

BAB 3

METODE PENELITIAN

3.1 *Software Development Lifecycle (SDLC)*

3.1.1 *Requirement Gathering*

Requirement gathering yang digunakan adalah kuesioner dan observasi. Kuesioner ini terdiri dari beberapa pertanyaan terkait dengan rumusan masalah yang sudah terbentuk. Kuesioner ini ditujukan untuk para penjual *online* yang sudah menjual produk-produk dalam jumlah yang cukup banyak. Observasi dilakukan dengan menganalisis aplikasi CMS sejenis seperti Shopify dan Wordpress.

3.1.2 *FDD (Feature Driven Development)*

Proses model yang digunakan dalam pembuatan aplikasi *web* ini adalah *Feature Driven Development* (FDD). Dalam implementasinya, tahapan pengembangan terbagi menjadi lima tahap, yaitu:

1. *Develop an overall model*

Pada tahap ini dilakukan diskusi dan rancangan terkait hasil yang diharapkan pada aplikasi melalui *requirement gathering*.

2. *Build a feature list*

Pengembangan model sistem tersebut dibuat berdasarkan dengan diagram UML, yakni *use case diagram*. Pada tahap ini dilakukan identifikasi fitur sesuai dengan modul *use case diagram*. Hasil identifikasi list fitur yang akan dikembangkan adalah sebagai berikut:

- *Register*
- *Login*
- *Forgot Password*
- *Logout*
- *Edit User*
- *View Single Page*
- *Add Section*
- *Edit Section*
- *Delete Section*

- *Manage Pages*
- *Add Page*
- *Delete Page*
- *View Products*
- *Add Product*
- *View Single Product*
- *Edit Product*
- *Delete Product*
- *Buy Product*
- *Customize Theme*
- *View Dashboard*

3. Plan by feature

Tahap ini mengidentifikasi urutan pengembangan fitur berdasarkan prioritas fitur. *Gantt chart* yang digunakan dalam pengembangan aplikasi ini dijabarkan sebagai berikut.

Tabel 3.1. *Gantt Chart*

	Februari			Maret				April	
	2	3	4	1	2	3	4	1	2
<i>View Products</i>									
<i>View Single Product</i>									
<i>Add Product</i>									
<i>Edit Product</i>									
<i>Delete Product</i>									
<i>View Single Page</i>									
<i>Add Section</i>									
<i>Edit Section</i>									
<i>Delete Section</i>									
<i>Register</i>									
<i>Login</i>									
<i>Logout</i>									
<i>Forgot Password</i>									
<i>Manage Pages</i>									
<i>Add Page</i>									
<i>Delete Page</i>									
<i>Buy Product</i>									
<i>Edit User</i>									
<i>Customize Theme</i>									
<i>View Dashboard</i>									

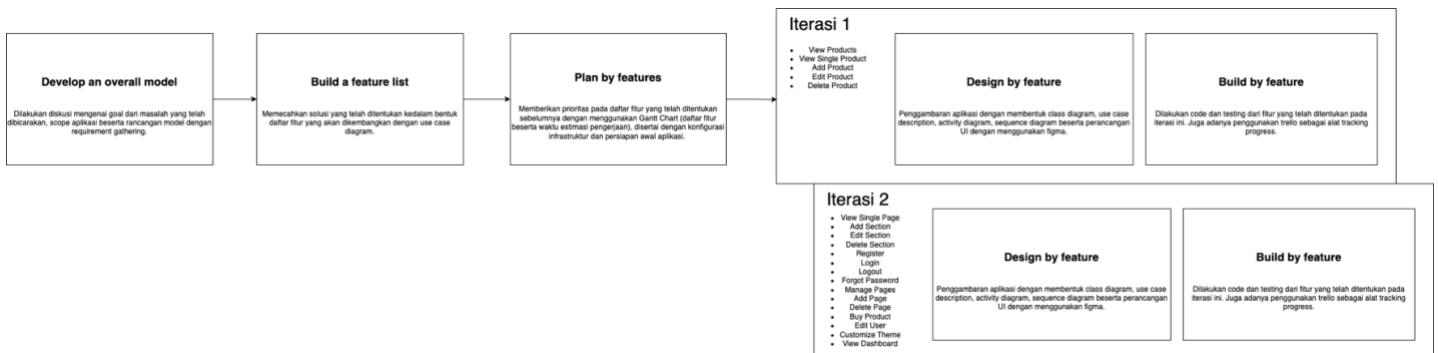
Pada tahap ini juga dilakukan konfigurasi dan penyediaan infrastruktur sistem untuk menunjang pengembangan aplikasi.

4. Design by feature

Pada tahap ini, proses desain yang dimaksud adalah penggambaran *class diagram*, *use case description*, *activity diagram*, dan *sequence diagram*. Selain itu, dilakukan juga *mockup* perancangan desain antarmuka (UI) aplikasi dengan menggunakan Figma.

5. Build by feature

Pada akhir tahapan FDD, pengembang melakukan *coding* dan *testing* aplikasi sesuai dengan bahasa pemrograman, framework, dan *tools* yang telah ditentukan. FDD yang digunakan memerlukan waktu 2 minggu sekali untuk melakukan *delivery* kepada *user* untuk di-*review* dan 1 minggu sekali untuk melakukan *update progress*. *Tools* seperti Trello juga digunakan untuk melakukan *tracking progress* (berbentuk list dan kartu yang dapat di-*assign* ke masing-masing anggota tim) setiap minggunya.



Gambar 3.1. Pelaksanaan FDD

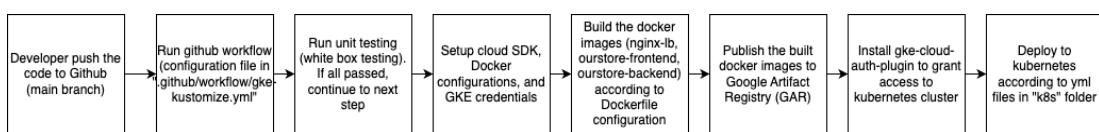
3.1.3 Testing

Semua tahap *testing* dilakukan mulai dari *unit testing*, *integration testing*, sampai dengan *end-to-end testing*. *Unit testing* dilakukan dengan membuat kode *testing* menggunakan *framework* Jest. *Unit testing* dijalankan pada CI/CD pipeline setelah *developer* melakukan *push* pada *code repository*. *Integration testing* dilakukan dengan menggunakan Postman. *Integration testing* melakukan *testing* kepada setiap API yang tersedia. *End-to-end testing* yaitu pelaksanaan *testing* pada *browser*. *End-to-end testing* dilakukan sesuai dengan *use case* yang dapat dilakukan oleh

actor mulai dari awal hingga akhir. *Integration testing* dan *end-to-end testing* termasuk bagian dari *black-box testing*.

3.1.4 CI/CD

Untuk mempercepat *delivery* terutama saat melakukan tahap *build by feature*, diterapkan *workflow Continuous Integration and Continuous Deployment* (CI/CD). CI/CD memudahkan proses *development* dengan melakukan automasi mulai dari kode sampai dengan *deployment* di *environment production*. Tool CI/CD yang digunakan adalah Github Action. Tahap *pipeline* automasi yang dilakukan di *project* ini adalah sebagai berikut.



Gambar 3.2. *Workflow* Github Action

3.2 Timeline

	2022				2023							
	November	Desember	Januari	Februari	Januari	Februari	Maret	April	Januari	Februari	Maret	April
	3	4	1	2	3	4	1	2	3	4	1	2
Develop an overall model												
Brainstorming ideas	■	■										
Requirement gathering	■	■										
Build a feature list												
Identify features			■									
Use case diagram				■								
Plan by feature					■							
Create gantt chart						■						
Provisioning development												
Github repos				■								
Trello					■							
API lists and flow						■						
Code and infrastructures												
MongoDB atlas			■		■							
Postman				■		■						
Godaddy					■	■						
GCP Infrastructure						■						
Github workflows							■					
Nginx proxy								■				
Design by feature												
Use Case Description					■							
Class Diagram						■						
Activity Diagram							■					
Sequence Diagram								■				
Perancangan UI									■			
Build by feature												
View Products							■		■			
Backend (API)												
Frontend												
View Single Product							■		■			
Backend (API)												
Frontend												
Add Product								■	■			
Backend (API)												
Frontend												
Edit Product								■	■			
Backend (API)												
Frontend												
Delete Product								■	■			
Backend (API)												
Frontend												
View Single Page									■			
Backend (API)												
Frontend												
Add Section									■	■		
Backend (API)												
Frontend												
Edit Section										■	■	
Backend (API)												
Frontend												
Delete Section									■	■		
Backend (API)												
Frontend												
Register									■	■		
Backend (API)												
Frontend												
Login									■	■		
Backend (API)												
Frontend												
Logout										■	■	
Backend (API)												
Frontend												
Forgot Password									■	■		
Backend (API)												
Frontend												
Manage Pages									■	■		
Backend (API)												
Frontend												
Add Page									■	■		
Backend (API)												
Frontend												
Delete Page										■	■	
Backend (API)												
Frontend												
Buy Product										■		
Backend (API)												
Frontend												
Edit User										■	■	
Backend (API)												
Frontend												
Customize Theme										■		
Backend (API)												
Frontend												
View Dashboard											■	
Backend (API)												
Frontend												
Testing											■	

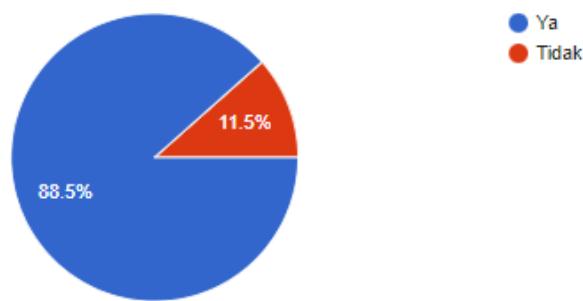
Tabel 3.2. *Timeline Pengembangan Aplikasi*

3.3 Analisis Kebutuhan *User*

3.3.1 Analisis Kebutuhan *User* berdasarkan Kuesioner

Metode kuesioner digunakan untuk mengidentifikasi masalah-masalah yang dialami oleh pengguna. Proses penyebaran kuesioner dilakukan secara daring dengan memanfaatkan *platform* Google Form. Waktu penyebaran form ini adalah dari minggu keempat bulan November 2022 sampai dengan minggu pertama bulan Desember 2022 selama 14 hari. Target kuesioner ini adalah para penjual pada *platform e-commerce*. Pada penelitian yang dilakukan, terkumpul hasil survei terhadap 26 responden. Pertanyaan dan hasil survei dijabarkan sebagai berikut.

1. Apakah fitur personalisasi *layout* produk sehingga *user interface* berbeda dari penjual yang lain dapat memberikan nilai tambah pada *branding*?

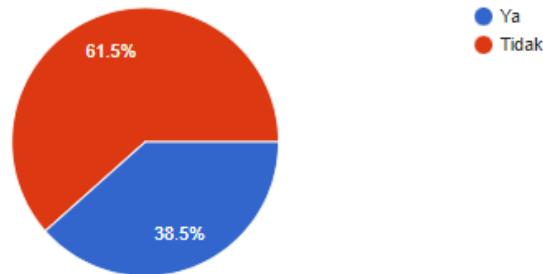


Gambar 3.3. Diagram Persentase Responden mengenai Fitur Personalisasi dapat memberikan Nilai Tambah pada *Branding*

Tabel 3.3. Hasil Survei mengenai Fitur Personalisasi dapat memberikan Nilai Tambah pada *Branding*

Jawaban	Jumlah Responden	Persentase
Ya	23	88.5%
Tidak	3	11.5%
Total	26	100%

2. Apakah sistem pemasaran lewat *email* dan SMS ke pelanggan dapat bersaing dengan sistem pemasaran *highlight* di *marketplace* pada umumnya?

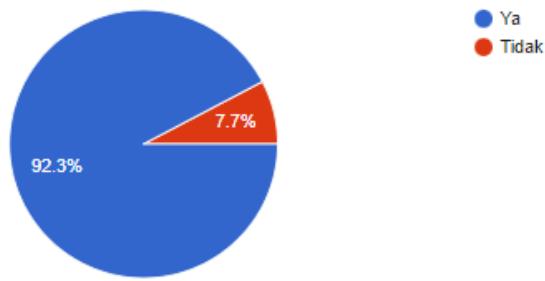


Gambar 3.4. Diagram Persentase Responden mengenai Pengembangan Sistem Pemasaran *Email* dan SMS untuk dapat bersaing dengan Sistem Pemasaran *Highlight*

Tabel 3.4. Hasil Survei mengenai Pengembangan Sistem Pemasaran *Email* dan SMS untuk dapat bersaing dengan Sistem Pemasaran *Highlight*

Jawaban	Jumlah Responden	Persentase
Tidak	16	61.5%
Ya	10	38.5%
Total	26	100%

3. Apakah adanya *dashboard monitoring* (jumlah pengunjung, jumlah produk terjual, dll) dapat membantu anda dalam menentukan pilihan dan rencana pengembangan toko?

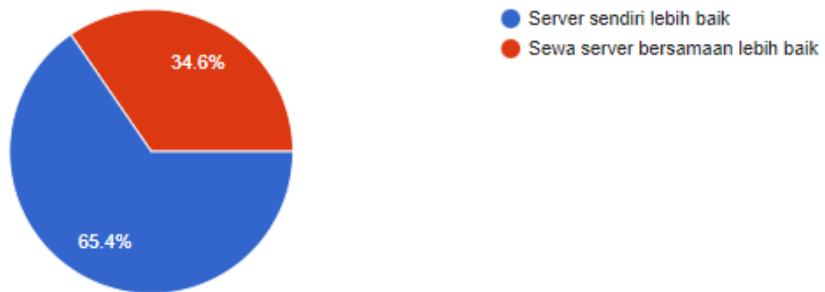


Gambar 3.5. Diagram Persentase Responden mengenai Keberadaan *Dashboard Monitoring* yang dapat Membantu Rencana Pengembangan Toko

Tabel 3.5. Hasil Survei mengenai Keberadaan *Dashboard Monitoring* yang dapat Membantu Rencana Pengembangan Toko

Jawaban	Jumlah Responden	Persentase
Ya	24	92.3%
Tidak	2	7.7%
Total	26	100%

4. Apakah dengan memiliki *server* aplikasi sendiri (dengan catatan lebih mahal dan lebih repot) lebih baik daripada menyewa *server* secara bersamaan? (dengan catatan apabila satu *server* mati maka semua *website* penjual juga akan mati)

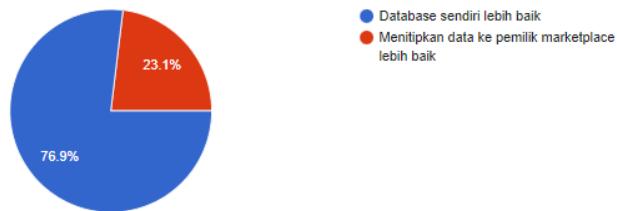


Gambar 3.6. Diagram Persentase Responden mengenai Pemilihan *Server* Aplikasi

Tabel 3.6. Hasil Survei mengenai Pemilihan *Server* Aplikasi

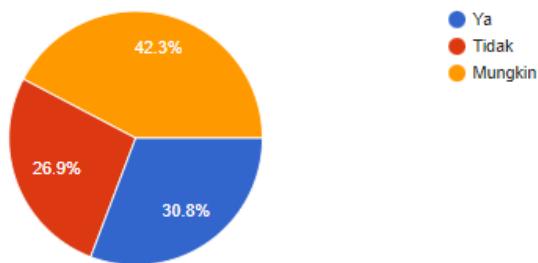
Jawaban	Jumlah Responden	Persentase
<i>Server</i> sendiri lebih baik	17	65.4%
Sewa <i>server</i> bersamaan lebih baik	9	34.6%
Total	26	100%

5. Apakah dengan memiliki *database* sendiri lebih baik (dengan catatan lebih mahal dan lebih repot) daripada menitipkan data ke pemilik *marketplace*? (dengan catatan apabila ada kebocoran data pada satu server akan berdampak pada semua penjual)

Gambar 3.7. Diagram Persentase Responden mengenai Pemilihan *Database* AplikasiTabel 3.7. Hasil Survei mengenai Pemilihan *Database* Aplikasi

Jawaban	Jumlah Responden	Persentase
<i>Database</i> sendiri lebih baik	20	76.9%
Menitipkan data ke pemilik <i>marketplace</i> lebih baik	6	23.1%
Total	26	100%

6. Apakah toko *online* membutuhkan fitur eksperimental (fitur tambahan yang dapat dikembangkan sendiri oleh *developer merchant*) sesuai dengan kebutuhan bisnis toko?

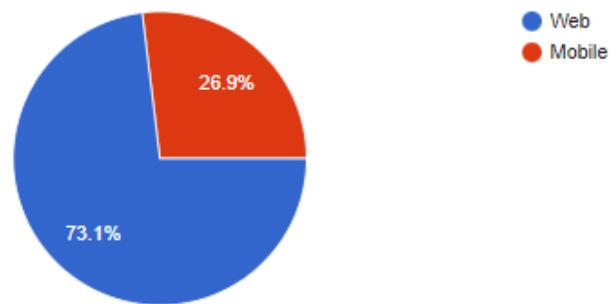


Gambar 3.8. Diagram Persentase Responden mengenai Kebutuhan untuk Mengembangkan Fitur Eksperimental sesuai Kebutuhan Bisnis Toko

Tabel 3.8. Hasil Survei mengenai Kebutuhan untuk Mengembangkan Fitur Eksperimental sesuai Kebutuhan Bisnis Toko

Jawaban	Jumlah Responden	Persentase
Ya	8	30.8%
Tidak	7	26.9%
Mungkin	11	42.3%
Total	26	100%

7. Apa *platform* yang lebih baik untuk aplikasi ini? *Web* atau *mobile*?



Gambar 3.9. Diagram Persentase Responden mengenai Pemilihan *Platform* Aplikasi

Tabel 3.9. Hasil Survei mengenai Pemilihan *Platform* Aplikasi

Jawaban	Jumlah Responden	Persentase
Web	19	73.1%
Mobile	7	26.9%
Total	26	100%

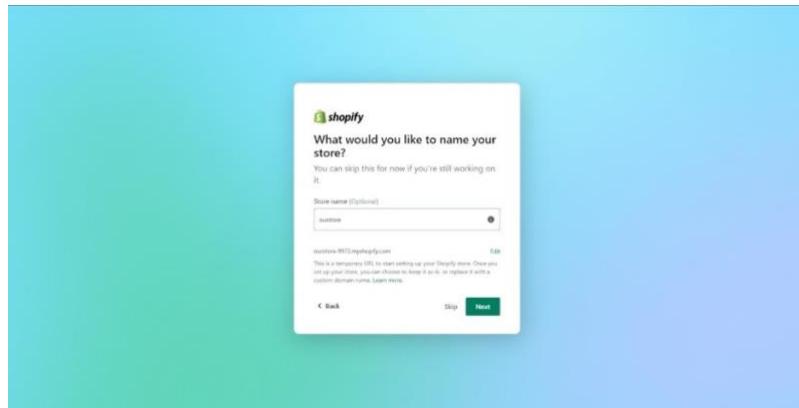
Berdasarkan data kuisioner yang diperoleh, dapat disimpulkan bahwa para pengguna khususnya penjual *online* menginginkan aplikasi berbasis *web* di mana aplikasi tersebut memperbolehkan pengguna untuk melakukan kustomisasi terhadap isi *website*-nya. Selain itu, mayoritas pengguna memilih untuk mengelola data dan *server* mereka sendiri, memiliki fitur *dashboard monitoring*, dan kebutuhan akan fitur eksperimental yang bebas dikembangkan oleh pengguna.

3.3.2 Analisis Aplikasi Sejenis

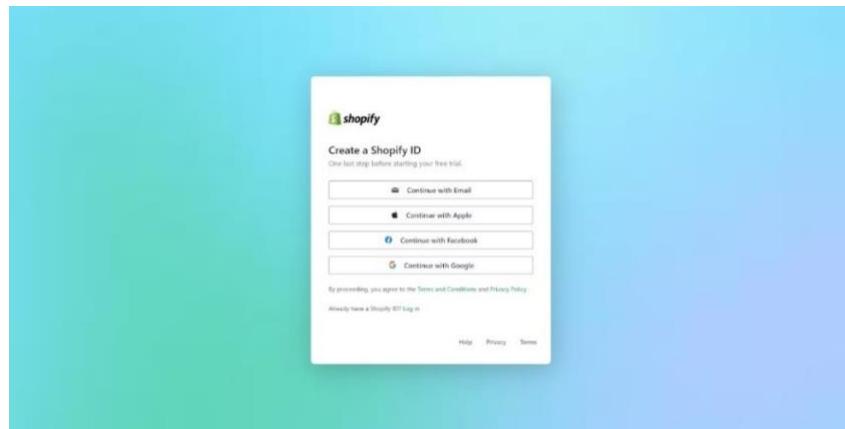
Sebagai bentuk metode pengumpulan data, observasi dilakukan terhadap aplikasi sejenis. Observasi ini dilakukan sebagai tambahan sumber data untuk mendukung hasil kuesioner yang telah diperoleh. Terdapat dua jenis aplikasi sejenis yaitu Shopify dan WordPress.

- **Shopify**

Shopify adalah *software e-commerce* berbasis *Software as a Service* (SaaS) yang populer. Shopify memungkinkan penjual untuk mendesain tampilan toko *online*-nya sendiri. Untuk menggunakan Shopify, pengguna diharuskan untuk membuat akun terlebih dahulu.

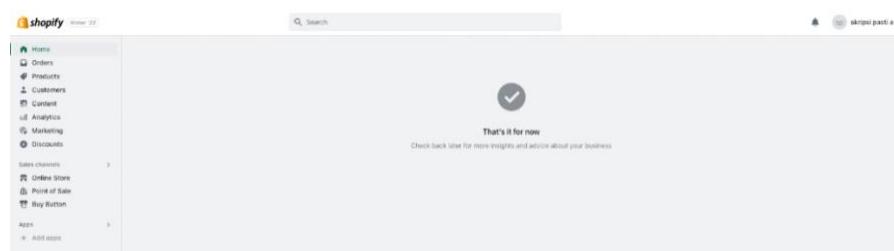


Gambar 3.10. Halaman *Register* Shopify 1



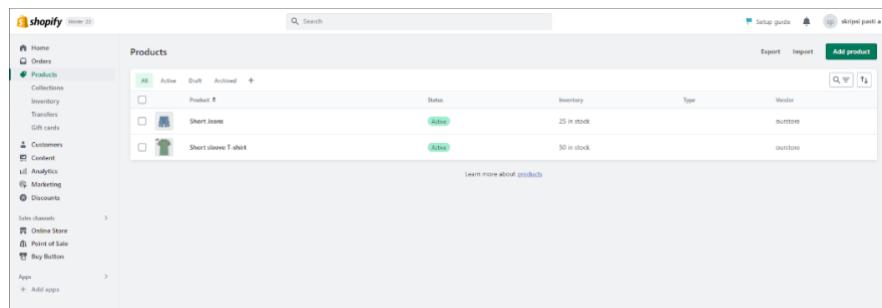
Gambar 3.11. Halaman *Register* Shopify 2

Setelah melakukan registrasi akun, pengguna akan diarahkan ke halaman *dashboard* Shopify. Pada dashboard terdapat beberapa menu seperti *product*, *domain*, *page*, *channel*, dan lainnya.

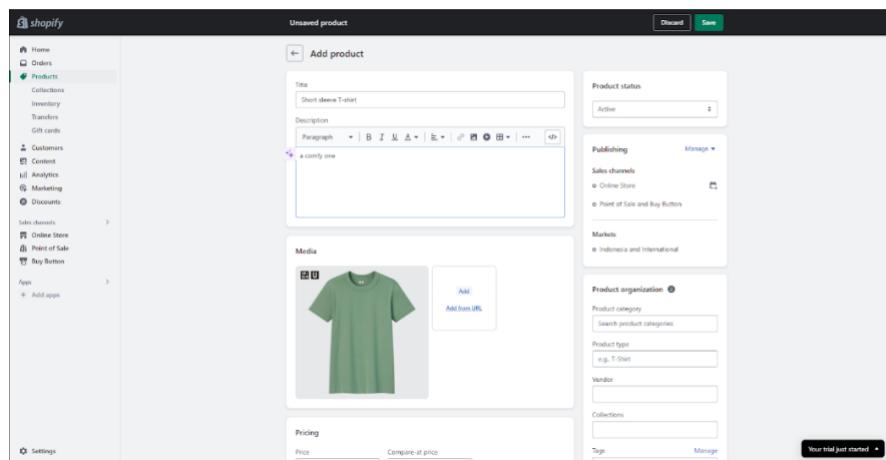


Gambar 3.12. Halaman *Dashboard* Shopify

Pada menu *product*, pengguna dapat melakukan manajemen produk dan juga menambahkan produk. Menambah produk dapat dilakukan dengan klik pada tombol “*Add product*”.

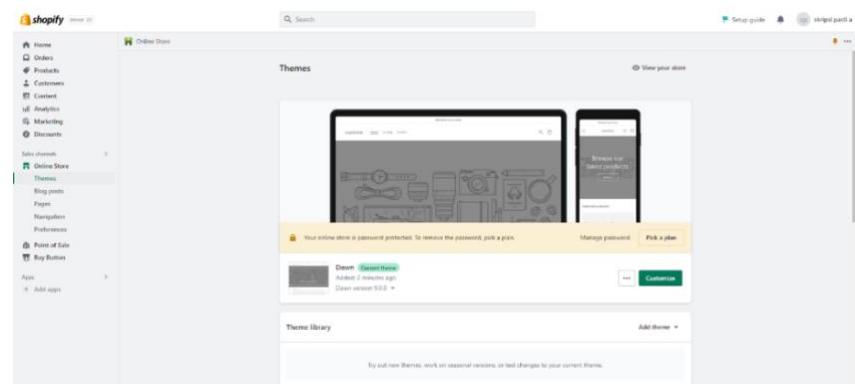


Gambar 3.13. Halaman *Product* Shopify

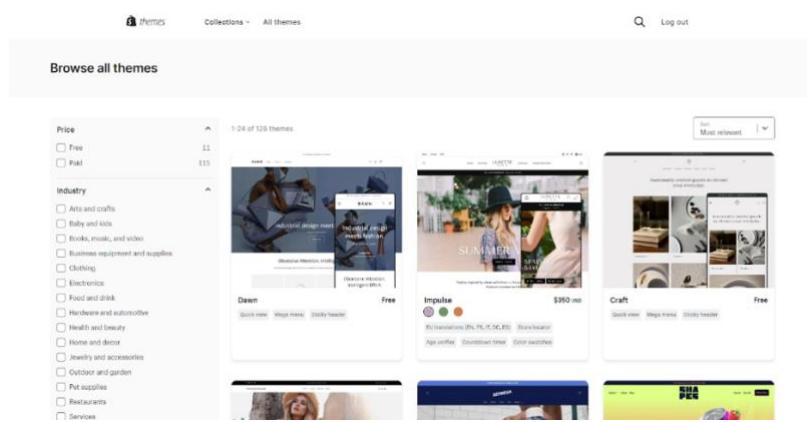


Gambar 3.14. Halaman *Add Product* Shopify

Selain produk, pengguna dapat mengganti tema dari *website* Shopify. Hal ini dapat dilakukan dengan menuju ke menu *Themes* yang ada pada *sidebar*. Pengguna juga dapat melakukan eksplorasi terhadap tema yang tersedia di *marketplace*.

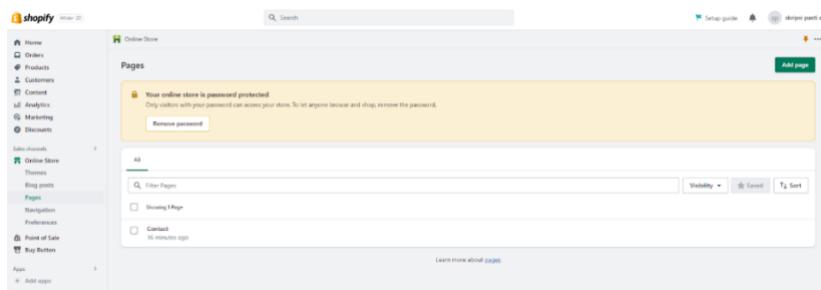


Gambar 3.15. *Theme* yang ter-*install* pada Shopify

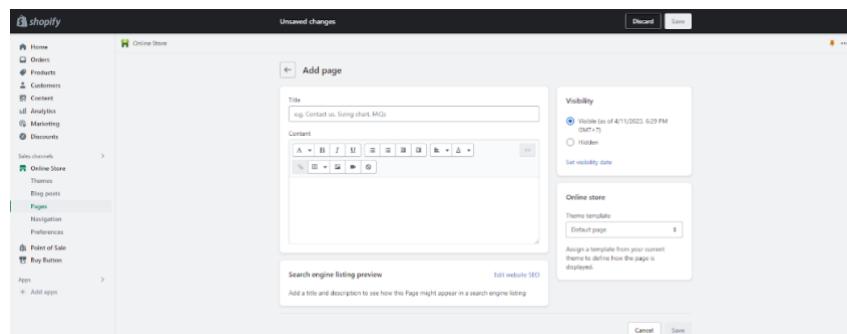


Gambar 3.16. *Theme* yang tersedia pada Shopify

Pengguna juga dapat melihat dan menambah halaman *web*. Caranya dengan menuju ke menu *Pages* pada menu *sidebar*. Menambah halaman *web* dapat dilakukan dengan klik tombol “*Add page*”.

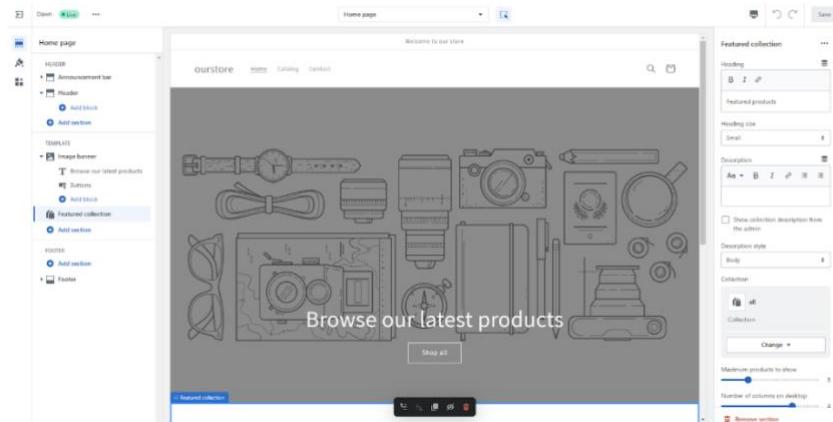


Gambar 3.17. Halaman *Pages* Shopify



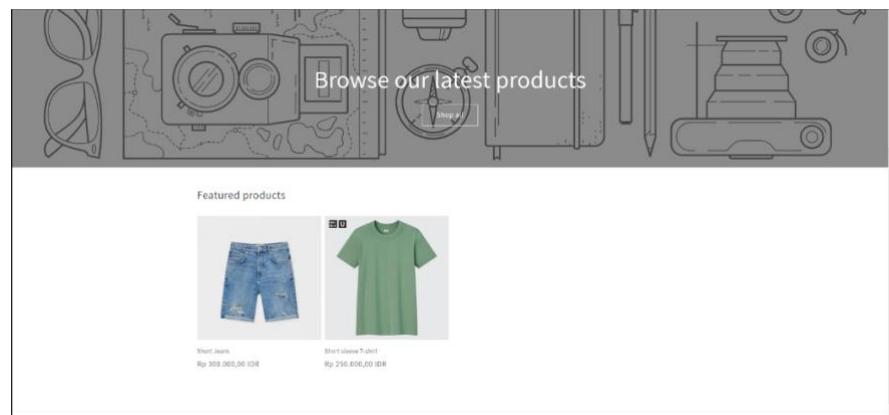
Gambar 3.18. Halaman *Add Page* Shopify

Terakhir, pengguna dapat mengubah isi halaman *web* dengan klik “*view your store*” di halaman *Theme*. Pengguna dapat mengubah teks, menambah *section*, melakukan *styling* pada teks, dan lain-lain.

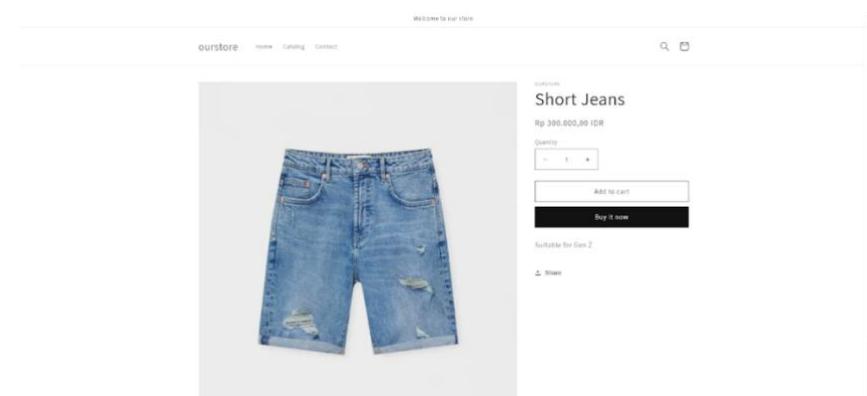


Gambar 3.19. Halaman *Edit Website* Shopify

Pelanggan dapat melihat *web* dengan menuju kepada link URL domain yang disediakan oleh Shopify dengan format <<nama toko>>-<<angka>>.myshopify.com. Pengguna juga dapat mengganti *domain* gratis ini sesuai yang diinginkan.



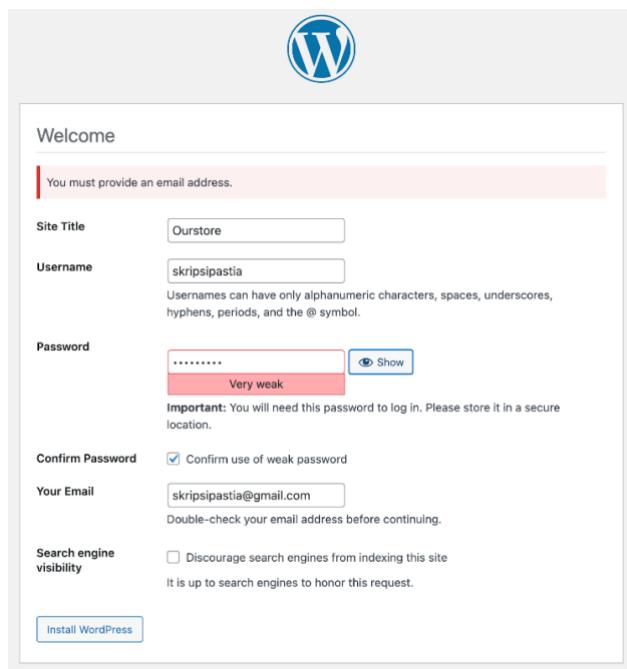
Gambar 3.20. Halaman *Website* Menggunakan Shopify 1



Gambar 3.21. Halaman *Website* Menggunakan Shopify 2

- **WordPress**

WordPress adalah salah satu *software open-source Content Management System* (CMS) yang paling populer dan gratis. WordPress dibuat menggunakan bahasa pemrograman PHP dan menggunakan *database MySQL* atau MariaDB. Instalasi WordPress dapat dilakukan dengan berbagai cara. Cara yang paling umum adalah melalui *hosting provider* yang langsung terekspos di *internet*. Cara lain dapat melakukan Docker agar dapat di-*install* secara lokal sehingga seluruh *environment* WordPress dapat dimiliki dan dieksplorasi. Pada saat mengakses WordPress melalui *browser*, sistem akan menampilkan halaman instalasi. Beberapa *field* perlu dimasukkan di halaman ini seperti *site title*, *username*, *email*, dan *password*.



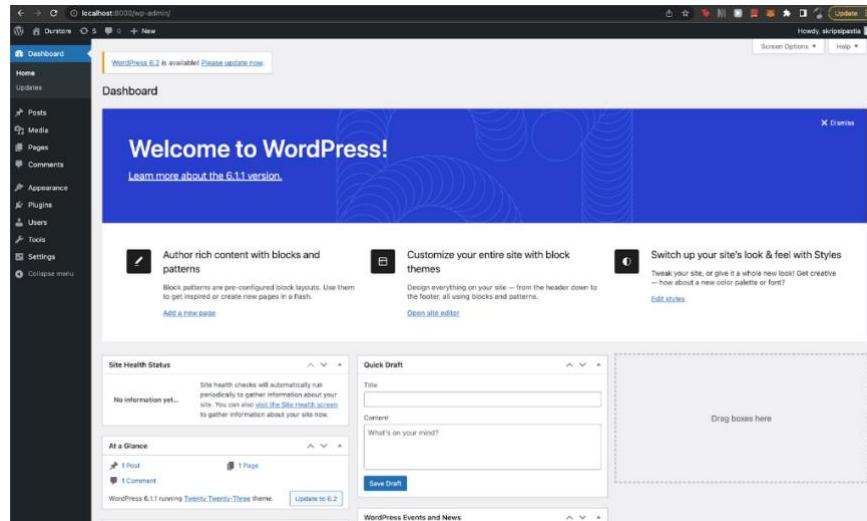
Gambar 3.22. Halaman Instalasi Wordpress

Setelah WordPress terinstalasi, pengguna dapat melakukan *login* sesuai dengan *username* dan *password* yang telah di-*input* di halaman instalasi sebelumnya.



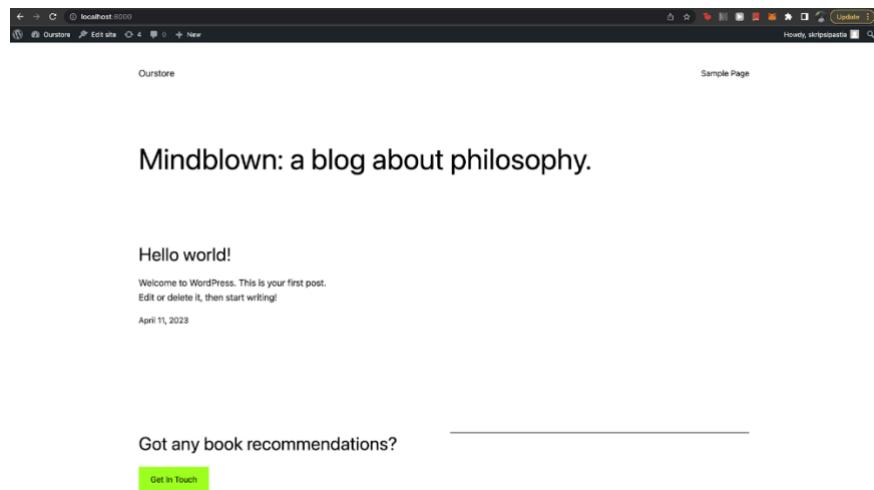
Gambar 3.23. Halaman *Login* Wordpress

Setelah pengguna (*admin*) melakukan *login*, WordPress menampilkan halaman *dashboard* yang terdiri dari beberapa menu pada *sidebar*.



Gambar 3.24. Halaman *Dashboard* Wordpress

Untuk menuju pada halaman *web* yang asli, pengguna dapat menuju menu *home* pada *bar* yang terletak di atas, kemudian memilih *visit site*.



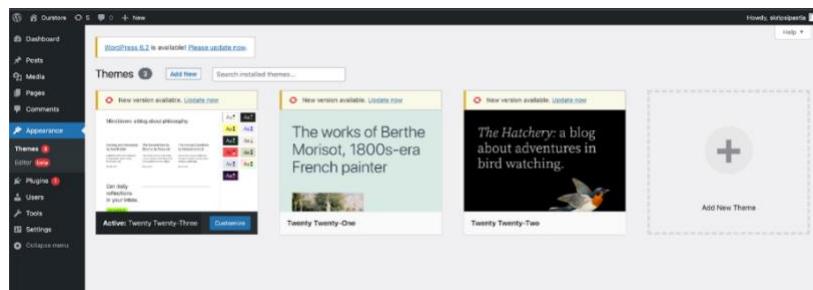
Gambar 3.25. Halaman *Website* yang Dibuat oleh Wordpress

Pengguna juga dapat melakukan perubahan halaman dengan melakukan klik tombol *edit site* pada bar yang terletak di menu atas.

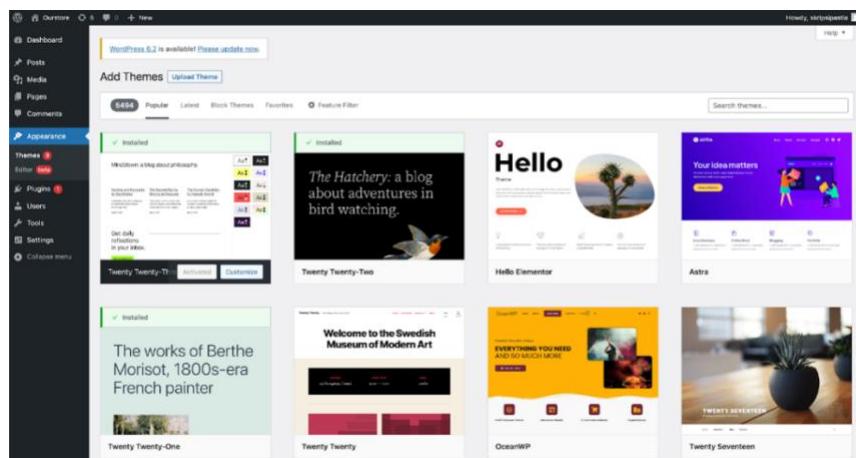


Gambar 3.26. *Edit* Halaman *Website* Wordpress

Ada beberapa tema juga yang dapat dipakai sehingga halaman dapat terkustomisasi sesuai dengan keinginan pengguna. Untuk melihat tema apa saja yang ter-*install* atau tersedia dapat melalui menu *apprearance* → *theme*.

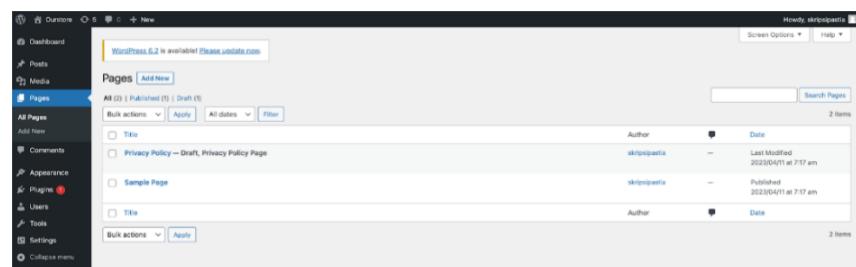


Gambar 3.27. *Theme* yang Ter-*install* pada Wordpress



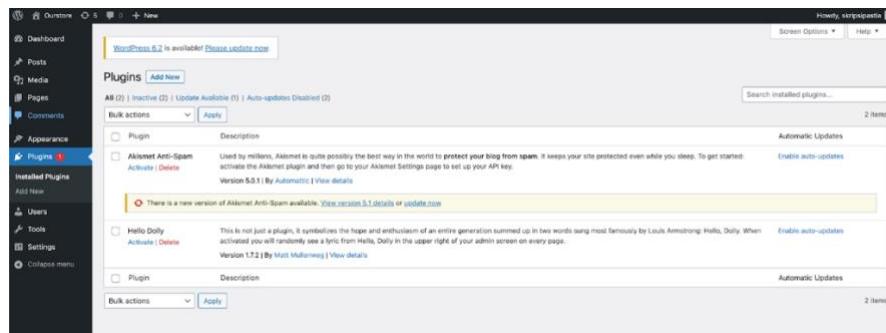
Gambar 3.28. *Theme* yang Tersedia pada Wordpress

Website ini adalah satu dari beberapa halaman yang dapat dibuat melalui WordPress. Pengguna dapat melakukan manajemen halaman melalui menu yang terdapat pada *sidebar*.

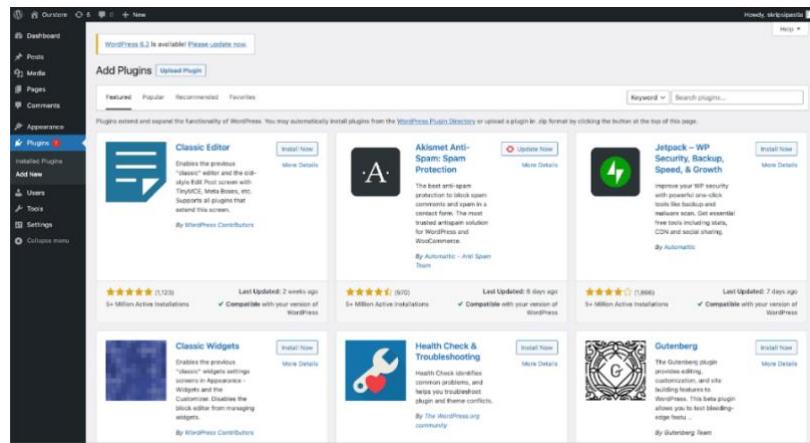


Gambar 3.29. Manajemen Menu Halaman *Dashboard* Wordpress

Fitur-fitur tambahan dapat di-*install* melalui *plugins*. Menu untuk melakukan manajemen *plugins* terdapat pada *sidebar*. Di sini pengguna dapat melihat *plugins* apa saja yang ter-*install* dan tersedia untuk ditambahkan pada *website*.



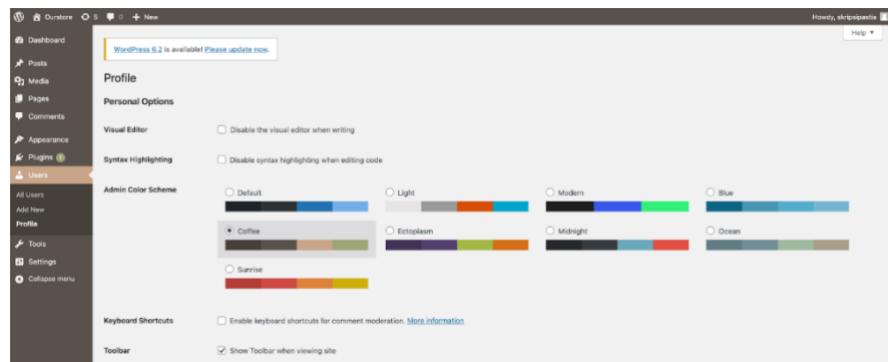
Gambar 3.30. *Plugins* yang Ter-install pada Wordpress



Gambar 3.31. *Plugins* yang Tersedia pada Wordpress

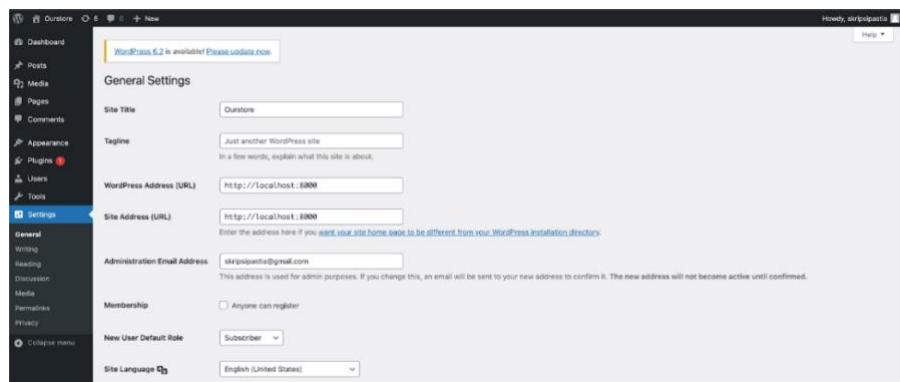
Plugins tersebut dikembangkan oleh *developer* independen dan ada yang gratis dan juga berbayar. *Developer* independen tidak memahami secara spesifik mengenai masalah atau *requirement* yang dimiliki oleh penjual. *Plugins* ini juga hanya dapat dikembangkan menggunakan bahasa pemrograman PHP.

Selain mengganti tema pada *website*, pengguna (*admin*) juga dapat mengganti tema pada tampilan *dashboard*. Cara menggantinya adalah dengan melalui menu *Users* → *Profile*. Di menu ini pengguna juga dapat mengganti nama, *email*, dan *username*.



Gambar 3.32. Menu *Profile Users* pada Dashboard Wordpress

Terakhir, pengguna (*admin*) juga dapat mengganti judul *website* dan juga URL *website* pada menu pengaturan (*settings*) yang terdapat pada *sidebar*.



Gambar 3.33. Menu *Settings* pada Dashboard Wordpress

Tabel 3.10. Analisa Perbandingan Aplikasi Sejenis

	Shopify	WordPress
Fitur personalisasi	V	V
Fitur <i>dashboard monitoring</i>	V	V
<i>Server</i> sendiri	X	V
<i>Database</i> sendiri	X	V
Fitur eksperimental oleh pemilik toko	X	X

3.3.3 Usulan atau Solusi Pemecahan Masalah

Dengan mengidentifikasi masalah-masalah dari pengguna dan melalui analisis aplikasi sejenis, dapat disimpulkan bahwa solusi yang diusulkan adalah sebagai berikut.

1. Membuat aplikasi CMS yang memiliki fitur-fitur utama yang dimiliki oleh *e-commerce* lain pada umumnya.
2. Membuat aplikasi CMS yang memiliki fitur personalisasi produk.
3. Membuat aplikasi CMS yang memiliki fitur *dashboard monitoring*.
4. Membuat aplikasi CMS yang *di-deploy* pada *server* sendiri.
5. Membuat aplikasi CMS yang memiliki *database server* sendiri.
6. Membuat aplikasi CMS yang menyediakan *platform* bagi *developer* untuk mengembangkan fitur spesifik dengan cepat.

3.4 Kebutuhan Teknologi

Dengan analisis dari masalah pengguna, membandingkan aplikasi sejenis, sampai pada menjabarkan usulan/solusi yang ditawarkan, maka terdapat kebutuhan teknologi untuk mendukung solusi tersebut. Teknologi yang dibutuhkan dapat dijabarkan sebagai berikut.

1. Project

Tool kanban-style seperti Trello dibutuhkan untuk melakukan *tracking progress project* secara periodik dengan bentuk kartu, *list*, dan *board*. *Developer* juga dapat menambahkan *backlog* pada list yang tersedia dan setiap kartu dapat di-*assign* ke masing-masing *developer*. Google Sheet dan Google Docs juga digunakan untuk melakukan *collaborative working* secara *real time*. *Discord* digunakan untuk berkomunikasi antar *developer*. *Discord* digunakan karena terdapat fitur *channel* yang membagi *thread* komunikasi sehingga konteks komunikasi dapat terbagi dengan baik. Selain itu, terdapat fitur *webhook* yang berguna untuk melakukan *tracking commit* pada Github (SCM).

2. Source Code Management (SCM)

SCM *tool* yang digunakan adalah Github karena paling populer. Selain itu, arsitektur Github yang *centralised (cloud)* dan *distributed (local git)* juga memudahkan *collaborative working* karena ada *centralised*

repository dan lebih fleksibel karena bebas melakukan *commit* di *local* yang tidak mengganggu *developer* lain.

3. *Wireframing UI*

Tool untuk membuat rancangan UI yang digunakan adalah Figma. Figma adalah UI *design tool* paling populer yang berbasis *web* dan *cloud* sehingga memungkinkan *collaborative working*.

4. *Development Tool*

Visual Studio Code digunakan sebagai *code editor* karena memiliki komunitas yang luas dan terdapat *extension* yang lengkap seperti *Live Server*, *REST client*, dan Docker. Selain itu, Visual Studio Code juga terintegrasi dengan *SCM tool* dan terdapat *built-in terminal* yang memudahkan untuk *running* dan *debugging*. Postman sebagai API *Platform* yang paling populer digunakan untuk melakukan *integration testing* pada API sekaligus melakukan *list* API apa saja yang tersedia. *Package manager* yang digunakan adalah Node Package Manager (NPM) karena bahasa pemrograman yang digunakan adalah JavaScript (NodeJS).

5. *Programming Language and Framework*

Framework yang digunakan adalah MERN stack yang terdiri dari MongoDB, ExpressJS, ReactJS, dan NodeJS. MongoDB sebagai *database*, NodeJS dan ExpressJS sebagai *backend framework*, dan ReactJS sebagai *frontend framework*. Bahasa pemrograman yang digunakan adalah JavaScript. *Framework* ini digunakan karena paling populer di komunitas *developer* untuk *web development* sehingga *framework* ini sering diperbarui. Kelebihan dari *framework* ini adalah *development* yang cepat, mudah, dan efisien karena hanya memakai satu bahasa pemrograman untuk *backend* dan *frontend* sehingga *learning curve* tidak terlalu tinggi. Selain itu, *driver database* untuk MongoDB dan beberapa *third party tool* untuk NodeJS/ExpressJS juga tersedia dan terdokumentasi dengan baik.

6. *Database*

Requirement database dapat dijabarkan sebagai berikut. Perubahan skema *database* harus dapat dilakukan dengan mudah karena *user* dapat sewaktu-waktu meminta penambahan atau perubahan fitur. Data yang

disimpan juga bukan merupakan *analytical*, melainkan *transactional* sehingga tidak perlu menggunakan *columnar database*. Data yang disimpan juga cukup kompleks sehingga tidak dapat menggunakan *key-value store*. *Database server* juga dapat diskalakan dengan mudah secara *horizontal* dan mementingkan *consistency* daripada *availability* karena menyangkut harga produk. Dapat disimpulkan bahwa, MongoDB digunakan karena MongoDB merupakan *document-based* NoSQL *database* yang paling populer.

7. *Third Party Tool*

Fitur-fitur seperti pengiriman *email* dan pembayaran tidak bisa dikembangkan secara *in-house* karena sangat kompleks dan membutuhkan banyak *layer* sehingga dibutuhkan *third party software* untuk fitur ini. Nodemailer adalah *tool* untuk mengirim *email* yang terintegrasi dengan NodeJS (tersedia di NPM). Stripe adalah *payment gateway* dan *payment processing platform* yang paling populer. Dependensi untuk Stripe juga tersedia pada NPM.

8. *CI/CD Tool*

Terdapat dua pembagian jenis CI/CD *tool*, yaitu *public (cloud-based)* dan juga *private*. *Public CI/CD tool* digunakan untuk projek ini karena kemudahan dan kecepatannya (tidak perlu melakukan instalasi dan konfigurasi yang banyak). Github workflow adalah *public CI/CD tool* yang digunakan karena terintegrasi dengan SCM Github sehingga tidak perlu membuat akun baru lagi (lebih mudah).

9. *Domain Name System (DNS)*

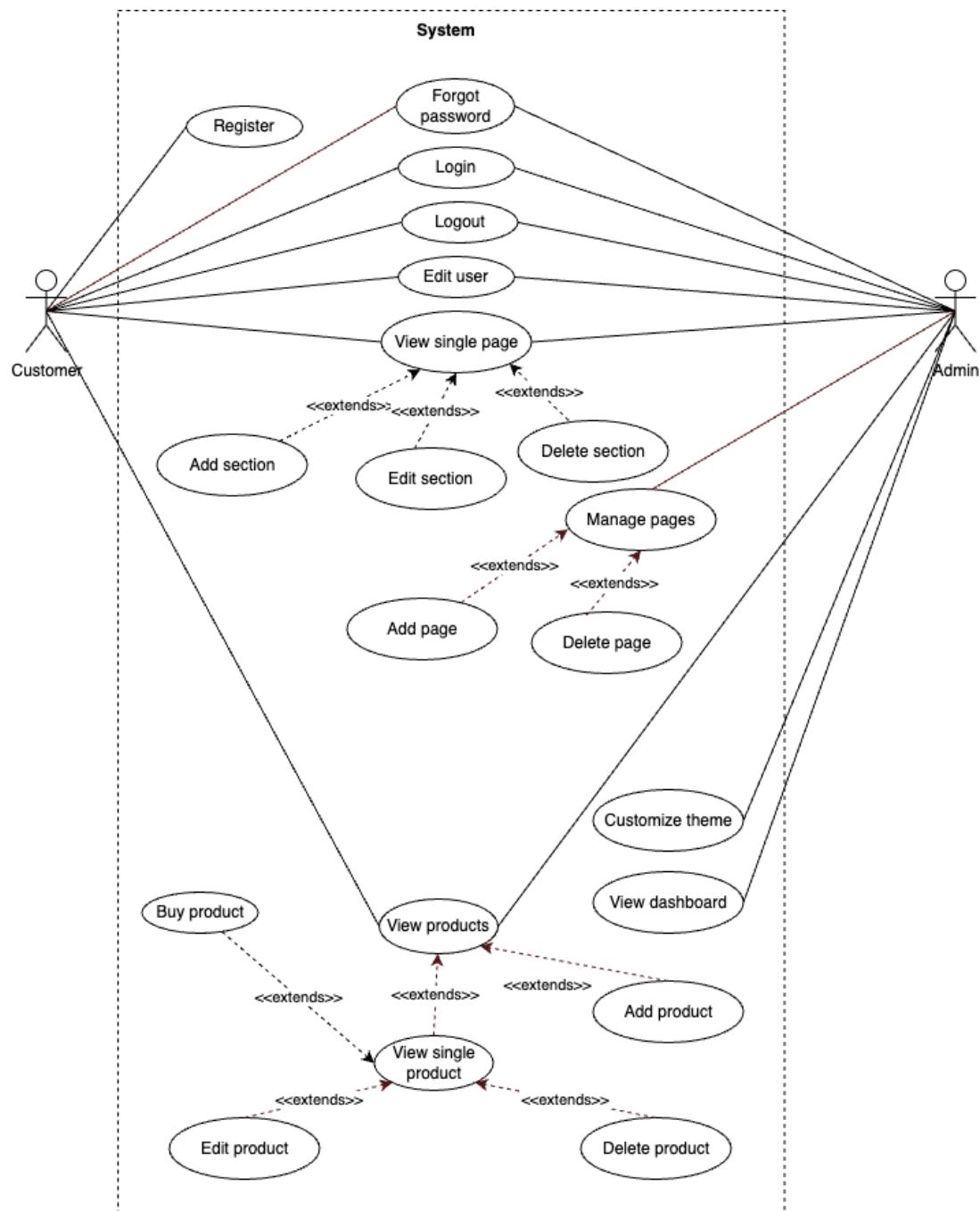
User kesulitan apabila memasukkan *input* berupa *IP address* pada *browser* karena hanya terdiri dari angka-angka yang lebih sulit untuk diingat dan situs *web* menjadi kurang kredibel. Dengan adanya DNS memungkinkan *user* untuk memasukkan nama *domain* saja daripada *IP address*. DNS juga memungkinkan untuk menambahkan sertifikat pada situs *web* sehingga koneksi lebih aman dan data terenkripsi. Untuk mendaftarkan nama *domain* perlu mendaftarkannya terlebih dahulu pada perusahaan *registrar*. Salah satu perusahaan *registrar* yang populer adalah GoDaddy dan digunakan juga untuk projek ini.

10. *Deployment*

Aplikasi ini ditujukan untuk pelaku bisnis *e-commerce* skala menengah dan kemungkinan untuk terjadinya lonjakan *traffic* tinggi sehingga diperlukan teknologi untuk melakukan *scaling* secara horizontal dan *real time*. Salah satu teknologi yang populer adalah Kubernetes. Kubernetes terdiri dari *pods* di mana setiap *pods* terdapat *container runtime* yang berjalan. Teknologi *container* yang digunakan adalah Docker karena merupakan *containerization tool* yang paling populer di kalangan *developer*. Melakukan *setup* dan instalasi Kubernetes tidak mudah sehingga produk dari *public cloud service* seperti Google Kubernetes Engine (GKE) juga digunakan.

3.5 Perancangan Sistem

3.5.1 Use Case Diagram



Gambar 3.34. Use Case Diagram

3.5.2 Use Case Description

3.5.2.1 Register

Tabel 3.11. *Use Case Description Register*

Use Case Name	Register
ID	UC01
Importance Level	High
Primary Actor	Customer
Use case type	Detail, Essential
Stakeholders and interest	
Customer - ingin mendaftarkan akun	
Brief description	
Use case ini menjelaskan bagaimana user melakukan registrasi akun pada aplikasi.	
Trigger	Type
User melakukan klik tombol "sign up" pada halaman Login.	External
Relationship	
Association: Customer	
Normal flow of events	
<ol style="list-style-type: none"> 1. User mengisi form yang berisi email, username, password, dan confirm password. 2. Sistem mengirim OTP kepada email user dan menampilkan halaman OTP. 3. User mengisi field OTP. 4. Sistem menampilkan halaman Products. 	
Alternate flow	
<ol style="list-style-type: none"> 2a. Apabila ada kesalahan input pada form, sistem akan menampilkan pesan error. 4a. Apabila OTP yang diisi user salah, sistem akan menampilkan pesan error. 	

3.5.2.2 Login

Tabel 3.12. *Use Case Description Login*

Use Case Name	Login
ID	UC02
Importance Level	High
Primary Actor	Customer, Admin
Use case type	Detail, Essential
Stakeholders and interest	
Customer, Admin - ingin masuk ke aplikasi	
Brief description	
Use case ini menjelaskan bagaimana user masuk ke dalam sistem menggunakan akun yang telah terdaftar untuk melakukan aktivitas..	
Trigger	Type
User menuju ke halaman Login.	External
Relationship	
Association: Customer, Admin	
Normal flow of events	
1. User mengisi form login yang berisi username dan password. 2. Sistem menampilkan halaman Products.	
Alternate flow	
2a. Apabila user mengisi username dan password yang salah, sistem akan menampilkan pesan error. 2b. Apabila user merupakan admin, sistem akan menampilkan halaman Dashboard.	

3.5.2.3 Forgot Password

Tabel 3.13. *Use Case Description Forgot Password*

Use Case Name	Forgot Password			
ID	UC03			
Importance Level	Medium			
Primary Actor	Customer, Admin			
Use case type	Detail, Real			
Stakeholders and interest				
Customer, Admin - ingin melakukan reset password karena user lupa password				
Brief description				
Use case ini menjelaskan bagaimana user melakukan reset password karena user lupa password yang berakibat user tidak dapat masuk ke sistem.				
Trigger	Type			
User melakukan klik tombol "forgot password" pada halaman Login.	External			
Relationship				
Association: Customer, Admin				
Normal flow of events				
<ol style="list-style-type: none"> 1. User mengisi form yang berisi email dan klik tombol "send". 2. Sistem mengirim reset token berupa link ke email yang diisi dan mengirimkan pesan sukses. 3. User masuk ke email dan klik pada link yang telah dikirim oleh sistem. 4. Sistem menampilkan halaman Reset. 5. User mengisi form yang berisi password dan confirm password, kemudian klik tombol "reset". 6. Sistem menampilkan pesan sukses. 				
Alternate flow				
4a. Apabila link yang diklik oleh user sudah expired atau tidak valid, sistem akan menampilkan pesan error. 6b. Apabila form yang diisi oleh user tidak valid, sistem akan menampilkan pesan error.				

3.5.2.4 Logout

Tabel 3.14. *Use Case Description Logout*

Use Case Name	Logout
ID	UC04
Importance Level	Low
Primary Actor	Customer, Admin
Use case type	Detail, Real
Stakeholders and interest	
Customer, Admin - ingin keluar dari aplikasi (logout)	
Brief description	
Use case ini menjelaskan bagaimana user melakukan logout pada sistem.	
Trigger	Type
User melakukan klik tombol "logout" pada header di beberapa halaman.	External
Relationship	
Association: Customer, Admin	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menghapus cookie pada browser. 2. Sistem menampilkan halaman Login. 	

3.5.2.5 Edit User

Tabel 3.15. *Use Case Description Edit User*

Use Case Name	Edit User
ID	UC05
Importance Level	Low
Primary Actor	Customer, Admin
Use case type	Detail, Real
Stakeholders and interest	
Customer, Admin - ingin mengubah email dan/atau username	
Brief description	
Use case ini menjelaskan bagaimana user mengubah email dan/atau username melalui menu Edit User.	
Trigger	Type
User melakukan klik tombol "Edit user" pada header di beberapa halaman.	External
Relationship	
Association: Customer, Admin	
Normal flow of events	
<ol style="list-style-type: none"> 1. User mengisi form yang telah diisi dengan current email dan username. 2. User melakukan klik tombol "save". 3. Sistem menampilkan pesan sukses. 	
Alternate flow	
3a. Apabila form yang diisi oleh user tidak valid, sistem akan menampilkan pesan error.	

3.5.2.6 View Single Page

Tabel 3.16. *Use Case Description View Single Page*

Use Case Name	View Single Page
ID	UC06
Importance Level	High
Primary Actor	Customer, Admin
Use case type	Detail, Essential
Stakeholders and interest	
Customer, Admin - ingin melihat atau mengubah (admin) halaman	
Brief description	
Use case ini menjelaskan bagaimana customer melihat suatu halaman atau bagaimana admin melihat atau mengubah suatu halaman.	
Trigger	Type
User menuju ke suatu halaman.	External
Relationship	
Association: Customer, Admin	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan halaman tertentu. 2. Sistem menampilkan tombol untuk melakukan manajemen pada halaman tertentu. 	
Alternate flow	
<ol style="list-style-type: none"> 1a. Apabila URL halaman tidak valid, sistem akan menampilkan pesan 404 error. 2a. Apabila user belum melakukan login atau user adalah customer, sistem tidak akan menampilkan tombol yang digunakan untuk mengatur halaman tertentu. 	

3.5.2.7 Add Section

Tabel 3.17. Use Case Description Add Section

Use Case Name	Add Section
ID	UC07
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin menambahkan section pada suatu halaman	
Brief description	
Use case ini menjelaskan bagaimana admin menambahkan section pada suatu halaman	
Trigger	Type
Admin klik pada tombol "Add section" di halaman View Single Page.	External
Relationship	
Association: Admin Extension: View single page	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan form "add section". 2. Admin mengisi form yang berisi column size, column type, header, content, dan lainnya. 3. Admin klik tombol "add". 4. Sistem menampilkan pesan sukses. 	
Alternate flow	
4a. Apabila form yang diisi oleh admin tidak valid, sistem akan menampilkan pesan error.	

3.5.2.8 Edit Section

Tabel 3.18. *Use Case Description Edit Section*

Use Case Name	Edit Section
ID	UC08
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin mengubah section pada suatu halaman	
Brief description	
Use case ini menjelaskan bagaimana admin mengubah section pada suatu halaman.	
Trigger	Type
Admin klik section spesifik pada suatu halaman	External
Relationship	
Association: Admin Extension: View single page	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan form "edit section". 2. Admin mengisi form yang telah diisi dengan column size, column type, header, content, dan lainnya. 3. Admin klik tombol "save". 4. Sistem menampilkan pesan sukses. 	
Alternate flow	
4a. Apabila form yang diisi oleh admin tidak valid, sistem akan menampilkan pesan error.	

3.5.2.9 Delete Section

Tabel 3.19. *Use Case Description Delete Section*

Use Case Name	Delete Section
ID	UC09
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin menghapus section pada suatu halaman	
Brief description	
Use case ini menjelaskan bagaimana admin menghapus section pada suatu halaman.	
Trigger	Type
Admin klik pada tombol "Delete section" saat form "Edit secton" muncul.	External
Relationship	
Association: Admin Extension: View single page	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan dialog konfirmasi. 2. Admin klik tombol "confirm". 3. Sistem menampilkan pesan sukses. 	

3.5.2.10 Manage Pages

Tabel 3.20. *Use Case Description Manage Pages*

Use Case Name	Manage Pages
ID	UC10
Importance Level	High
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin melihat dan mengatur halaman	
Brief description	
Use case ini menjelaskan bagaimana admin melihat dan mengatur halaman	
Trigger	Type
Admin menuju halaman Manage Page.	External
Relationship	
Association: Admin	
Normal flow of events	
1. Sistem menampilkan halaman Manage Page.	
Alternate flow	
1a. Apabila user masih belum melakukan login, sistem akan membawa ke halaman Login. 1b. Apabila role user bukan admin, sistem akan menampilkan pesan error unauthorized.	

3.5.2.11 Add Page

Tabel 3.21. *Use Case Description Add Page*

Use Case Name	Add Page
ID	UC11
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin menambah suatu halaman.	
Brief description	
Use case ini menjelaskan bagaimana admin menambah suatu halaman.	
Trigger	Type
Admin klik pada tombol "Add page" pada halaman Manage Page.	External
Relationship	
Association: Admin Extension: Manage pages	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan form "Add page". 2. Admin mengisi form yang berisi title, checkbox active, dan checkbox add to header, kemudian klik tombol "Add". 3. Sistem menampilkan pesan sukses. 	
Alternate flow	
3a. Apabila form yang diisi oleh admin tidak valid, sistem akan menampilkan pesan error.	

3.5.2.12 Delete Page

Tabel 3.22. *Use Case Description Delete Page*

Use Case Name	Delete Page
ID	UC12
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin menghapus suatu halaman.	
Brief description	
Use case ini menjelaskan bagaimana admin menghapus suatu halaman.	
Trigger	Type
Admin klik pada tombol "Delete page" pada suatu item halaman di halaman Manage Page.	External
Relationship	
Association: Admin Extension: Manage pages	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan form "Add page". 2. Admin mengisi form yang berisi title, checkbox active, dan checkbox add to header, kemudian klik tombol "Add". 3. Sistem menampilkan pesan sukses. 	
Alternate flow	
3a. Apabila form yang diisi oleh admin tidak valid, sistem akan menampilkan pesan error.	

3.5.2.13 View Products

Tabel 3.23. *Use Case Description View Products*

Use Case Name	View Products
ID	UC13
Importance Level	High
Primary Actor	Customer, Admin
Use case type	Detail, Essential
Stakeholders and interest	
Customer, Admin - ingin melihat produk-produk yang tersedia	
Brief description	
Use case ini menjelaskan bagaimana user melihat produk-produk yang tersedia.	
Trigger	Type
User menuju ke halaman Products.	External
Relationship	
Association: Customer, Admin	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan halaman Products. 2. Sistem menampilkan tombol untuk melakukan manajemen pada halaman Products. 	
Alternate flow	
1a. Apabila user belum melakukan login atau user adalah customer, sistem tidak akan menampilkan tombol untuk manajemen produk.	

3.5.2.14 Add Product

Tabel 3.24. *Use Case Description Add Product*

Use Case Name	Add Product
ID	UC14
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin menambah suatu produk	
Brief description	
Use case ini menjelaskan bagaimana admin menambahkan suatu produk untuk ditawarkan kepada customer.	
Trigger	Type
Admin klik tombol "Add product" pada halaman Products	External
Relationship	
Association: Admin Extension: View Products	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan form "Add product". 2. Admin mengisi form yang berisi nama, harga, deskripsi, dan gambar-gambar produk, kemudian klik tombol "Add". 3. Sistem menampilkan pesan sukses. 	
Alternate flow	
3a. Apabila form yang diisi oleh admin tidak valid, sistem akan menampilkan pesan error.	

3.5.2.15 View Single Product

Tabel 3.25. *Use Case Description View Single Product*

Use Case Name	View Single Product
ID	UC15
Importance Level	High
Primary Actor	Customer, Admin
Use case type	Detail, Essential
Stakeholders and interest	
Customer, Admin - ingin melihat suatu produk yang ditawarkan.	
Brief description	
Use case ini menjelaskan bagaimana user melihat suatu produk yang ditawarkan pada katalog.	
Trigger	Type
User menuju ke halaman Products.	External
Relationship	
Association: Customer, Admin Extension: View Products	
Normal flow of events	
1. Sistem menampilkan halaman Products. 2. Sistem menampilkan tombol untuk melakukan manajemen pada halaman Products.	
Alternate flow	
1a. Apabila user belum melakukan login atau user adalah customer, sistem tidak akan menampilkan tombol untuk manajemen produk.	

3.5.2.16 Edit Product

Tabel 3.26. *Use Case Description Edit Product*

Use Case Name	Edit Product
ID	UC16
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin mengubah suatu produk yang ditawarkan.	
Brief description	
Use case ini menjelaskan bagaimana admin mengubah suatu produk yang ditawarkan.	
Trigger	Type
User klik tombol "Edit product" pada suatu halaman produk.	External
Relationship	
Association: Admin Extension: View Single Product	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan form "Edit product". 2. Admin mengubah form yang telah diisi dengan keterangan produk. 3. Admin klik tombol "Save". 4. Sistem menampilkan pesan sukses. 	
Alternate flow	
4a. Apabila form yang diisi oleh admin tidak valid, sistem akan menampilkan pesan error.	

3.5.2.17 Delete Product

Tabel 3.27. *Use Case Description Delete Product*

Use Case Name	Delete Product
ID	UC17
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin menghapus suatu produk yang ditawarkan.	
Brief description	
Use case ini menjelaskan bagaimana admin menghapus suatu produk yang ditawarkan.	
Trigger	Type
Admin klik tombol "Delete product" pada suatu halaman produk.	External
Relationship	
Association: Admin Extension: View Single Product	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan dialog konfirmasi. 2. Admin klik tombol "Delete". 3. Sistem menampilkan pesan sukses. 	

3.5.2.18 Buy Product

Tabel 3.28. *Use Case Description Buy Product*

Use Case Name	Buy Product
ID	UC18
Importance Level	Medium
Primary Actor	Customer
Use case type	Detail, Essential
Stakeholders and interest	
Customer - ingin membeli suatu produk yang ditawarkan.	
Brief description	
Use case ini menjelaskan bagaimana customer membeli suatu produk yang ditawarkan.	
Trigger	Type
Customer klik tombol "Buy product" pada suatu halaman produk.	External
Relationship	
Association: Customer Extension: View Single Product	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan dialog konfirmasi. 2. Customer klik tombol "Confirm" 3. Sistem mengarahkan customer pada halaman Checkout 4. Customer mengisi form Checkout dan klik tombol "Pay" 5. Sistem menampilkan pesan sukses 	
Alternate flow	
5a. Apabila form yang diisi customer tidak valid atau Stripe mengalami error, sistem akan menampilkan pesan error.	

3.5.2.19 Customize Theme

Tabel 3.29. *Use Case Description Customize Theme*

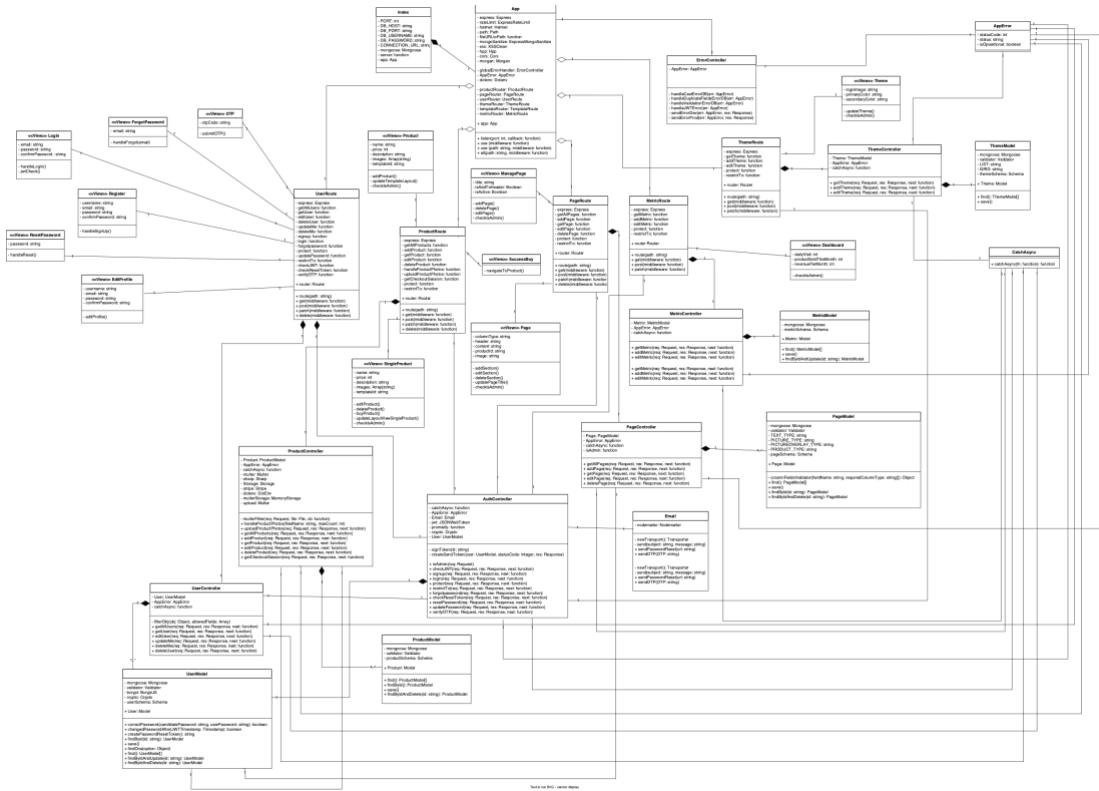
Use Case Name	Customize Theme
ID	UC19
Importance Level	Medium
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin mengubah theme pada tampilan aplikasi web.	
Brief description	
Use case ini menjelaskan bagaimana admin mengubah theme pada tampilan aplikasi web agar warna web seragam dan senada dengan logo.	
Trigger	Type
Admin menuju halaman Theme	External
Relationship	
Association: Admin	
Normal flow of events	
<ol style="list-style-type: none"> 1. Sistem menampilkan halaman Theme. 2. Admin melakukan perubahan pada halaman Theme (logo dan warna). 	
Alternate flow	
<ol style="list-style-type: none"> 1a. Apabila user belum melakukan login, sistem akan mengarahkan user kepada halaman Login. 1b. Apabila user bukan merupakan admin, sistem akan menampilkan pesan error unauthorized. 	

3.5.2.20 View Dashboard

Tabel 3.30. *Use Case Description View Dashboard*

Use Case Name	View Dashboard
ID	UC20
Importance Level	High
Primary Actor	Admin
Use case type	Detail, Essential
Stakeholders and interest	
Admin - ingin melihat metrik-metrik performa pada dashboard.	
Brief description	
Use case ini menjelaskan bagaimana admin melihat metrik-metrik performa pada dashboard sehingga admin dapat mendapat insight atau dapat melakukan aksi sesuai dengan data yang dilihat.	
Trigger	Type
Admin menuju halaman Dashboard	External
Relationship	
Association: Admin	
Normal flow of events	
1. Sistem menampilkan halaman Dashboard	
Alternate flow	
1a. Apabila user belum melakukan login, sistem akan mengarahkan user kepada halaman Login. 1b. Apabila user bukan merupakan admin, sistem akan menampilkan pesan error unauthorized.	

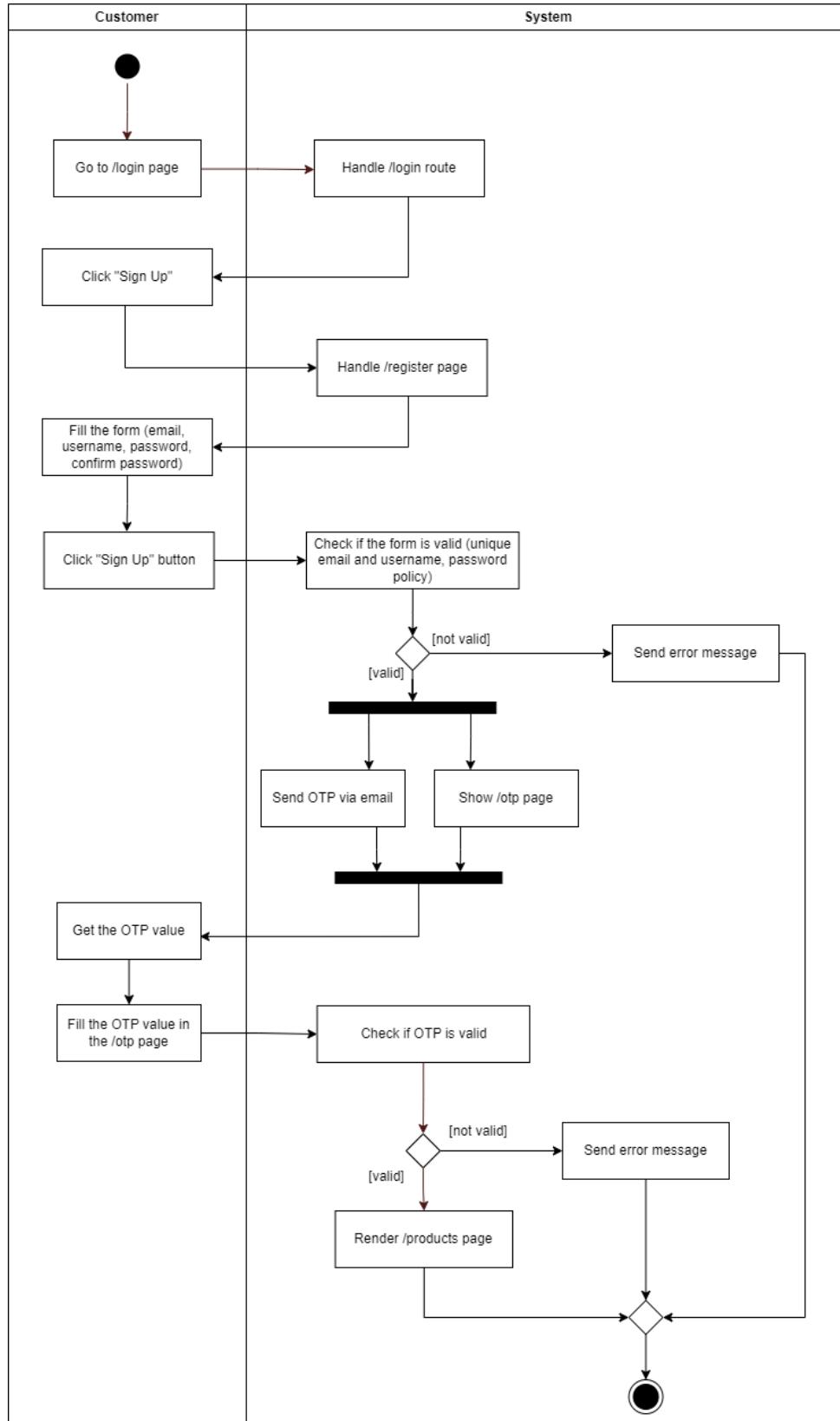
3.5.3 Class Diagram



Gambar 3.35. *Class Diagram*

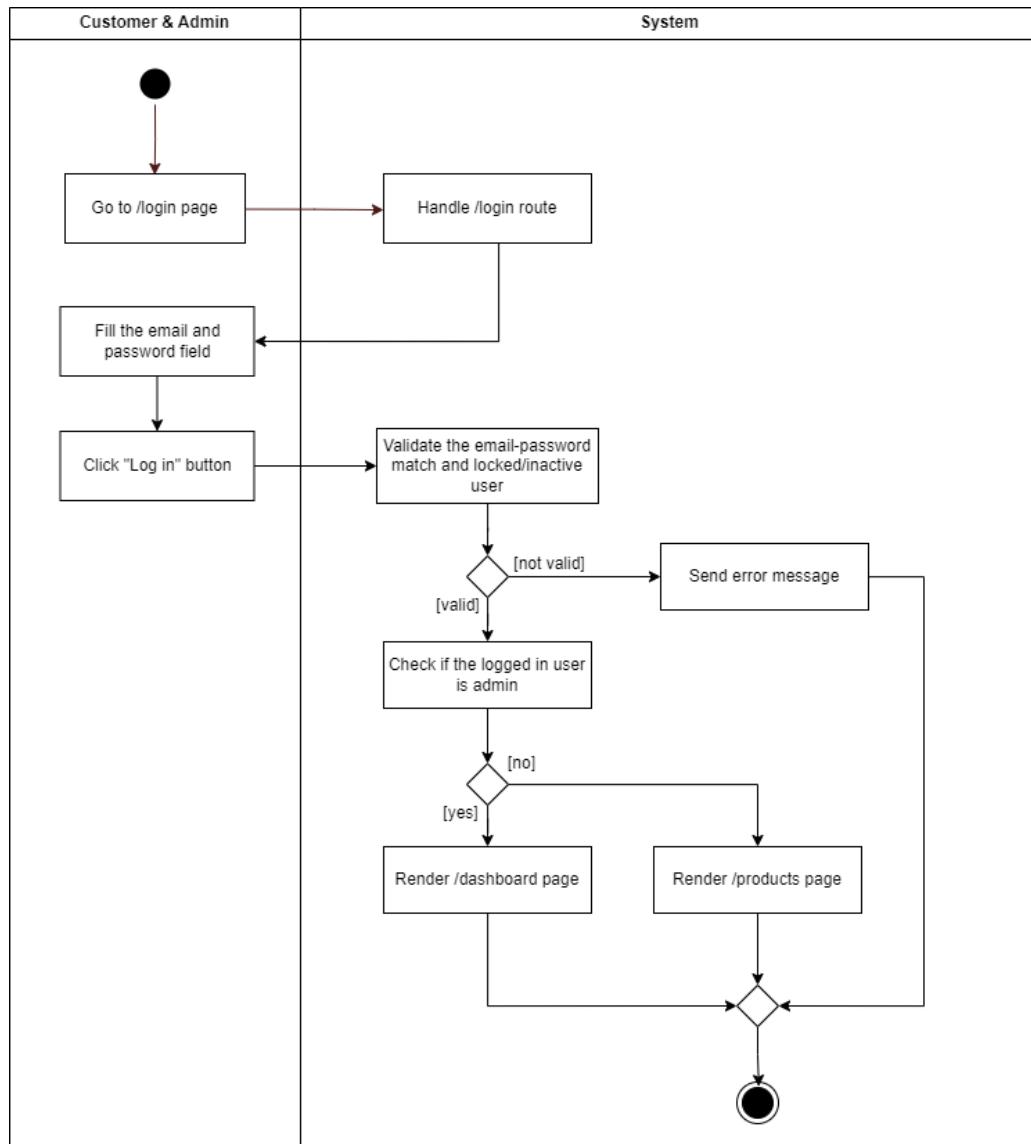
3.5.4 Activity Diagram

3.5.4.1 Register



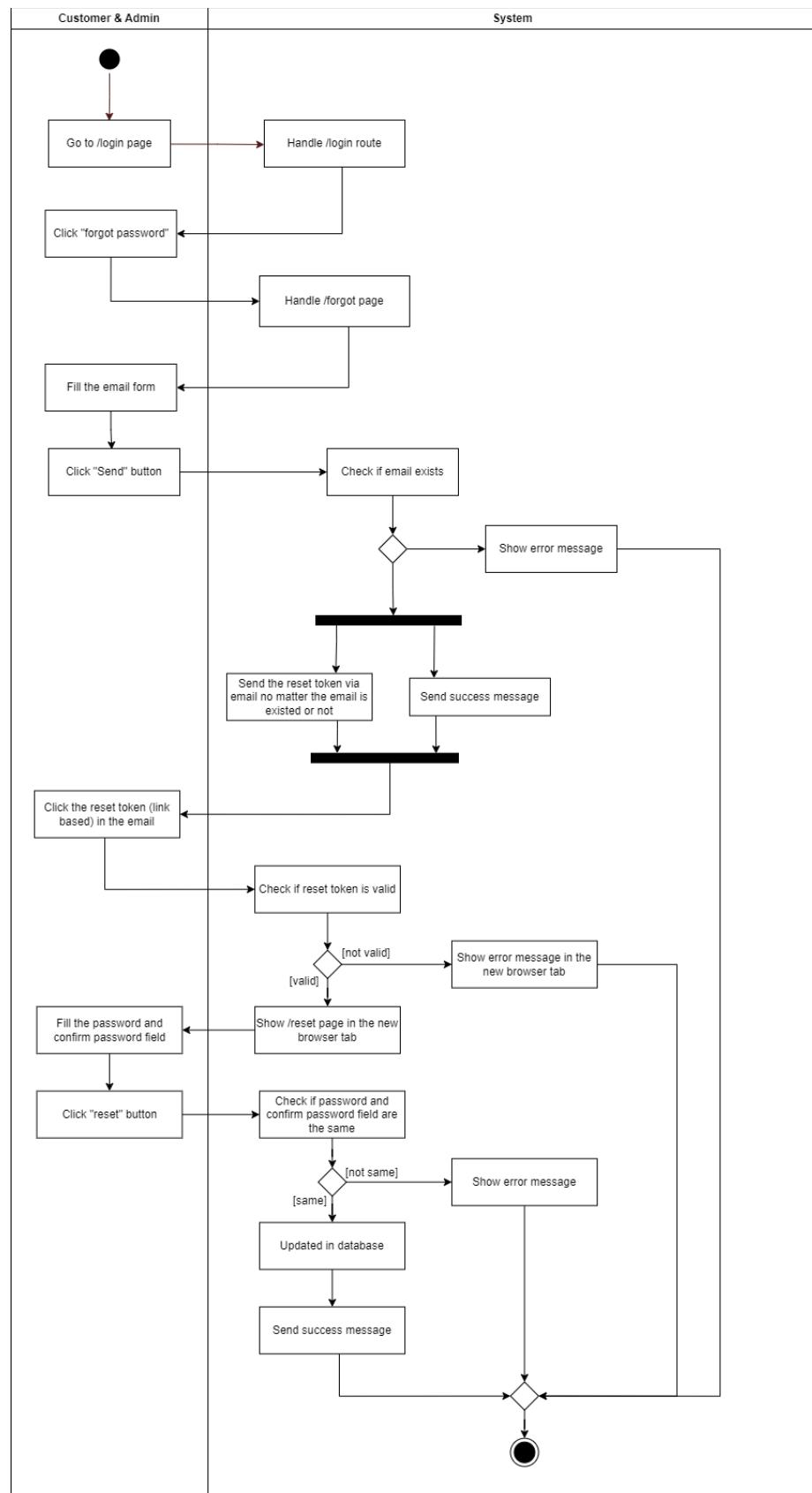
Gambar 3.36. Activity Diagram Register

3.5.4.2 Login



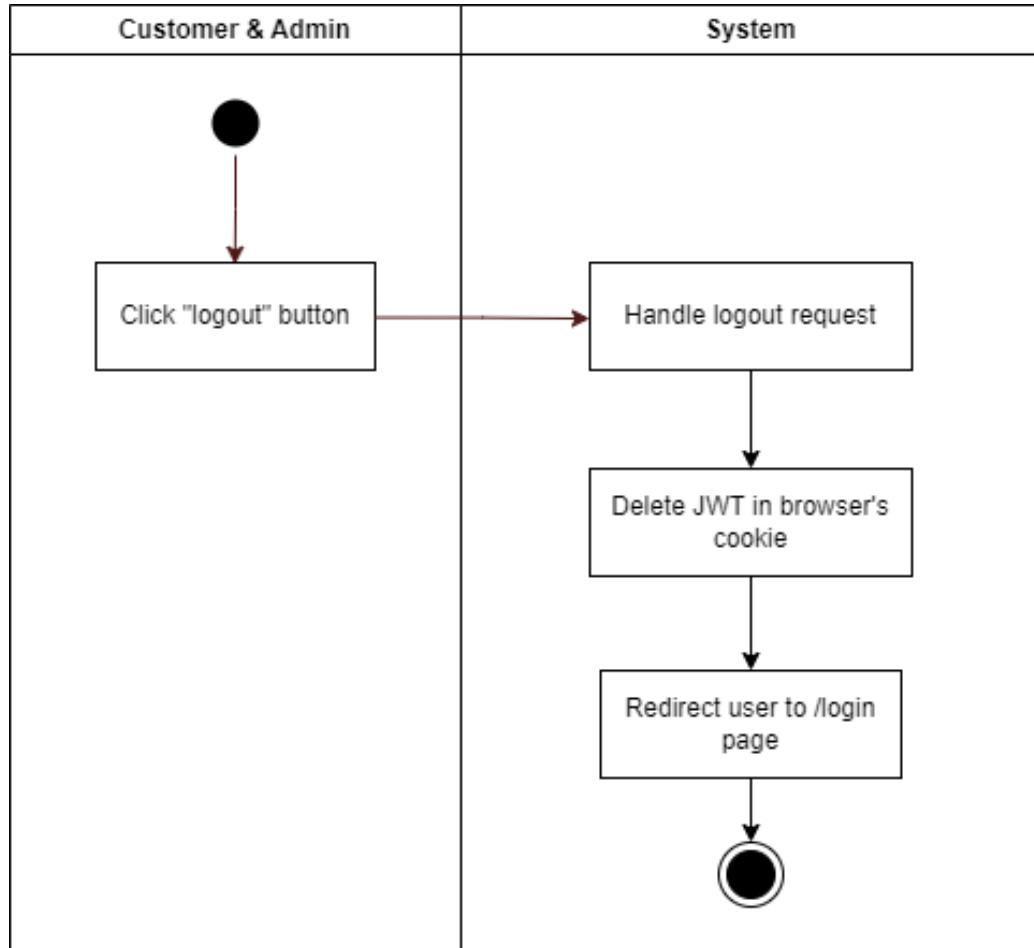
Gambar 3.37. Activity Diagram Login

3.5.4.3 Forgot Password

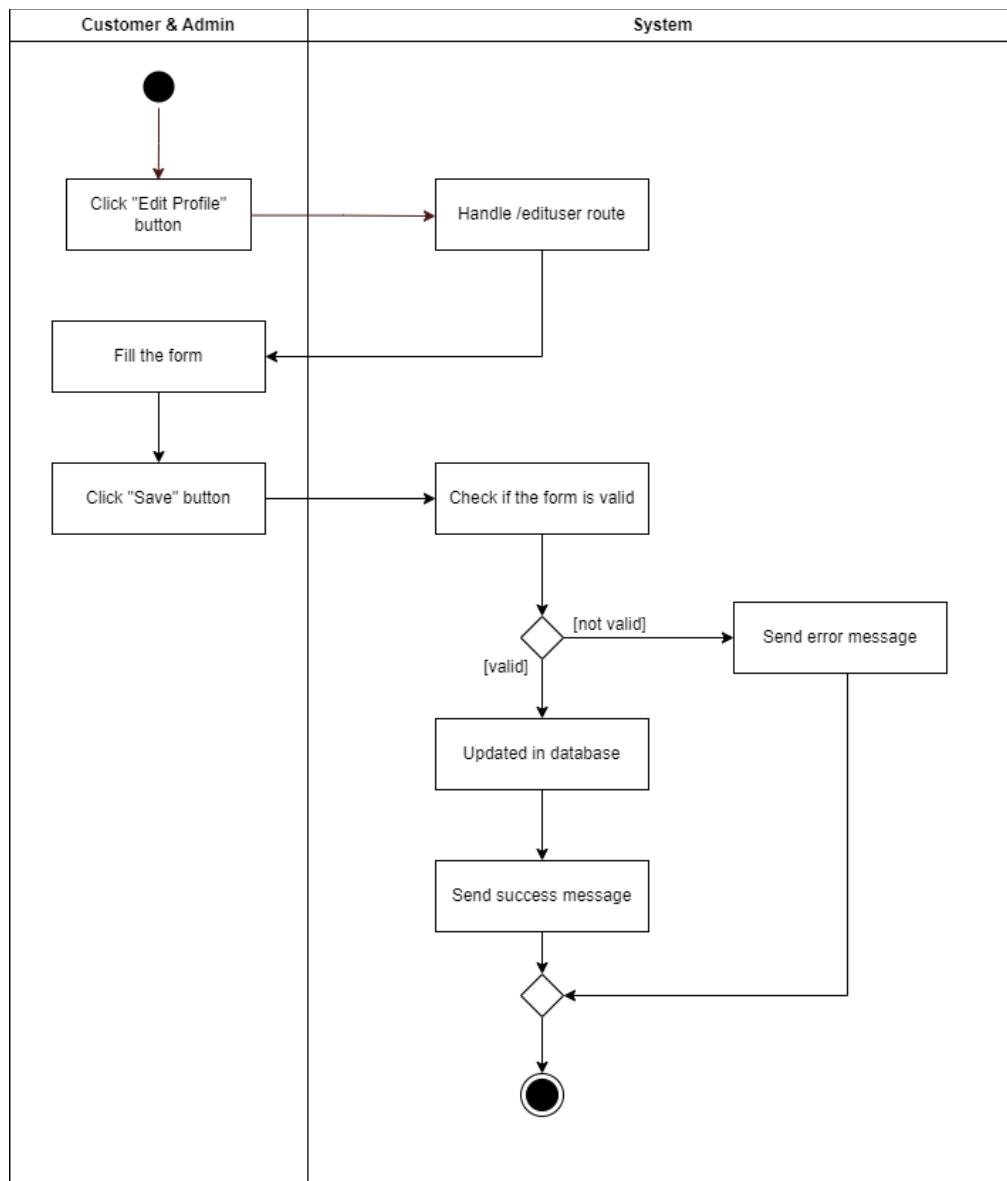


Gambar 3.38. Activity Diagram Forgot Password

3.5.4.4 Logout

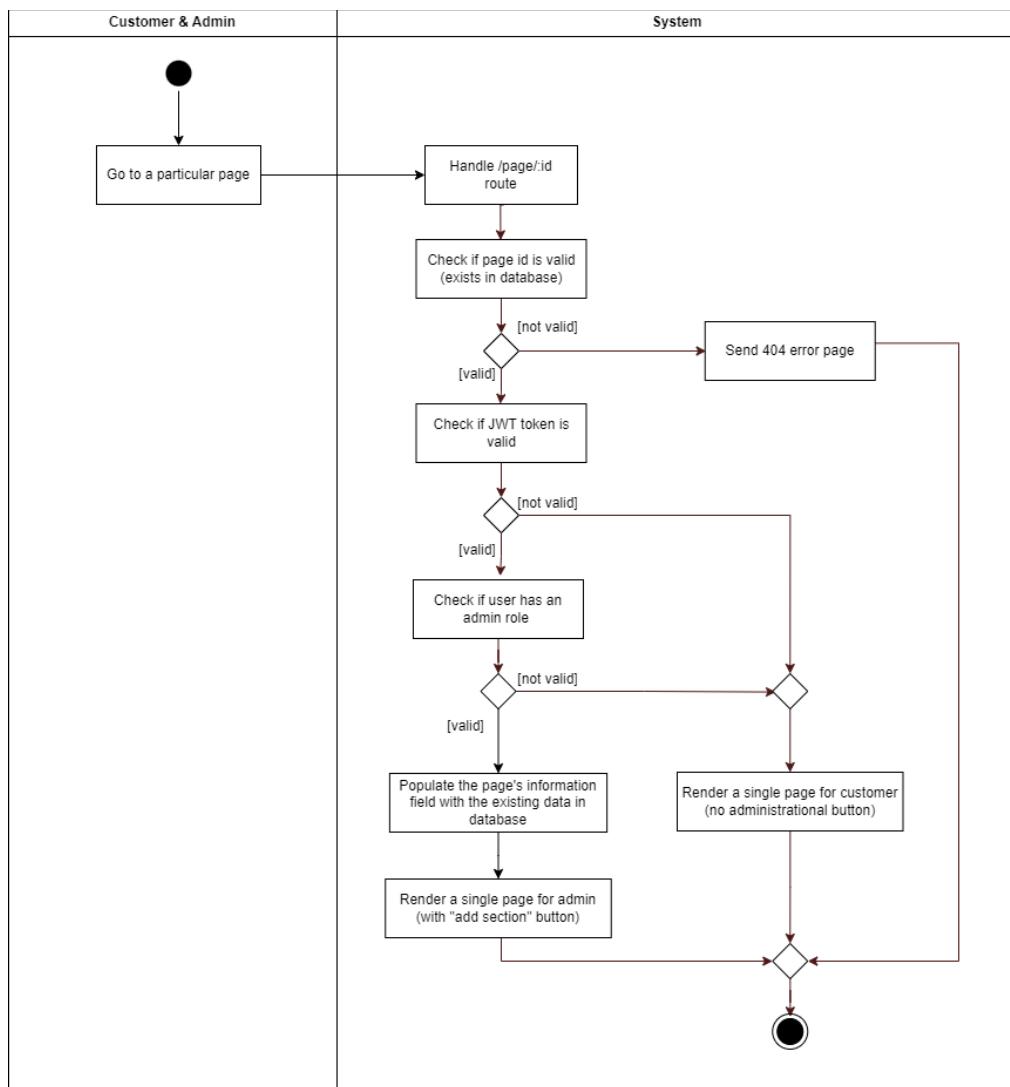
Gambar 3.39. *Activity Diagram Logout*

3.5.4.5 Edit User



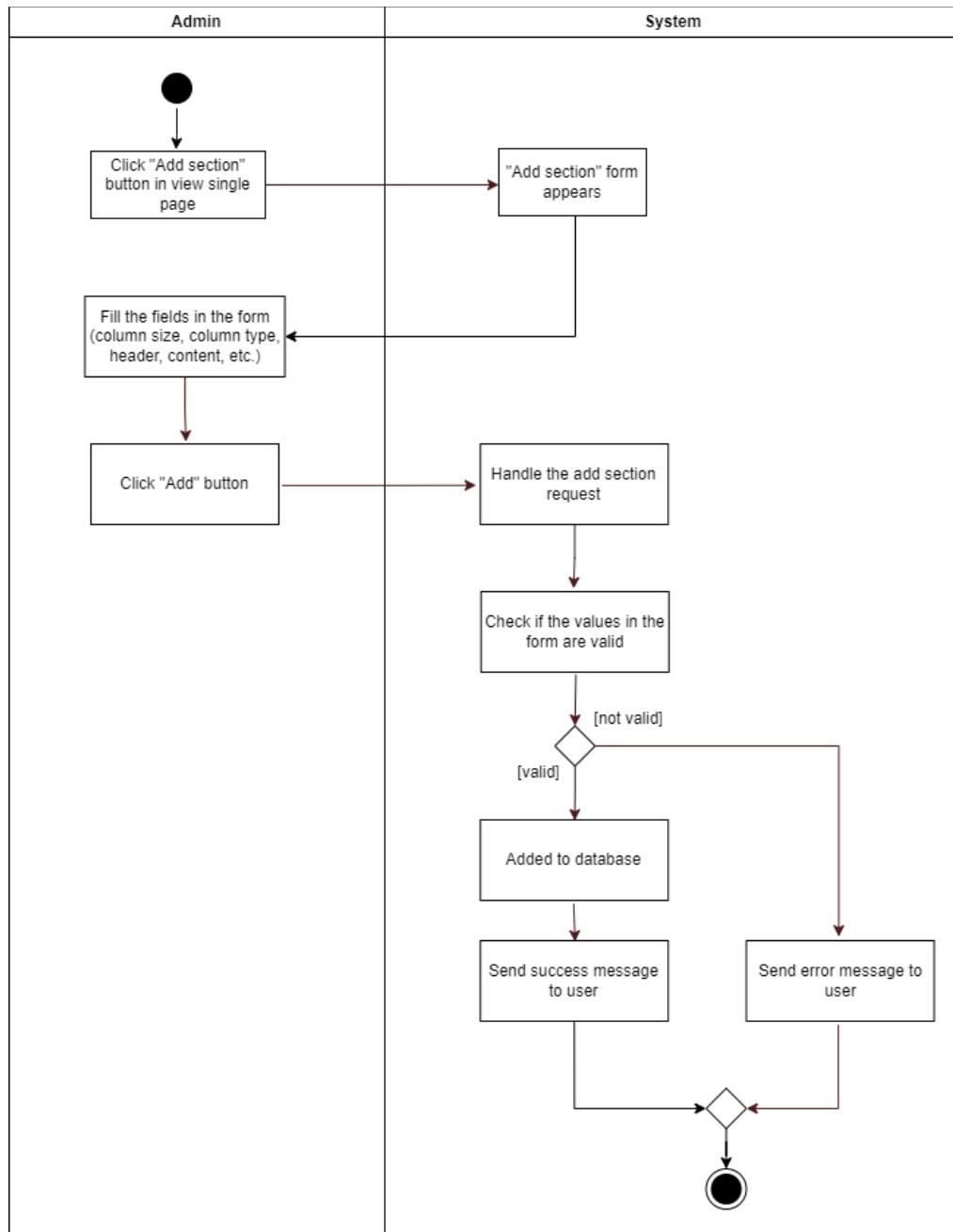
Gambar 3.40. *Activity Diagram Edit User*

3.5.4.6 View Single Page



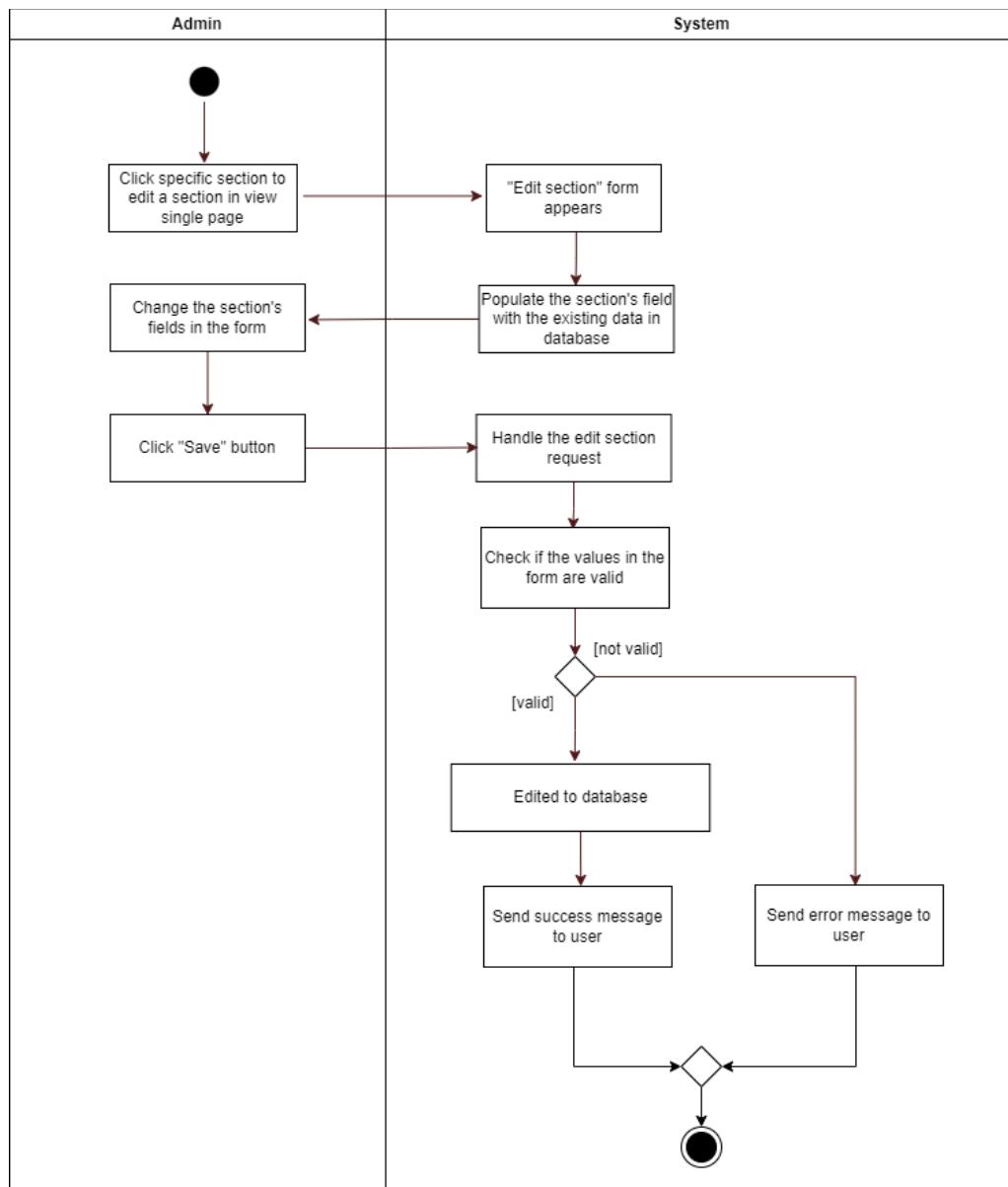
Gambar 3.41. Activity Diagram View Single Page

3.5.4.8 Add Section



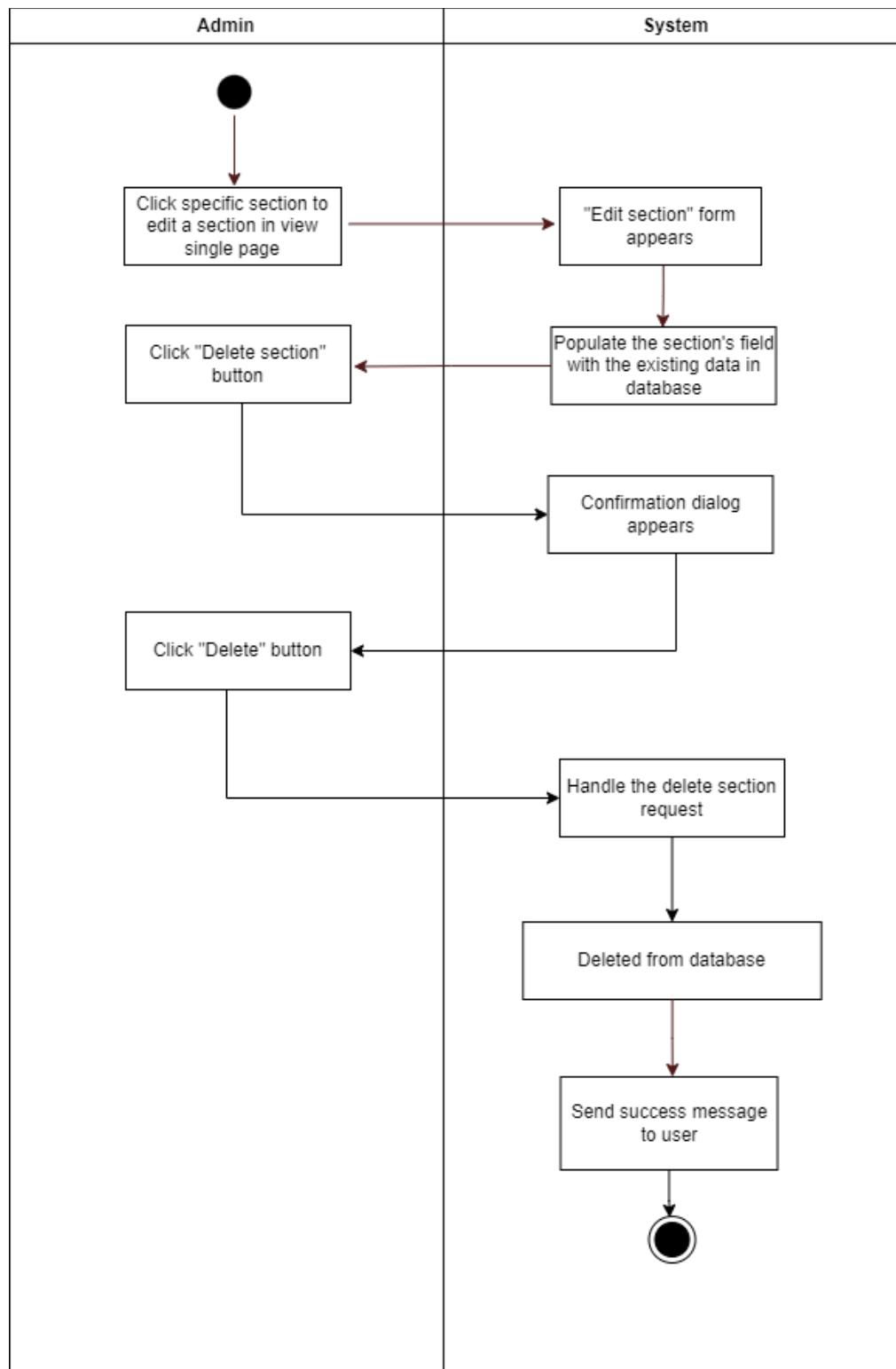
Gambar 3.42. Activity Diagram Add Section

3.5.4.9 Edit Section



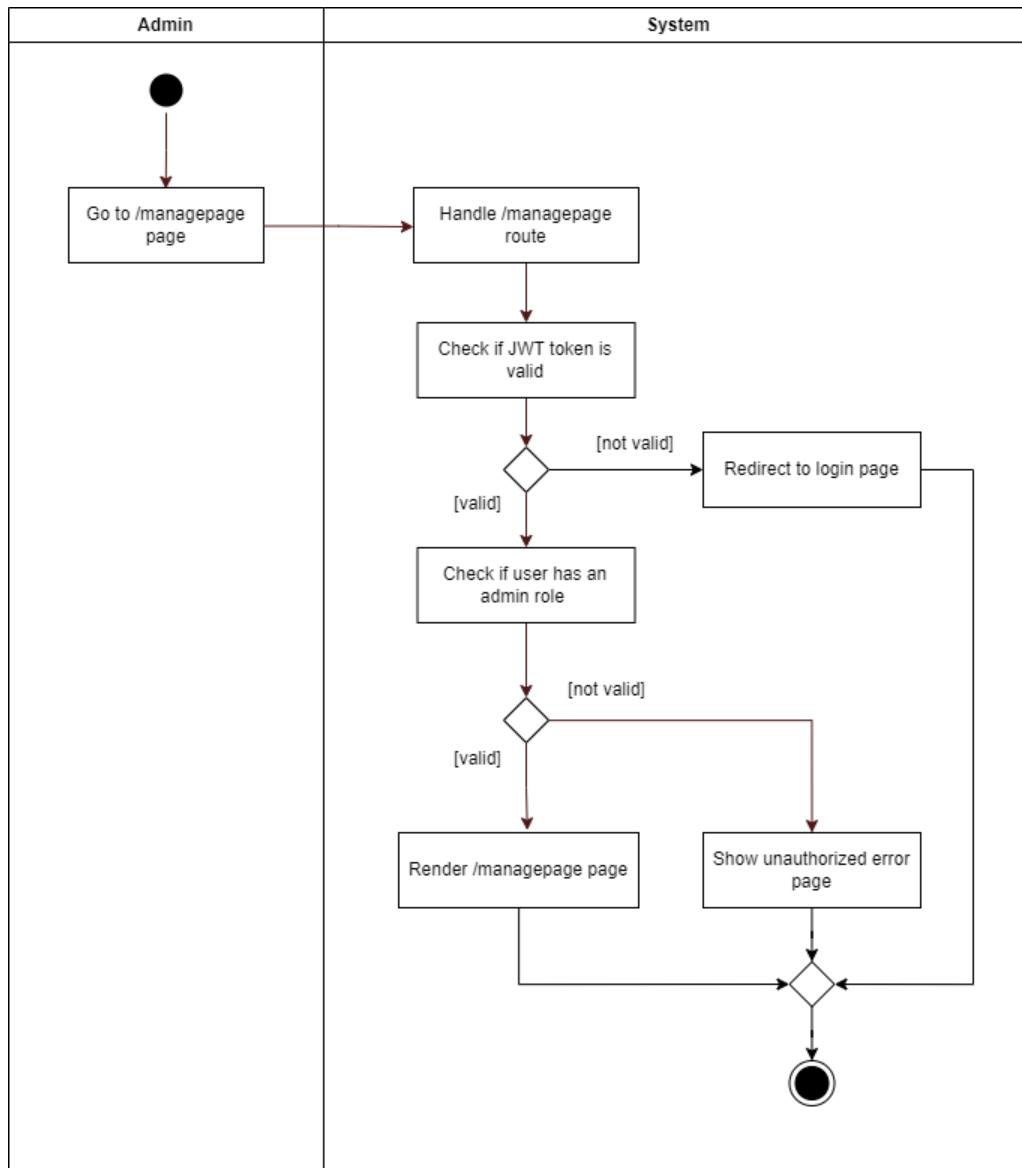
Gambar 3.43. *Activity Diagram Edit Section*

3.5.4.10 Delete Section



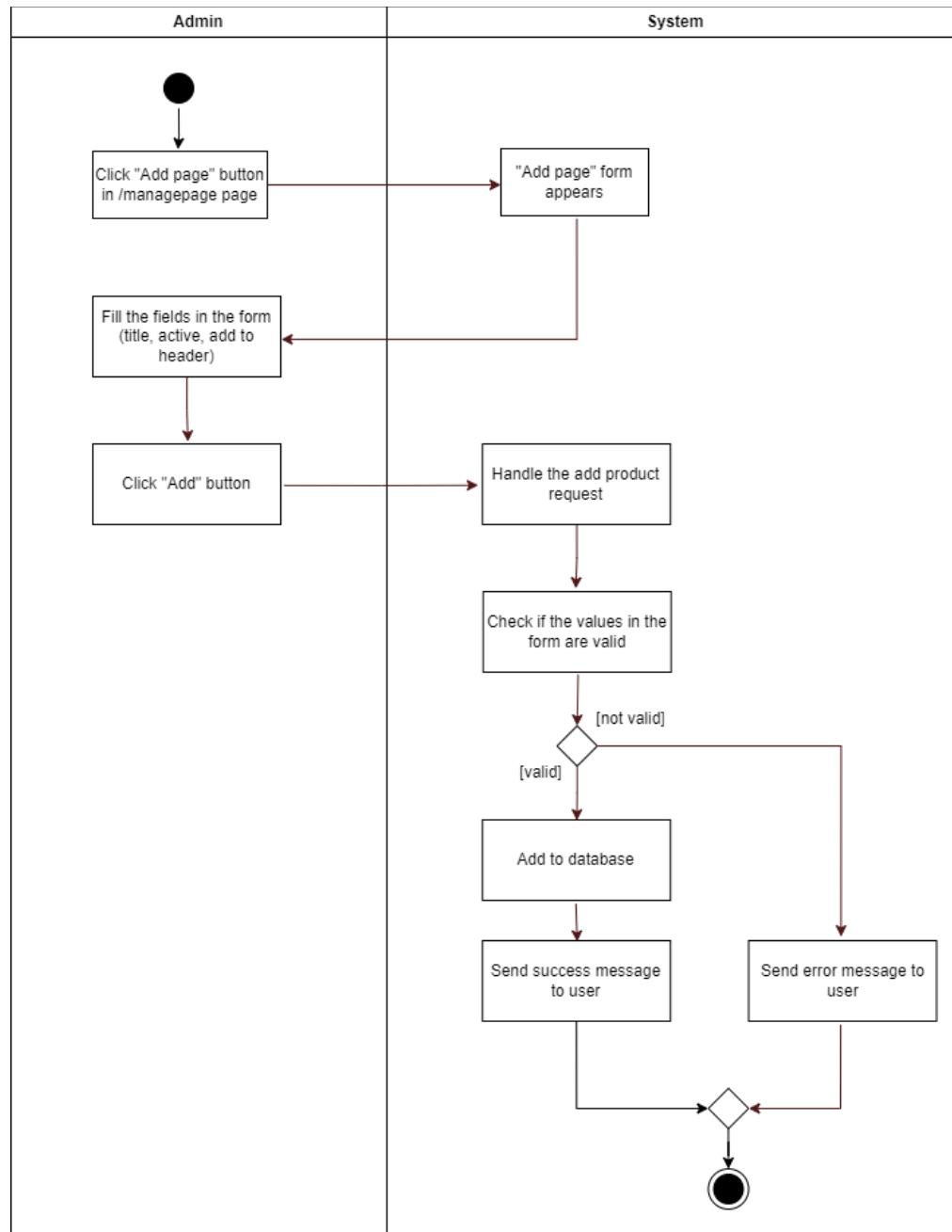
Gambar 3.44. Activity Diagram Delete Section

3.5.4.11 Manage Pages



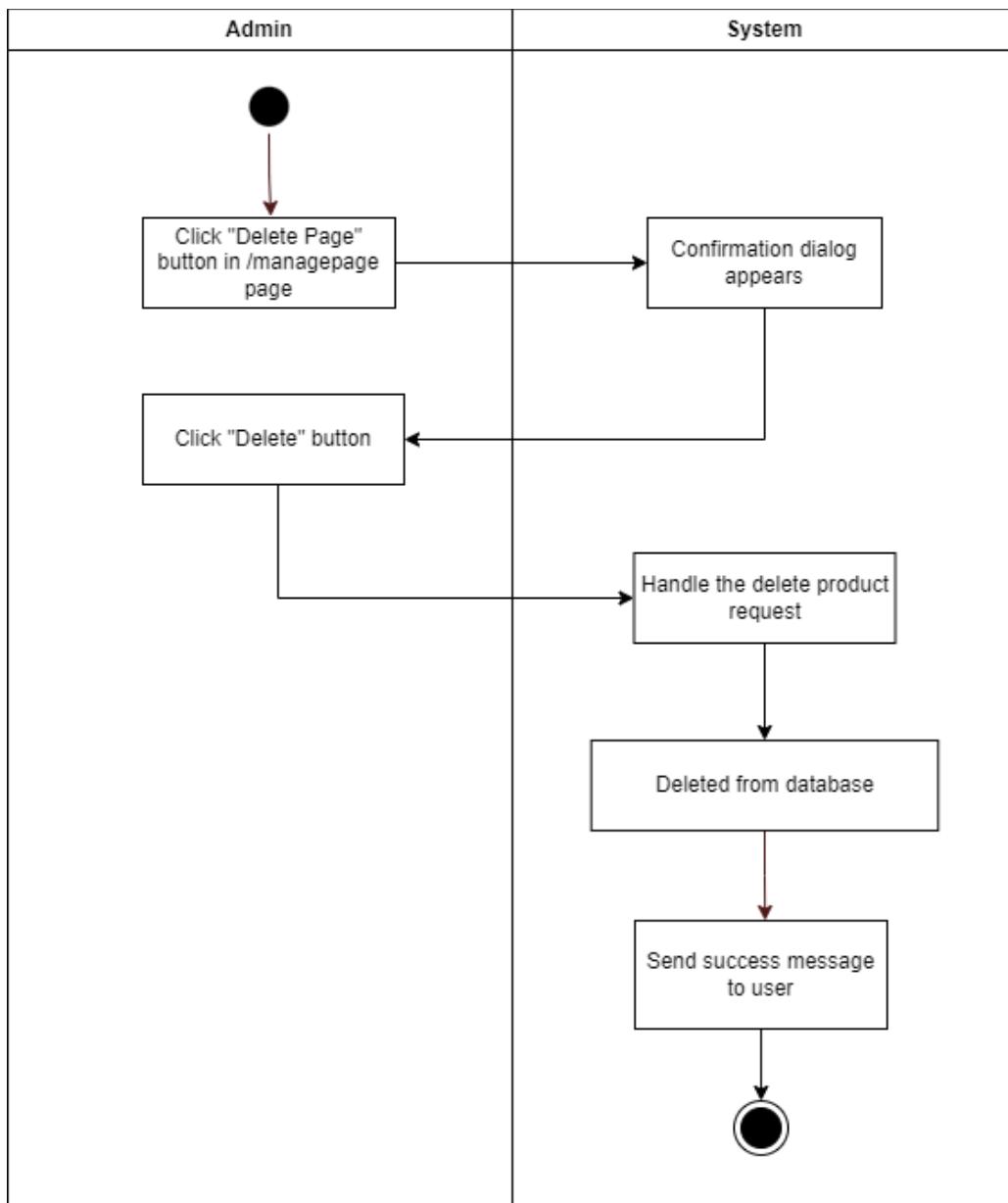
Gambar 3.45. *Activity Diagram Manage Pages*

3.5.4.12 Add Page



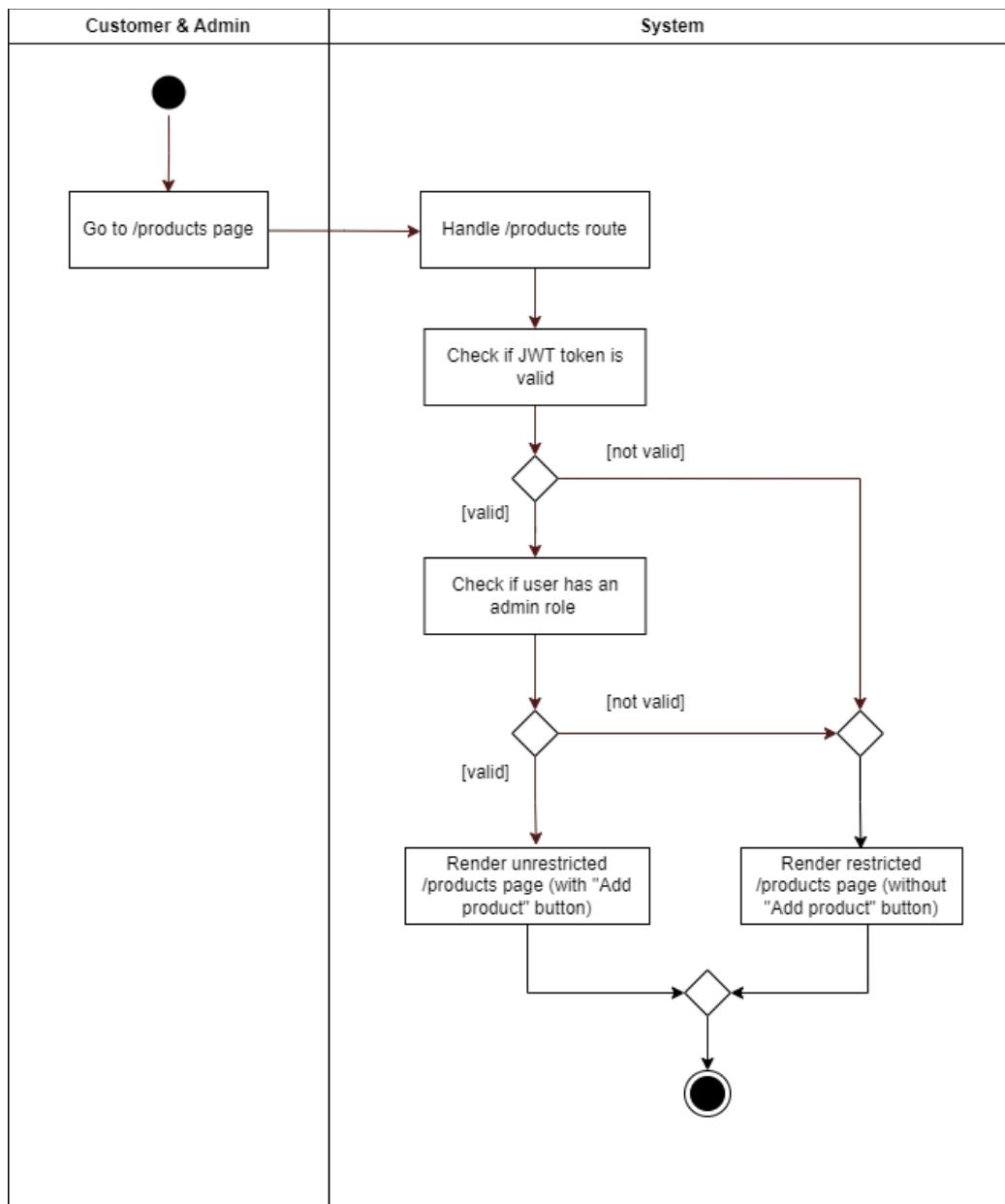
Gambar 3.46. Activity Diagram Add Page

3.5.4.13 Delete Page



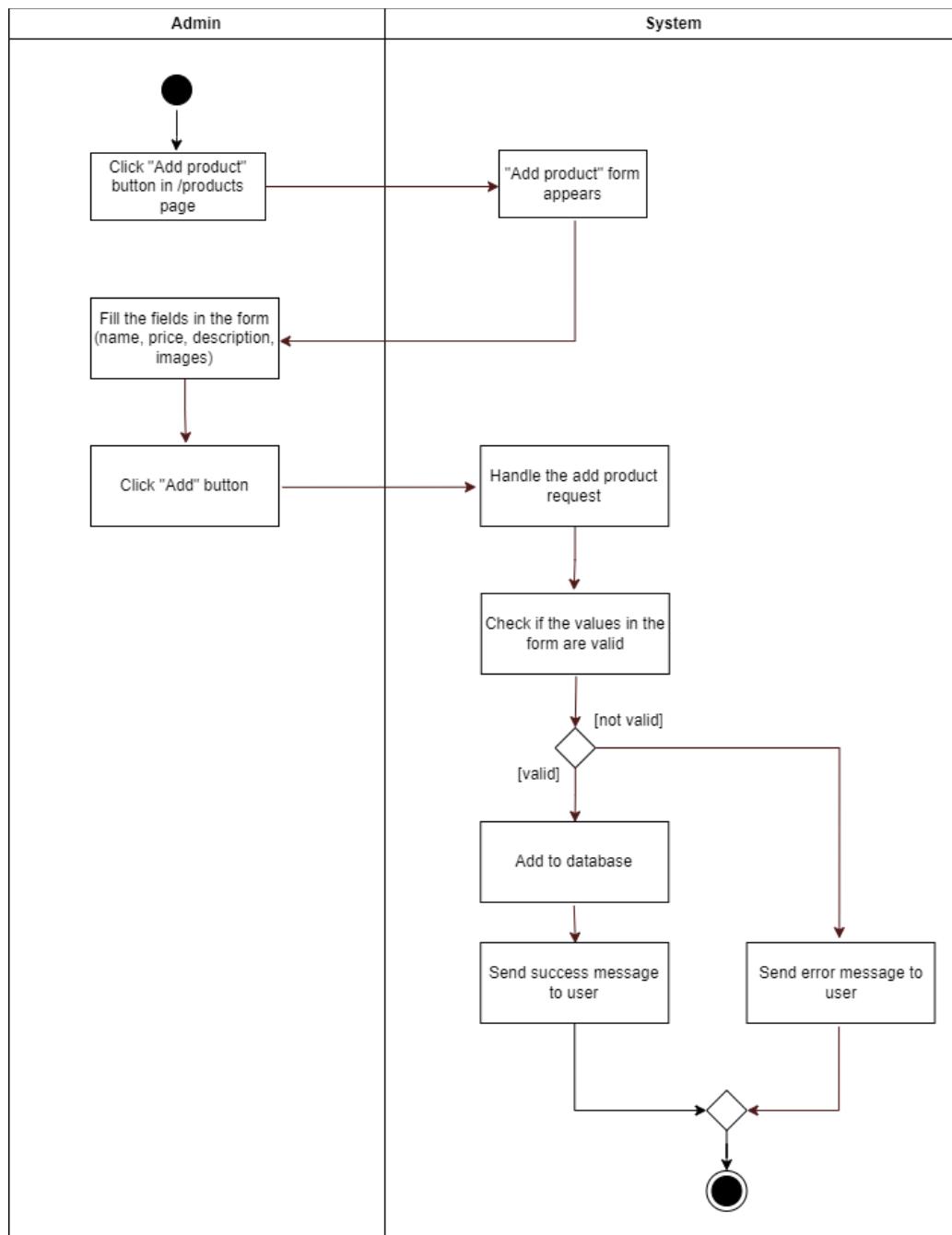
Gambar 3.47. Activity Diagram Delete Page

3.5.4.14 View Products



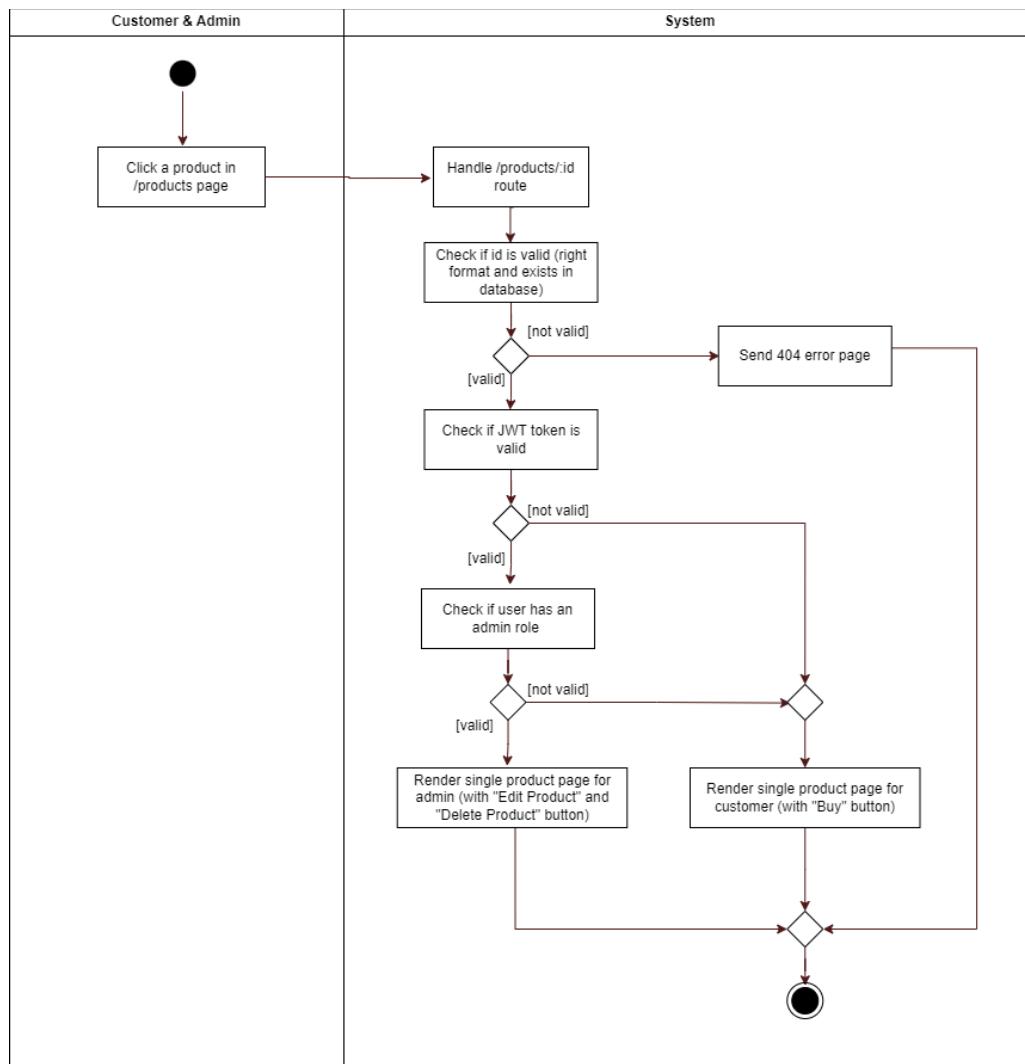
Gambar 3.48. *Activity Diagram View Products*

3.5.4.15 Add Product



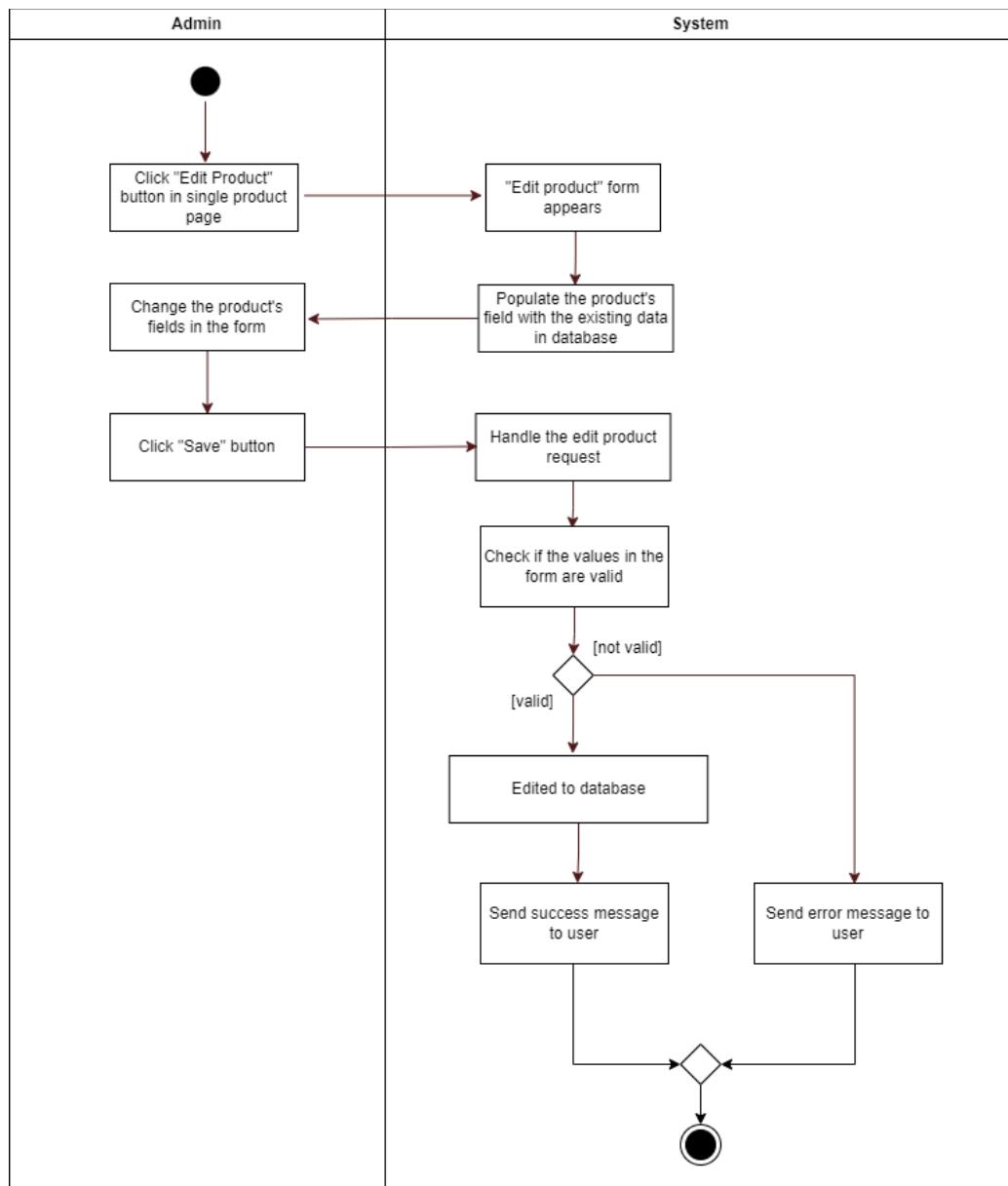
Gambar 3.49. Activity Diagram Add Product

3.5.4.16 View Single Product



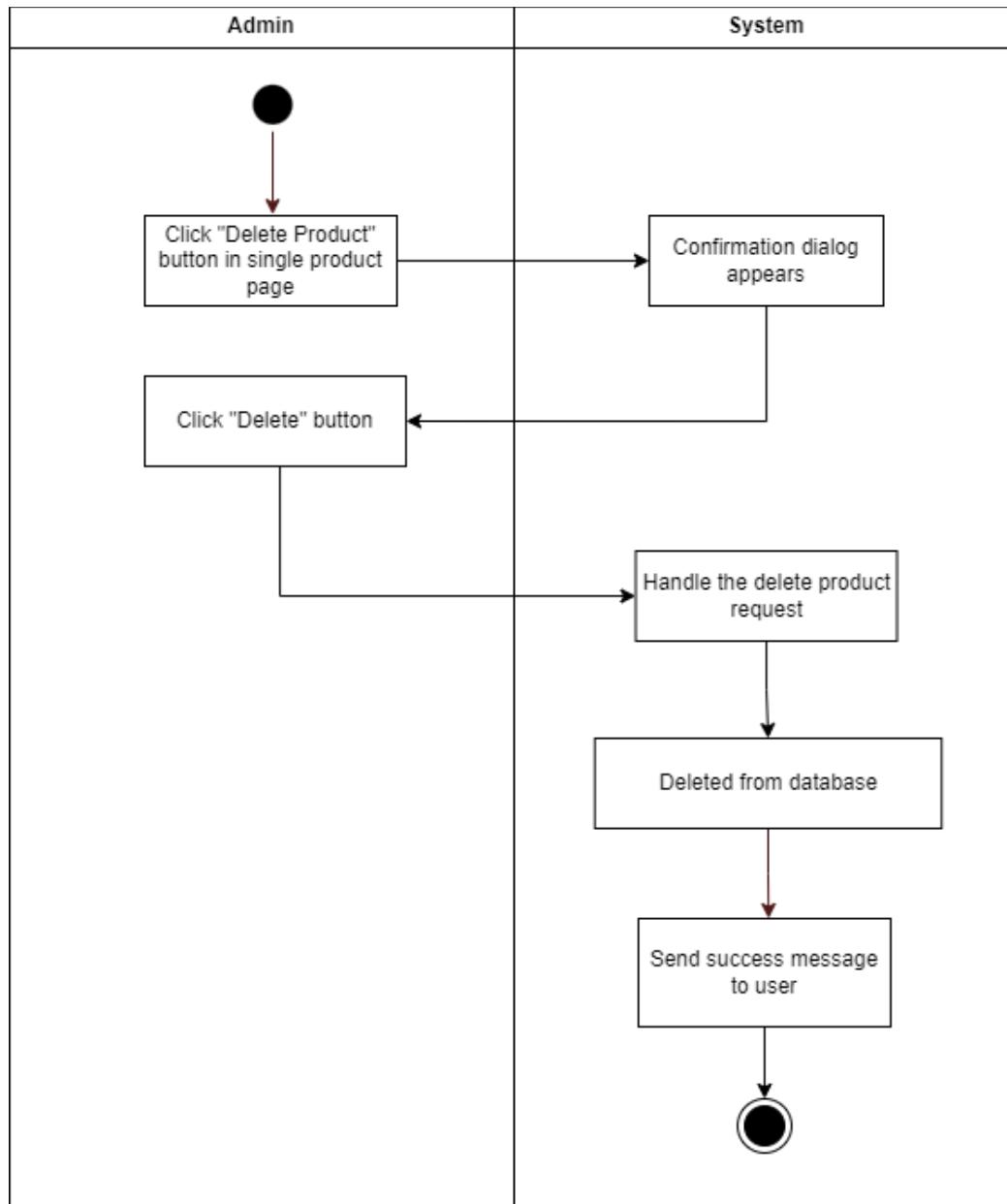
Gambar 3.50. *Activity Diagram View Single Product*

3.5.4.17 Edit Product



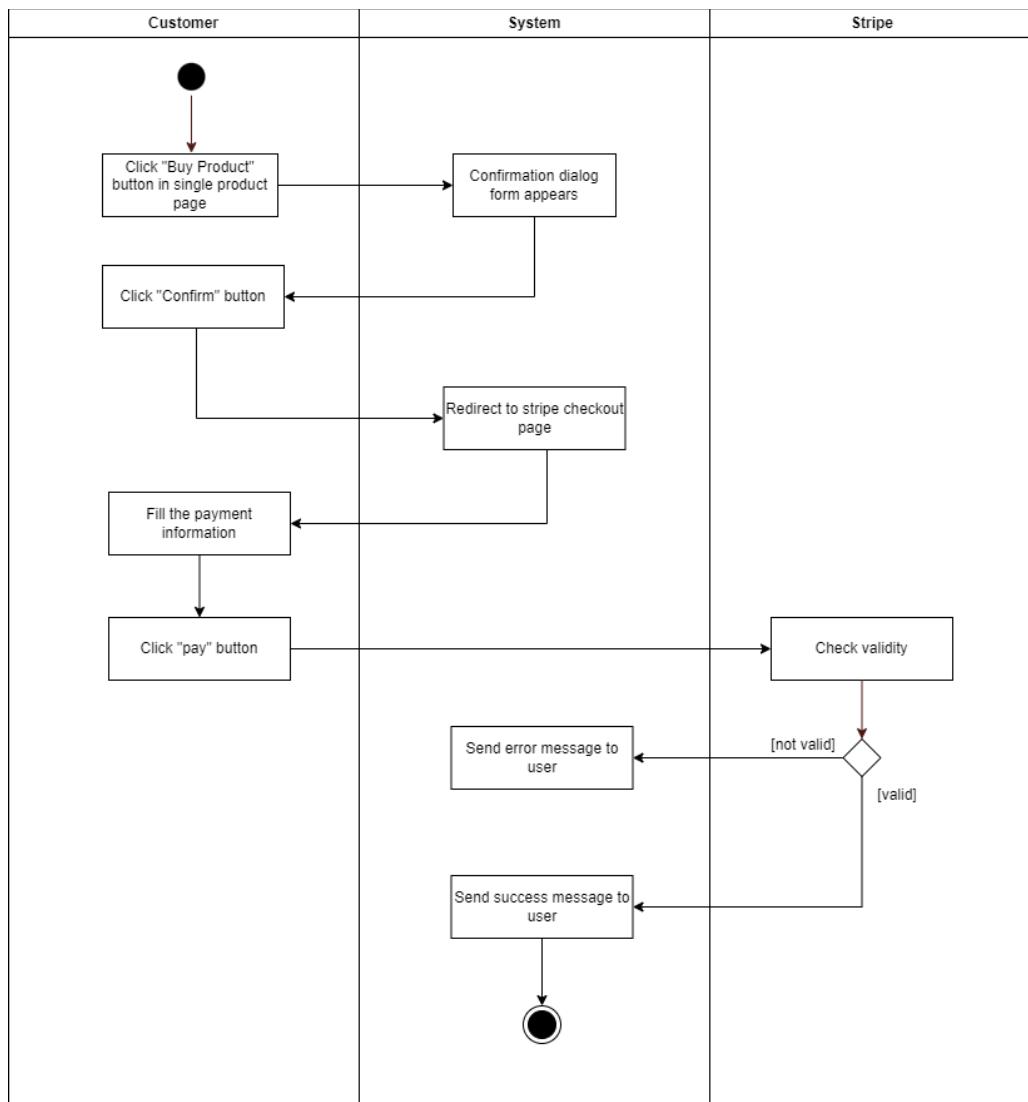
Gambar 3.51. Activity Diagram Edit Product

3.5.4.18 Delete Product



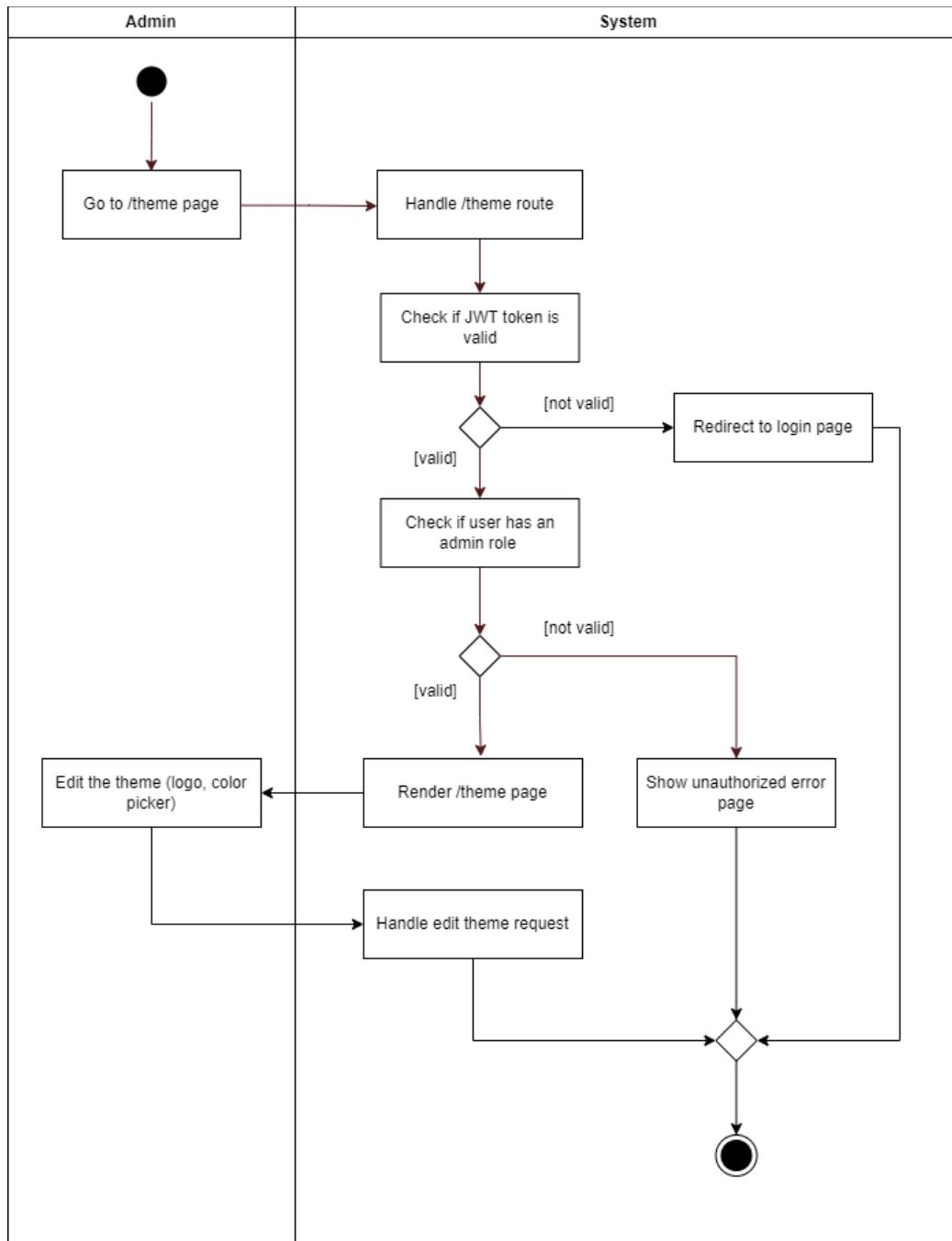
Gambar 3.52. Activity Diagram Delete Product

3.5.4.19 Buy Product



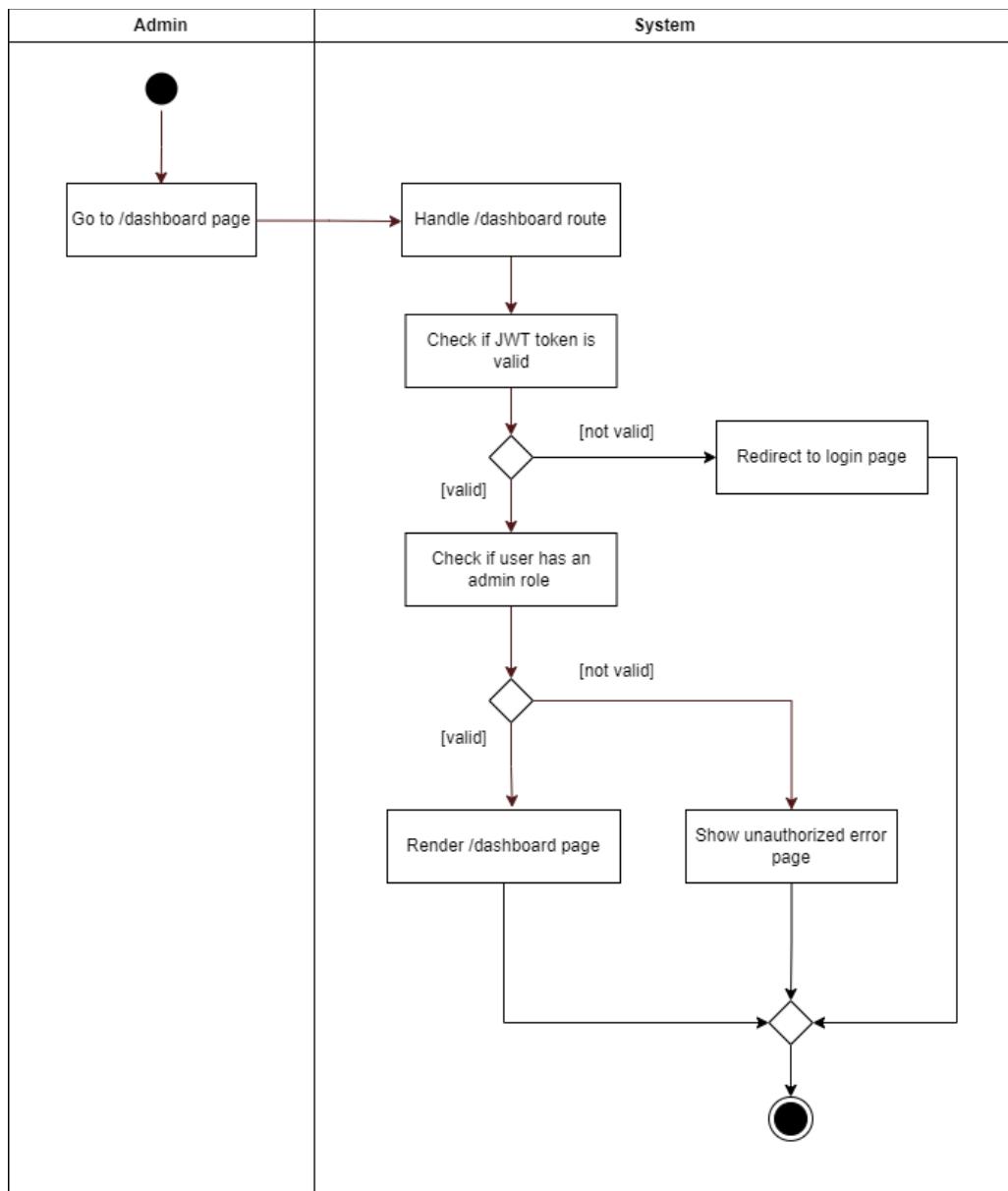
Gambar 3.53. Activity Diagram Buy Product

3.5.4.20 Customize Theme



Gambar 3.54. Activity Diagram Customize Theme

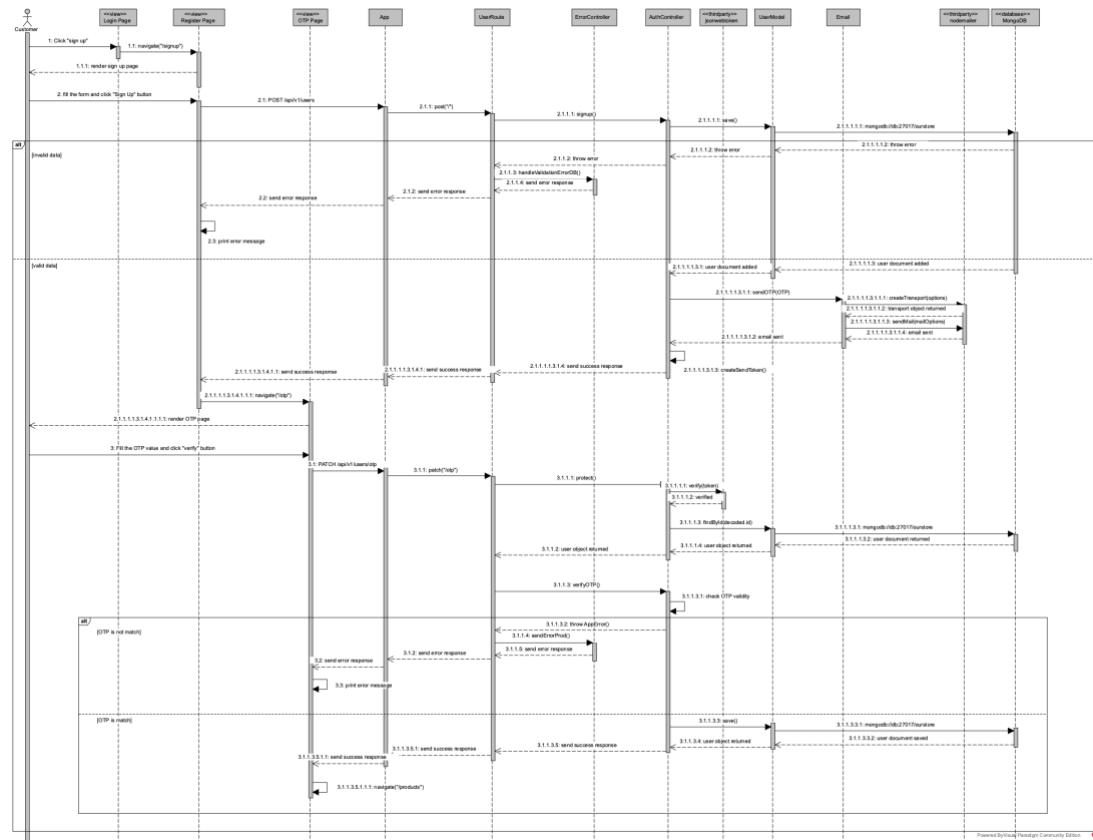
3.5.4.21 View Dashboard



Gambar 3.55. Activity Diagram View Dashboard

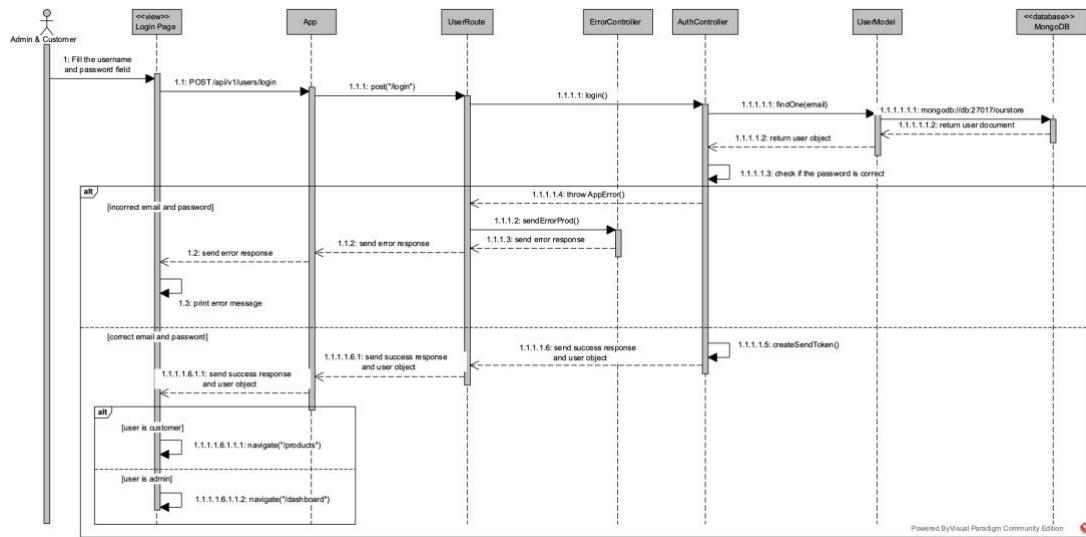
3.5.5 Sequence Diagram

3.5.5.1 Register



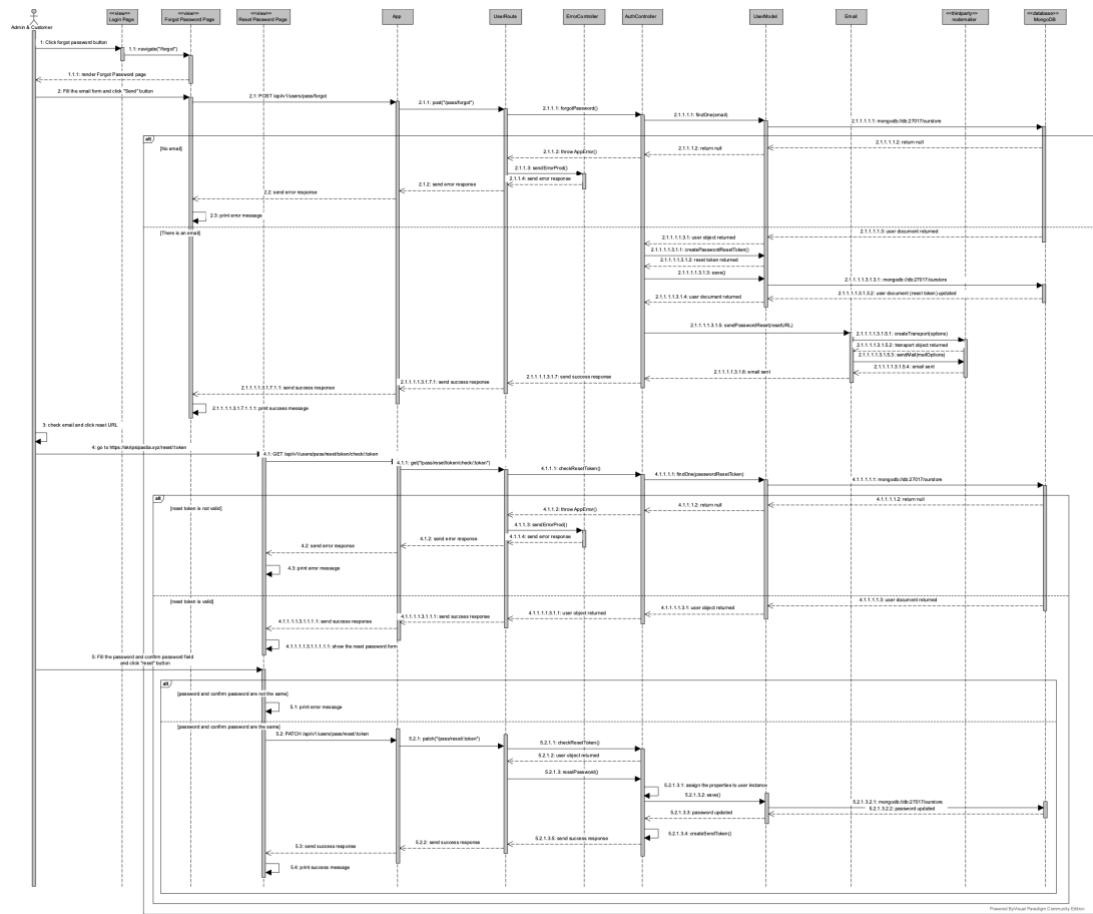
Gambar 3.56. *Sequence Diagram Register*

3.5.5.2 Login



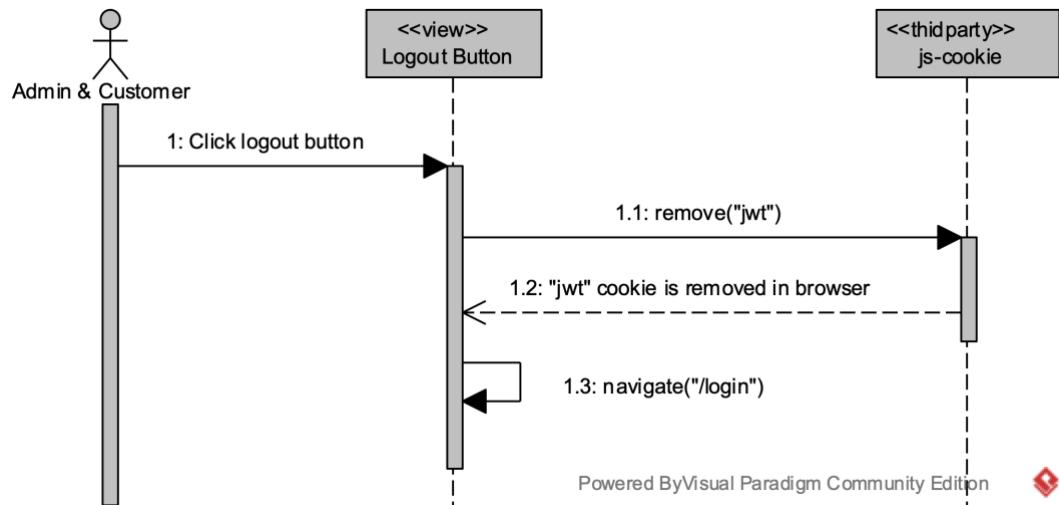
Gambar 3.57. Sequence Diagram Login

3.5.5.3 Forgot Password

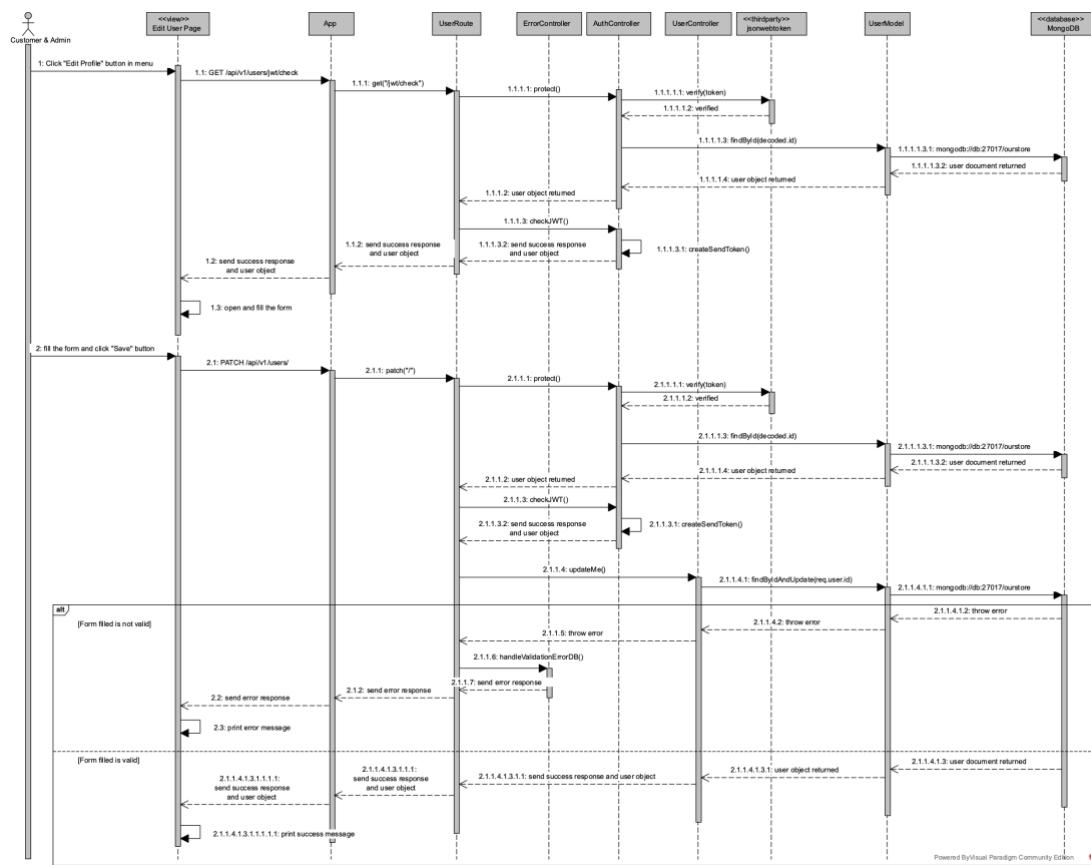


Gambar 3.58. Sequence Diagram Forgot Password

3.5.5.4 Logout

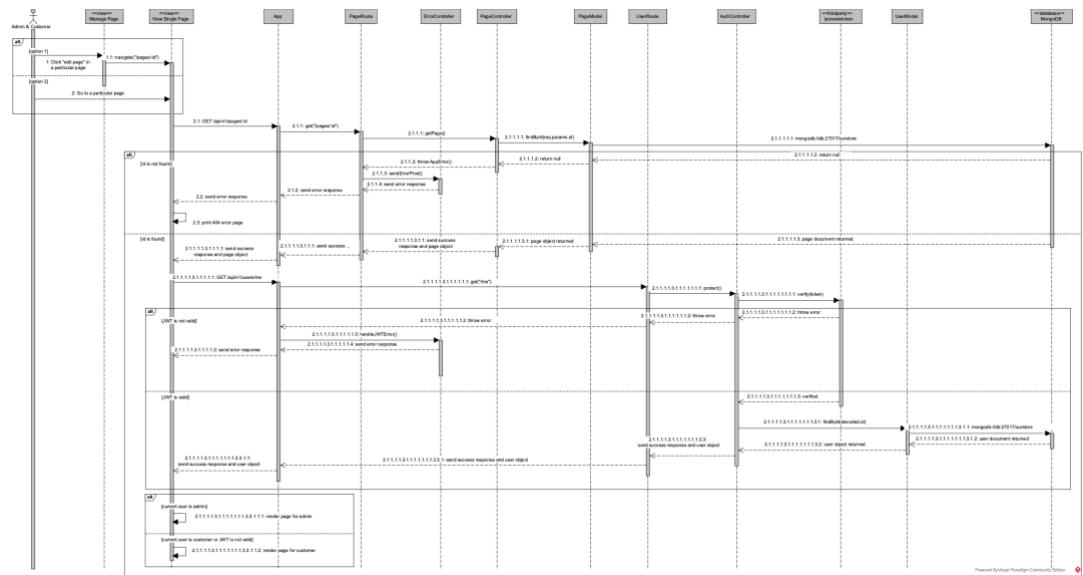
Gambar 3.59. *Sequence Diagram Logout*

3.5.5.5 Edit User



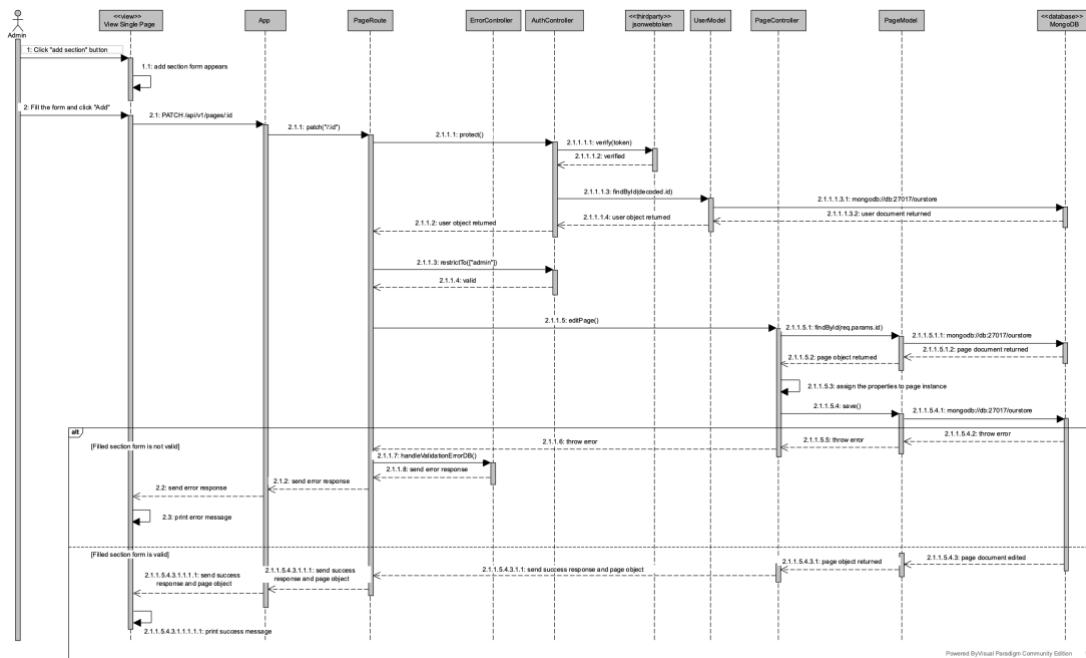
Gambar 3.60. Sequence Diagram Edit User

3.5.5.6 View Single Page



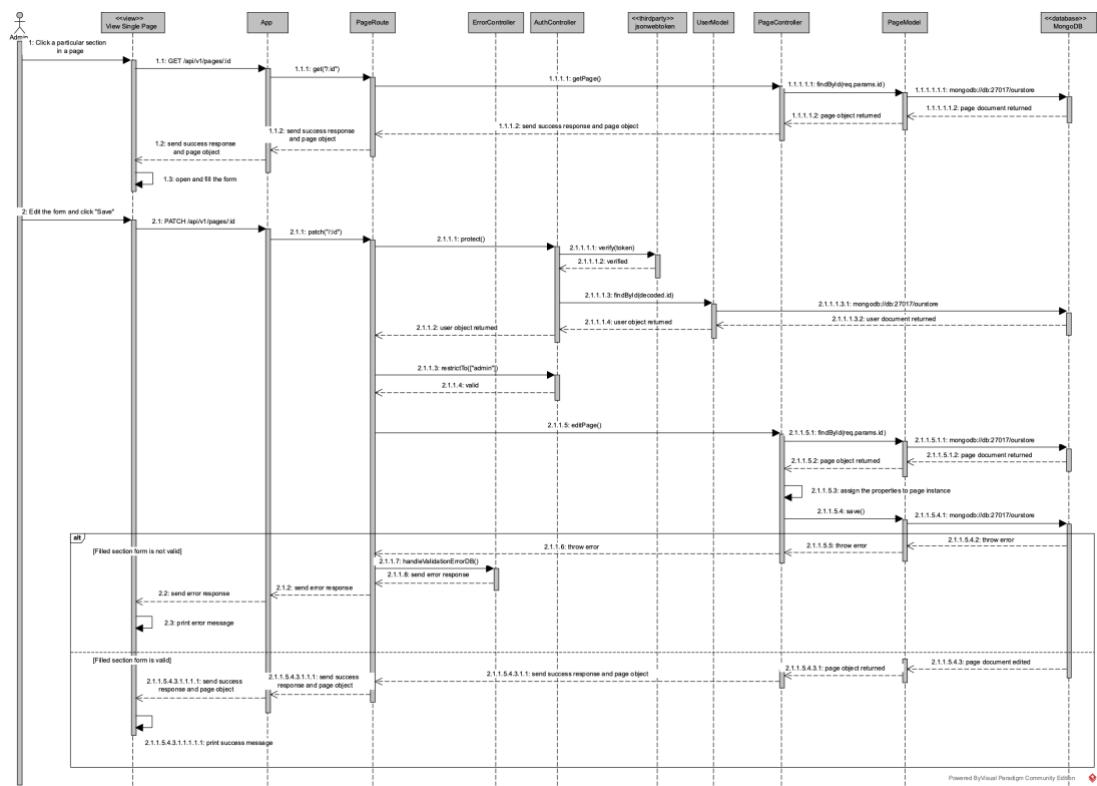
Gambar 3.61. Sequence Diagram View Single Page

3.5.5.7 Add Section



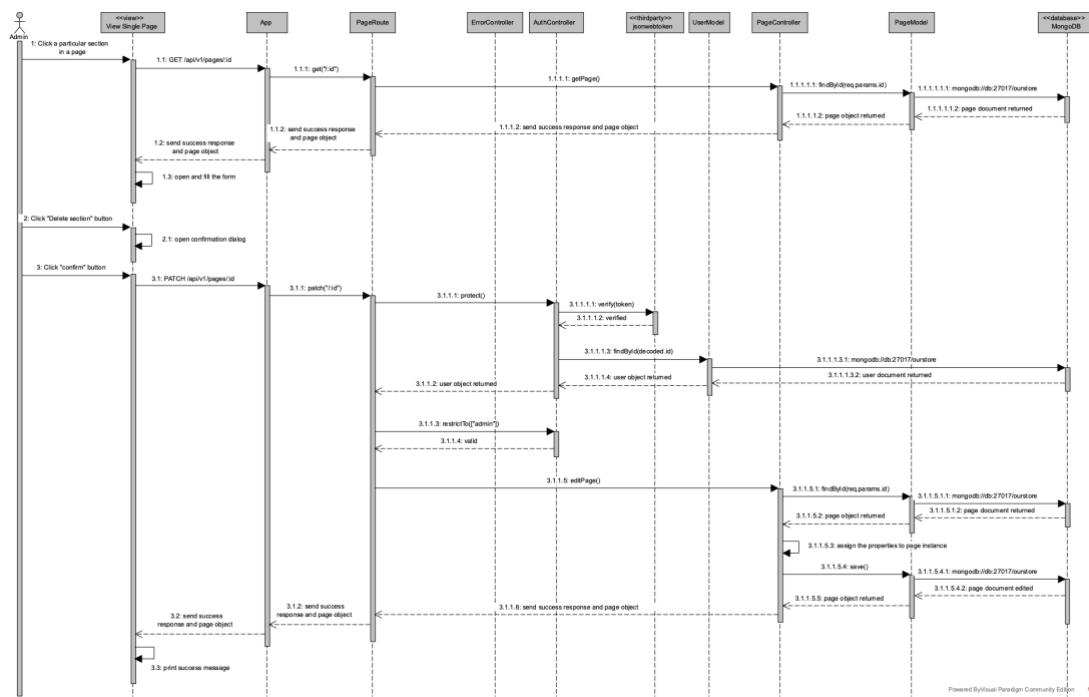
Gambar 3.62. Sequence Diagram Add Section

3.5.5.8 Edit Section



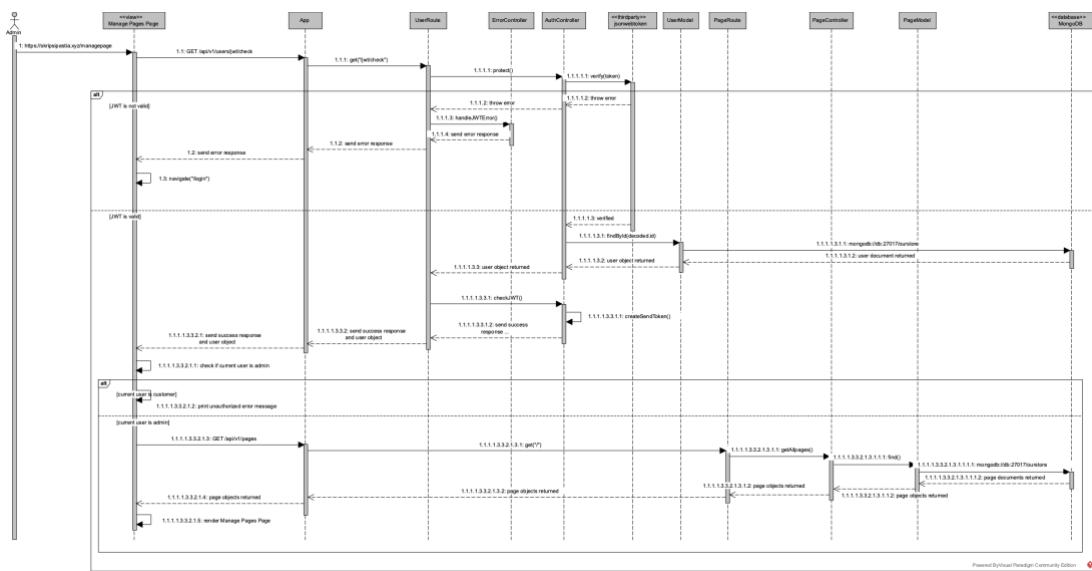
Gambar 3.63. Sequence Diagram Edit Section

3.5.5.9 Delete Section



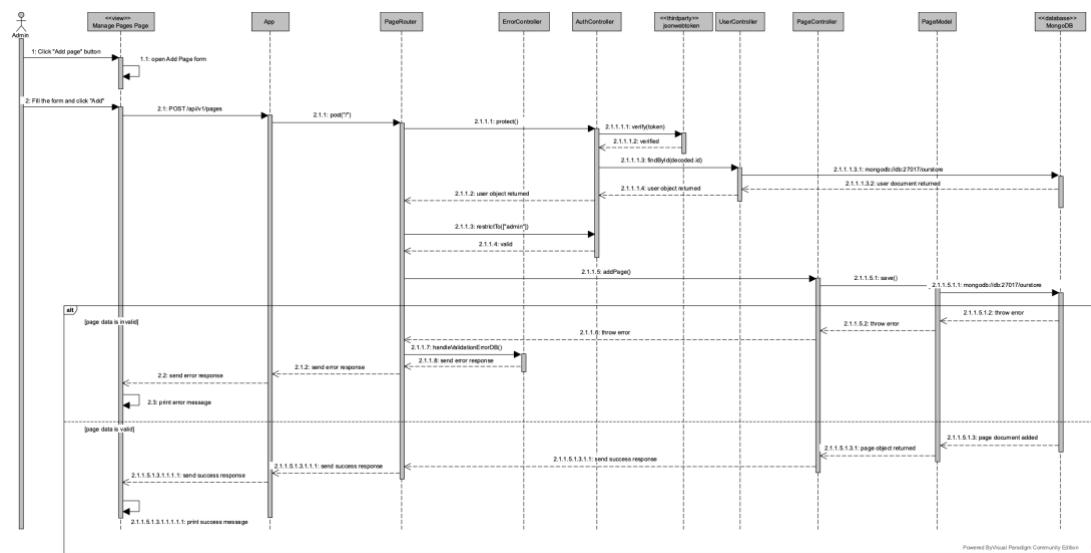
Gambar 3.64. Sequence Diagram Delete Section

3.5.5.10 Manage Page



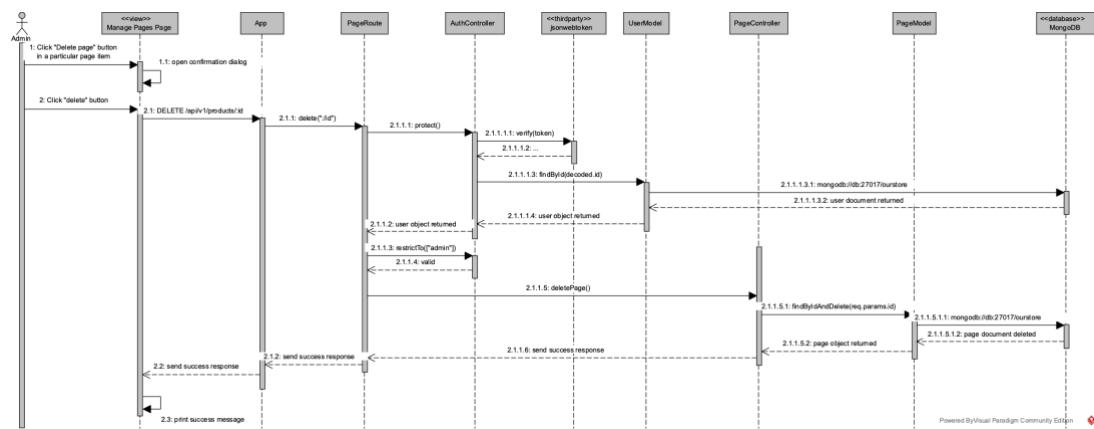
Gambar 3.65. Sequence Diagram Manage Page

3.5.5.11 Add Page



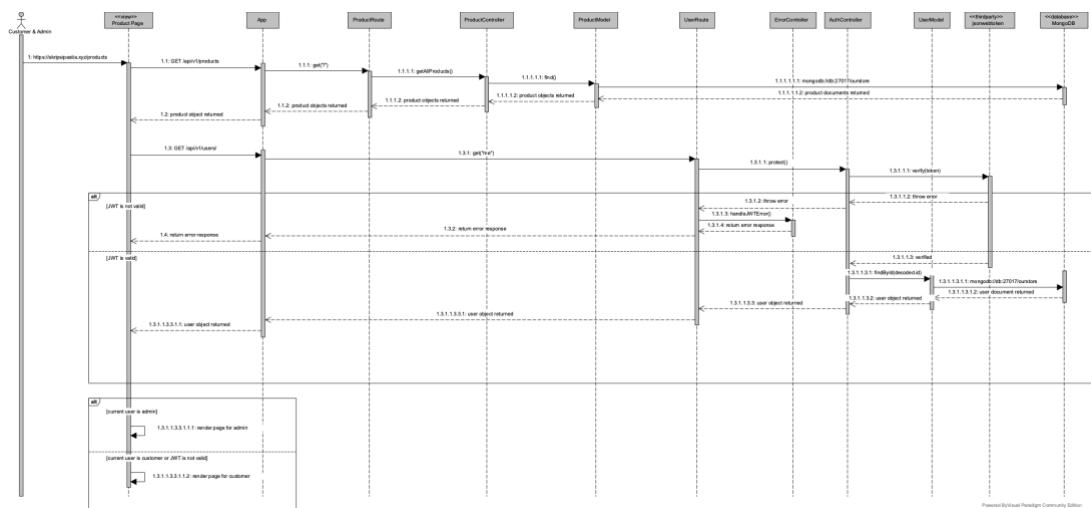
Gambar 3.66. Sequence Diagram Add Page

3.5.5.12 Delete Page



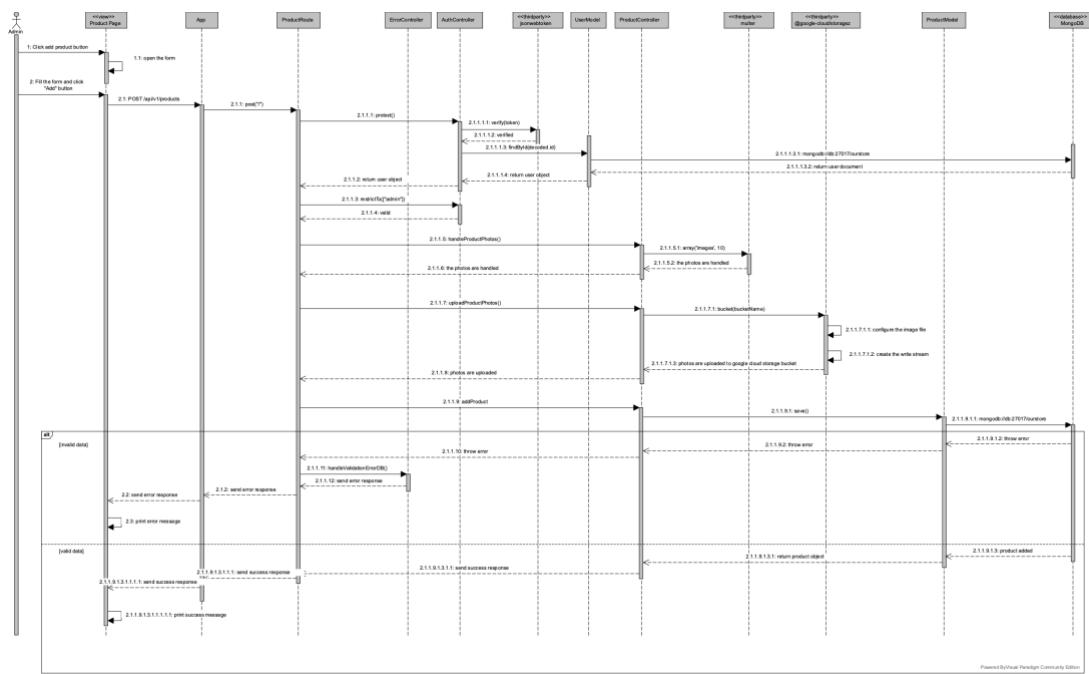
Gambar 3.67. Sequence Diagram Delete Page

3.5.5.13 View Products



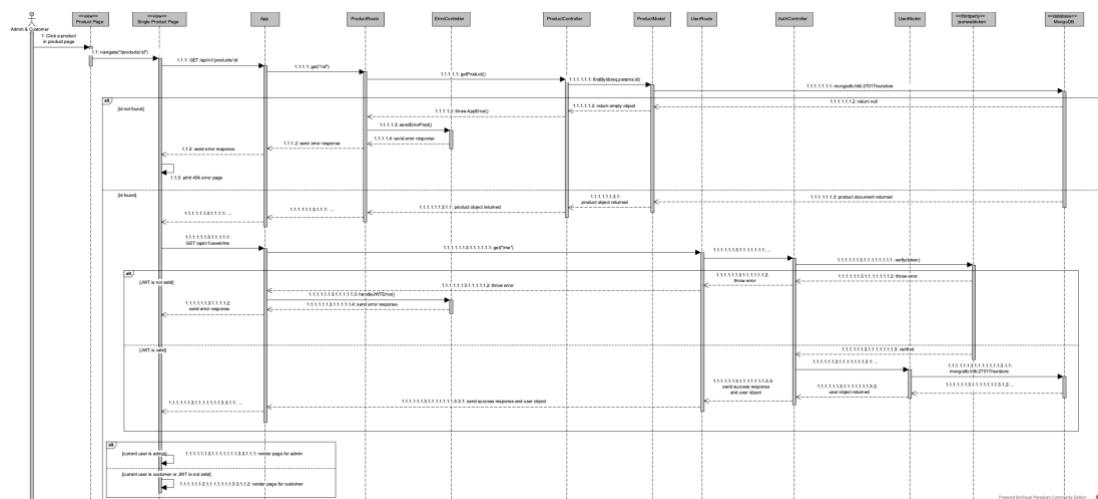
Gambar 3.68. Sequence Diagram View Products

3.5.5.14 Add Product



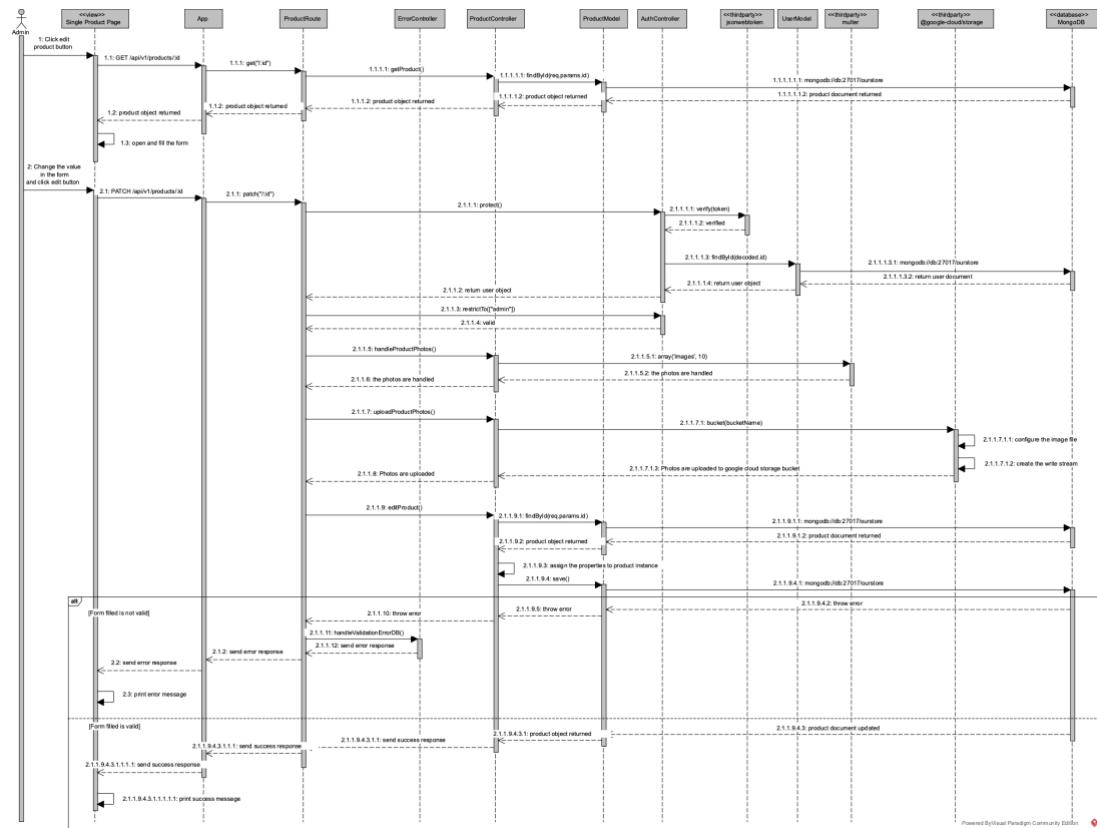
Gambar 3.69. Sequence Diagram Add Product

3.5.5.15 View Single Product



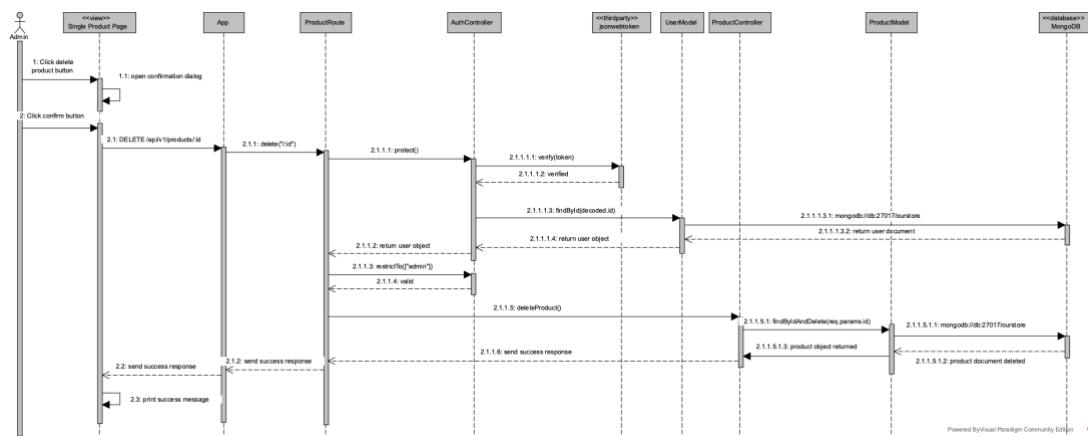
Gambar 3.70. Sequence Diagram View Single Product

3.5.5.16 Edit Product



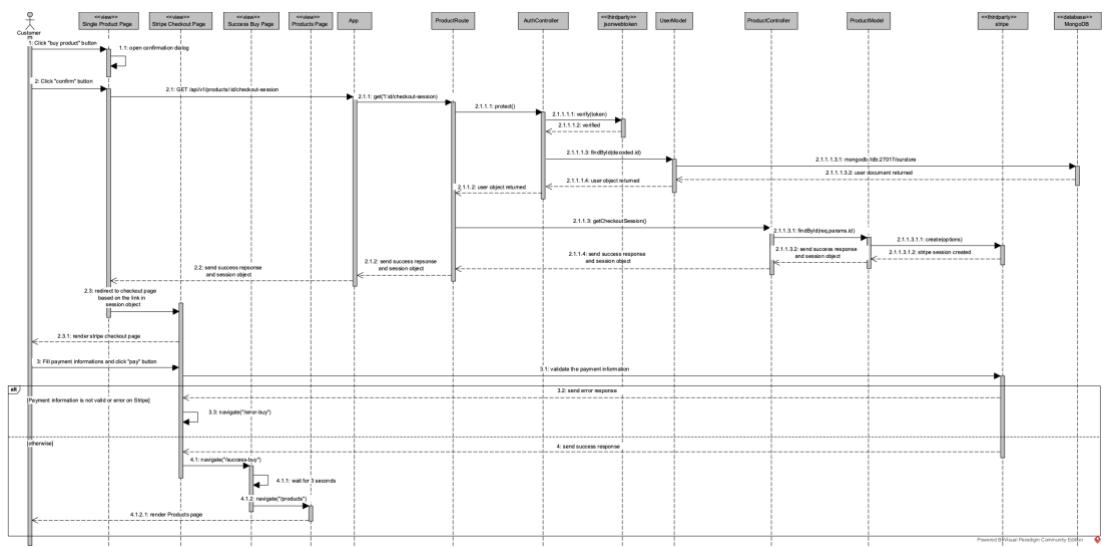
Gambar 3.71. Sequence Diagram Edit Product

3.5.5.17 Delete Product



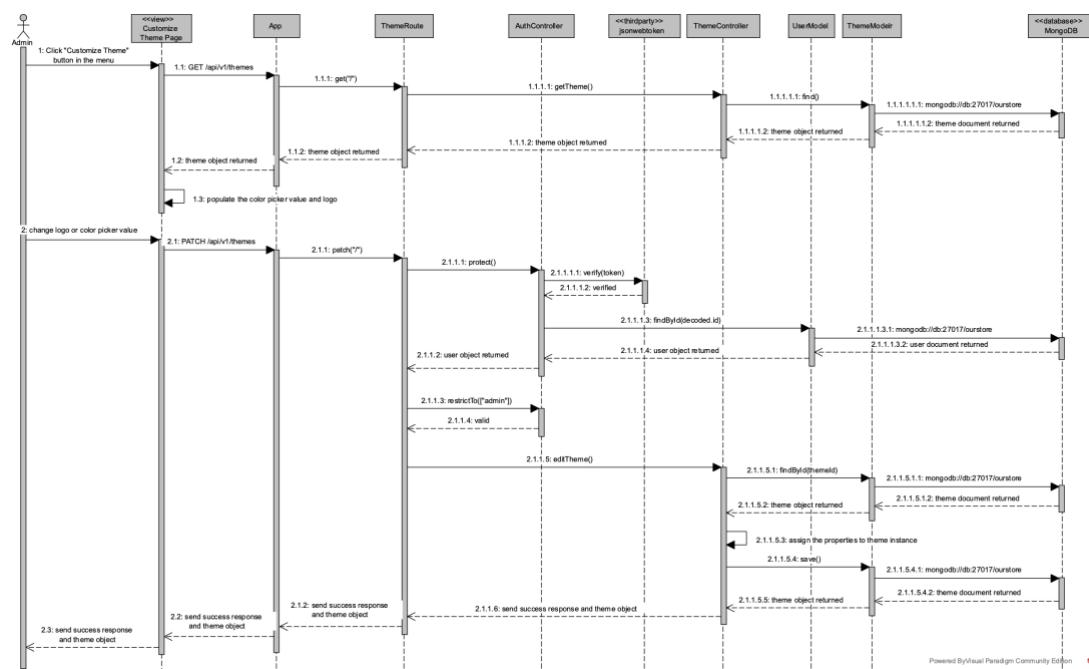
Gambar 3.72. Sequence Diagram Delete Product

3.5.5.18 Buy Product



Gambar 3.73. Sequence Diagram Buy Product

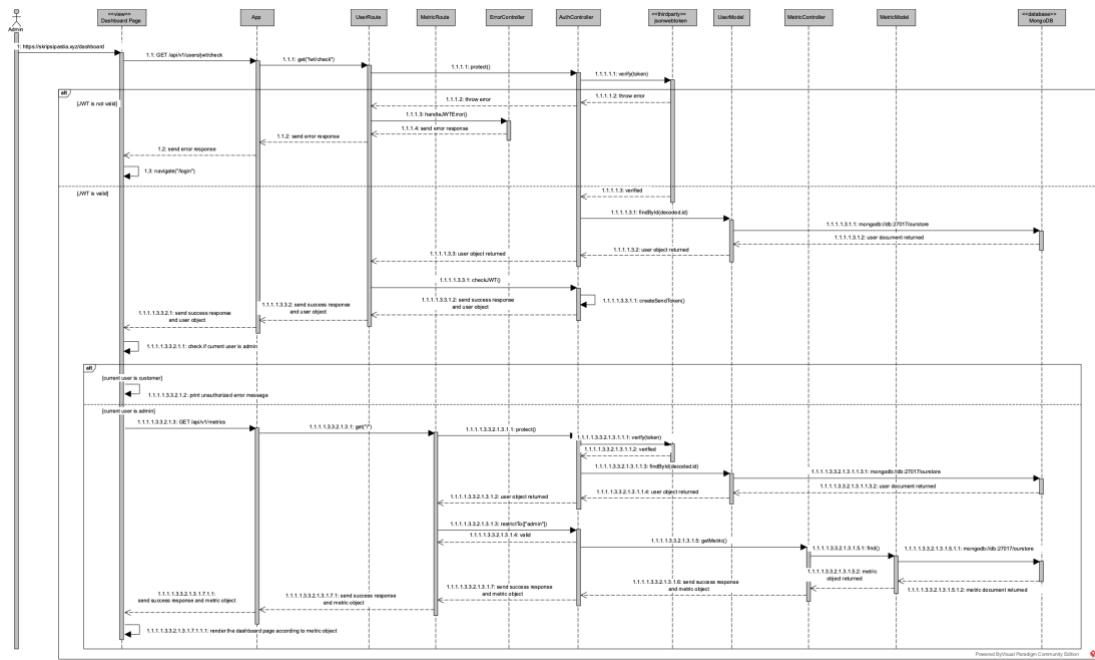
3.5.5.19 Customize Theme



Gambar 3.74. Sequence Diagram Customize Theme

Powered ByVisual Paradigm Community Edition

3.5.5.20 View Dashboard



Gambar 3.75. Sequence Diagram View Dashboard

3.6 Perancangan Layar

3.6.1 Halaman *Login*

The wireframe shows a login interface. At the top left is a circular logo placeholder labeled "LOGO". Below it is the text "Log in.". There are two input fields: one for "Email" and one for "Password". Below these is a "Log in" button. At the bottom left, there is a note: "Don't have an account yet? [Sign up](#)" and "[Forgot Password](#)". The right side of the wireframe is a large, empty rectangular area.

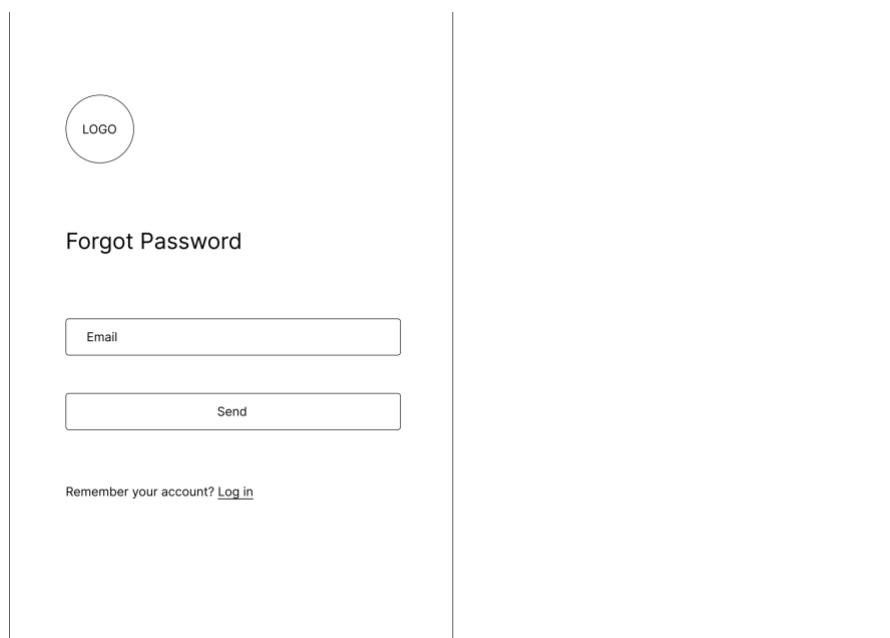
Gambar 3.76. Perancangan Halaman *Login*

3.6.2 Halaman *Register*

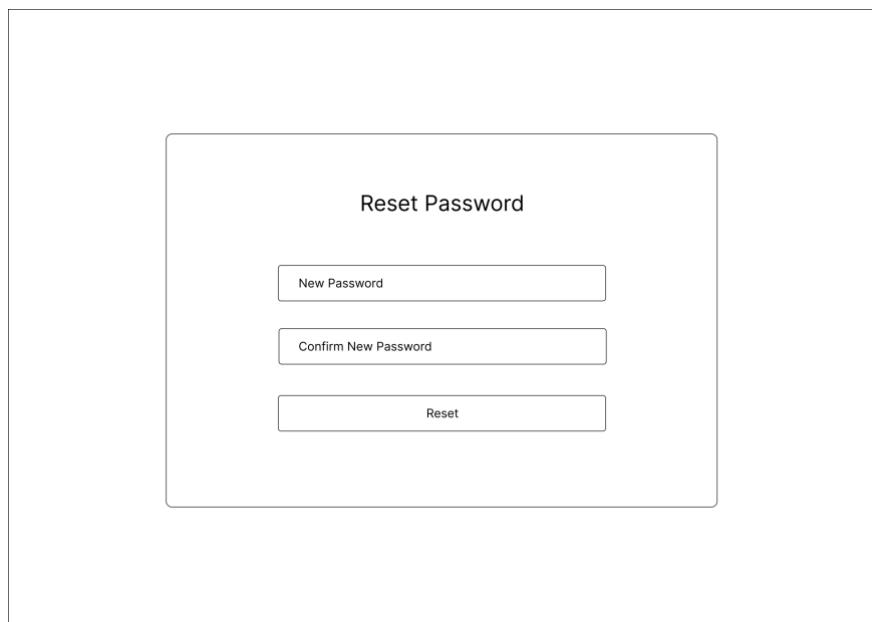
The wireframe shows a registration interface. At the top right is a circular logo placeholder labeled "LOGO". Below it is the text "Sign Up". There are four input fields: "Email", "Username", "Password", and "Confirm Password". Below these is a "Sign Up" button. At the bottom right, there is a note: "Already have an account? [Log in](#)". The left side of the wireframe is a large, empty rectangular area.

Gambar 3.77. Perancangan Halaman *Register*

3.6.3 Halaman *Forgot Password*

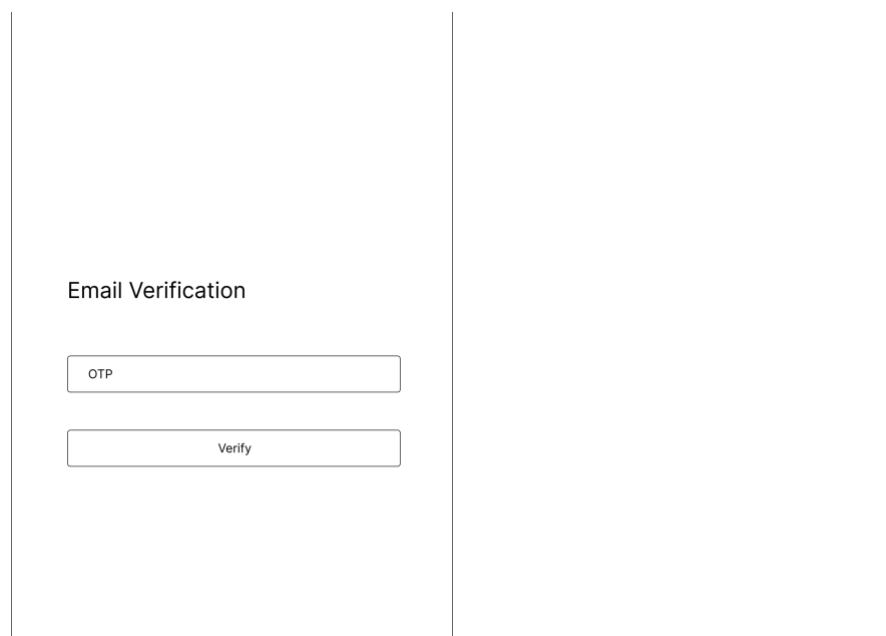


Gambar 3.78. Perancangan Halaman *Forgot Password*



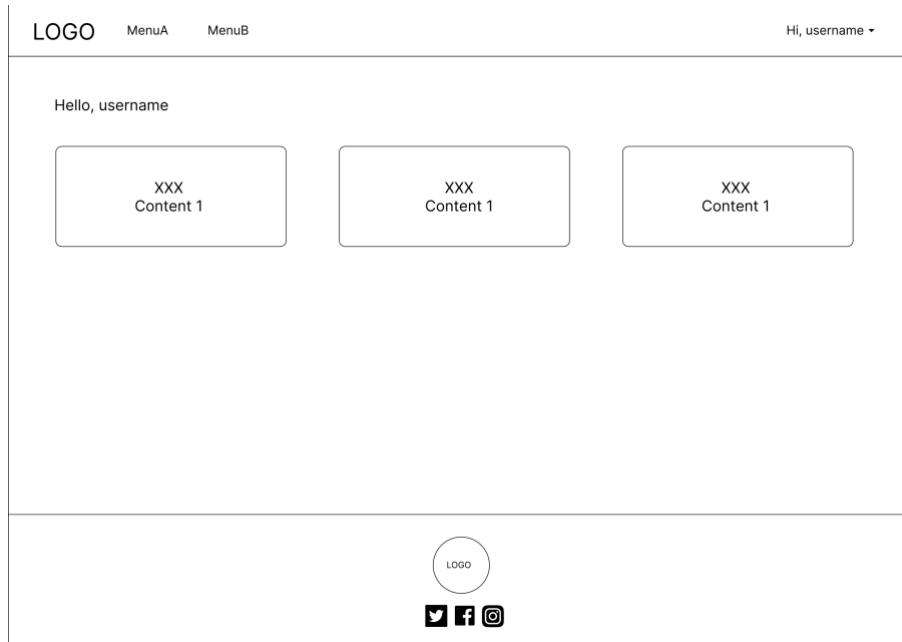
Gambar 3.79. Perancangan Halaman *Reset Password*

3.6.4 Halaman *Verify* OTP

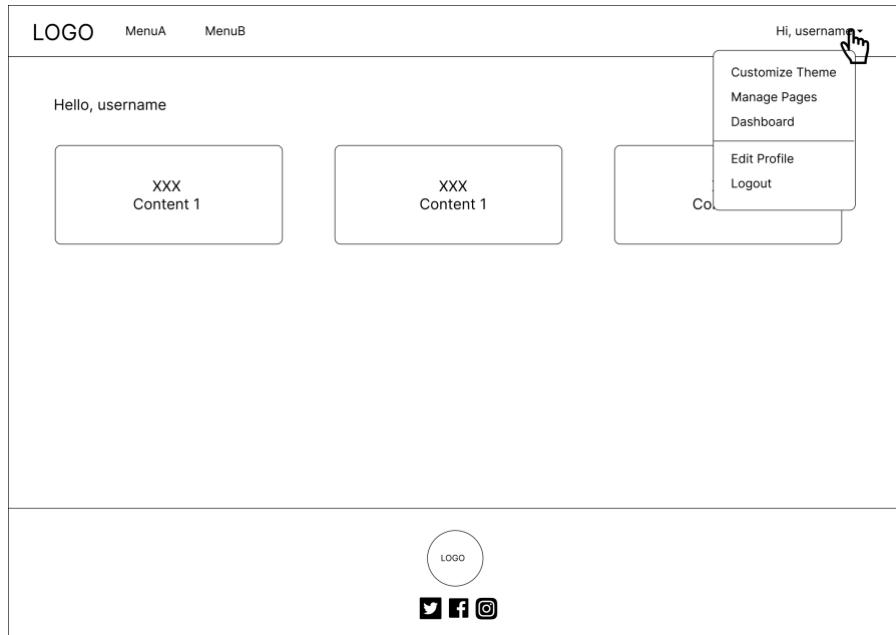


Gambar 3.80. Perancangan Halaman *Verify* OTP

3.6.5 Halaman Utama *Dashboard*

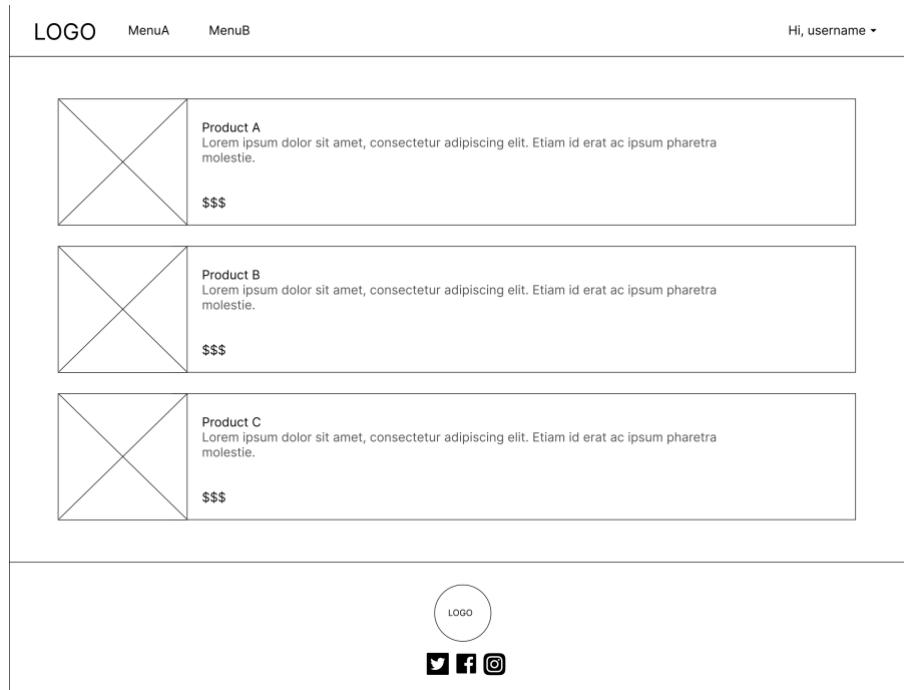


Gambar 3.81. Perancangan Halaman Utama *Dashboard*

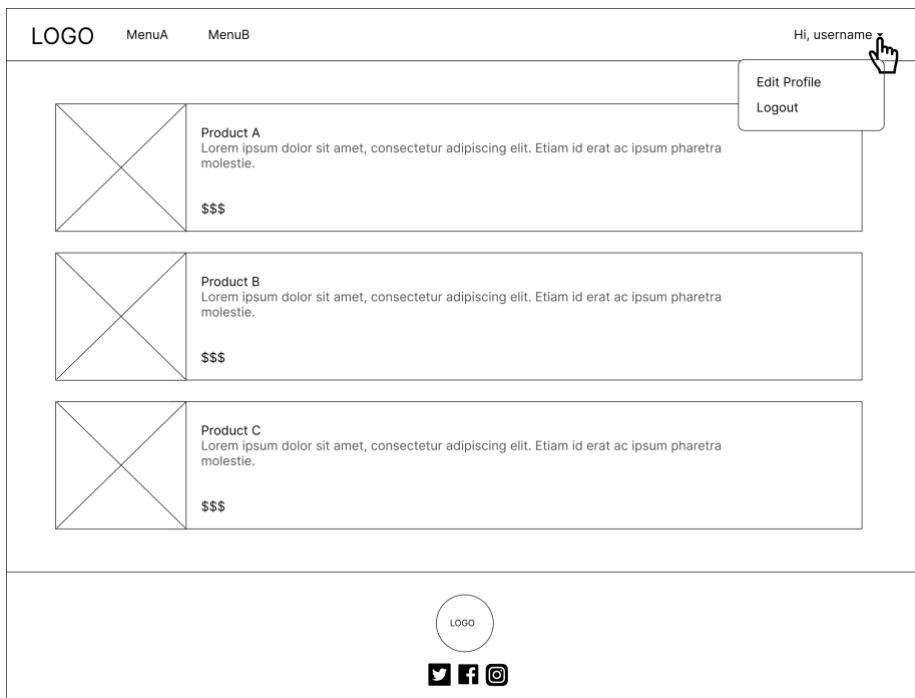


Gambar 3.82. Perancangan Halaman Utama *Dashboard* (*Dropdown Akun*)

3.6.6 Halaman Utama *Products* untuk *Customer*

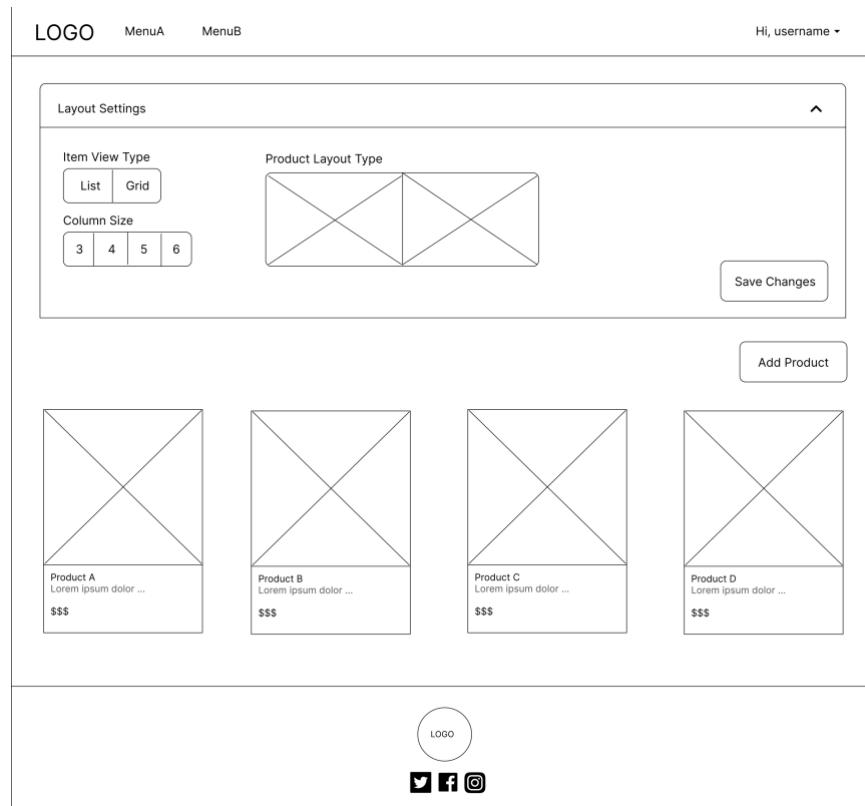


Gambar 3.83. Perancangan Halaman Utama *Products* untuk *Customer*



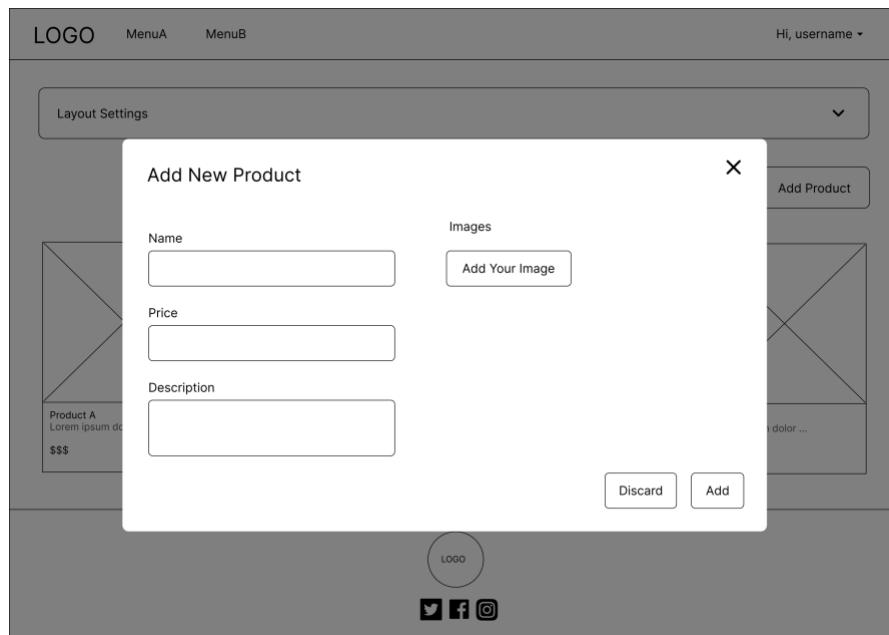
Gambar 3.84. Perancangan Halaman Utama *Products* untuk *Customer*
(*Dropdown Akun*)

3.6.7 Halaman *View Products* untuk Admin



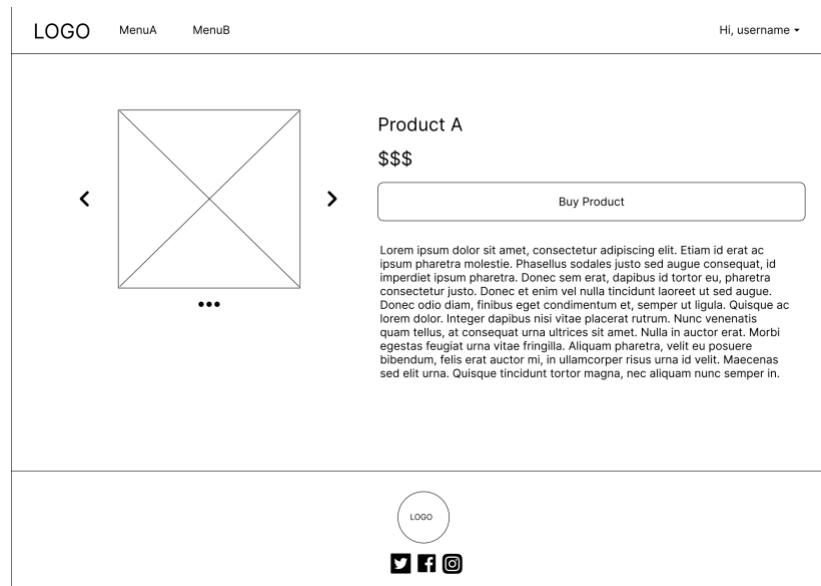
Gambar 3.85. Perancangan Halaman *View Products* untuk *Admin*

3.6.8 Halaman *Add Product* untuk Admin



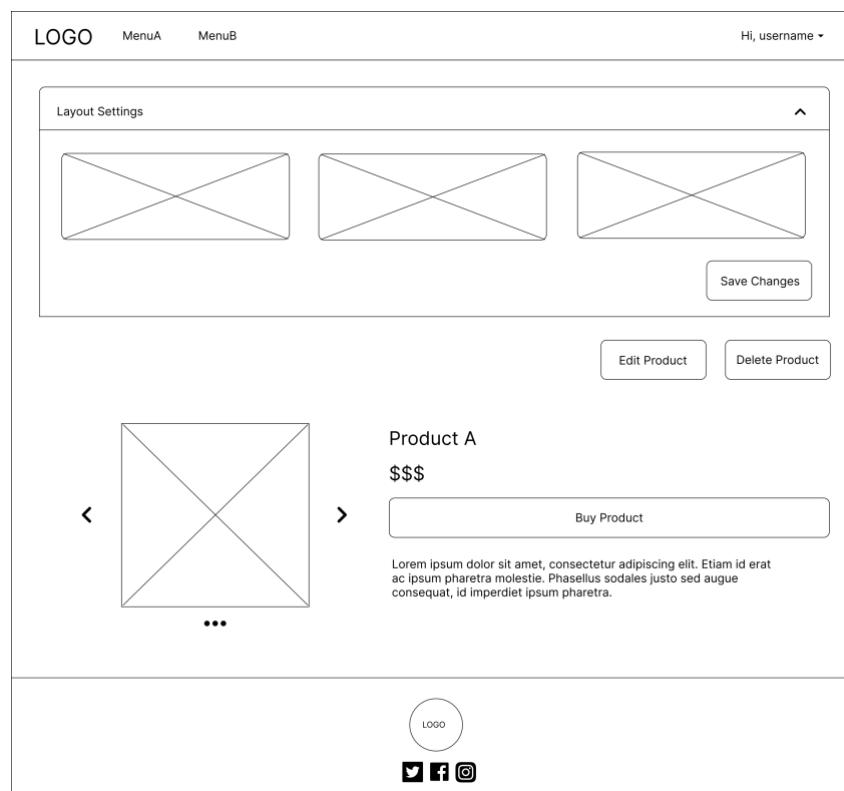
Gambar 3.86. Perancangan Halaman *Add Product* untuk *Admin*

3.6.9 Halaman *View Single Product* untuk *Customer*



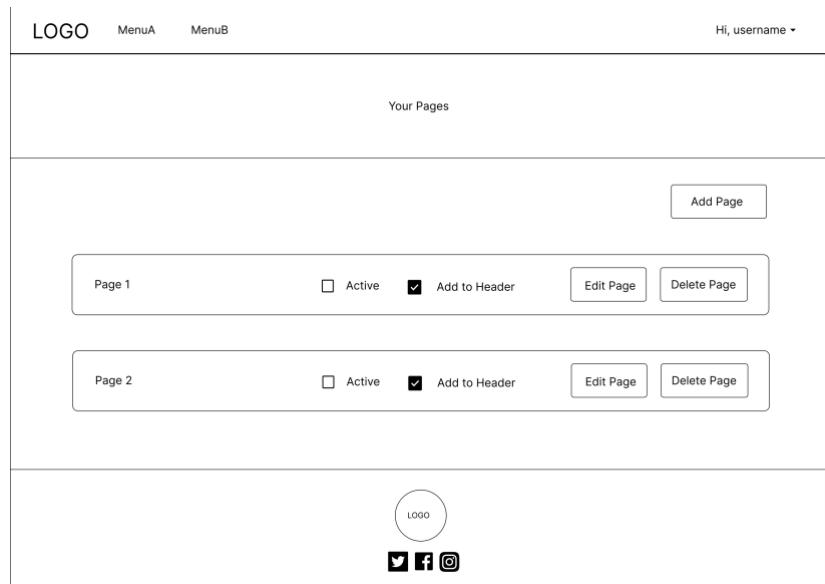
Gambar 3.87. Perancangan Halaman *View Single Product* untuk *Customer*

3.6.10 Halaman *View Single Product* untuk *Admin*



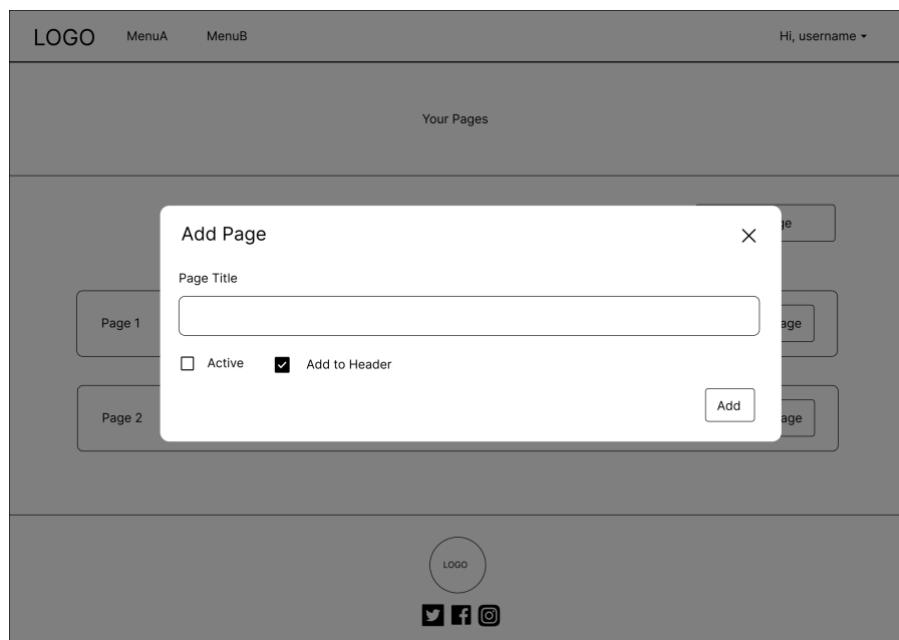
Gambar 3.88. Perancangan Halaman *View Single Product* untuk *Admin*

3.6.11 Halaman *Manage Pages*



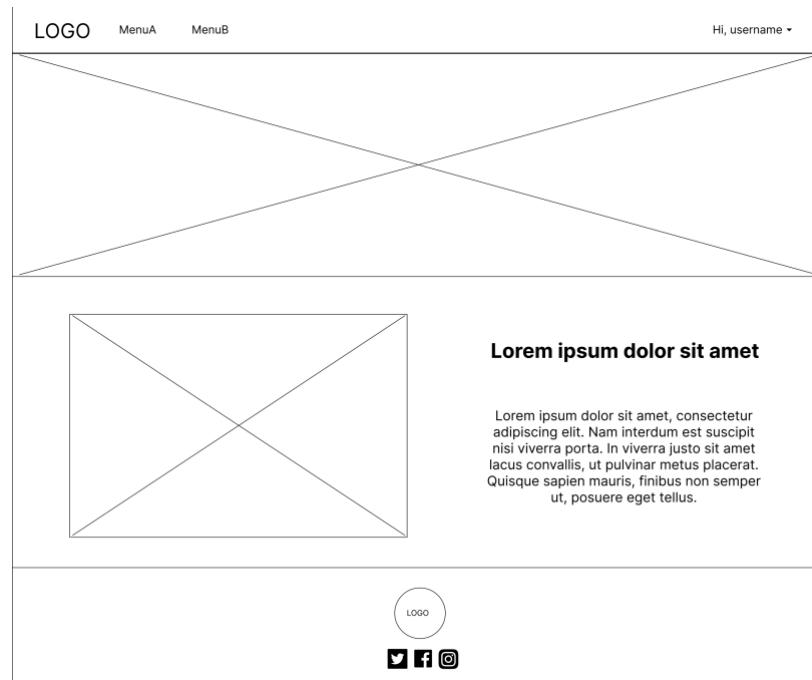
Gambar 3.89. Perancangan Halaman *Manage Pages*

3.6.12 Halaman *Add Page*



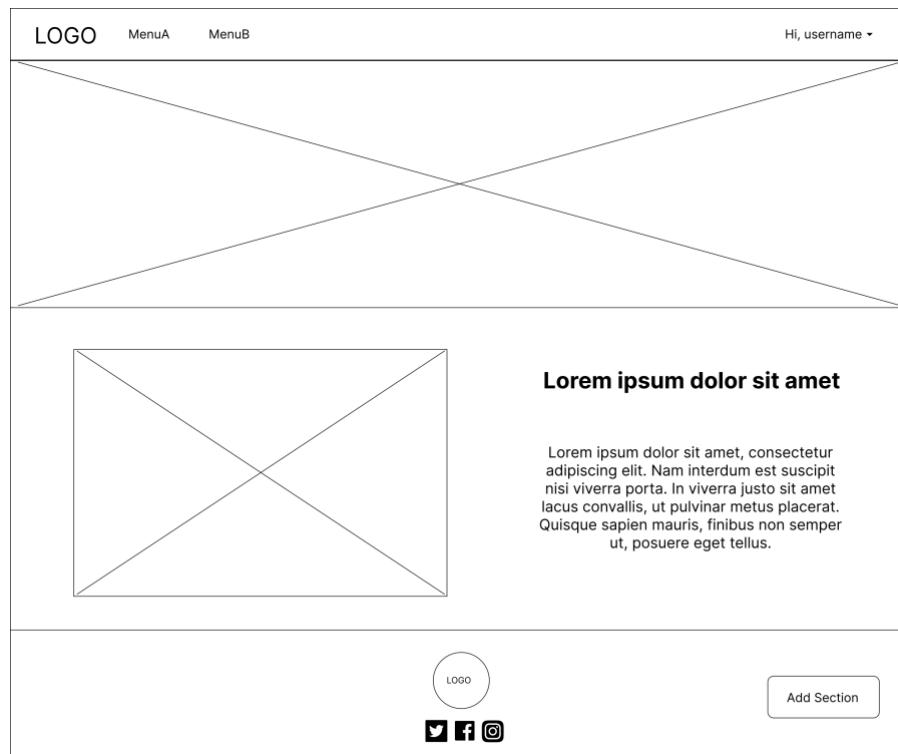
Gambar 3.90. Perancangan Halaman *Add Page*

3.6.13 Halaman *View Page* untuk *Customer*



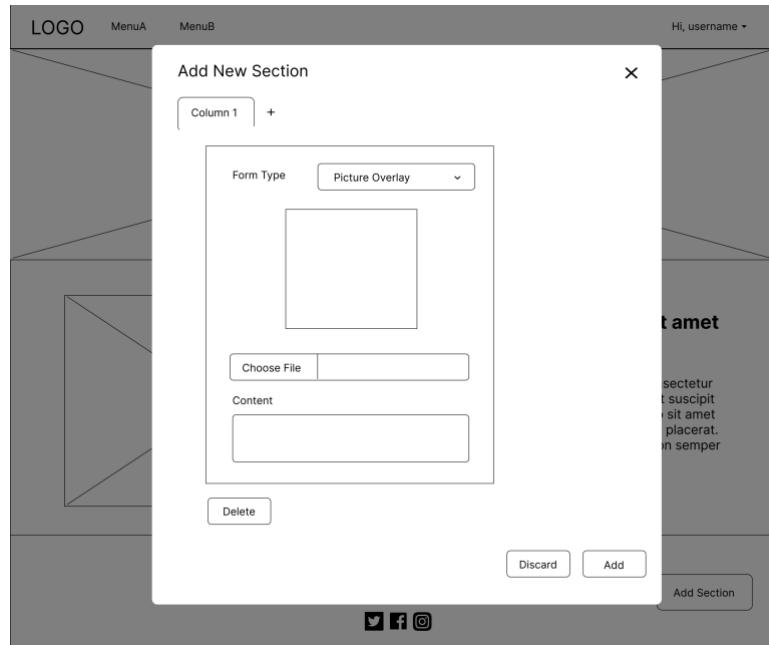
Gambar 3.91. Perancangan Halaman *View Page* untuk *Customer*

3.6.14 Halaman *View Page* untuk Admin



Gambar 3.92. Perancangan Halaman *View Page* untuk *Admin*

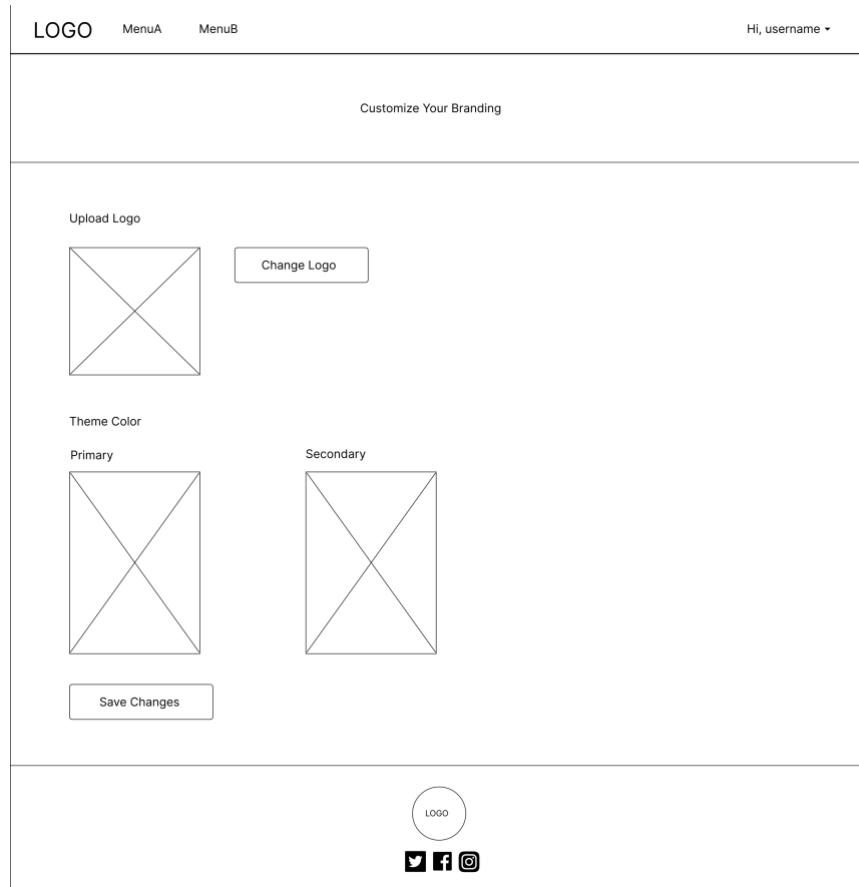
3.6.15 Halaman *Add Section* untuk Admin



Gambar 3.93. Perancangan Halaman *Add Section* untuk *Admin*

3.6.16 Halaman *Edit Profile*

Gambar 3.94. Perancangan Halaman *Edit Profile*

3.6.17 Halaman *Customize Theme*Gambar 3.95. Perancangan Halaman *Customize Theme*

3.7 Perancangan Database

3.7.1 Data Dictionary

3.7.1.1 Dokumen User

Tabel 3.31. *Data Dictionary User*

Field	Data Type	Description	Constraint
Email	String	The user's email	- Required field - Unique - Email format (enforced by validator library)
Username	String	The user's username	- Required field - Unique - Max length is 20
Password	String	The user's password	- Required field - Hashed + salted (in business logic)
Role	String	The role to restrict users to access certain resources	- The value can only be "customer" or "admin"
PasswordChangedAt	Date	The date when password changed to ensure the integrity of jwt	-
PasswordResetToken	String	The token user uses to recover the account when the user forgot the password	-
PasswordResetExpires	Date	The date when PasswordResetToken expires	-
Otp	String	The OTP for account verification when registered at the first time	-
Verified	Boolean	The data to state if the user is verified or not	-
isActive	Boolean	The data to state if the account is active or not (deleted)	-

3.7.1.2 Dokumen *Product*

Tabel 3.32. *Data Dictionary Product*

Field	Data Type	Description	Constraint
Name	String	The name of the product	- Required field - Max length is 100
Price	Number	The price of the product (in rupiah)	- Required field - Minimum 0 (stripe enforced to set minimum of 1\$ but it keeps 0 in db schema in case if it supports another currencies in the future) - Maximum 10B (1e+10)
Description	String	The description of the product	- Required column - max length is 2000
Images	List	The list of images attached	-
	imageLink	String	The link of each image attached - URL format (enforced by validator library)

3.7.1.3 Dokumen Page

Tabel 3.33. *Data Dictionary Page*

Field	Data Type	Description	Constraint
Name	String	The name of the page	- Required field
IsActive	Boolean	The data to state if the page is active or not (under development)	- Required field
IsInHeader	Boolean	The data to state if the page is in the header or not	- Required field
Sections	List	The list of customizable sections that exist in the page	-
Columns	Object	The column of each section in the page	-
	ColumnType	String	The type of column - Required field - The value can only "TEXT_TYPE", "PICTURE_TYPE", "PICTUREOVERLAY_TYPE", or "PRODUCT_TYPE"
	ImageLink	String	The link of image attached in a column - Have to exist only if the ColumnType is either "PICTURE_TYPE" or "PICTUREOVERLAY_TYPE" - URL format (enforced by validator library)
	Header	String	The header (text) of a column - Have to exist only if the ColumnType is "TEXT_TYPE"
	Content	String	The content (text) of a column - Have to exist only if the ColumnType is either "TEXT_TYPE" or "PICTUREOVERLAY_TYPE"
	ProductId	String	The product id of product attached to a column - Have to exist only if the ColumnType is "PRODUCT_TYPE"

3.7.1.4 Dokumen *Theme*

Tabel 3.34. *Data Dictionary Theme*

Field	Data Type	Description	Constraint
Primary	String	The primary theme color (6-digit RGB hex)	- Required field
Secondary	String	The secondary/accent theme color (6-digit RGB hex)	- Required field
TemplateId	String	The template id related to template document in product page	- Required field
SingleProductTemplateId	String	The template id for view single product page related to template document in product page	- Required field
ViewType	String	The view type applied in product page	- Required field - The value can only "list" or "grid"
ColumnSize	Number	The column size applied in product page	- Required field - Minimum 3 - Maximum 6
LogoLink	String	The link of the brand logo	- URL format (enforced by validator library)

3.7.1.5 Dokumen *Metric*

Tabel 3.35. *Data Dictionary Metric*

Field	Data Type	Description	Constraint
ProductSoldPerMonth	Number	The amount of product quantity sold per month	- Required field
RevenuePerMonth	Number	The amount of revenue (product price) sold per month	- Required field
DailyActiveUser	Number	The averaged DAU (total user logged in divided by the day elapsed from the first time the app launched)	- Required field