

Tabel 3.1. - Uraian tugas yang dikerjakan (Lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
13	Menampilkan data dari <i>API</i> (<i>Application Programming Interface</i>) sesuai <i>mapping</i> .
14	Implementasi <i>Submit activity</i> dan <i>show activity detail</i> sesuai dengan data <i>mapping</i> oleh <i>API</i> .
15	Mengerjakan revisi dan <i>debug</i> dari sistem PPBJS dan LP3ET.
16	Mengerjakan revisi LP3ET menjadi <i>multiple</i> dan mengerjakan penambahan alur dan <i>input</i> di aplikasi Rhea Sales Activity .
17	Mengerjakan <i>progress ring</i> pada dan revisi <i>bug</i> pada aplikasi Rhea Sales Activity .
18	Menambahkan penanggalan pada proses ditunda LP3ET dan menambahkan <i>flow</i> diubah pada PPBJS.

3.4 Detail Pelaksanaan Magang

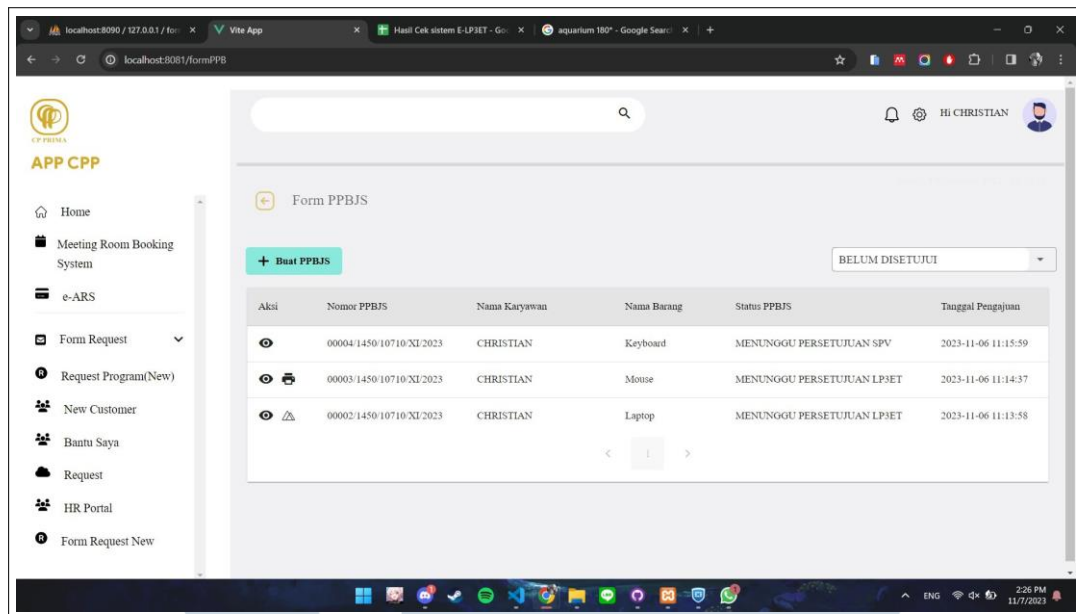
Detail tugas yang dikerjakan saat magang dan hasil dari pekerjaan yang dikerjakan di bagian *Front End*.

3.4.1 Pengembangan Sistem PPBJS dan LP3ET

Sistem PPBJS dan LP3ET merupakan sistem perkantoran untuk melakukan semua kegiatan pekerjaan kantor. Salah satunya adalah pengadaan barang dan jasa, Sistem ini sudah ada sebelumnya namun sekarang masih dikembangkan dan memiliki banyak *bug* dan perlu diperbaiki sesuai dengan *System Analyst*.

1. Revisi Tampilan *Website* Sesuai *Request*

Melakukan Revisi tampilan sesuai revisi bertahap dari *Analyst*. Revisi berupa hal yang sederhana seperti letak komponen, ukuran tulisan, warna, dan posisi.



Gambar 3.1. Tampilan PPBJS

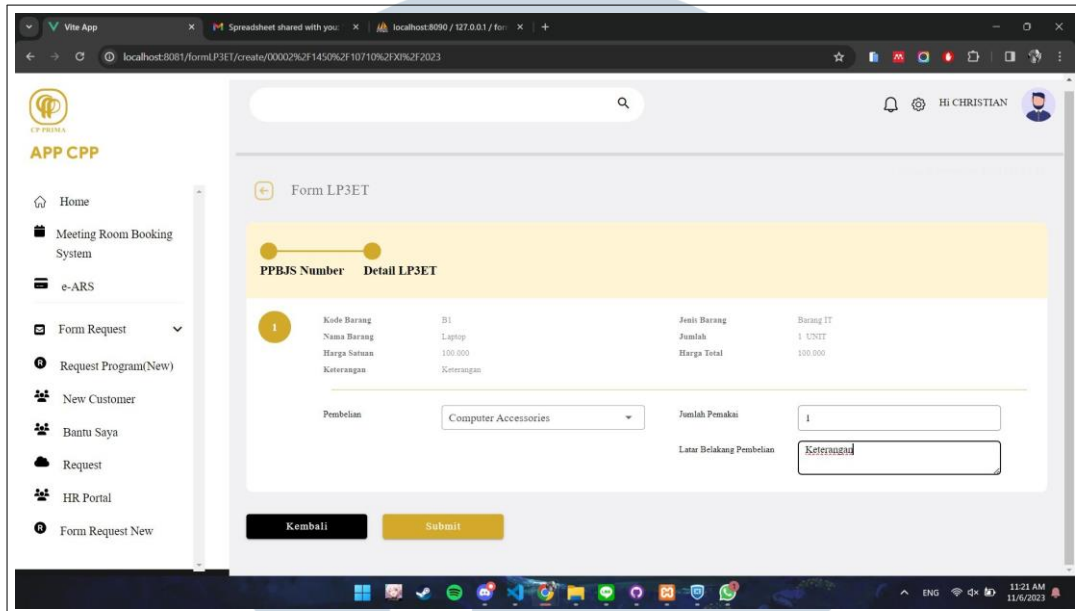
Gambar 3.1 merupakan hasil implementasi dari revisi tampilan pada sistem PPBJS, revisi yang dimaksud berupa revisi yang sangat simpel. Revisi pada sistem berupa letak sebuah komponen, ukuran tulisan, warna komponen, posisi, dan juga penyetaraan bahasa.

2. Menghubungkan PPBJS dan LP3ET

Menghubungkan PPBJS dan LP3ET tanpa harus pindah menu untuk melihat LP3ET yang dibuat berdasarkan Nomor PPBJS, Dapat langsung melihat dari menu PPBJS.

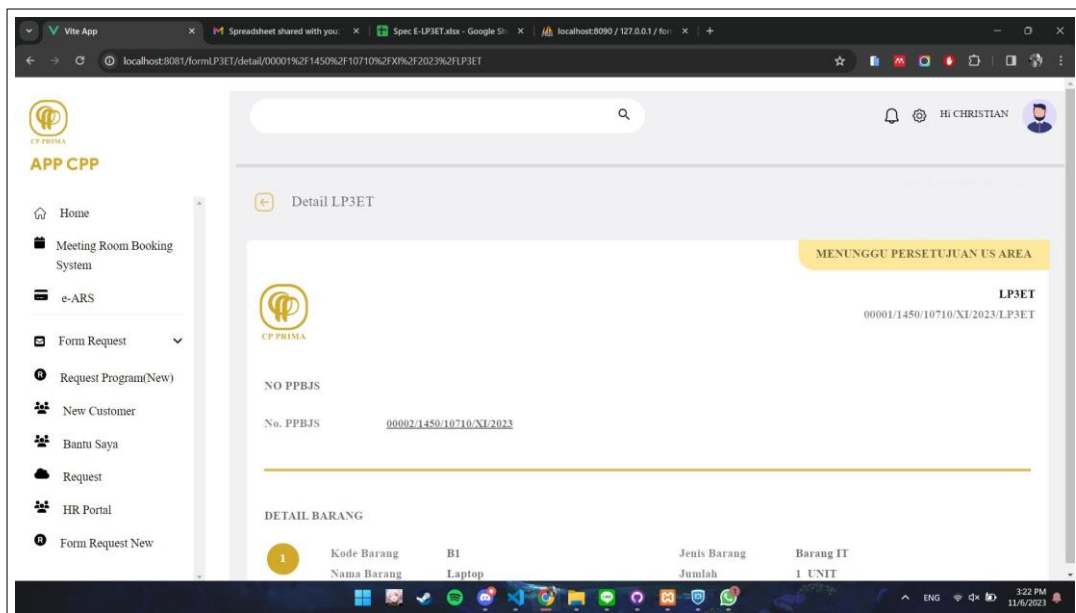
UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tampilan ini muncul ketika *user* memilih *icon* mata. Pada tampilan ini *user* dapat melakukan persetujuan PPBJS.



Gambar 3.4. Tampilan Membuat LP3ET

Gambar 3.4 merupakan tampilan ketika *user* memilih *icon printer* dan dapat membuat LP3ET dari nomor PPBJS.

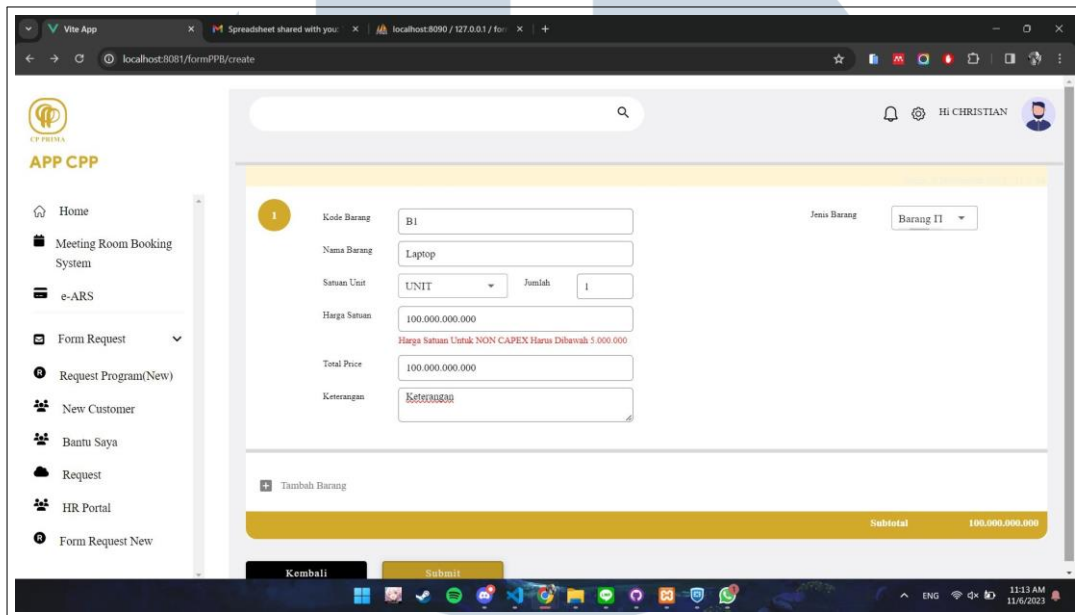


Gambar 3.5. Tampilan Detail LP3ET

Gambar 3.5 merupakan tampilan ketika *user* memilih *icon segitiga* dan dapat melihat detail dari LP3ET yang telah dibuat.

3. Perubahan Validator Field Input PPBJS

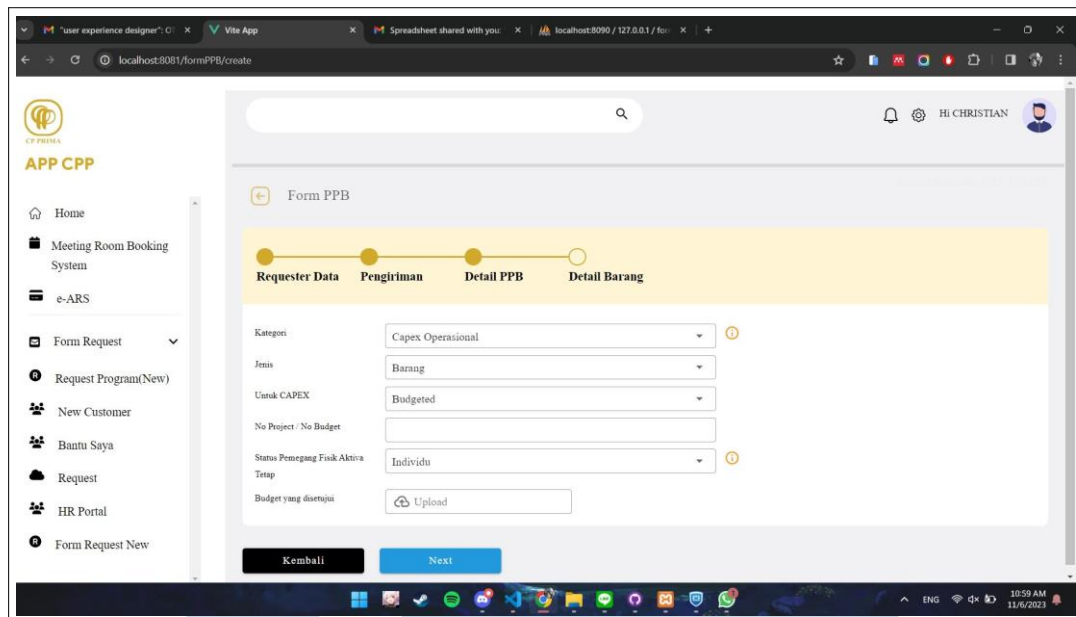
Menambahkan beberapa *validator* pada *input field* pembuatan PPBJS dan menghilangkan beberapa *input field* apabila kategori tertentu yang dipilih berdasarkan kode dari kategori tersebut.



Gambar 3.6. Validator

Gambar 3.6 merupakan salah satu tampilan dari implementasi validasi. Sistem memiliki beberapa aturan untuk *input* tertentu. Semua validasi yang memiliki cara kerja seperti ini memiliki logika yang sama dimana sistem melakukan validasi berdasarkan data yang di *input*.

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



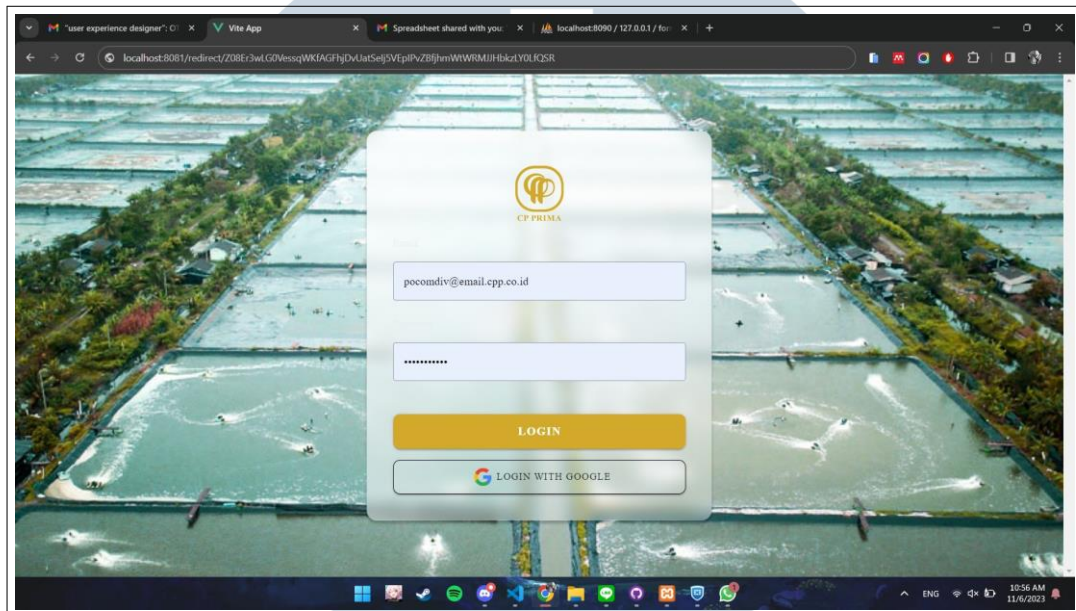
Gambar 3.7. Input Validator

Gambar 3.7 merupakan salah satu tampilan dari implementasi *input*. Sistem menampilkan *input field* untuk pilihan tertentu. Pada contoh ini *input field* untuk gambar akan muncul apabila kategori yang dipilih adalah *budgeted* dan tidak akan muncul apabila *unbudgeted*. Implementasi logika pada sistem ini menggunakan kode tertentu bagi setiap kategorinya dan menyembunyikan komponen berdasarkan kode kategori yang dipilih.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

4. Redirect Token

Redirect Token adalah fitur tambahan untuk *login*. *User* dapat masuk dari aplikasi lain tanpa melakukan *login* terlebih dahulu.



Gambar 3.8. Redirect

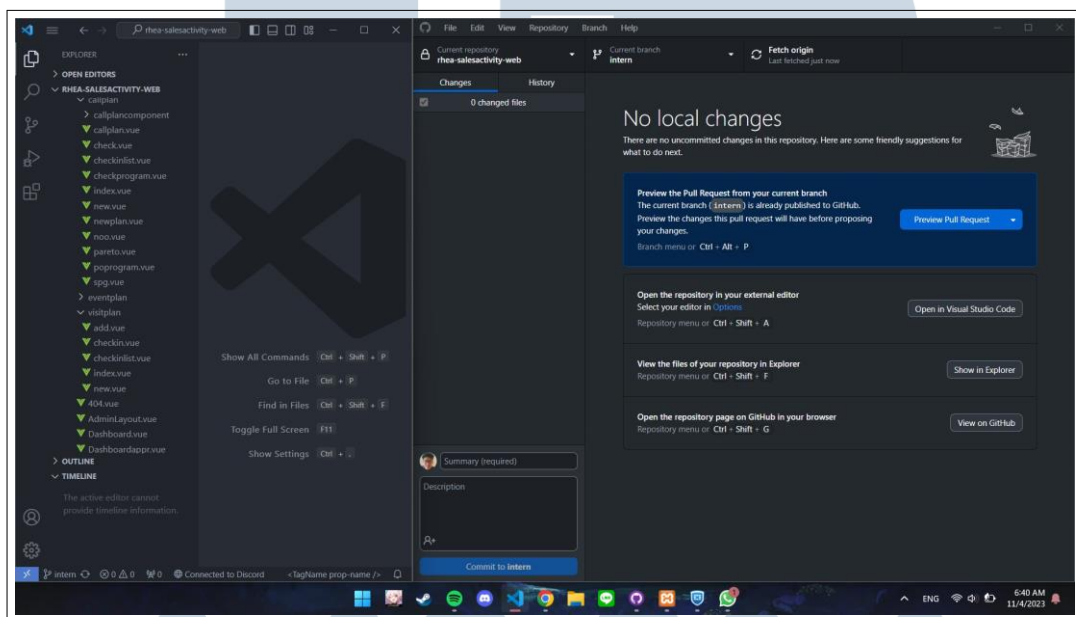
Gambar 3.8 merupakan sebuah simulasi, ketika *user* melakukan *redirect* dari aplikasi kantor yang lain ke aplikasi ini. *Token* akan ada di *URL (Uniform Resource Locator) website* yang kemudian akan divalidasi oleh sistem. Jika *user* yang melakukan *redirect* memiliki *token* yang sama dengan yang ada pada *database* maka *user* berhasil *login*.

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

3.4.2 Pengembangan Sistem Rhea Sales Activity Module

Aplikasi *Rhea Sales Activity* merupakan sebuah aplikasi yang digunakan oleh *sales* dari perusahaan Rhea dimana aplikasi tersebut mencatat kegiatan dari *sales* seperti aktivitas harian dan aktivitas acara. Terdapat penambahan modul baru dimana *sales* dapat mencatat *activity* secara mingguan.

1. Membaca kode pada aplikasi sebelum memulai

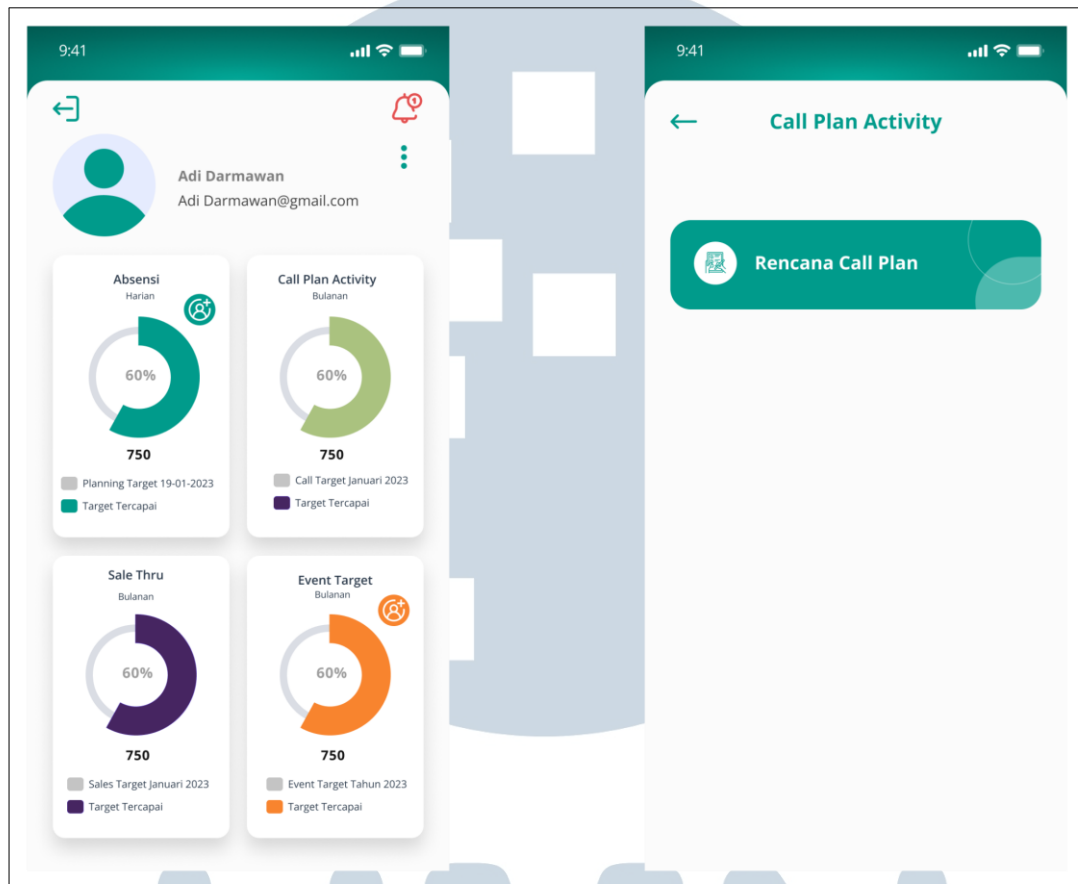


Gambar 3.9. Mempelajari Aplikasi Rhea Sales Activity

Gambar 3.9 merupakan *screenshot* dari *file github rhea-sales-activity* dimana sudah *diclone* ke *device* dan mendapatkan tugas untuk melakukan eksplorasi dari aplikasi sebelum memulai pengerjaan.

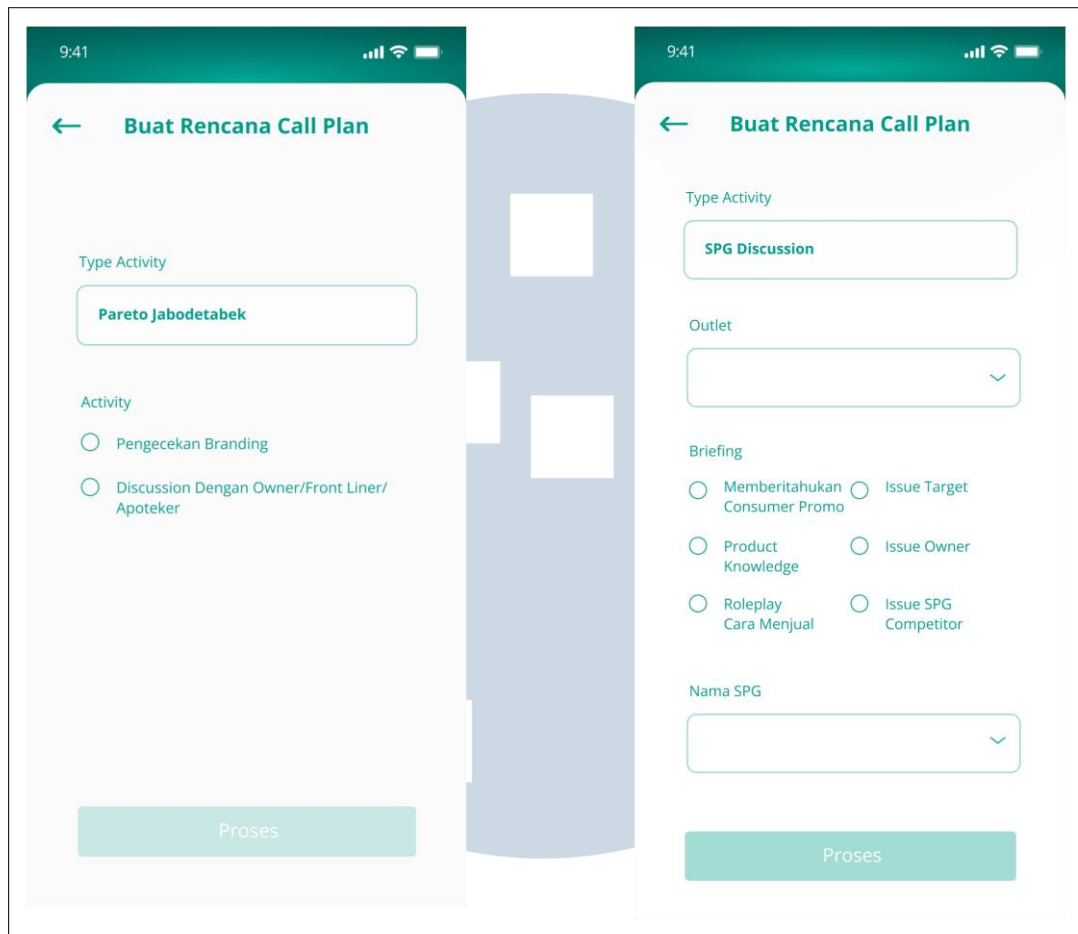
UNIVERSITAS
MULTIMEDIA
NUSANTARA

2. Melakukan *slicing* Figma dari *team UI/UX* menjadi halaman menggunakan *Vue.js*



Gambar 3.10. Figma Halaman Awal

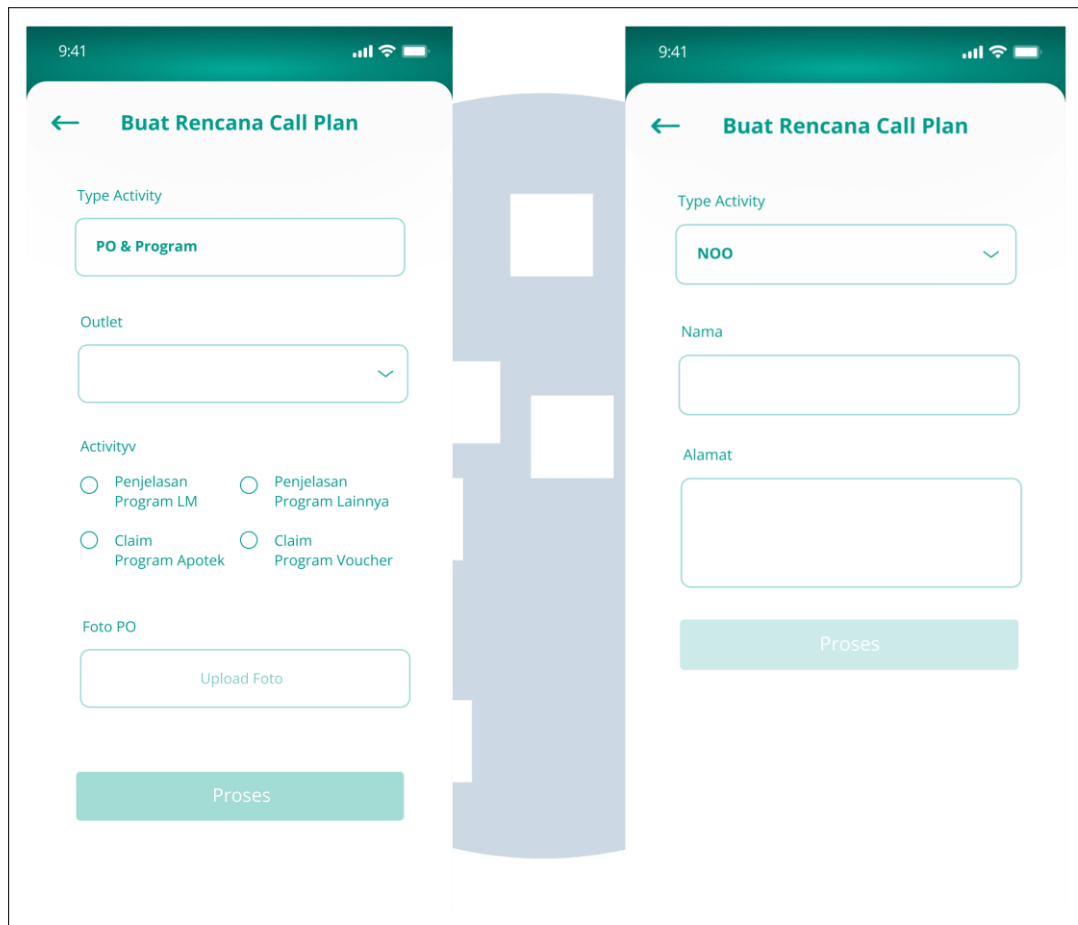
Gambar 3.10. merupakan desain antar muka dari halaman awal saat pertama kali masuk ke aplikasi dan ingin membuat rencana *call plan*. Pada tampilan awal terdapat informasi tentang pengguna berupa nama dan juga alamat *email* dan persentase dari *call plan* yang telah dilakukan. Sedangkan di tampilan pembuatan rencana *call plan* hanya terdapat *button* yang akan menghubungkan ke halaman berikutnya.



Gambar 3.11. Figma pareto dan SPG

Gambar 3.11. merupakan desain antar muka dari halaman input pareto dan SPG. Pada halaman pareto hanya meminta memilih *activity* yang akan dilakukan dan pada halaman *SPG* diminta memilih *outlet*, *activity*, dan *SPG*. Dimana semua datanya akan didapatkan dari *mapping*.

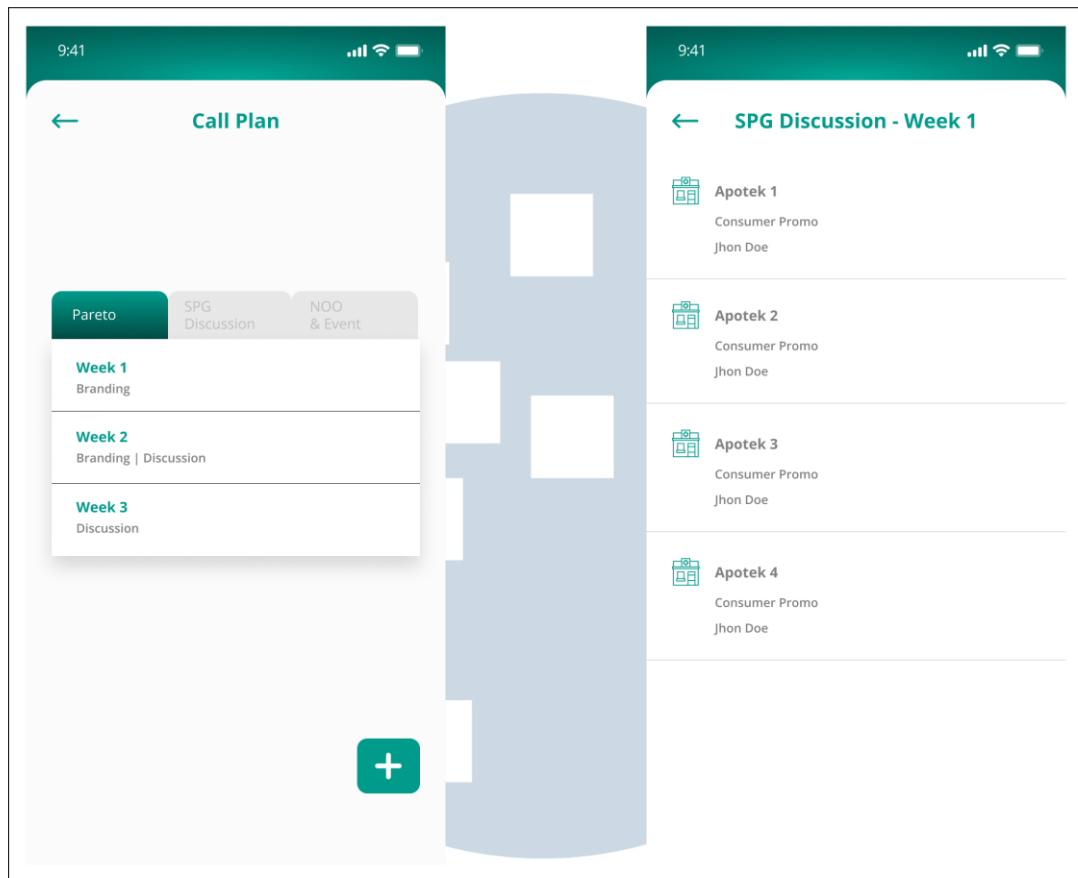
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.12. Figma PO dan Noo/Event

Gambar 3.12. merupakan desain antar muka dari halaman *input PO* dan *NOO/Event*. Pada halaman *PO* diminta memilih *outlet*, *activity*, dan foto sedangkan pada halaman *NOO/Event* akan diminta berupa teks nama *outlet* atau *event* dan alamat *oulet/event*. Semua data akan didapatkan dari *mapping*.

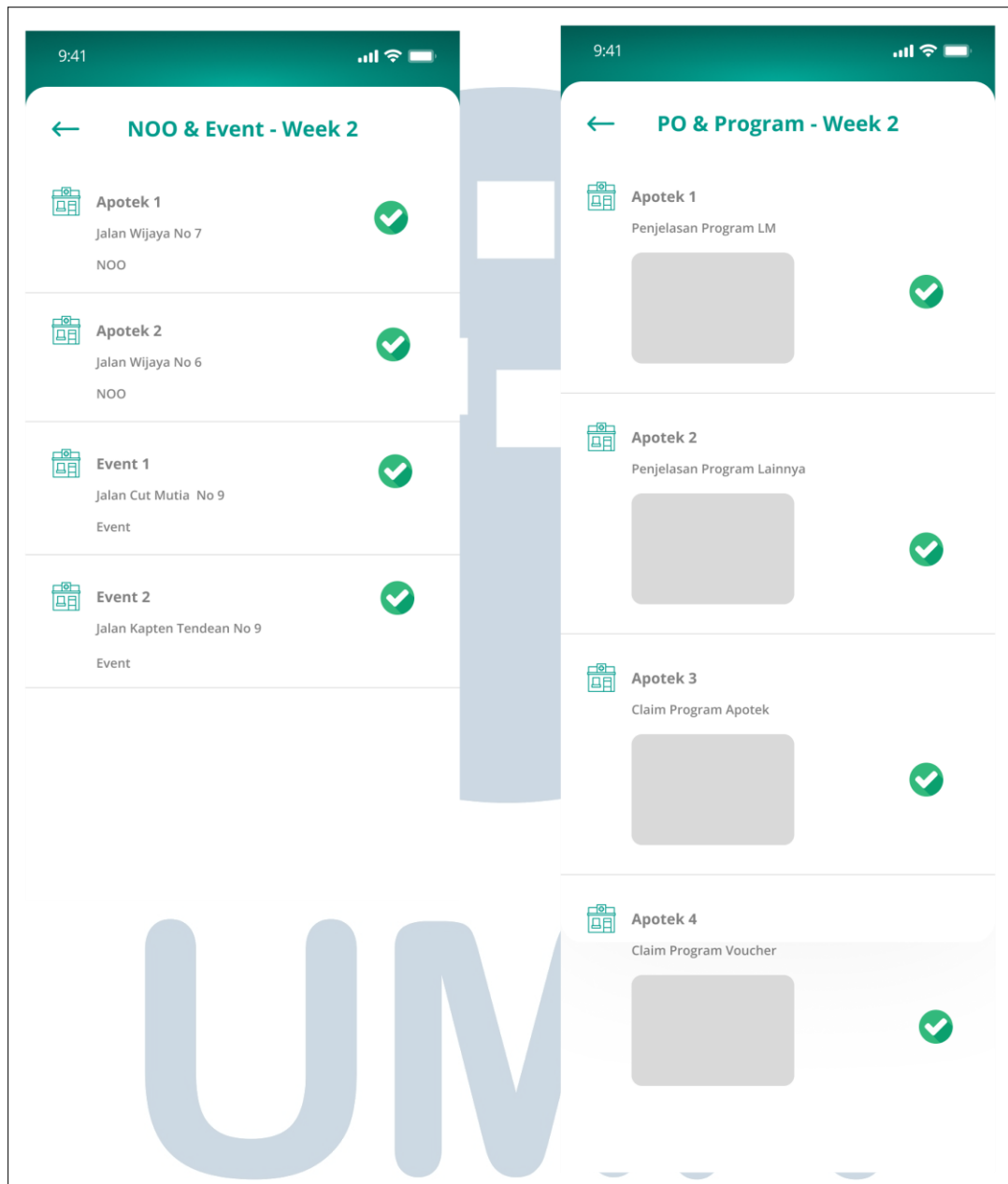
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.13. Figma Detail tab dan SPG Detail

Gambar 3.13. merupakan desain antar muka dari halaman detail *tab* dan *SPG detail*. *Detail tab* merupakan halaman yang berisikan *sales activity* yang sudah di *input* pada minggu ini dan hanya akan muncul apabila semua *activity* telah selesai. Pada *SPG detail* menampilkan *detail* yang dipilih saat *input SPG* yaitu nama *outlet*, *activity* yang dipilih, dan nama *SPG*.

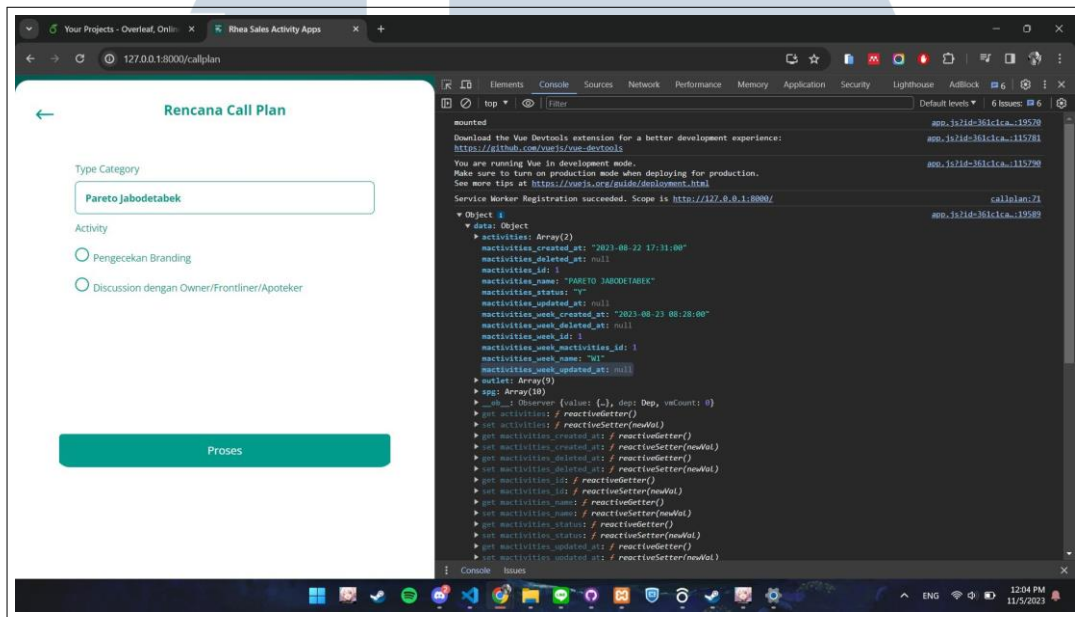
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



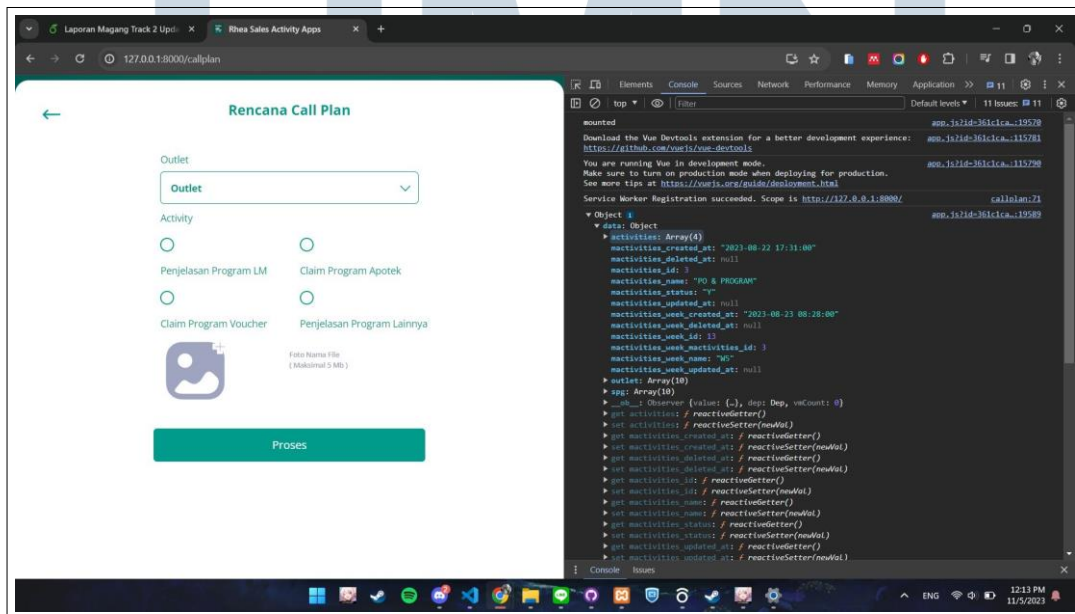
Gambar 3.14. Figma Noo Detail dan Po detail

Gambar 3.14. merupakan desain antar muka dari halaman detail *NOO/Event* dan *PO detail*. Pada *NOO/Event detail* menampilkan detail data yang sudah di *input* pada halaman *NOO/Event* yaitu nama *outlet/event*, alamat *outlet/event*, dan nama kegiatan, Sedangkan pada *PO detail* menampilkan data yang di *input* pada halaman *PO* yaitu nama *outlet*, *activity* yang dipilih, dan gambar yang diunggah.

3. Pada Gambar 3.15. dan Gambar 3.16. merupakan implementasi dari *mapping* komponen sesuai dari data yang dikirim oleh *API* dan *database*, dimana komponen *sales activity* yang ditampilkan sesuai aktivitas mingguan. Aktivitas ini memiliki tampilan yang berbeda yaitu Pareto, *SPG*, *PO*, dan *Noo/Event*. Dimana setiap *sales activity* ini memiliki mekanismya sendiri yang mengharuskan *input* yang berbeda.

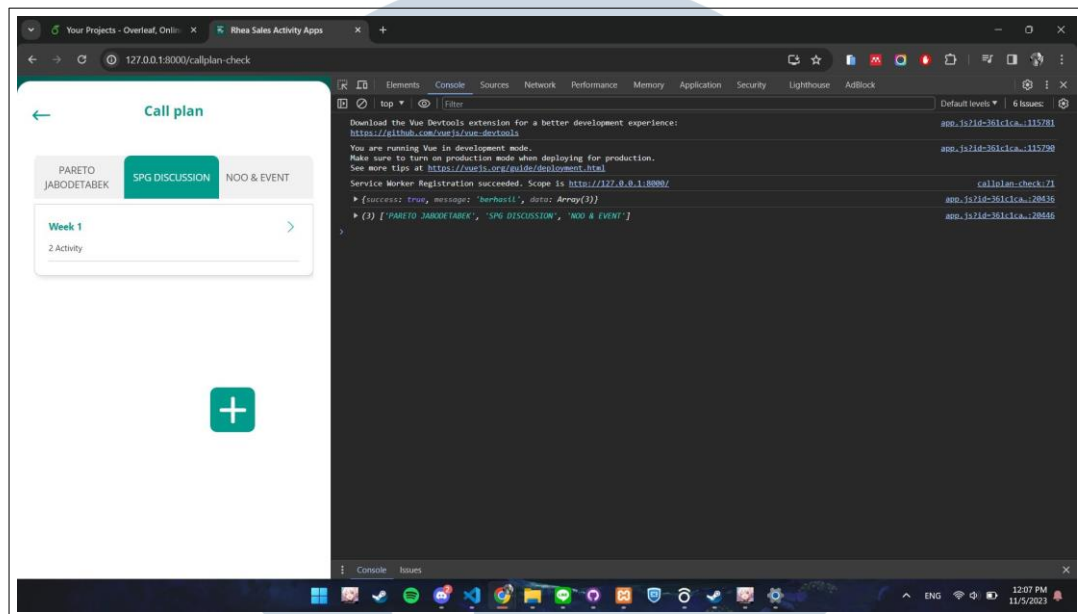


Gambar 3.15. Implementasi Mapping



Gambar 3.16. Implementasi Mapping

4. Melakukan *mapping sales activity* yang sudah di *submit* untuk minggu ini dan ditampilkan dalam bentuk *tab list*.

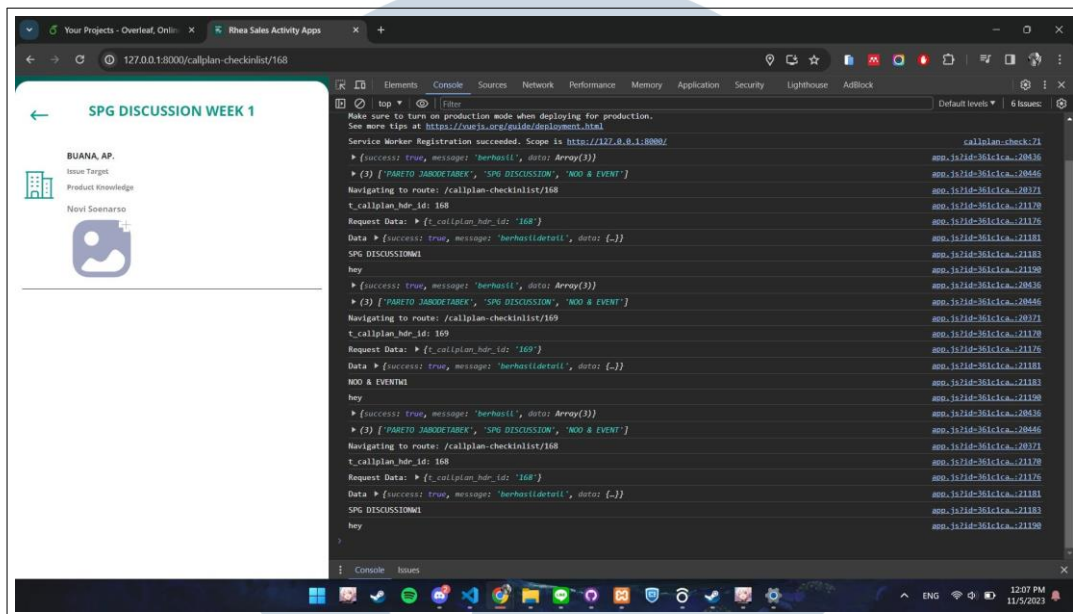


Gambar 3.17. Implementasi Tab Mapping

Gambar 3.17. merupakan tampilan *tab list* yang berisikan *activity* yang ada pada minggu tertentu, yang di bawahnya terdapat *tab detail* yang dapat melihat *detail* dari *activity* yang dipilih.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

5. Melakukan *mapping sales activity detail* memilih *icon* panah yang akan menampilkan *detail* dari *sales activity* yang dipilih.



Gambar 3.18. Implementasi Detail Mapping

Gambar 3.18. merupakan tampilan dari detail *list activity*. Pada tampilan ini terdapat data yang telah di *input* oleh *user* sesuai *form* pada bagian awal.

3.5 Kendala dan Solusi yang Ditemukan

Dalam proses kerja magang ditemukan beberapa kendala sebagai berikut:

1. Membutuhkan waktu lebih untuk mengintegrasikan antara *framework laravel* dan *vue.js*. *framework* ini belum pernah dipelajari sebelumnya.
2. Kurangnya pengetahuan tentang pengembangan dan alur sistem pada PT. Central Proteina Prima. Pengetahuan ini tidak mudah didapatkan dari supervisor karena memiliki kesibukkan lain.

Solusi yang ditemukan:

1. Mempelajari *framework Laravel* dan *Vue.js* secara bersamaan dengan proses pengerjaan. Jika terdapat kendala bisa segera menghubungi supervisor. Selain itu dapat membaca kode dari modul proyek lain yang memiliki fitur yang sama.

2. Secara rutin membaca dan mengeksplorasi proyek yang diberikan sebelum bertanya pada supervisor. Sehingga kesempatan dapat dimanfaatkan dengan baik.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA