

## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dan Koordinasi

Kedudukan yang di tempatkan oleh perusahaan adalah menjadi salah satu bagian dari EDP (*Electronic Data Processing*) yang berada di tim IT *controller*. Dalam EDP terdapat 3 tim, yaitu infrastruktur yang memiliki *jobdesk* sebagai pemegang dan penanggung jawab untuk mengelola dan menjaga stabilitas infrastruktur sistem, termasuk database, sistem operasi, serta komponen lainnya yang menunjang keberlanjutan dan keandalan sistem yang dikembangkan. *Project leader*, memiliki *jobdesk* untuk mengelola proyek, seperti *testing* untuk memastikan sistem berjalan sesuai dengan spesifikasi dan bebas dari *bug* sebelum di *publish* pada *production*. IT *controller* memiliki *jobdesk* untuk memelihara dan mengembangkan aplikasi.

Sebagai bagian dari tim IT *controller* yang berada langsung dibawah pengawasan Bapak Rizky Alfatahillah, salah satu kontrobusi adalah membantu dalam pembuatan dan perancangan ulang aplikasi yang sedang dikembangkan seperti Milky Verse dan Risto untuk melakukan migrasi menggunakan flutter. Dalam kegiatan ini, pekerjaan yang diberikan adalah membantu untuk membuat UI (*User Interface*) pada tiap aplikasi yang sedang dikerjakan.

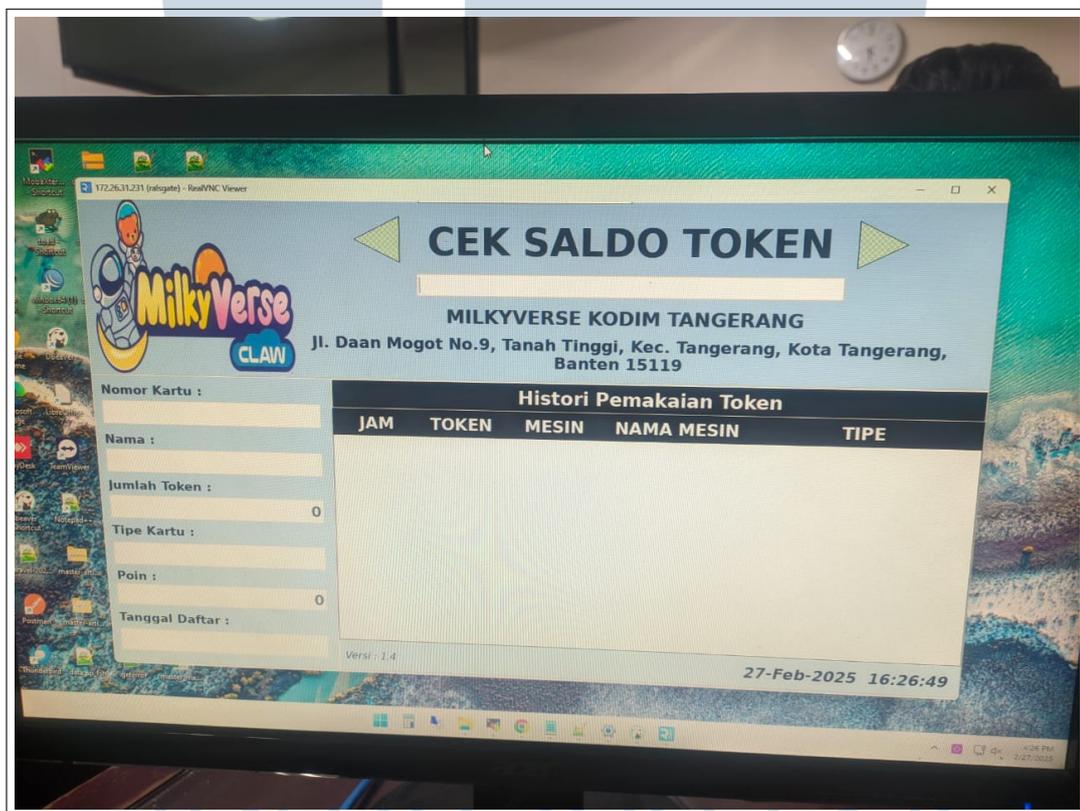
Selain itu, membantu dalam penyelesaian permasalahan jika terjadi kendala atau eror, membuat fitur baru dan merapihkan tampilan yang *overflowed* dari laporan *project leader*. Kegiatan ini biasanya dari pihak *project leader* mendapatkan laporan dari berbagai divisi diluar EDP untuk menambahkan fitur atau terkendala, seperti data yang tidak muncul di halaman web hingga ingin menambahkan sebuah menu dan fitur baru untuk menampilkan data tertentu.

Proses koordinasi dengan pembimbing lapangan dilakukan dengan cara langsung selama jam kerja (WFO). Jika ada task yang diberikan dari *project leader*, maka akan dilakukan secara langsung. Melakukan diskusi tentang permasalahan atau kendala yang dihadapi dari suatu project hingga melakukan *meeting* dengan kepala divisi EDP untuk menunjukkan hasil dari pengerjaan proyek.

## 3.2 Tugas yang Dilakukan

### 3.2.1 Desain Ulang Tampilan MilkyVerse

Pada aplikasi Milky Verse dilakukan perubahan dan perbaikan UI (*User Interface*), merupakan Proses yang bertujuan untuk memperbaiki tampilan agar lebih baik dan lebih terstruktur. Aplikasi Milky Verse sendiri berfungsi sebagai platform digital bagi pengguna untuk melihat saldo mereka serta melakukan pengecekan terhadap berbagai detail *user*. Fitur-fitur utama yang ditampilkan dalam aplikasi meliputi, informasi saldo pengguna dan detail dari wahana yang telah dimainkan. Berikut adalah tampilan aplikasi sebelum dilakukan proses redesain:



Gambar 3.1. *User Interface* Milky Verse (sebelum flutter)

Dari gambar gambar 3.1, dapat dilihat bahwa desain dari tampilan aplikasi diatas memiliki kekurangan, seperti tata letak pada desain yang kurang rapih, pemilihan warna yang kurang harmonis dengan *background* dari MilkyVerse, serta penyajian informasi yang kurang rapih.

Berdasarkan teori 10 *Usability Heuristics*, khususnya pada nilai *Aesthetic and Minimalist Design*, tampilan seharusnya berisikan informasi yang penting

sesuai dengan kebutuhan user dan tidak mengganggu user dalam melihat tampilan tersebut[5]. Pada bagian *header* terdapat tempat untuk menampilkan nomor kartu dan juga tulisan sehingga terkesan ramai dan kurang rapih. Desain tabel dan tampilan informasi dibagian kiri dari tabel terlihat seperti menumpuk yang membuat itu tidak nyaman untuk dilihat. Oleh karena itu, diperlukan perbaikan dalam tata letak, pemilihan warna serta efisiensi penyajian informasi yang tidak terlalu banyak dan sesuai dengan apa yang dibutuhkan oleh *user*. Selain merubah tampilan, aplikasi Milky Verse melakukan perubahan dari *desktop based* menjadi *web based*. Hal ini dilakukan untuk meningkatkan fleksibilitas dan aksesibilitas.

### 3.2.2 Membuat Aplikasi Risto

Aplikasi risto dibuat dengan melakukan integrasi dari aplikasi fashion, bazar dan scanning yang dilakukan demi mempermudah penggunaan aplikasi yang tidak terlalu banyak dan mobilitas jadi lebih efisien. Risto merupakan aplikasi yang dapat melakukan pencatatan produk, penerimaan konsinyasi dan laporan hasil penjualan dari setiap toko fashion atau bazar. Dari setiap toko fashion atau bazar dapat melakukan scanning terhadap SKU yang bertujuan untuk melihat berapa harga barang, promo, dan detail lain dari tiap produk tergantung dari tokonya. Berikut adalah tampilan dari tiap aplikasi yang dilakukan integrasi kedalam Risto



Gambar 3.2. Aplikasi Fashion



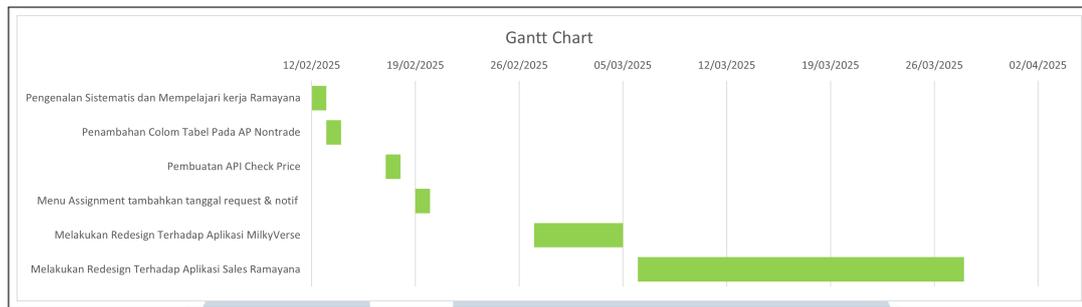
Gambar 3.3. Tampilan Aplikasi Scanning

Gambar diatas merupakan tampilan dari aplikasi bazar, fashion dan scanning. ketiga aplikasi tersebut akan dilakukan integrasi kedalam aplikasi risto yang akan dibuat menggunakan flutter. Tampilan dari tiap aplikasi juga kurang menarik karena tata letak yang kurang rapih seperti tampilan menu yang kurang efisien . Setiap menu yang di klik akan muncul tampilan submenu, namun submenu yang muncul memenuhi tampilan sehingga itu mengurangi efisiensi tampilan. Penggunaan tipografi yang kurang menarik terkesan seperti aplikasi yang sangat tua.

Berdasarkan teori 10 *Usability Heuristics*, pada aspek *Consistency and Standards*, tampilan dari tiap aplikasi tersebut menunjukkan bahwa tampilan masih kurang konsisten dalam menyajikan bagaimana sebuah kata ditulis, seperti pemilihan warna yang berbeda-beda. Pada nilai *Aesthetic and Minimalist Design*, tata letak dari tampilan membuat nilai dari *aesthetic* menjadi berkurang dikarenakan penggunaan tipo grafi yang kurang menarik, pemilihan *background* yang terkesan monoton serta layout yang sudah tertinggal jaman. Oleh karena itu, diperlukan tata letak yang lebih optimal dan pemilihan warna yang efisien.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Gantt Chart berikut.



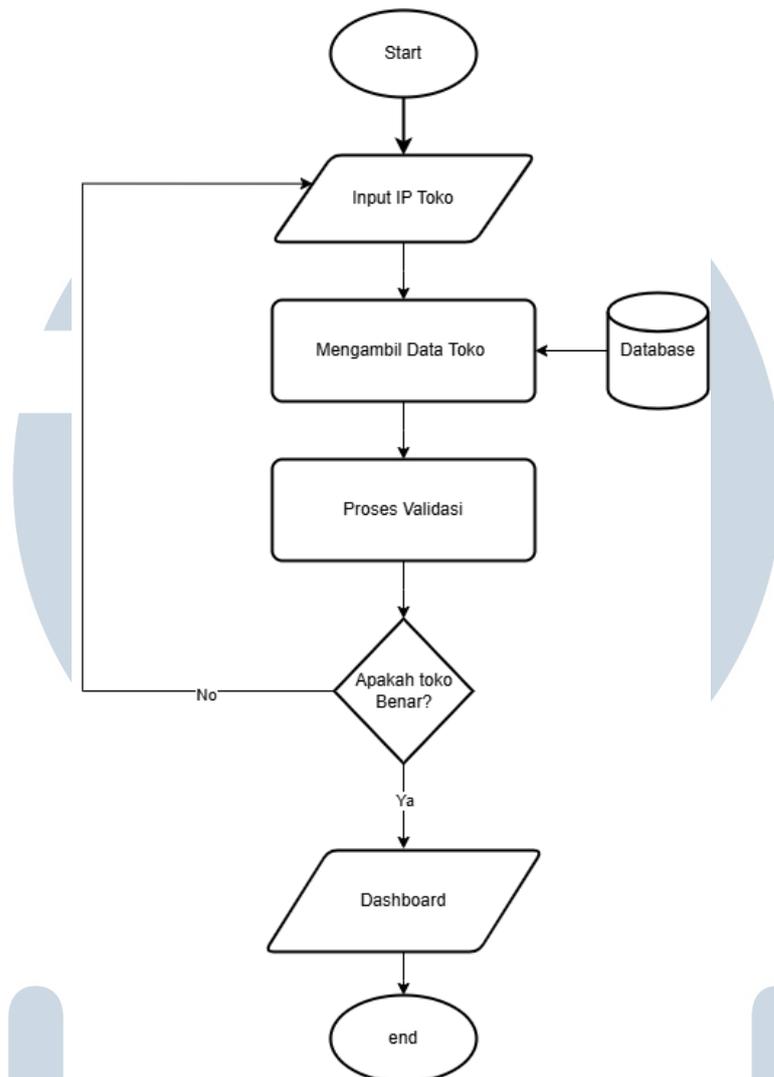
Gambar 3.4. *Gantt Chart Pelaksanaan Magang*

Pada divisi Electronic Data Processing (EDP) melaksanakan berbagai tugas seperti yang terlihat pada (gambar3.4) dengan melakukan pengembangan aplikasi sesuai dengan jadwal yang terstruktur dalam *Gantt Chart*. Tugas-tugas tersebut mencakup penambahan kolom tabel pada AP Nontrade, pembuatan API untuk pengecekan harga, serta penambahan fitur notifikasi pada menu *assignment*. Selain itu, proses redesain juga diterapkan pada dua aplikasi utama, yaitu Milky Verse dan aplikasi Risto.

### 3.3.1 Milky Verse

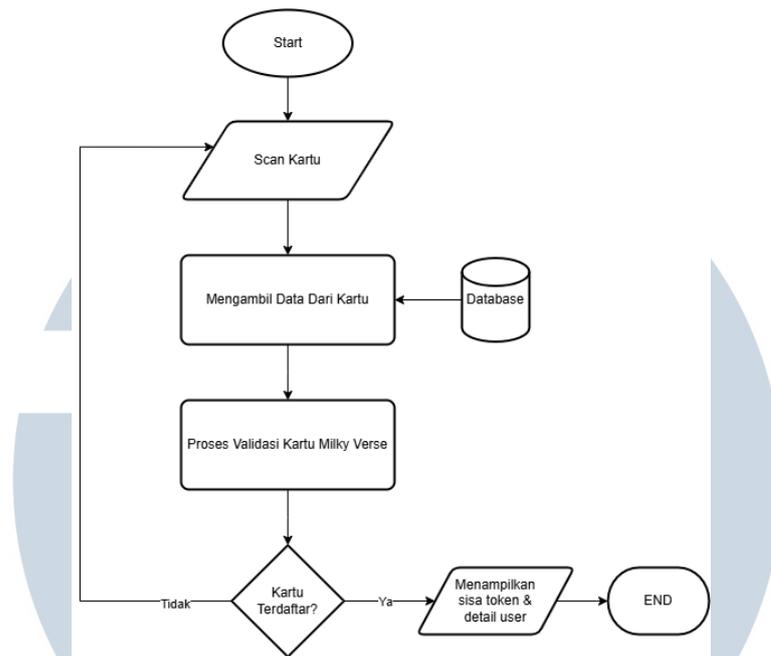
Melakukan redesain terhadap aplikasi Milky Verse merupakan upaya yang dilakukan untuk meningkatkan kualitas tampilan. Perbaikan yang dilakukan dalam proses redesain meliputi pemilihan warna yang sesuai dengan tema Milky Verse yaitu luar angkasa, merapihkan tata letak tipografi, melakukan perbaikan terhadap tabel dan informasi menjadi lebih sederhana dan hanya menampilkan informasi yang dibutuhkan serta memberikan elemen tambahan yaitu sebuah gambar yang menampilkan promo yang sedang berjalan. Berikut merupakan alur dari aplikasi Milky Verse.

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 3.5. Flowchart login milky verse

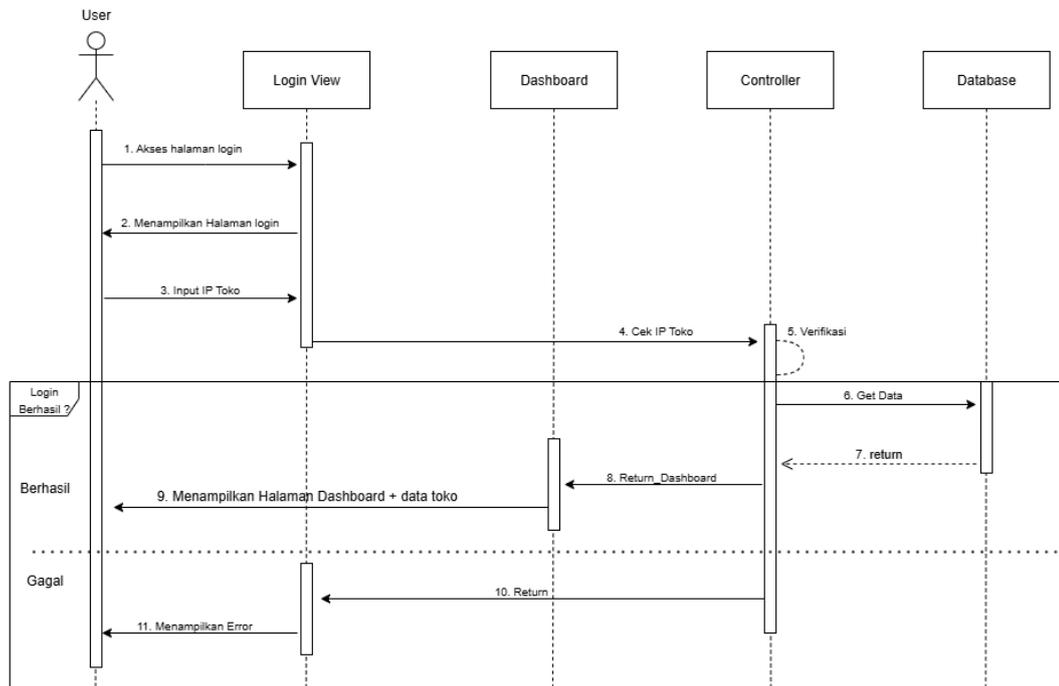
Gambar 3.5 menampilkan alur dari login pada aplikasi milky verse. User melakukan login dengan menginput IP dari toko Milky Verse. Setelah input, maka proses setelahnya adalah melakukan validasi terhadap ip tersebut dari database. jika toko terdaftar, maka akan menampilkan data toko dan menuju dashboard. Namun jika IP tidak terdaftar maka akan kembali ke tampilan input IP.



Gambar 3.6. Flowchart Scan Kartu

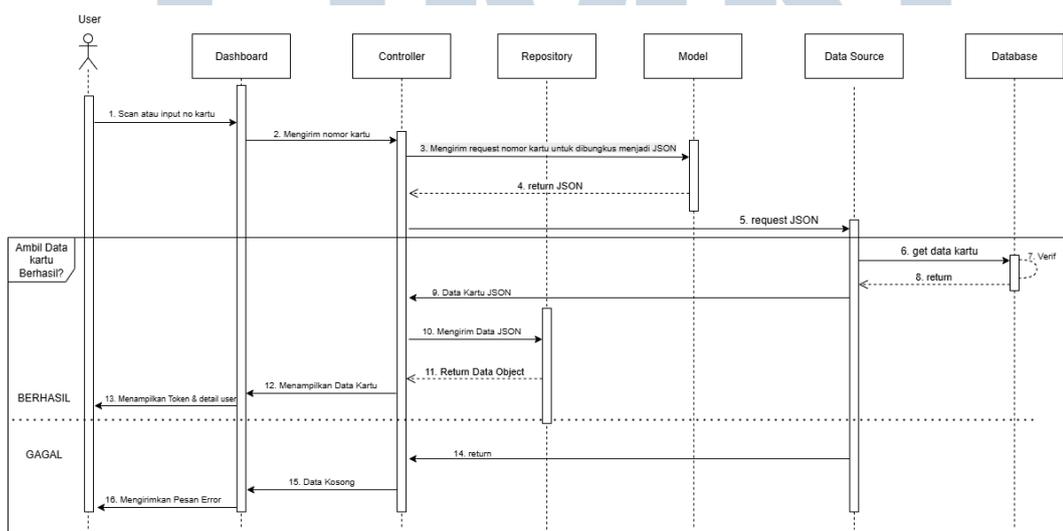
Gambar 3.6 Menjelaskan alur *scan* kartu berjalan. Kasir melakukan *scan* terhadap kartu yang dimiliki oleh user. setelah itu melakukan validasi terhadap nomor kartu apakah terdaftar atau tidak. Jika kartu terdaftar maka akan menampilkan sisa token dan detail user. Apabila tidak terdaftar akan mengirimkan pesan error, kartu tidak ditemukan. Dari alur proses ini, dapat diperhatikan ada proses *logic* yang berjalan. Untuk melihat seperti apa proses *logic* yang berjalan, berikut adalah sequence diagram yang membahas logika dari tiap modul atau fitur yang ada.

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



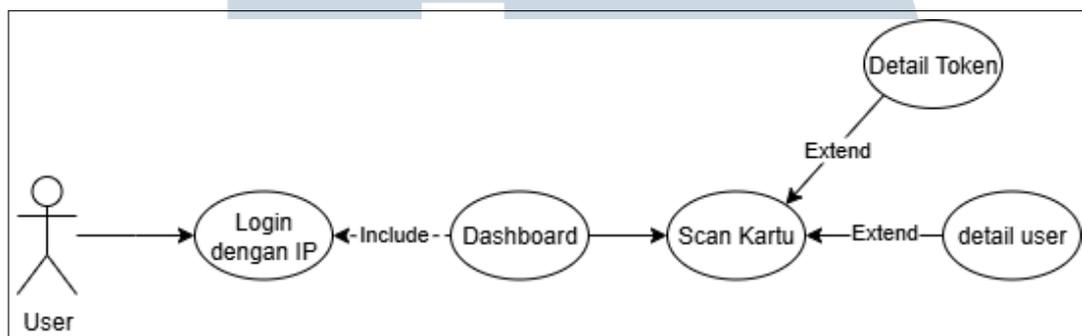
Gambar 3.7. Sequence Diagram Login Milky Verse

Gambar 3.7 menggambarkan prosesi yang terjadi pada login. Kasir membuka login page dan melakukan input IP toko. Request masuk ke server untuk dilakukan validasi dengan database. Jika IP valid, maka dari database dan server akan mengirimkan data toko dan akses pada tampilan milky verse. Jika IP tidak terdaftar maka akan menampilkan pesan error dan kembali ke input IP.



Gambar 3.8. Sequence Diagram Scan Kartu Milky Verse

Gambar 3.8 Menjelaskan bagaimana prosesi yang terjadi pada saat kasir melakukan scan kartu. Kasir melakukan scan terhadap kartu, lalu server akan menerima request dan melajutkannya ke database untuk melakukan validasi terhadap request yang dikirim. Jika nomor kartu valid, maka akan mengembalikan token dan detail user. Namun, jika tidak valid maka akan mengirimkan pesan error kartu tidak terdaftar. Untuk menggambarkan interaksi antara pengguna dan mesin, dijelaskan melalui *use case*. Berikut adalah use case dari Milky Verse.



Gambar 3.9. Use Case Milky Verse

Gambar 3.9 menjelaskan bagaimana *user* berinteraksi dengan mesin. *User* memasukan IP untuk menampilkan halaman utama dan melakukan scan terhadap kartu untuk menampilkan jumlah wahana dan jumlah saldo.

Redesain tidak hanya melakukan perubahan terhadap tampilan, tetapi mencakup perubahan dari *desktop based application* menjadi *web based application*. Berikut adalah kelemahan dari *desktop based application*:

- **Aksesibilitas dan Ketergantungan Pada Perangkat**

Aplikasi Milky Verse memiliki ketergantungan terhadap perangkat dikarenakan *desktop based application* itu harus dijalankan pada OS (*operating system*) yang ada, sehingga itu membuat ketergantungan terhadap spesifikasi dari device apakah mempunyai atau tidak.[6] Selain itu, aksesibilitas menjadi terbatas dikarenakan aplikasi harus di *download* pada perangkat yang ingin digunakan dan itu akan membutuhkan waktu untuk siap digunakan.

- **Update and Maintenance**

Jika Terjadi persoalan terhadap Milky Verse, maka harus mengirimkan pihak dari infrastruktur untuk melakukan *maintenance*, sehingga ini membuat aplikasi kurang fleksibel dan membutuhkan waktu[6]. Begitupun jika terjadi

*update*, aplikasi tidak bisa melakukan *update* langsung dikarenakan harus *download* versi terbaru untuk mendapatkan fitur yang baru ditambahkan. Jika terjadi *update* yang besar, seperti versi yang berubah ini akan menghambat kinerja.

Karena *desktop based application* sangatlah kurang optimal dan efisien, aplikasi Milky Verse melakukan perubahan dengan menggunakan *web based application*. Perubahan ini membuat aplikasi Milky Verse menjadi lebih optimal dan efisien. Berikut adalah kelebihan dari *web based application*:

- ***Centralized***

Aplikasi Milky Verse menggunakan *centralized* sehingga seluruh request dan data hanya terpusat pada satu server yang berada di pusat. *Centralized* membuat aplikasi menjadi optimal dan efisien, terutama pada *maintenance*. Karena aplikasi Milky Verse terletak pada 1 server, ini membuat *update*, perbaikan pada bug atau penambahan fitur menjadi lebih efisien dan tidak perlu melakukan secara manual pada tiap device yang ada.[7]

- **Aksesibilitas yang mudah**

Aksesibilitas dapat meningkat karena sudah menggunakan *web based application* yang membutuhkan internet untuk menjalankan aplikasi sehingga yang bisa meningkatkan efisien karena tidak perlu melakukan *download* secara manual pada setiap device[7]. Milky Verse tidak perlu tergantung dengan spesifikasi device dan OS (*operating system*).

Setelah melakukan perubahan dari *desktop based application* menjadi *web based application*, melakukan perubahan pada tampilan dengan prototipe terhadap aplikasi Milky Verse menggunakan perangkat figma, hal ini mempermudah pada pembuatan desain karena akan lebih terstruktur jika ada prototipe. Dengan adanya prototipe ini, proses redesain menjadi lebih cepat dan optimal, karena jika ada kekurangan terhadap desain seperti tata letak yang kurang baik, pemilihan warna yang tidak sesuai dengan tema dapat disesuaikan dan dibenarkan di dalam prototipe. Selain itu, prototipe yang sudah sesuai, dijadikan sebagai desain akhir yang tidak akan diubah lagi dan melakukan desain tampilan seperti yang ada pada prototipe. Berikut adalah prototipe dari aplikasi Milky Verse.



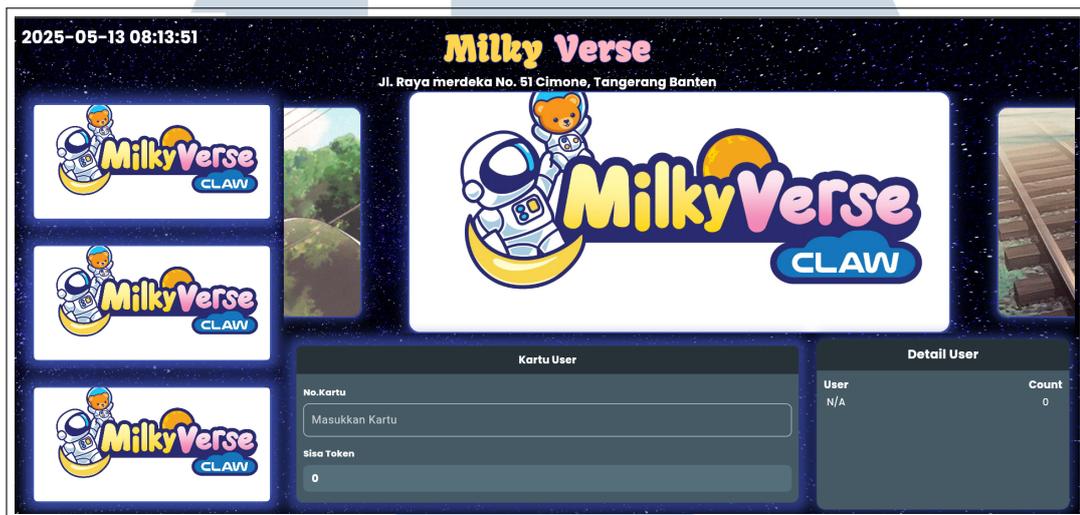
Gambar 3.10. *Prototipe Milky Verse*

Gambar 3.10 merupakan desain yang ada di prototipe, perubahan dari aplikasi Milky Verse terletak pada:

- Tabel info yang terletak pada sebelah kiri, diubah menjadi menampilkan promo yang tersedia pada Milky Verse.
- Tabel yang sebelumnya menampilkan terlalu banyak, diganti dengan yang lebih baik dan hanya menampilkan no kartu, sisa saldo, berapa jumlah wahana yang dinaiki.
- Tata letak tulisan dan logo Milky Verse yang sebelumnya kurang rapih, diperbarui dengan lebih rapih dan tipo grafi yang lebih menarik serta menghilangkan *icon* panah.
- Mengubah warna latar belakang dan warna dari tiap bagian menjadi sesuai dengan tema.

Setelah melakukan desain pada prototipe, melakukan penerapan langsung menggunakan flutter terhadap aplikasi Milky Verse. Flutter adalah *cross platform-framework* aplikasi *mobile* yang diciptakan oleh Google. Flutter dapat menciptakan aplikasi dengan desain yang menarik dengan *codebase*. Kelebihan dari flutter adalah aplikasi dapat digunakan pada berbagai platform, dari Android, IOS, website, hingga desktop[8].

Penerapan flutter pada aplikasi Milky Verse adalah untuk membuat tampilan lebih menarik. koleksi widget yang sudah ada untuk digunakan pada tampilan aplikasi, dari layout, tema, font. Widget ini bisa dikustomisasi agar menyesuaikan dengan kebutuhan, Sehingga aplikasi yang dibuat menghasilkan tampilan unik dengan cara yang mudah. Berikut adalah aplikasi yang telah dilakukan migrasi menggunakan flutter.



Gambar 3.11. User Interface MilkyVerse (flutter)

Gambar 3.11 memperlihatkan aplikasi Milky Verse yang sudah diperbarui dari tampilannya dan juga sudah menggunakan *web based application*. Berdasarkan teori 10 *Usability Heuristics*, khususnya pada nilai *Aesthetic and Minimalist Design* yang mengacu pada tampilan sebelumnya juga, berikut beberapa perubahannya :

1. Memberikan informasi yang penting  
tampilan ini sudah berisikan informasi yang penting dan sesuai dengan kebutuhan user, karena fokus dari tampilan ini adalah memperlihatkan saldo dan berapa banyak wahana yang sudah dimainkan oleh user.
2. Perubahan Tata Letak  
Tata letak dari tampilan Milky Verse sudah diperbaiki, warna yang sudah selaras dengan temanya yaitu Milky Verse yang mengambil tema luar angkasa, tampilan tabel yang disederhanakan dimana hanya menampilkan nomor kartu, sisa saldo, detail user dan jumlah wahana yang dimainkan. Menambahkan tempat untuk menampilkan promo.

Dengan Migrasi ini, aplikasi Milky Verse sudah menjadi lebih modern, lebih menarik untuk dilihat serta meningkatkan fleksibilitas dan aksesibilitas.

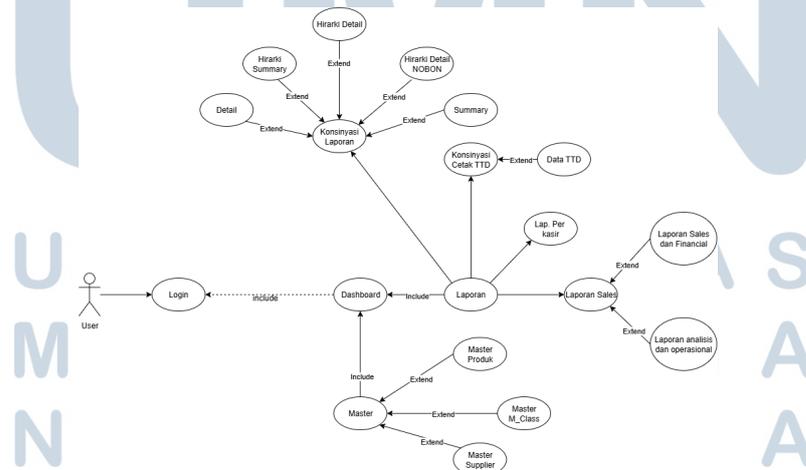
### 3.3.2 Risto Ramayana

Risto merupakan aplikasi yang dapat melakukan pencatatan produk, penerimaan konsinyasi dan laporan hasil penjualan dari setiap toko fashion atau bazar. Dari setiap toko fashion atau bazar dapat melakukan *scan* SKU yang bertujuan untuk melihat berapa harga barang, promo, dan detail lain dari tiap produk tergantung dari tokonya.

Aplikasi ini merupakan integrasi dari 3 aplikasi lain yang sebelumnya sudah ada, yaitu aplikasi fashion, bazar dan scanning. Tujuan melakukan integrasi adalah untuk mempermudah proses pencatatan, pengelolaan, *monitoring* dan transaksi yang hanya dilakukan oleh 1 aplikasi besar, sehingga membuat mobilitas dan kinerja lebih maksimal tidak perlu menjalankan 3 aplikasi sekaligus dalam waktu yang bersamaan. Berikut Penjelasan dari tiap aplikasi dan perubahan yang akan diterapkan pada aplikasi risto:

- **Aplikasi Fashion**

Aplikasi fashion dapat melakukan *report*, pencatatan, laporan dan mencetak tanda terima dari setiap produk seperti, produk konsinyasi, produk ramayana dan transaksi yang terjadi di dalam toko. Berikut *use case* diagram dari aplikasi Fashion:



Gambar 3.12. Use Case Fashion

Gambar 3.12 menjelaskan cara interaksi mesin dengan pengguna. User dapat

melihat semua laporan yang tersedia dan melakukan *download* terhadap laporan. Pada bagian master, user dapat melakukan input terhadap data master dan master supplier. Input ini bertujuan untuk menambah data baru dari produk dan juga supplier jika ada pihak konsinyasi yang sesuai SOP bekerja sama dengan toko terkait. Berikut penjelasan lebih detail untuk integrasi ke Flutter per modul:

– **Master Supplier Fashion**

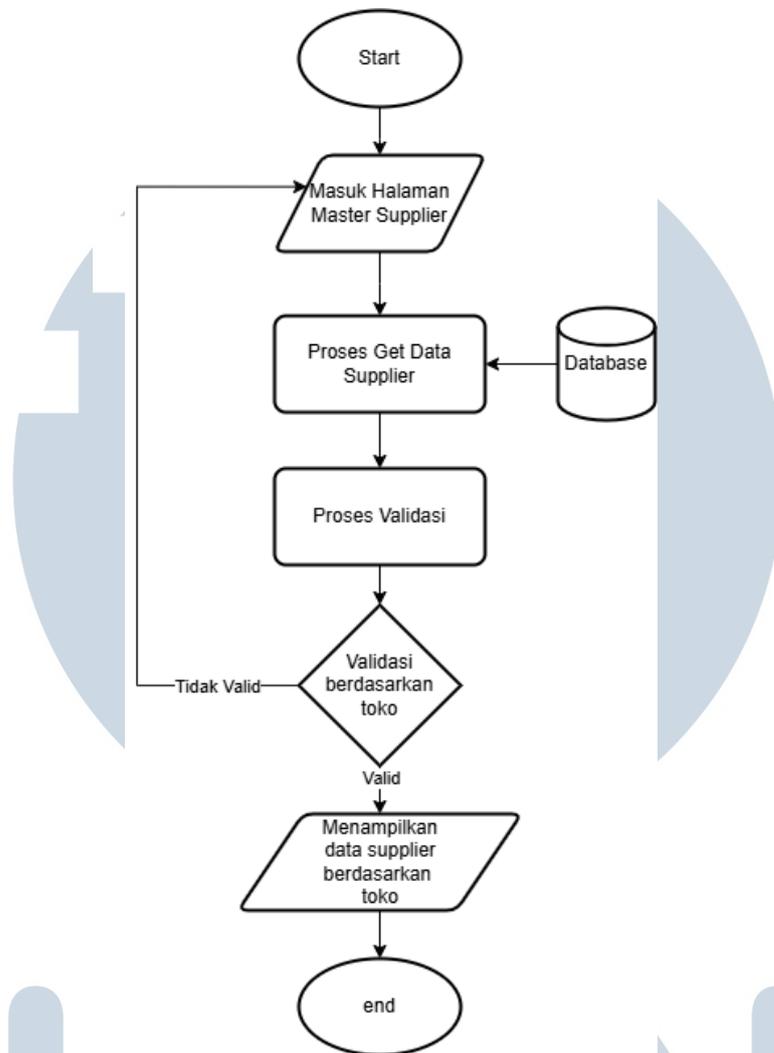
Perubahan yang dilakukan pada Master Supplier Fashion meliputi layout dan pemilihan warna yang disesuaikan dengan warna perusahaan. Berikut adalah prototipe dari Master Supplier:



Gambar 3.13. Prototipe Master Supplier Fashion

Pada Gambar 3.13 menjelaskan dengan prototipe yang menjadi acuan untuk dilakukan perubahan terhadap modul Master Supplier yang akan diintegrasikan ke dalam aplikasi Risto. Berikut flowchart dari Master Supplier:

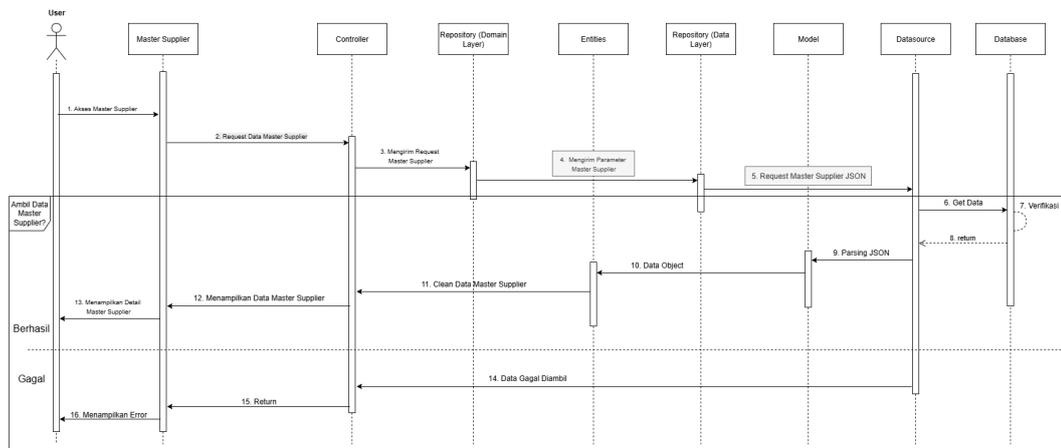
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.14. Flowchart Get Supplier Fashion

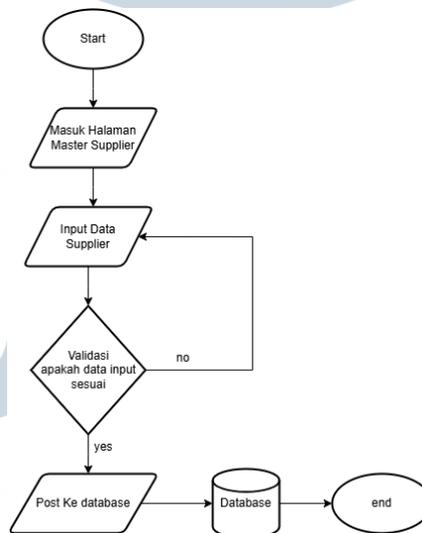
Gambar 3.14 merupakan flowchart yang menjelaskan bagaimana proses mendapatkan data dari Master Supplier. Saat masuk ke tampilan Master Supplier, akan dilakukan *get request* yang divalidasi berdasarkan toko, kemudian ditampilkan datanya. Berikut adalah alur dari sequence diagram-nya:

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



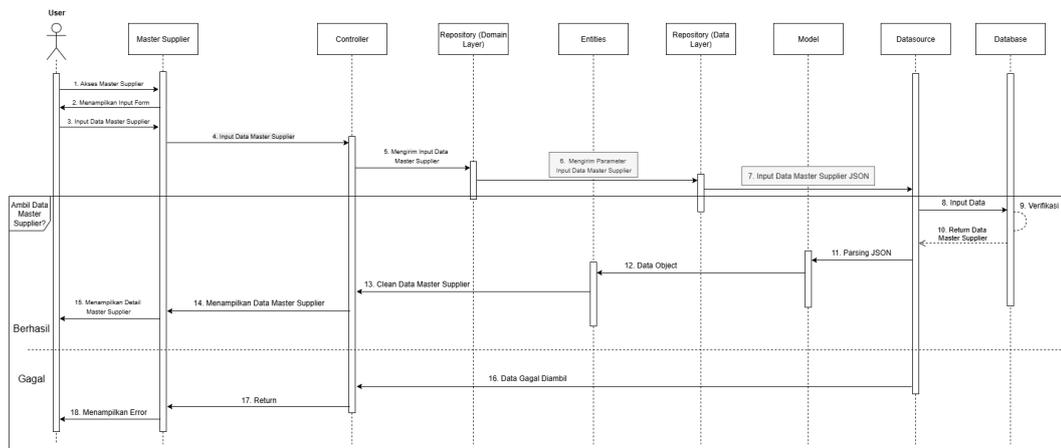
Gambar 3.15. Sequence Diagram Get Supplier

Gambar 3.15 menjelaskan bagaimana alur dari proses *request* yang terjadi ketika user masuk kedalam halaman Master Supplier. Jika request berhasil dikirimkan, maka akan menampilkan data master supplier dan jika gagal maka akan mengembalikan error. Selanjutnya, untuk proses input supplier:



Gambar 3.16. Flowchart Input Supplier Fashion

Gambar 3.16 memperlihatkan bagaimana alur prosesi yang terjadi pada saat mengisi form input. Data yang di input akan dilakukan validasi. Jika valid, maka akan disimpan ke database. Berikut sequence diagram-nya:

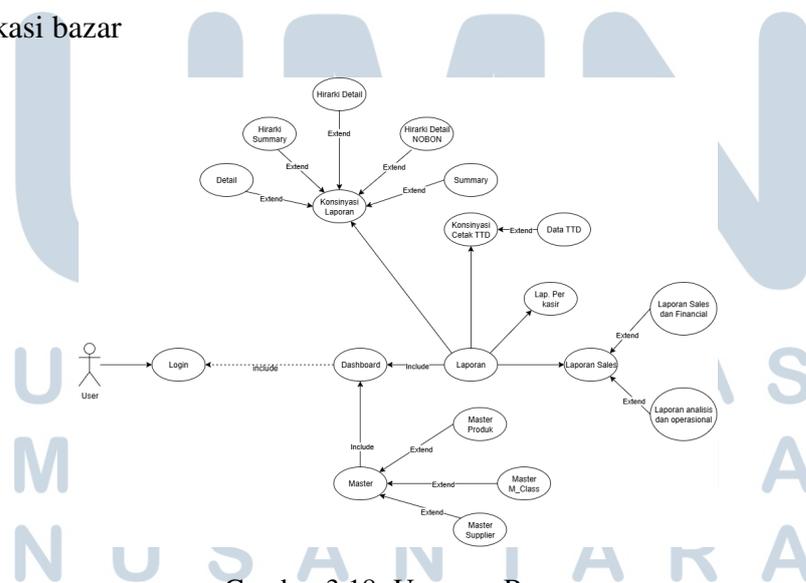


Gambar 3.17. Sequence Diagram Input Supplier

Pada gambar 3.17 menjelaskan bagaimana alur dari input data master supplier ke dalam database lalu akan kembali di tampilkan. Jika input berhasil maka data master supplier yang paling baru akan di tampilkan menjadi yang paling atas, jika gagal maka akan mengembalikan error.

• **Aplikasi Bazar**

Aplikasi Bazar dapat melakukan report, pencatatan, laporan dan mencetak tanda terima dari setiap produk seperti, produk konsinyasi, produk ramayana dan transaksi yang terjadi di dalam toko bazar. Berikut adalah *usecase* dari aplikasi bazar



Gambar 3.18. Use case Bazar

Gambar 3.18 menjelaskan cara interaksi mesin dengan pengguna. User dapat melihat semua laporan yang tersedia dan melakukan *download* terhadap

laporan. Pada bagian master, user dapat melakukan input terhadap data master dan master supplier. Input ini bertujuan untuk menambah data baru dari produk dan juga supplier jika ada pihak konsinyasi yang sesuai SOP bekerja sama dengan toko terkait. Berikut penjelasan lebih detail untuk integrasi ke Flutter per modul:

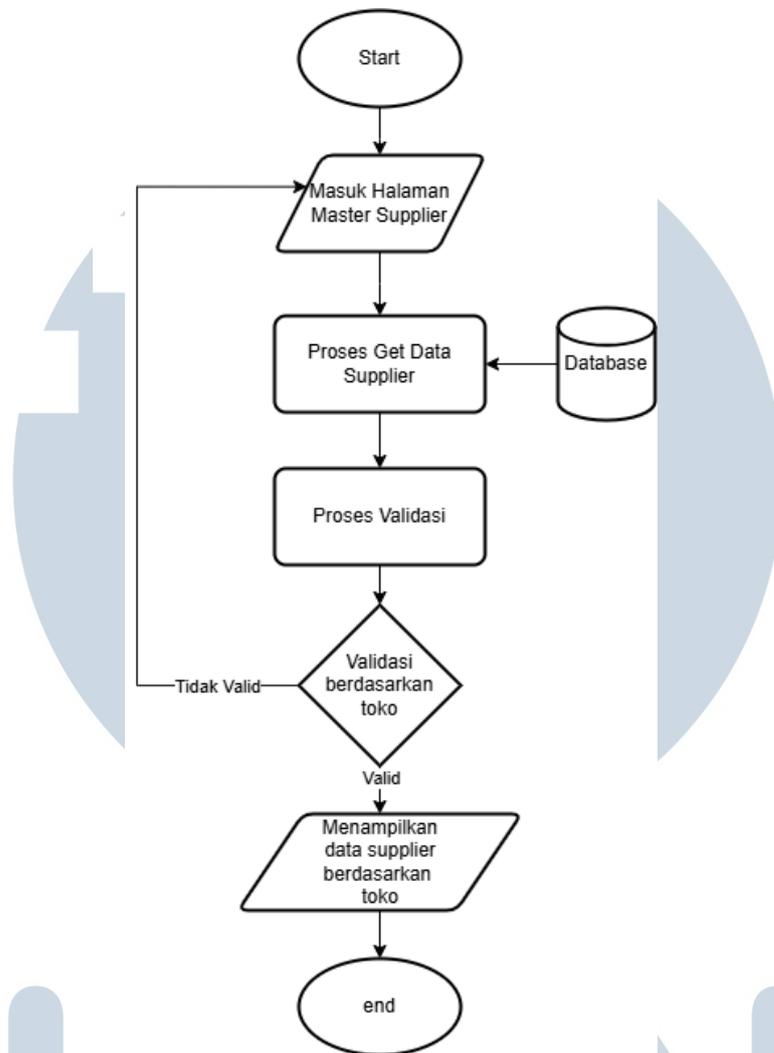
- Master Supplier bazar perubahan yang dilakukan pada Master Supplier bazar adalah perubahan tata letak dan pemilihan warna yang disesuaikan dengan karakter perusahaan. Berikut adalah prototipe dari master supplier bazar

Kode Supplier	Nama Supplier	Alamat	Contact	Email
9120701	AYONY TANJUNG PT	Jl. Kampung Bali no 10	08129876543	ayony@gmail.com
9125401	MULTI GARMENJAYA P	Jl. Kejaksaan no 205	08129876543	garmen@gmail.com
9128301	CARDINAL SHOES LAD	Jakarta Pusat, tanah abang	08129876543	cardinal@gmail.com
9130901	VIDIAELOK LESTARI	Veteran, Tangerang	08129876543	vidaelok@gmail.com
9130905	STAR ASIA BROTHERS	Panglima Polim	08129876543	starasia@gmail.com
9134201	COKRO HADIWIJAYA	Jendral Soedirman 10	08129876543	cokro@gmail.com

Gambar 3.19. Prototipe master supplier bazar

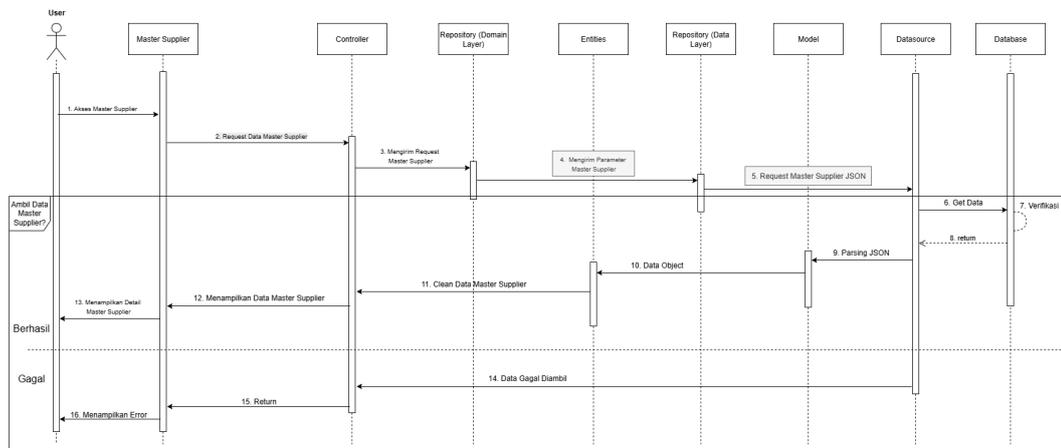
Pada gambar 3.19 merupakan prototipe yang akan menjadi acuan dari perubahan modul master supplier yang akan di integrasikan kedalam aplikasi risto. Berikut adalah flowchart dari master supplier:

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



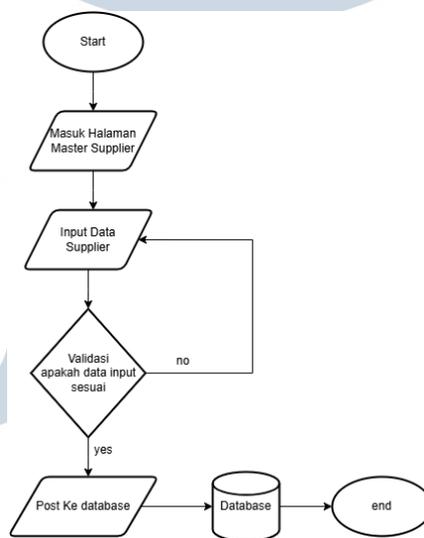
Gambar 3.20. Flowchart Get Supplier bazar

Gambar 3.20 menjelaskan proses untuk mendapatkan data dari master supplier bazar. Pada saat masuk kedalam tampilan master supplier, sistem akan melakukan Get request untuk mendapatkan data yang ada pada master supplier. Setelah melakukan request akan dilakukan validasi berdasarkan toko. Setelah validasi selesai maka akan menampilkan data supplier berdasarkan toko. Dari flowchart ini, terdapat proses sistem yang terjadi. berikut adalah proses dari sequence diagramnya:



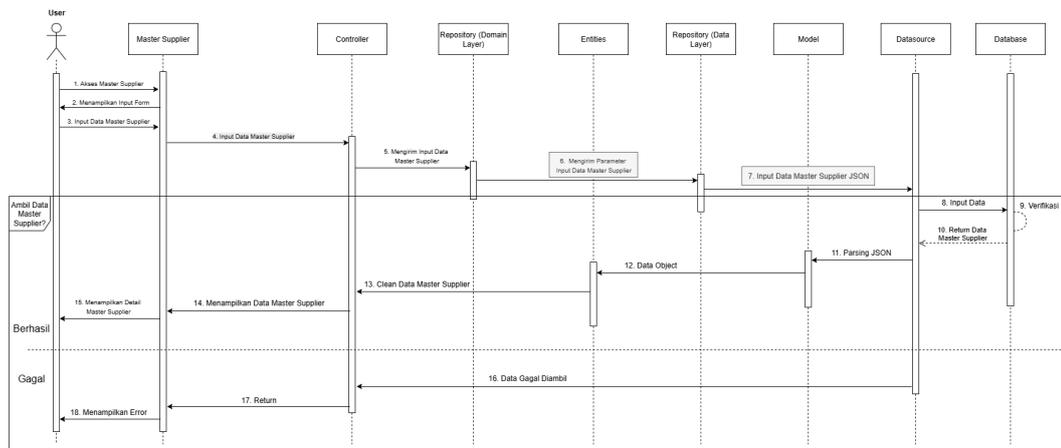
Gambar 3.21. Sequence Diagram master supplier bazar

Sequence diagram yang ada pada (Gambar 3.21) menjelaskan prosesi yang terjadi ketika masuk kedalam tampilan master supplier. User mengakses halaman master supplier, Jika request berhasil dikirimkan, maka akan menampilkan data master supplier dan jika gagal maka akan mengembalikan error. Selanjutnya, untuk proses input supplier.



Gambar 3.22. Flowchart Input Supplier bazar

Flowchart pada (gambar 3.22) menjelaskan alur untuk input data supplier baru kedalam database. Setelah melakukan input akan dilakukan validasi terhadap data yang di input, jika data yang di input benar maka akan tersimpan kedalam database. Dari flowchart ini, terdapat proses sistem yang terjadi. berikut adalah alur dari sequence diagramnya.



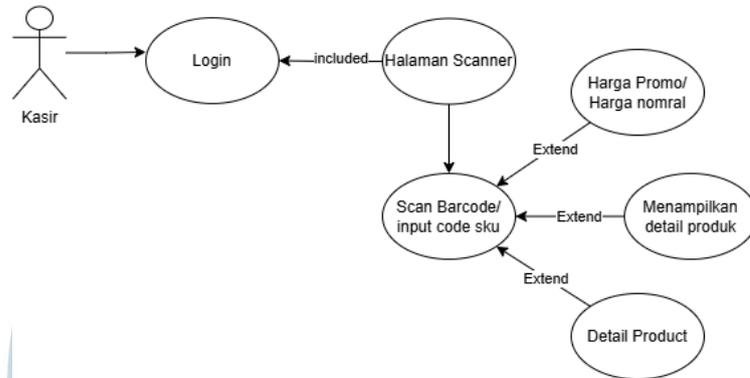
Gambar 3.23. Sequence Diagram input Supplier bazar

Gambar 3.23 memperlihatkan sequence diagram untuk menjelaskan prosesi yang terjadi ketika input data supplier baru. User mengakses halaman master supplier, dari halaman melakukan input untuk menyimpan data supplier yang baru. setelah itu mengirimkan data ke database untuk menyimpan data master supplier yang ada sesuai toko ke dalam database. Jika berhasil maka akan menampilkan kembali supplier yang baru saja di input, jika salah maka akan memunculkan error.

- **Aplikasi Scanning**

Scanning merupakan aplikasi price check dari tiap SKU (stock keeping unit) atau produk. Ketika barcode di scan atau memasukan kode sku, maka akan menampilkan nama produk, harga dari produk (apakah diskon atau tidak) dan detail dari produk tersebut. Berikut adalah Use Case diagram dari aplikasi scanning:

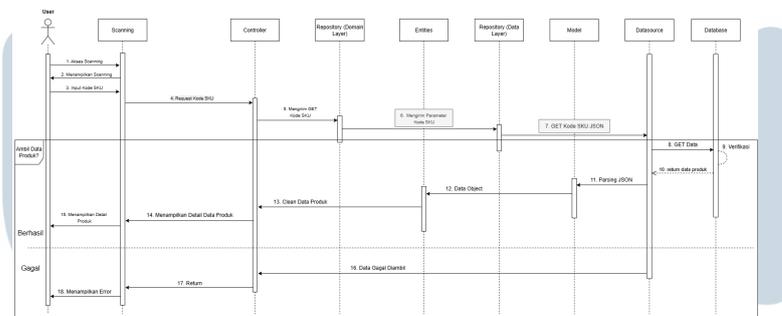
### Use Case Scanning



Gambar 3.24. Use Case Scanning

Dari diagram use case yang dapat dilihat pada (gambar 3.24), menjelaskan bahwa hanya terdapat 1 aktor yaitu kasir. Kasir dapat melakukan *scan* atau memasukan kode sku untuk produk tertentu. Setelah melakukan *scan* atau input code sku, maka akan tampil harga, nama dan detail dari produk yang di input.

Pada aplikasi ini terdapat fungsi modulasi aplikasi scanner, berikut sequence diagram dari aplikasi scanner:



Gambar 3.25. Sequence Diagram Scanner

Gambar 3.25 merupakan sequence diagram untuk menjelaskan proses yang terjadi pada aplikasi scanning, dimana user mengakses halaman aplikasi untuk melakukan *scan* terhadap produk. User melakukan input SKU (*Stock Keeping Unit*) untuk melihat produk tersebut tersedia atau tidak dan sedang terdapat potongan harga atau tidak. Jika SKU yang dimasukan benar, maka produk yang sesuai dengan SKU tersebut akan muncul namun jika tidak sesuai maka akan menampilkan error.

Pembuatan prototipe juga diperlukan untuk melakukan perubahan tampilan terhadap aplikasi scanning menggunakan perangkat figma. Hal ini mempermudah pada pembuatan desain karena akan lebih terstruktur dan optimal jika dibuat prototipe terlebih dahulu, karena jika ada kekurangan terhadap desain seperti tata letak yang kurang baik, pemilihan warna yang tidak sesuai dengan tema dapat disesuaikan dan dibenarkan di dalam prototipe. Selain itu, prototipe yang sudah sesuai, dijadikan sebagai desain final yang tidak akan diubah lagi dan melakukan desain tampilan seperti yang ada pada prototipe. Berikut adalah prototipe dari aplikasi Scanning.

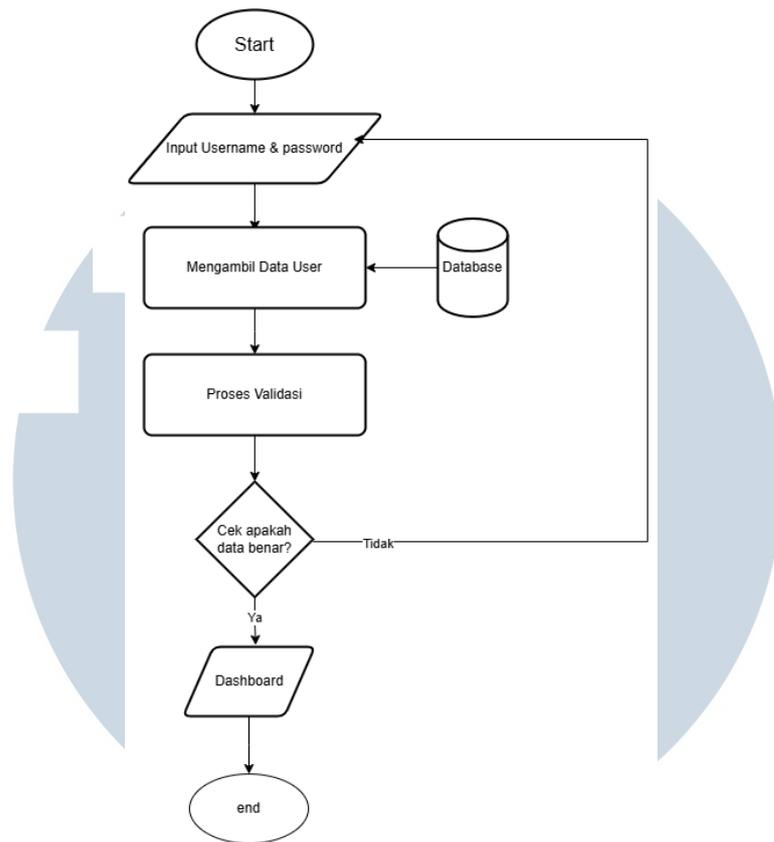


Gambar 3.26. prototipe aplikasi scanner

Dari desain yang ada di prototipe (gambar 3.26), perubahan dari aplikasi scanning terletak pada:

- Layout yang lebih modern tidak menumpuk seperti desain sebelumnya
- Tipografi yang dibuat sederhana dan data yang di tampilkan hanya yang diperlukan saja, seperti menghilangkan deskripsi produk yang sama.
- Pemilihan warna *background* yang disesuaikan dengan warna karakter dari perusahaan ramayana, yaitu merah dan putih.

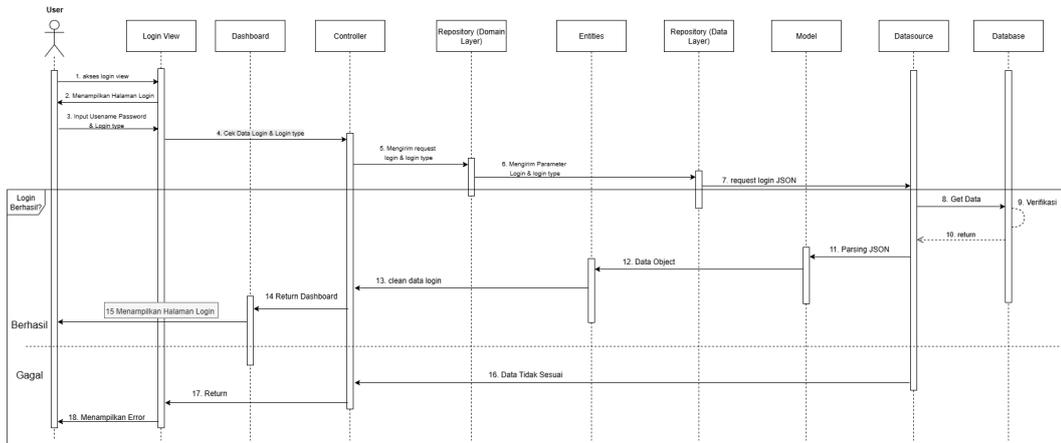
Aplikasi fashion, bazar dan scanning akan dilakukan integrasi kedalam aplikasi risto, sehingga aplikasi risto memiliki fungsi untuk mencakup semua pencatatan dan melakukan scanning dari berbagai toko yang ada di ramayana. Aplikasi risto memiliki 2 login, yaitu sebagai fashion dan bazar. Berikut adalah flowchart dari login Risto:



Gambar 3.27. Flowchart Login Risto

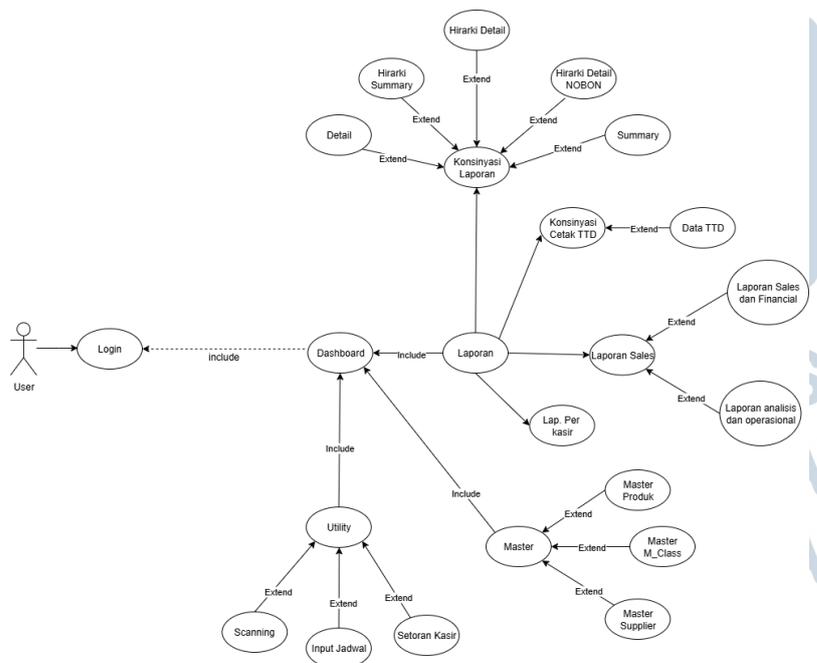
Gambar 3.27 merupakan flowchart yang menjelaskan alur login dari aplikasi risto, dimana pada saat login user akan input username dan password serta memilih jenis dari toko mereka. Setelah mereka memilih, proses validasi terjadi untuk melakukan pengecekan terhadap user. Apakah user benar berada di jenis toko yang mereka pilih, jika salah maka akan kembali ke tampilan login dan jika benar maka akan masuk kedalam dashboard sesuai dengan jenis toko yang dipilih. Berikut adalah prosesi yang terjadi pada login Risto.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.28. sequence diagram login risto

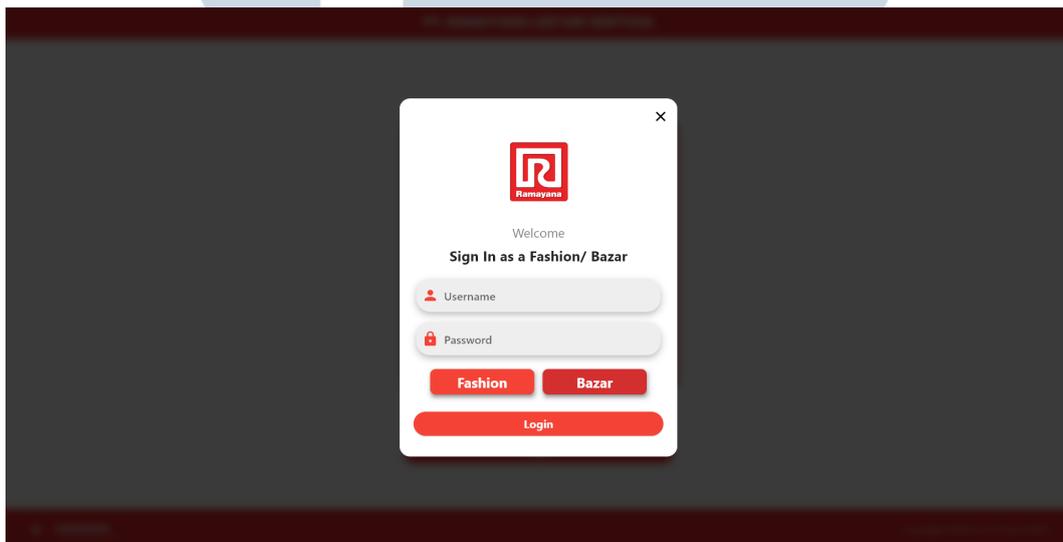
Sequence diagram pada (gambar 3.28) menjelaskan prosesi yang terjadi pada saat login kedalam aplikasi risto. user melakukan input dan memilih tipe toko, setelah memilih dari halaman akan mengirimkan request login ke server. Setelah itu akan dilakukan validasi terhadap user dan tipe toko. Jika user tervalidasi, maka user bisa masuk kedalam tampilan sesuai dengan kode toko yang dipilih, jika tidak valid maka akan mengirimkan pesan error. Untuk menggambarkan interaksi antara pengguna dan mesin, dijelaskan melalui usecase. Berikut adalah usecase dari Aplikasi Risto :



Gambar 3.29. Usecase Aplikasi Risto

Gambar 3.29 merupakan sequence diagram dari aplikasi Risto yang sudah diintegrasikan dengan aplikasi lainnya. User tetap dapat melakukan hal yang sama seperti melihat dan melakukan *download* terhadap laporan. User juga dapat mengakses scanning untuk melakukan *scan terhadap* SKU pada produk. Produk akan tampil sesuai toko yang di input pada saat user login dan laporan serta data lainnya juga akan tampil sesuai dengan toko yang disesuaikan dengan user.

Setelah melakukan desain prototipe, melakukan penerapan langsung menggunakan flutter terhadap aplikasi risto. Penerapan flutter pada aplikasi risto adalah untuk membuat tampilan lebih menarik, koleksi widget yang sudah ada dapat digunakan pada tampilan aplikas. Dari Widget ini bisa dikustomisasi agar menyesuaikan dengan kebutuhan, Sehingga aplikasi yang dibuat menghasilkan tampilan unik dengan cara yang mudah. Berikut adalah aplikasi yang telah dilakukan migrasi menggunakan flutter.



Gambar 3.30. tampilan login risto

Dari (gambar 3.30), dapat dilihat login dari aplikasi risto yang sudah diperbarui dari tampilannya. Berdasarkan teori 10 *usability heuristic*, khususnya pada nilai *aesthetic and minimalist design*, dapat dilihat bahwa tampilan memiliki design yang minimalis, dengan pemilihan warna yang disesuaikan dengan karakter perusahaan dan tidak terlalu banyak widget.

Kode Supplier	Nama Supplier	Alamat	Contact	Email
9141101	PT. DHIYA MAHARANI	Jl. Kebon Jeruk raya 42	(053) 8742	<a href="mailto:dhiya@gmail.com">dhiya@gmail.com</a>
9125401	MULTI GARMENJAYA P	Jl. Kampung Bandan 15	(0341) 804382	<a href="mailto:garmen@gmail.com">garmen@gmail.com</a>

Gambar 3.31. tampilan master supplier risto

Tampilan dari master supplier sudah diperbarui, dari (gambar 3.31) berdasarkan teori 10 *usability heuristic*, khususnya pada nilai *aesthetic and minimalist design*, dapat dilihat bahwa tampilan memiliki desain yang minimalis, memberikan informasi yang tidak terlalu banyak sehingga tidak membingungkan pengguna dalam melihat data yang tampil. Pada master supplier juga bisa melakukan *search* berdasarkan kode supplier atau nama supplier, sehingga dapat mempermudah pencarian pada data supplier secara cepat.

**Price Check**

SKU / Barcode:  Barcode:

Sku: - Dept: -

BP  
MSC

---

*Deskripsi*

COCA COLA 1 LITER

---

*Promosi*

Rp 7.000

Rule: 3 periode 01.05.2025 sd 31.05.2025  
Disc 1 : 30.00% Disc 2 : - Disc 3 : -  
H. Normal : Rp **10.000**  
Stock: 20 pcs

Gambar 3.32. tampilan scanner risto

Tampilan dari scanner sudah diperbarui, perubahan layout dan tipografi dari

tampilan sebelumnya. Dari (gambar 3.32) dapat dilihat dimana tampilan warna dibuat mengikuti karakter perusahaan dan layout tidak menumpuk seperti tampilan sebelumnya. Dengan migrasi ini, aplikasi risto menjadi lebih modern, lebih menarik untuk dilihat.

### **3.4 Kendala dan Solusi yang Ditemukan**

#### **3.4.1 Kendala kerja Magang**

Kurangnya dokumentasi menjadi hambatan yang sangat signifikan dalam proses pengembangan sistem di Divisi EDP. Banyak proyek yang ditangani oleh bagian IT Controller tidak disertai dengan panduan teknis maupun dokumentasi prosedural yang memadai. Akibatnya, proses pengerjaan menjadi kurang terarah karena pengembang harus mencari tahu sendiri alur sistem dan cara implementasinya, yang menyebabkan waktu penyelesaian menjadi lebih lama dan efisiensi menurun.

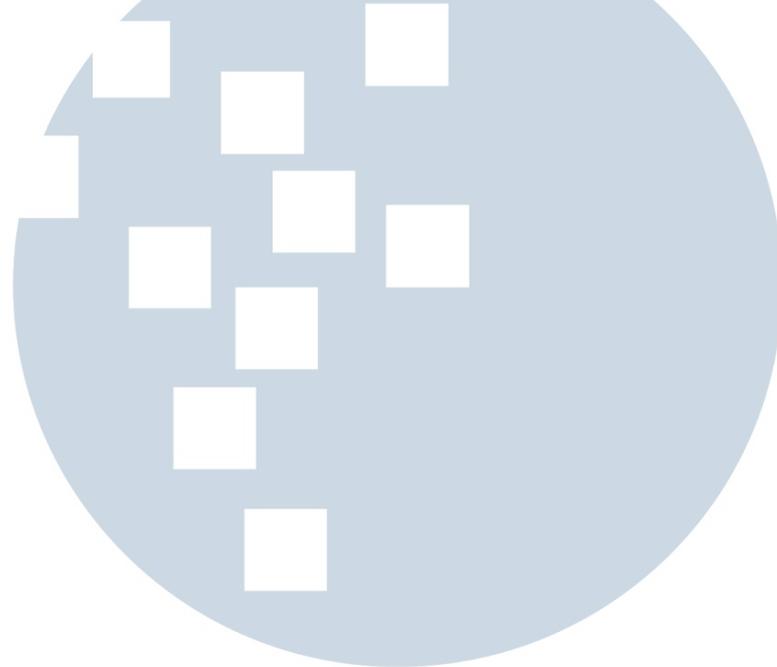
Selain itu, tidak adanya blueprint atau kerangka kerja standar untuk setiap proyek turut menambah kompleksitas. Tanpa arahan yang jelas mengenai apa yang harus dikerjakan dan bagaimana tahapan pengerjaannya, tim mengalami kebingungan dalam menentukan prioritas serta alur pengembangan. Permasalahan ini semakin diperparah dengan keterbatasan pengalaman tim dalam menangani proyek-proyek dengan tingkat kompleksitas bisnis logic yang tinggi. Kurangnya pengalaman tersebut menjadi hambatan tambahan dalam memahami sistem dan menyusun solusi yang tepat secara cepat dan akurat.

#### **3.4.2 Solusi yang ditemukan**

Untuk mengatasi berbagai kendala tersebut, pendekatan yang diambil adalah dengan melakukan eksplorasi dan investigasi secara langsung terhadap sistem yang akan dikerjakan. Langkah awal dimulai dengan menggali informasi teknis yang berkaitan, baik dari dokumentasi yang tersedia maupun dari pengamatan langsung terhadap sistem berjalan. Kemudian dilakukan pemetaan alur kerja (flow) serta identifikasi kebutuhan teknis yang diperlukan.

Setelah itu, proses pembelajaran dilakukan secara mandiri melalui referensi yang relevan, baik berupa dokumentasi resmi, forum pengembang, maupun proyek serupa yang pernah dikerjakan sebelumnya. Pendekatan ini memungkinkan tim untuk memahami sistem secara bertahap dan menyusun strategi pengembangan

secara lebih sistematis. Di samping itu, hasil eksplorasi ini dijadikan bahan untuk membuat dokumentasi dan blueprint internal, agar dapat digunakan sebagai panduan pada proyek-proyek berikutnya dan mengurangi ketergantungan pada pengalaman individu.



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA