

**PENGEMBANGAN SISTEM INFORMASI TOKO ONLINE NUE
PINS BERBASIS WEB DENGAN MEMANFAATKAN API
MIDTRANS SEBAGAI MEDIA PEMBAYARAN**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dorojatun Chandrabumi

NIM: 165150707111012



**PROGRAM STUDI TEKNOLOGI INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

**MALANG
2021**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI TOKO *ONLINE* NUE PINS BERBASIS *WEB*
DENGAN MEMANFAATKAN *API MIDTRANS* SEBAGAI MEDIA PEMBAYARAN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Dorojatun Chandrabumi

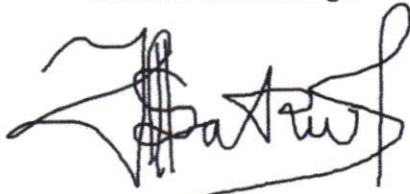
NIM: 165150707111012

Skripsi ini telah diuji dan dinyatakan lulus pada

13 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing 1



Issa Arwani, S.Kom., M.Sc.
NIP: 19830922 201212 1 003

Dosen Pembimbing 2




Buce Trias Hanggara, S.Kom., M.Kom.
NIP: 19890426 201903 1 009

Mengetahui

Ketua Jurusan Sistem Informasi




Issa Arwani, S.Kom., M.Sc. yw
NIP: 19830922 201212 1 003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Jakarta, 11 Juni 2021



Dorojatun Chandrabumi

NIM : 165150707111012



ABSTRAK

Nue Pins merupakan merek atau *brand* yang bergerak dalam bidang aksesoris wanita seperti cincin, anting, kalung, *hairpins*, *strap* pada kacamata dan masker. Penjualan produk Nue Pins, dilakukan dengan secara langsung dengan membuka gerai pada *department store* di Jakarta, dan menjual secara *online* melalui media social *Instagram*. Untuk melakukan pemesanan secara *online*, pembeli harus menanyakan terlebih dahulu mengenai informasi produk dan harga produk. Proses transaksi yang dilakukan, pembeli melakukan pengiriman foto bukti *transfer* kepada pihak dari Nue Pins. Hal ini menjadi kurang efektif dikarenakan pembeli harus menanyakan langsung terkait produk yang masih tersedia dan produk yang sudah tidak tersedia atau habis terjual dan pihak dari Nue Pins harus melakukan pengecekan kembali terhadap foto bukti transfer yang telah dikirimkan.

Bedasarkan permasalahan diatas, maka peneliti akan membangun sistem informasi toko *online* berbasis *website* yang memanfaatkan teknologi *payment gateway* atau media pembayaran dengan menggunakan *framework code igniter* yang dapat memberikan informasi terkait produk yang dijual oleh Nue Pin dan dapat melakukan proses pembelian dari produk Nue Pins, sekaligus dapat memberikan informasi mengenai status pemesanan produk yang dibeli. Sistem yang dibangun, juga dapat memberikan pengoperasian produk, dan pemesanan bagi pihak Nue Pins. Proses transaksi memanfaatkan teknologi *payment gateway* dengan menggunakan *API Midtrans*. Metode pengembangan sistem menggunakan metode *waterfall*, dan pendefinisian kebutuhan sistem akan dilakukan dengan wawancara kepada pemilik dari *brand* Nue Pins. Berdasarkan hasil dari pengujian dengan menggunakan metode *blackbox*, kebutuhan yang telah diimplementasikan menghasilkan status valid. Kemudian berdasarkan hasil pengujian *usability* dengan menggunakan tipe skoring *system usability scale* (SUS) dengan jumlah responden 5, sistem menghasilkan penilaian rata – rata sebesar 73.5 yang termasuk dalam *grade C*, dan tingkat *acceptability* adalah *acceptable* yang menunjukkan sistem dapat diterima oleh pengguna.

Kata kunci : Nue Pins, Sistem Informasi, *Payment Gateway*, *Website*, *Waterfall*, *Blackbox*, *System Usability Scale*



ABSTRACT

Nue pins is a brand that launch a females accessories products, like rings, necklace, hairpins, mask and glasses strap. The process of selling products in Nue Pins, is run by opening the store at some department store held in Jakarta, and for online selling, Nue Pins make Instagram account to sell the products. To order products from Instagram, buyer must ask the availability of products and product price. For the transaction process, buyer must send a photo of payment receipt to Nue Pins contact. This is less effective, because buyer didn't know the information of products which is available or not. It makes buyer must ask first to buy the products, and Nue Pins side must check the receipt of payment transfer.

Based on problems above, the researcher will develop a web based information system by using payment gateway technology and use code igniter frame work, that can give the information of Nue Pins products, and can make process of selling, then the system will give order status. The system also can make operation for controlling products and orders. Transaction process will take the advantages of payment gateway technology by using Midtrans API. System will use waterfall methods as system development lifecycle, and the needs of system will defined by do interview with Nue Pins owner. Based on testing results by using blackbox method, the system needs that already implemented, has been valid. That can be said, the system function are fulfill the needs of system there were defined in needs analytics before. Then, based on usability test with using system usability scale scoring type for 5 response, system have 73.5 score rate. And the score is included in the C grade, which C grade is categorized as acceptable system for Users.

Keywords : *Nue Pins, Information System, Payment Gateway, Web Based, Waterfall, Blackbox Method, System Usability Scale Method*

DAFTAR ISI

| | |
|--|----|
| PENGESAHAN | 2 |
| PERNYATAAN ORISINALITAS | 3 |
| KATA PENGANTAR | 4 |
| ABSTRAK | 5 |
| ABSTRACT | 6 |
| DAFTAR ISI | 7 |
| DAFTAR TABEL | 14 |
| DAFTAR GAMBAR | 17 |
| BAB 1 PENDAHULUAN | 20 |
| 1.1 Latar Belakang | 20 |
| 1.2 Rumusan Masalah | 21 |
| 1.3 Tujuan | 22 |
| 1.4 Manfaat | 22 |
| 1.4.1 Bagi Pemilik Nue Pins | 22 |
| 1.4.2 Calon Pelanggan Nue Pins | 22 |
| 1.5 Batasan Masalah | 23 |
| 1.6 Sistematika Pembahasan | 23 |
| Bab 2 LANDASAN KEPUSTAKAAN | 25 |
| 2.1 Kajian Pustaka | 25 |
| 2.2 Toko Online | 25 |
| 2.3 Nue Pins | 25 |
| 2.4 <i>Midtrans</i> | 26 |
| 2.5 <i>Application Programming Interface (API)</i> | 26 |
| 2.6 <i>Javascript Object Notation (JSON)</i> | 27 |
| 2.7 Code Igniter | 27 |
| 2.8 MySQL | 27 |
| 2.9 Unified Modelling Language (UML) | 27 |
| 2.9.1 Use Case Diagram | 28 |
| 2.9.2 Activity Diagram | 29 |
| 2.9.3 <i>Sequence Diagram</i> | 29 |
| 2.9.4 <i>Class Diagram</i> | 30 |

| | | |
|---------------------------------------|---|-----------|
| 2.10 | Physical Data Model | 31 |
| 2.11 | Waterfall | 32 |
| 2.11 | Black Box Testing | 32 |
| 2.12 | Usability Testing..... | 32 |
| BAB 3 METODOLOGI | | 34 |
| 3.1 | Studi Literatur | 34 |
| 3.2 | Analisis Kebutuhan Sistem | 35 |
| 3.3 | Perancangan Sistem | 35 |
| 3.4 | Implementasi Sistem | 35 |
| 3.5 | Pengujian Sistem | 35 |
| 3.6 | Kesimpulan dan Saran..... | 36 |
| Bab 4 Analisis Kebutuhan | | 37 |
| 4.1 | Gambaran Umum Sistem | 37 |
| 4.2 | Identifikasi Aktor | 38 |
| 4.3 | Identifikasi Kebutuhan Sistem | 38 |
| 4.3.1 | Kebutuhan Fungsional | 39 |
| 4.3.2 | Kebutuhan Non Fungsional | 41 |
| 4.4 | Use Case Diagram..... | 42 |
| 4.5 | Use Case Scenario | 43 |
| 4.5.1 | Use Case Scenario Login User | 43 |
| 4.5.2 | Use Case Scenario Login Admin | 43 |
| 4.5.3 | Use Case Scenario Register | 44 |
| 4.5.4 | Use Case Scenario Logout User..... | 45 |
| 4.5.5 | Use Case Scenario Logout Admin..... | 46 |
| 4.5.6 | Use Case Scenario Menampilkan Produk | 46 |
| 4.5.7 | Use Case Scenario Keranjang Belanja | 47 |
| 4.5.8 | Use Case Scenario Menghapus Produk..... | 47 |
| 4.5.9 | Use Case Scenario Menentukan Jumlah Pembelian | 48 |
| 4.5.10 | Use Case Scenario Membuat Pemesanan..... | 49 |
| 4.5.11 | Use Case Scenario Track Pemesanan | 50 |
| 4.5.12 | Use Case Scenario Informasi Produk..... | 50 |
| 4.5.13 | Use Case Scenario Menambah Data Produk..... | 51 |

| | | |
|--------|--|----|
| 4.5.14 | <i>Use Case Scenario</i> Mengubah Data Produk..... | 52 |
| 4.5.15 | <i>Use Case Scenario</i> Menghapus Data Produk..... | 52 |
| 4.5.16 | <i>Use Case Scenario</i> Informasi Pemesanan..... | 53 |
| 4.5.17 | <i>Use Case Scenario</i> Verifikasi Pemesanan..... | 54 |
| 4.5.18 | <i>Use Case Scenario</i> Dashboard..... | 54 |
| 4.6 | <i>Activity Diagram</i> | 55 |
| 4.6.1 | <i>Activity Diagram</i> Login User..... | 55 |
| 4.6.2 | <i>Activity Diagram</i> Login Admin..... | 56 |
| 4.6.3 | <i>Activity Diagram</i> Register..... | 57 |
| 4.6.4 | <i>Activity Diagram</i> Logout User..... | 58 |
| 4.6.5 | <i>Activity Diagram</i> Logout Admin..... | 59 |
| 4.6.6 | <i>Activity Diagram</i> Menampilkan Produk..... | 60 |
| 4.6.7 | <i>Activity Diagram</i> Keranjang Belanja..... | 61 |
| 4.6.8 | <i>Activity Diagram</i> Menghapus Produk..... | 62 |
| 4.6.9 | <i>Activity Diagram</i> Menentukan Jumlah Pembelian..... | 63 |
| 4.6.10 | <i>Activity Diagram</i> Membuat Pemesanan..... | 64 |
| 4.6.11 | <i>Activity Diagram</i> Track Pemesanan..... | 65 |
| 4.6.12 | <i>Activity Diagram</i> Informasi Produk..... | 66 |
| 4.6.13 | <i>Activity Diagram</i> Menambah Data Produk..... | 67 |
| 4.6.14 | <i>Activity Diagram</i> Mengubah Data Produk..... | 68 |
| 4.6.15 | <i>Activity Diagram</i> Menghapus Data Produk..... | 68 |
| 4.6.16 | <i>Activity Diagram</i> Informasi Pemesanan..... | 69 |
| 4.6.17 | <i>Activity Diagram</i> Verifikasi Pemesanan..... | 70 |
| 4.6.18 | <i>Activity Diagram</i> Dashboard..... | 70 |
| Bab 5 | Perancangan sistem..... | 72 |
| 5.1 | Perancangan Arsitektur Sistem..... | 72 |
| 5.2 | Perancangan <i>Sequence Diagram</i> | 73 |
| 5.2.1 | <i>Sequence Diagram</i> Login User..... | 73 |
| 5.2.3 | <i>Sequence Diagram</i> Login Admin..... | 74 |
| 5.2.3 | <i>Sequence Diagram</i> Register..... | 75 |
| 5.2.4 | <i>Sequence Diagram</i> Logout User..... | 75 |
| 5.2.5 | <i>Sequence Diagram</i> Logout Admin..... | 76 |



| | | |
|--------|--|-----|
| 5.2.6 | Sequence Diagram Menampilkan Produk..... | 77 |
| 5.2.7 | Sequence Diagram Keranjang Belanja..... | 77 |
| 5.2.8 | Sequence Diagram Menghapus Produk..... | 78 |
| 5.2.9 | Sequence Diagram Menentukan Jumlah Pembelian..... | 79 |
| 5.2.10 | Sequence Diagram Membuat Pemesanan..... | 80 |
| 5.2.11 | Sequence Diagram Track Pemesanan..... | 81 |
| 5.2.12 | Sequence Diagram Informasi Produk..... | 81 |
| 5.2.13 | Sequence Diagram Menambah Data Produk..... | 82 |
| 5.2.14 | Sequence Diagram Mengubah Data Produk..... | 83 |
| 5.2.15 | Sequence Diagram Menghapus Data Produk..... | 84 |
| 5.2.16 | Sequence Diagram Informasi Pemesanan..... | 85 |
| 5.2.17 | Sequence Diagram Verifikasi Pemesanan..... | 85 |
| 5.2.18 | Sequence Diagram Dashboard..... | 86 |
| 5.3 | Perancangan Class Diagram..... | 86 |
| 5.4 | Perancangan Physical Data Model..... | 88 |
| 5.5 | Perancangan Antarmuka..... | 89 |
| 5.5.1 | Perancangan Antarmuka Login User..... | 90 |
| 5.5.2 | Perancangan Antarmuka Login Admin..... | 91 |
| 5.5.3 | Perancangan Antarmuka Register..... | 92 |
| 5.5.4 | Perancangan Antarmuka Logout User..... | 93 |
| 5.5.5 | Perancangan Antarmuka Logout Admin..... | 94 |
| 5.5.6 | Perancangan Antarmuka Menampilkan Produk..... | 95 |
| 5.5.7 | Perancangan Antarmuka Keranjang Belanja..... | 97 |
| 5.5.8 | Perancangan Antarmuka Menghapus Produk..... | 98 |
| 5.5.9 | Perancangan Antarmuka Menentukan Jumlah Pembelian..... | 99 |
| 5.5.10 | Perancangan Antarmuka Membuat Pemesanan..... | 100 |
| 5.5.11 | Perancangan Antarmuka Track Pemesanan..... | 101 |
| 5.5.12 | Perancangan Antarmuka Informasi Produk..... | 102 |
| 5.5.13 | Perancangan Antarmuka Menambah Data Produk..... | 102 |
| 5.5.14 | Perancangan Antarmuka Mengubah Data Produk..... | 103 |
| 5.5.15 | Perancangan Antarmuka Menghapus Data Produk..... | 103 |
| 5.5.16 | Perancangan Antarmuka Informasi Pemesanan..... | 104 |

| | | |
|---------------------------------|---|-----|
| 5.5.17 | Perancangan Antarmuka Verifikasi Pemesanan | 104 |
| 5.5.18 | Perancangan Antarmuka Dashboard | 105 |
| Bab 6 Implementasi sistem | | 106 |
| 6.1 | Spesifikasi Sistem | 106 |
| 6.1.1 | Spesifikasi Perangkat Keras dan Perangkat Lunak | 106 |
| 6.2 | Implementasi Basis Data | 107 |
| 6.2.1 | Implementasi Tabel User | 107 |
| 6.2.2 | Implementasi Tabel Admin | 107 |
| 6.2.3 | Implementasi Tabel produk | 108 |
| 6.2.4 | Implementasi Tabel Keranjang Belanja | 108 |
| 6.2.5 | Implementasi Tabel Pemesanan | 109 |
| 6.3 | Implementasi API Midtrans | 109 |
| 6.4 | Implementasi Kode Program | 112 |
| 6.4.1 | Implementasi Login User | 112 |
| 6.4.2 | Implementasi Login Admin | 113 |
| 6.4.3 | Implementasi Register | 113 |
| 6.4.4 | Implementasi Logout User | 114 |
| 6.4.5 | Implementasi Logout Admin | 115 |
| 6.4.6 | Implementasi Menampilkan Produk | 115 |
| 6.4.7 | Implementasi Keranjang Belanja | 116 |
| 6.4.8 | Implementasi Menghapus Produk | 117 |
| 6.4.9 | Implementasi Menentukan Jumlah Pembelian | 117 |
| 6.4.10 | Implementasi Membuat Pemesanan | 118 |
| 6.4.11 | Implementasi Track Pemesanan | 120 |
| 6.4.12 | Implementasi Informasi Produk | 121 |
| 6.4.13 | Implementasi Menambah Data Produk | 121 |
| 6.4.14 | Implementasi Mengubah Data Produk | 122 |
| 6.4.15 | Implementasi Menghapus Data Produk | 124 |
| 6.4.16 | Implementasi Informasi Pemesanan | 125 |
| 6.4.17 | Implementasi Verifikasi Pemesanan | 125 |
| 6.4.18 | Implementasi Dashboard | 126 |
| 6.5 | Implementasi Antarmuka | 127 |

| | |
|---|-----|
| 6.5.1 Antarmuka Login User..... | 128 |
| 6.5.2 Antarmuka Login Admin | 128 |
| 6.5.3 Antarmuka Register | 129 |
| 6.5.4 Antarmuka Logout User | 130 |
| 6.5.5 Antarmuka Logout Admin..... | 130 |
| 6.5.6 Antarmuka Menampilkan Produk..... | 131 |
| 6.5.7 Antarmuka Keranjang Belanja | 131 |
| 6.5.8 Antarmuka Menghapus Produk..... | 132 |
| 6.5.9 Antarmuka Menentukan Jumlah Pembelian | 132 |
| 6.5.10 Antarmuka Membuat Pemesanan..... | 133 |
| 6.5.11 Antarmuka Track Pemesanan | 134 |
| 6.5.12 Antarmuka Informasi Produk | 135 |
| 6.5.13 Antarmuka Menambah Data Produk | 135 |
| 6.5.14 Antarmuka Mengubah Data Produk | 136 |
| 6.5.15 Antarmuka Menghapus Data Produk..... | 136 |
| 6.5.16 Antarmuka Informasi Pemesanan..... | 137 |
| 6.5.17 Antarmuka Verifikasi Pemesanan | 137 |
| 6.5.18 Antarmuka Dashboard | 138 |
| Bab 7 Pengujian sistem | 139 |
| 7.1 Pengujian <i>Blackbox</i> | 139 |
| 7.1.1 Pengujian <i>Blackbox Login User</i> | 139 |
| 7.1.2 Pengujian <i>Blackbox Login Admin</i> | 139 |
| 7.1.3 Pengujian <i>Blackbox Register</i> | 140 |
| 7.1.4 Pengujian <i>Blackbox Logout User</i> | 141 |
| 7.1.5 Pengujian <i>Blackbox Logout Admin</i> | 141 |
| 7.1.6 Pengujian <i>Blackbox Menampilkan Produk</i> | 142 |
| 7.1.7 Pengujian <i>Blackbox Keranjang Belanja</i> | 142 |
| 7.1.8 Pengujian <i>Blackbox Menghapus Produk</i> | 143 |
| 7.1.9 Pengujian <i>Blackbox Menentukan Jumlah Pembelian</i> | 144 |
| 7.1.10 Pengujian <i>Blackbox Membuat Pemesanan</i> | 144 |
| 7.1.11 Pengujian <i>Blackbox Track Pemesanan</i> | 145 |
| 7.1.12 Pengujian <i>Blackbox Informasi Produk</i> | 146 |

| | |
|---|-----|
| 7.1.13 Pengujian <i>Blackbox</i> Menambah Data Produk..... | 146 |
| 7.1.14 Pengujian <i>Blackbox</i> Mengubah Data Produk..... | 147 |
| 7.1.15 Pengujian <i>Blackbox</i> Menghapus Data Produk..... | 148 |
| 7.1.16 Pengujian <i>Blackbox</i> Informasi Pemesanan..... | 149 |
| 7.1.17 Pengujian <i>Blackbox</i> Verifikasi Pemesanan..... | 149 |
| 7.1.18 Pengujian <i>Blackbox</i> Dashboard..... | 150 |
| 7.2 Pengujian <i>Usability</i> | 150 |
| Bab 8 Penutup..... | 153 |
| 8.1 Kesimpulan..... | 153 |
| 8.2 Saran..... | 153 |
| Daftar Pustaka..... | 154 |
| LAMPIRAN HASIL WAWANCARA..... | 157 |
| LAMPIRAN KUISIONER <i>SYSTEM USABILITY SCALE</i> | 158 |



DAFTAR TABEL

| | |
|---|-----|
| Tabel 2.1 Simbol <i>Use Case Diagram</i> | 28 |
| Tabel 2.2 Simbol <i>Activity Diagram</i> | 29 |
| Tabel 2.3 Simbol <i>Sequence Diagram</i> | 29 |
| Tabel 2.4 Simbol <i>Class Diagram</i> | 30 |
| Tabel 2.5 Contoh Kuisisioner <i>System Usability Scale</i> | 33 |
| Tabel 4.1 Identifikasi Aktor | 38 |
| Tabel 4.2 Kebutuhan Fungsional | 39 |
| Tabel 4.3 Kebutuhan Non Fungsional | 42 |
| Tabel 4.4 <i>Use Case Scenario Login User</i> | 43 |
| Tabel 4.5 <i>Use Case Scenario Login Admin</i> | 44 |
| Tabel 4.6 <i>Use Case Scenario Register</i> | 44 |
| Tabel 4.7 <i>Use Case Scenario Logout User</i> | 45 |
| Tabel 4.8 <i>Use case scenario logout Admin</i> | 46 |
| Tabel 4.9 <i>Use Case Scenario</i> Menampilkan Produk | 46 |
| Tabel 4.9 <i>Use Case Scenario</i> keranjang belanja | 47 |
| Tabel 4.10 <i>Use Case Scenario</i> menghapus produk | 48 |
| Tabel 4.11 <i>Use Case Scenario</i> menentukan jumlah pembelian | 48 |
| Tabel 4.12 <i>Use Case Scenario</i> Membuat Pemesanan | 49 |
| Tabel 4.13 <i>Use Case Scenario</i> Track Pemesanan | 50 |
| Tabel 4.14 <i>Use Case Scenario</i> Melihat Data Produk | 50 |
| Tabel 4.15 <i>Use Case Scenario</i> Menambah Data Produk | 51 |
| Tabel 4.16 <i>Use Case Scenario</i> Mengubah Data Produk | 52 |
| Tabel 4.17 <i>Use Case Scenario</i> Menghapus Data Produk | 53 |
| Tabel 4.18 <i>Use Case Scenario</i> Melihat Data Pemesanan | 53 |
| Tabel 4.19 <i>Use Case Scenario</i> Verifikasi Pemesanan | 54 |
| Tabel 4.20 <i>Use Case Scenario</i> Dashboard | 54 |
| Tabel 6.1 Spesifikasi Perangkat Keras | 106 |
| Tabel 6.2 Spesifikasi Perangkat Lunak | 106 |
| Tabel 6.3 Implementasi Tabel User | 107 |
| Tabel 6.4 Implementasi Tabel Admin | 107 |
| Tabel 6.5 Implementasi Tabel Produk | 108 |

| | |
|--|-----|
| Tabel 6.6 Implementasi Tabel Keranjang Belanja..... | 108 |
| Tabel 6.7 Implementasi Tabel Pemesanan..... | 109 |
| Tabel 6.8 Impementasi Server Key API Midtrans..... | 110 |
| Tabel 6.9 Implementasi Library Midtrans..... | 110 |
| Tabel 6.10 Implementasi Kode Sumber Login User..... | 112 |
| Tabel 6.11 Implementasi Kode Sumber Login Admin..... | 113 |
| Tabel 6.12 Implementasi Kode Sumber Register..... | 114 |
| Tabel 6.13 Implementasi Kode Sumber Logout User..... | 115 |
| Tabel 6.14 Implementasi Kode Sumber Logout Admin..... | 115 |
| Tabel 6.15 Implementasi Kode Sumber Menampilkan Produk..... | 115 |
| Tabel 6.16 Implementasi Kode Sumber Keranjang Belanja..... | 116 |
| Tabel 6.17 Implementasi Kode Sumber Menghapus produk..... | 117 |
| Tabel 6.18 Implementasi Kode Sumber Menentukan Jumlah pembelian..... | 117 |
| Tabel 6.19 Implementasi Kode Sumber Membuat Pemesanan..... | 118 |
| Tabel 6.20 Implementasi Kode Sumber Track Pemesanan..... | 120 |
| Tabel 6.21 Implementasi Kode Sumber Informasi Produk..... | 121 |
| Tabel 6.22 Implementasi Kode Sumber Menambah Data Produk..... | 121 |
| Tabel 6.23 Implementasi Kode Sumber Mengubah Data Produk..... | 123 |
| Tabel 6.24 Implementasi Kode Sumber Menghapus Data Produk..... | 124 |
| Tabel 6.25 Implementasi Kode Sumber Informasi Pemesanan..... | 125 |
| Tabel 6.26 Implementasi Kode Sumber Verifikasi Pemesanan..... | 125 |
| Tabel 6.27 Implementasi Dashboard..... | 126 |
| Tabel 7.1 Pengujian Blackbox Login User..... | 139 |
| Tabel 7.2 Pengujian Blackbox Login Admin..... | 139 |
| Tabel 7.3 Pengujian Blackbox Register..... | 140 |
| Tabel 7.4 Pengujian Blackbox Logout User..... | 141 |
| Tabel 7.5 Pengujian Blackbox Logout Admin..... | 141 |
| Tabel 7.6 Pengujian Blackbox Menampilkan Data Produk..... | 142 |
| Tabel 7.7 Pengujian Blackbox Keranjang Belanja..... | 143 |
| Tabel 7.8 Pengujian Blackbox Menghapus Produk..... | 143 |
| Tabel 7.9 Pengujian Blackbox Menentukan Jumlah Pembelian..... | 144 |
| Tabel 7.10 Pengujian Blackbox Membuat Pemesanan..... | 145 |

| | |
|--|-----|
| Tabel 7.11 Pengujian <i>Blackbox Track</i> Pemesanan..... | 146 |
| Tabel 6.12 Pengujian <i>Blackbox</i> Informasi Produk..... | 146 |
| Tabel 7.13 Pengujian <i>Blackbox</i> Menambah Data Produk..... | 147 |
| Tabel 7.14 Pengujian <i>Blackbox</i> Mengubah Data Produk..... | 147 |
| Tabel 7.15 Pengujian <i>Blackbox</i> Menghapus Data Produk..... | 148 |
| Tabel 7.16 Pengujian <i>Blackbox</i> Informasi Pemesanan..... | 149 |
| Tabel 7.17 Pengujian <i>Blackbox</i> Verifikasi Pemesanan..... | 149 |
| Tabel 7.18 Pengujian <i>Blackbox Dashboard</i> | 150 |
| Tabel 7.19 Tabel Daftar Responden..... | 150 |
| Tabel 7.20 Hasil Pengujian <i>System Usability Scale</i> | 151 |
| Tabel 7.21 Hasil Perhitungan <i>System Usability Scale</i> | 152 |



DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Logo Nue Pins..... | 26 |
| Gambar 2.2 Metode Waterfall..... | 32 |
| Gambar 2.3 Rumus Perhitungan Skor <i>System Usability Scale</i> | 33 |
| Gambar 3.1 Diagram Metodologi..... | 34 |
| Gambar 4.1 Penomoran Kebutuhan Sistem..... | 39 |
| Gambar 4.2 <i>Use Case Diagram</i> | 43 |
| Gambar 4.3 <i>Activity Diagram Login User</i> | 56 |
| Gambar 4.4 <i>Activity Diagram Login Admin</i> | 57 |
| Gambar 4.5 <i>Activity Diagram Register</i> | 58 |
| Gambar 4.6 <i>Activity Diagram Logout User</i> | 59 |
| Gambar 4.7 <i>Activity Diagram Logout Admin</i> | 60 |
| Gambar 4.8 <i>Activity Diagram Menampilkan Produk</i> | 61 |
| Gambar 4.9 <i>Activity Diagram Keranjang Belanja</i> | 62 |
| Gambar 4.10 <i>Activity Diagram Menghapus Produk</i> | 63 |
| Gambar 4.11 <i>Activity Diagram Menentukan Jumlah Pembelian</i> | 64 |
| Gambar 4.12 <i>Activity Diagram Membuat Pemesanan</i> | 65 |
| Gambar 4.13 <i>Activity Diagram Track Pemesanan</i> | 66 |
| Gambar 4.14 <i>Activity Diagram Melihat Data Produk</i> | 67 |
| Gambar 4.15 <i>Activity Diagram Menambah Data Produk</i> | 68 |
| Gambar 4.16 <i>Activity Diagram Mengubah Data Produk</i> | 68 |
| Gambar 4.17 <i>Activity Diagram Menghapus Data Produk</i> | 69 |
| Gambar 4.18 <i>Activity Diagram Melihat Data Pemesanan</i> | 70 |
| Gambar 4.19 <i>Activity Diagram Verifikasi Pemesanan</i> | 70 |
| Gambar 4.20 <i>Activity Diagram Dashboard</i> | 71 |
| Gambar 5.1 Arsitektur Sistem..... | 72 |
| Gambar 5.2 <i>Sequence Diagram Login User</i> | 73 |
| Gambar 5.3 <i>Sequence Diagram Login Admin</i> | 74 |
| Gambar 5.4 <i>Sequence Diagram Register</i> | 75 |
| Gambar 5.5 <i>Sequence Diagram Logout User</i> | 76 |
| Gambar 5.6 <i>Sequence Diagram Logout Admin</i> | 76 |
| Gambar 5.7 <i>Sequence Diagram Menampilkan Produk</i> | 77 |

| | |
|--|-----|
| Gambar 5.8 <i>Sequence</i> Diagram Keranjang Belanja..... | 78 |
| Gambar 5.9 <i>Sequence</i> Diagram Menghapus Produk | 78 |
| Gambar 5.10 <i>Sequence</i> Diagram Menentukan Jumlah Pembelian..... | 79 |
| Gambar 5.11 <i>Sequence</i> Diagram Membuat Pemesanan | 80 |
| Gambar 5.12 <i>Sequence</i> Diagram <i>Track</i> Pemesanan | 81 |
| Gambar 5.13 <i>Sequence</i> Diagram Melihat Data Produk..... | 82 |
| Gambar 5.14 <i>Sequence</i> Diagram Menambah Data Produk..... | 82 |
| Gambar 5.15 <i>Sequence</i> Diagram Mengubah Data Produk | 83 |
| Gambar 5.16 <i>Sequence</i> Diagram Menghapus Data Produk..... | 84 |
| Gambar 5.17 <i>Sequence</i> Diagram Melihat Data Pemesanan | 85 |
| Gambar 5.18 <i>Sequence</i> Diagram Verifikasi Pemesanan | 85 |
| Gambar 5.19 <i>Sequence</i> Diagram <i>Dashboard</i> | 86 |
| Gambar 5.20 <i>Class</i> Diagram Sistem Informasi <i>E-Commerce</i> Nue Pins | 88 |
| Gambar 5.21 <i>Physical</i> Data Model..... | 89 |
| Gambar 5.22 Perancangan Antarmuka <i>Login User</i> | 90 |
| Gambar 5.23 Perancangan Antarmuka <i>Login Admin</i> | 91 |
| Gambar 5.24 Perancangan Antarmuka Register..... | 92 |
| Gambar 5.25 Perancangan Antarmuka <i>Logout User</i> | 93 |
| Gambar 5.26 Perancangan Antarmuka <i>Logout Admin</i> | 94 |
| Gambar 5.27 Perancangan Antarmuka Menampilkan Produk (1)..... | 95 |
| Gambar 5.28 Perancangan Antarmuka Menampilkan Produk (2)..... | 96 |
| Gambar 5.29 Perancangan Antarmuka Keranjang Belanja (1) | 97 |
| Gambar 5.30 Perancangan Antarmuka Keranjang Belanja (2) | 98 |
| Gambar 5.31 Perancangan Antarmuka Menghapus Produk..... | 98 |
| Gambar 5.32 Perancangan Antarmuka Menentukan Jumlah Pembelian | 99 |
| Gambar 5.33 Perancangan Antarmuka Membuat Pemesanan (1)..... | 100 |
| Gambar 5.34 Perancangan Antarmuka Membuat Pemesanan (2)..... | 100 |
| Gambar 5.35 Perancangan Antarmuka <i>Track</i> Pemesanan | 101 |
| Gambar 5.36 Perancangan Antarmuka Melihat Data Produk..... | 102 |
| Gambar 5.37 Perancangan Antarmuka Menambah Data Produk..... | 102 |
| Gambar 5.38 Perancangan Antarmuka Mengubah Data Produk..... | 103 |
| Gambar 5.39 Perancangan Antarmuka Menghapus Data Produk..... | 103 |

| | |
|---|-----|
| Gambar 5.40 Perancangan Antarmuka Melihat Data Pemesanan..... | 104 |
| Gambar 5.41 Perancangan Antarmuka Verifikasi Pemesanan..... | 104 |
| Gambar 5.42 Perancangan Antarmuka <i>Dashboard</i> | 105 |
| Gambar 6.1 <i>Result Data JSON API Midtrans</i> | 111 |
| Gambar 6.2 Antarmuka <i>Login User</i> | 128 |
| Gambar 6.3 Antarmuka <i>Login Admin</i> | 129 |
| Gambar 6.4 Antarmuka <i>Register</i> | 129 |
| Gambar 6.5 Antarmuka <i>Logout User</i> | 130 |
| Gambar 6.6 Antarmuka <i>Logout Admin</i> | 130 |
| Gambar 6.7 Antarmuka Menampilkan Produk..... | 131 |
| Gambar 6.8 Antarmuka Keranjang Belanja..... | 132 |
| Gambar 6.9 Antarmuka Menghapus Produk..... | 132 |
| Gambar 6.10 Antarmuka Menentukan Jumlah pembelian..... | 133 |
| Gambar 6.11 Antarmuka Membuat Pemesanan..... | 134 |
| Gambar 6.12 Antarmuka Track Pemesanan..... | 135 |
| Gambar 6.13 Antarmuka Informasi Produk..... | 135 |
| Gambar 6.14 Antarmuka Menambah Data Produk..... | 136 |
| Gambar 6.15 Antarmuka Mengubah Data Produk..... | 136 |
| Gambar 6.16 Menghapus Data Produk..... | 137 |
| Gambar 6.17 Antarmuka Informasi Pemesanan..... | 137 |
| Gambar 6.18 Antarmuka Verifikasi Pemesanan..... | 138 |
| Gambar 6.19 Antarmuka <i>Dashboard</i> | 138 |

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Toko merupakan tempat yang digunakan sebagai lahan untuk menawarkan barang dan jasa. Didalam toko terdapat aktifitas memajang, menyimpan dan menjual produk yang telah diproduksi atau produk yang telah didistribusikan pada toko tersebut. Terdapat konsumen sebagai pembeli produk dan pengusaha toko sebagai penjual produk. Fungsi toko secara garis besar adalah wadah untuk memasarkan produk dengan memperkenalkan harga dan kualitas dari produk yang disajikan pada toko itu sendiri. Toko juga dapat diperkenalkan secara *online* yang biasanya terdapat di forum jual beli *online*, *marketplace*, dan *website* dari toko itu sendiri. Toko *online* merupakan wadah untuk menawarkan barang atau jasa melalui internet, sehingga dapat menampilkan informasi ataupun melakukan pembelian dari barang atau jasa yang ditawarkan bagi pengunjung toko *online* (Sari, 2015). Nue Pins merupakan brand yang menyediakan produk berupa aksesoris wanita. Nue Pins memiliki toko yang dibuka secara *offline* pada bilangan department store di Jakarta, seperti di Pondok Indah Mall, Plaza Senayan, dan Central Park dan online shop pada social media Instagram. Toko Nue Pins menjual aksesoris berupa cincin, anting, bros, penjepit rambut dan kancing. Untuk melakukan pembelian produk dari Nue Pins, dapat mengunjungi toko *offline* Nue Pins secara langsung atau melakukan pembelian melalui Social Media Instagram. Pembelian produk Nue Pins melalui Social Media Instagram dilakukan dengan cara mengirimkan pesan kepada kontak yang tertera pada akun Instagram Nue Pins lalu melakukan pengisian data yang diperlukan seperti nama pembeli, nama produk yang akan dibeli, jumlah yang akan dibeli, dan alamat pembeli.

Berdasarkan hasil wawancara dengan Ekadriani Fitria selaku pemilik dari *Brand* Nue Pins, permasalahan yang didapat adalah *brand* Nue Pins yang sedang tidak beroperasi dalam waktu sementara akan melakukan perubahan strategi pemasaran dan penjualan produk dan memerlukan sebuah *website* sebagai wadah untuk melakukan pemasaran produk dan penjualan produk. Permasalahan berikutnya yang didapat adalah pada saat proses pembelian melalui *social media instagram*, Berdasarkan pengalaman dari pemilik Nue Pins, proses penjualan pada *social media* masih dikatakan kurang efektif, dikarenakan pembeli harus menanyakan langsung terkait produk yang masih tersedia dan produk yang sudah tidak tersedia atau habis terjual. Permasalahan berikutnya adalah pada saat pencatatan penjualan. Pencatatan penjualan dilakukan secara manual dengan cara mencatat menggunakan notes pada smartphone lalu dipindahkan kedalam microsoft excel.

Pemasaran produk Nue Pins diharapkan dapat dilaksanakan melalui *website* yang dapat menampilkan informasi dari produk Nue Pins secara langsung. Pencatatan penjualan dapat dilakukan secara mudah dengan tidak diharuskan memindahkan data dari *notes* pada *smartphone* ke *Microsoft Excel*. Saat ini, pemasaran produk Nue Pins hanya dilakukan melalui *social media Instagram*.

Proses pembelian produk Nue Pins melalui *social media Instagram*, masih dilakukan dengan cara menghubungi langsung kontak yang tertera pada akun *Instagram* Nue Pins. Stock produk yang tersedia, hanya diinformasikan jika pembeli menanyakan langsung ke akun *social media Instagram* Nue Pins.

Berdasarkan pada masalah diatas, maka dibangun Sistem Informasi Toko *Online* berbasis *Web* dengan memanfaatkan teknologi *Payment Gateway* atau media pembayaran, menggunakan *API Midtrans*, yang dapat melakukan pembelian secara *online*, dan menampilkan informasi produk dari Nue Pins. Tujuan pemanfaatan *API Midtrans* adalah sebagai media pembayaran dalam proses pembayaran, sehingga pembeli dapat melakukan proses pembayaran melalui *bank transfer* dengan menggunakan kode pembayaran yang akan diberikan oleh sistem. Pengembangan sistem informasi toko *online* berbasis web, bertujuan untuk memasarkan dan menjual produk dari Brand Nue Pins secara *online*. Sehingga produk Nue Pins dapat dipromosikan kepada calon pembeli aksesoris wanita yang belum mengetahui Brand Nue Pins dan tidak mempersulit pembeli dalam melakukan proses pembayaran dan mendapatkan informasi stok barang yang tersedia ataupun tidak. Keunggulan pengembangan sistem informasi pada platform website adalah pengaksesan website dapat dilakukan melalui web browser personal computer dan smartphone tanpa harus melakukan download pada application store terlebih dahulu. Penggunaan *API Midtrans* bertujuan sebagai media pembayaran dan dikarenakan *Midtrans* telah menyediakan 24 metode pembayaran yang dapat dipakai sehingga pemilik dari Nue Pins dapat memilih metode pembayaran yang akan digunakan didalam sistem.

Penelitian akan dilaksanakan dengan jenis penelitian implementatif - pengembangan. Pengambilan data analisis kebutuhan akan dilakukan melalui wawancara kepada pemilik Brand Nue Pins. Pada pemodelan perangkat lunak, akan digunakan metode *waterfall* dengan tujuan agar pengerjaan perangkat lunak tidak tumpang tindih dengan proses pengerjaan sebelumnya. Penggunaan metode *waterfall* dikarenakan pengambilan kebutuhan perangkat lunak telah diberikan secara langsung oleh pemilik brand Nue Pins, sehingga tidak akan diperlukan penambahan analisis kebutuhan perangkat lunak, saat proses pembangunan sistem informasi toko *online* sedang berlangsung. Pengambilan data kepuasan pengguna akan dilakukan secara kuantitatif kepada calon sistem informasi toko online Nue Pins berbasis web.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah di paparkan di atas, dapat di peroleh beberapa rumusan masalah pada penelitian ini, antara lain :

1. Bagaimana hasil dari Analisis Kebutuhan Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran?

2. Bagaimana hasil dari Perancangan Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran?
3. Bagaimana hasil dari Implementasi Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran?
4. Bagaimana hasil dari Pengujian Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran?

1.3 Tujuan

Terdapat beberapa tujuan berdasarkan rumusan masalah diatas, antara lain :

1. Mendapatkan hasil dari Analisis Kebutuhan Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran.
2. Mendapatkan hasil Perancangan dari Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran
3. Mendapatkan hasil Implementasi dari Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran
4. Mendapatkan hasil pengujian dari Pengembangan Sistem Informasi Toko Online Nue Pins Berbasis Web Dengan Memanfaatkan *API Midtrans* Sebagai Media Pembayaran

1.4 Manfaat

Dengan adanya penelitian ini, di harapkan adanya manfaat bagi pihak – pihak yang terkait, diantaranya :

1.4.1 Bagi Pemilik Nue Pins

1. Membantu pemilik *brand* Nue Pins dalam mempromosikan produk sekaligus nama *brand* Nue Pins.
2. Membantu pemilik *brand* Nue Pins dalam memberikan informasi dari produk dari *brand* Nue Pins.
3. Mempermudah rekaptulasi hasil penjualan produk Nue Pins.

1.4.2 Calon Pelanggan Nue Pins

1. Mempermudah calon pelanggan untuk mendapatkan informasi dari masing – masing produk *brand* Nue Pins.

2. Mempermudah calon pelanggan untuk melakukan transaksi secara online.

1.5 Batasan Masalah

Batasan Masalah yang dapat di paparkan berdasarkan permasalahan yang ada dalam penelitian ini, antara lain :

1. Sistem informasi yang dikembangkan mengambil studi kasus pada proses bisnis penjualan produk dari *brand* Nue Pins secara *online*.
2. Sistem informasi yang di kembangkan merupakan sistem informasi yang berfokus pada *Platform Website*.
3. Sistem yang dikembangkan menggunakan *framework Code Igniter* yang berbahasa pemrograman *PHP* dan menggunakan basis data *MySQL*.
4. Sistem yang dikembangkan memanfaatkan *API Midtrans* sebagai media pembayaran.

1.6 Sistematika Pembahasan

BAB I PENDAHULUAN

Pada bab ini dibahas mengenai ide dari dasar permasalahan yang diangkat untuk dilaksanakannya penelitian sebagai latar belakang. Bab ini juga membahas mengenai perumusan masalah, tujuan dari pelaksanaan penelitian, manfaat bagi beberapa pihak yang terkait, Batasan masalah, dan sistematika penulisan.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini, membahas definisi dari teori-teori yang berhubungan dengan penelitian dan diperoleh dari berbagai sumber.

BAB III METODOLOGI PENELITIAN

Pada bab ini, membahas metode yang akan di gunakan dalam penelitian, yang terdiri dari studi literatur, Analisa Kebutuhan dan Perancangan, Implementasi Sistem, Pengujian Sistem, Kesimpulan dan Saran.

BAB IV ANALISIS KEBUTUHAN

Pada bab ini, membahas pendefinisian dari aspek kebutuhan pengguna yang hasilnya akan digunakan dalam proses perancangan.

BAB V PERANCANGAN SISTEM

Pada bab ini, membahas penjelasan dari perancangan sistem yang akan dibangun berdasarkan kebutuhan pengguna.

BAB VI IMPLEMENTASI SISTEM

Pada bab ini, membahas hasil dari implementasi sistem berdasarkan perancangan yang telah dilakukan sebelumnya

BAB VII PENGUJIAN SISTEM

Pada bab ini, membahas hasil dari proses pengujian sistem dengan menggunakan beberapa metode penelitian

BAB VIII Pada bab ini, membahas hasil kesimpulan dari penelitian dan saran mengenai kekurangan yang dapat diperbaiki untuk penelitian selanjutnya



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian mengenai toko online sebelumnya pernah dilakukan oleh Sri Handayani, Anofrizen, dan M.Jazman. Penelitian yang dilakukan oleh Sri Handayani, Anofrizen, dan M.Jazman bertujuan untuk membangun toko *online* Adira Finance yang telah bergerak dalam pembiayaan dana pada *show room* penjualan kendaraan otomotif bekas. Pengembangan sistem informasi *e-commerce* untuk jaringan penjualan sepeda motor bekas kabupaten kampar menjadi solusi, dikarnakan dapat memberikan promosi pada penjualan motor bekas secara efektif (Handayani et al., 2016).

Penelitian mengenai toko online telah dilakukan juga oleh Andi Ridho Rahman, Beny, dan Erick Fernando. Penelitian yang dilakukan oleh Andi Ridho Rahman, Beny, dan Erick Fernando bertujuan untuk merancang sebuah sistem informasi toko *online* untuk menjadikan prose penjualan secara *online*. Perancangan *e-commerce* berbasis *website* menjadi solusi pada toko batik Mirabella, dikarnakan membantu promosi toko Mirabella dan mempermudah pembeli untuk memesan produk dari toko batik Mirabella tanpa harus berkunjung langsung ke toko (Rachman Andi et al., 2017).

Penelitian ketiga mengenai toko online telah dilakukan oleh Diki Susandi dan Sukisno. Penelitian yang dilakukan Diki Susandi dan Sukisno, bertujuan untuk membantu Dlapak *Street wear* dalam memperluas cakupan promosi, membantu pelanggan yang berasal dari luar kota agar dapat berbelanja pada *Distro Dlapak Street Wear* dan memaksimalkan pemasaran produk. Sistem penjualan berbasis *e-commerce* menjadi solusi, dikarnakan dapat mewujudkan promosi yang maksimal, meningkatkan penjualan produk, dan memberikan kemudahan pembeli dari *Distro Dlapak Street Wear* (Susandi & Sukisno, 2017).

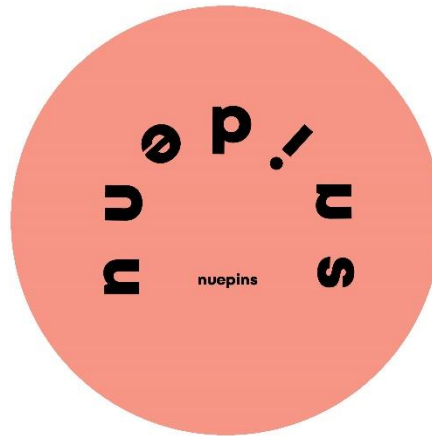
2.2 Toko Online

Menurut (Sari, 2015) toko *online* merupakan wadah untuk menawarkan barang atau jasa melalui internet, sehingga dapat menampilkan informasi ataupun melakukan pembelian dari barang atau jasa yang ditawarkan bagi pengunjung toko *online*. Abdul Gaffar Khan, mendefinisikan toko *online* atau *e-commerce* sebagai : “Menjual dan membeli barang melalui jaringan internet, dan dapat melakukan perbandingan harga terhadap toko lain melalui jaringan internet” (Khan, 2016). Dari definisi tersebut, terdapat 2 poin utama dalam toko *online* yaitu, pertama adanya pembelian dan penjualan barang atau jasa. Kedua, dalam sebagian orang, *Internet* dapat digunakan untuk mendapatkan informasi harga barang, sebelum melakukan proses pembelian secara *online* atau secara langsung di toko.

2.3 Nue Pins

Nue Pins merupakan merek atau *brand* yang bergerak dalam bidang aksesoris wanita seperti cincin, anting, kalung, *hairpins*, *strap* pada kacamata dan masker.

Nue Pins membuka toko secara *offline* pada bilangan *departement store* di Jakarta, diantaranya di Pondok Indah *Mall*, Plaza Senayan, *Mall of Indonesia*, Grand Indonesia dan toko *online* pada *social media Instagram*. Merek atau *brand* Nue Pins didirikan pada tahun 2019 oleh Saudari Ekadriani Fitria. Gambar 2.1 akan menunjukkan logo dari merek atau *brand* Nue Pins.



Gambar 2.1 Logo Nue Pins

Sumber : (Dokumen *Company Profile* Nue Pins)

2.4 Midtrans

Midtrans merupakan salah satu *start up* yang memberikan layanan *Payment Gateway* untuk memberikan fasilitas kebutuhan dalam masalah pembayaran secara *online* dan *offline*. *Midtrans* menyediakan berbagai macam pembayaran secara *online* diantaranya dengan menggunakan *e-money*, melalui *bank transfer*, pembayaran dengan *kartu kredit*. Sedangkan secara *offline*, *Midtrans* memfasilitasi pembayaran di *counter Minimarket* yang terdaftar. *Midtrans* dahulunya merupakan *start up* yang bernama *veritrans* yang didirikan pada tahun 2012, dan *direbranding* menjadi *Midtrans* pada tahun 2016 (Midtrans, 2016). Dengan pelayanan yang diberikan diharapkan para pelaku bisnis dapat terbantu dalam menjalankan bisnisnya dan dapat lebih optimal serta dapat meningkatkan penjualannya. Disisi lain, terdapat berbagai macam keuntungan yang diberikan oleh *Midtrans* bagi pelanggan yang gemar berbelanja online dengan menawarkan macam-macam pilihan metode pembayaran, seperti *kartu debit*, *e-wallet*, *transfer* melalui *bank*, *kartu kredit*, dan lain-lain.

2.5 Application Programming Interface (API)

Application Programming Interface (API) merupakan teknologi antarmuka yang dapat menghubungkan perangkat lunak yang *berbeda platform*, sehingga dapat melakukan pertukaran data (Sunardi et al., 2019). *API* bekerja diantara *client* dan *server* dimana *client* akan meminta request ke *server* melalui *API* dan *server* akan memberikan respon melalui *API*. Contoh penggunaan *API* adalah pada saat membuat fitur registrasi dengan menggunakan *Google API*. Ketika pengguna

aplikasi melakukan registrasi dengan akun *Google*, data akan langsung didapatkan berupa nama pengguna atau *email*, sehingga pengembang aplikasi tidak memerlukan waktu untuk membuat fitur registrasi, cukup dengan memanggil *Google API*(Eril, 2020).

2.6 Javascript Object Notation (JSON)

Javascript Object Notation adalah desain format untuk pertukaran data yang dapat dibaca mudah oleh manusia dan mempunyai pasangan atribut dan data yang dipisahkan dengan tanda kurung (Joshi & Dr.Chetan R, 2015). Sama seperti XML, menurut penelitian tersebut *JSON* mempunyai kelengkapan fitur dalam pengiriman data antara *server* dan aplikasi, seperti *JSON* dapat menggunakan *array* untuk menyimpan banyak data dalam 1 objek.

2.7 Code Igniter

Code Igniter merupakan *framework* open source yang dibuat dengan menggunakan bahasa PHP, dan dapat digunakan untuk pengembangan aplikasi berbasis *website* dengan menggunakan konsep *Model, View, dan Controller* (Afuan Lasmedi, 2010). Menurut Afuan Lasmedi pada jurnal yang berjudul Pemanfaatan *Framework Code Igniter* dalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed, *Models* dalam *Framework Code Igniter* dapat diartikan sebagai kelas PHP yang bekerja dengan informasi yang ada dalam basis data seperti operasi *Create, Read, Update, dan Delete*. *Views* pada *Code Igniter*, dapat diartikan sebagai tampilan dari *interface* dengan menggunakan kode HTML yang datanya berisikan dari basis data. Sementara *Controllers* pada *Code Igniter*, dapat diartikan sebagai logika bisnis yang menginstruksikan *models* untuk mengambil data dari basis data dan kemudian akan ditampilkan pada *views*.

2.8 MySQL

Menurut (Febio, 2011) *Database* secara sederhana dapat kita sebut sebagai gudang data. Secara teori, *database* adalah kumpulan data atau informasi yang kompleks, data-data tersebut disusun menjadi beberapa kelompok dengan tipe data yang sejenis disebut *table/entity*, di mana setiap datanya dapat saling berhubungan satu sama lain atau dapat berdiri sendiri, sehingga mudah diakses. *MySQL* adalah *Database Management System (DBMS)* yang memiliki fungsi sebagai *relational database system (RDBMS)* yang bekerja dengan arsitektur *client server*(Turban, 2017). *Structured Query Language (SQL)* merupakan bahasa standar yang digunakan *MySQL*. *Relational Database System* merupakan hubungan antar tabel pada *database*.

2.9 Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan bahasa pemodelan yang dibuat untuk memvisualisasikan perancangan perangkat lunak berdasarkan grafik dan gambar (Heriyanto, 2018). Pemodelan sebuah perangkat lunak merupakan bagian


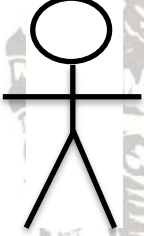

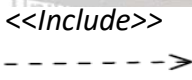
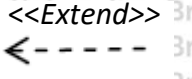
penting yang menjadi dasar sukses dari suatu perangkat lunak agar perangkat lunak berjalan sesuai dengan kebutuhan pengguna dan kebutuhan sistem itu sendiri (Wira et al., 2019). Adapun jenis-jenis diagram UML dalam memodelkan perangkat lunak menurut (Putra, 2018), yaitu *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*, *Class Diagram*.

2.9.1 Use Case Diagram

Use Case Diagram mempresentasikan sebuah interaksi antara aktor, *use case* dan relasinya (Haviluddin, 2011). *Use Case Diagram* merupakan pemodelan interaksi yang dilakukan aktor kepada sistem berdasarkan kebutuhan fungsional sistem. Adapun simbol-simbol yang digunakan dalam *Use Case Diagram* yang akan dijelaskan pada Tabel 2.1.

Tabel 2.1 Simbol Use Case Diagram

Sumber : (Urva & Siregar, 2015)







| Gambar | Keterangan |
|---|---|
|  | <i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang berinteraksi dengan aktor. |
|  | Aktor menggambarkan pengguna atau sistem yang akan menjalankan fungsi yang ada pada sistem. |
|  | Asosiasi merupakan simbol hubungan antara aktor dan <i>use case</i> untuk mengetahui fungsi yang dikerjakan oleh aktor. |
|  | <i>Include</i> menghubungkan <i>use case</i> yang membutuhkan fungsi dari <i>use case</i> lainnya. |
|  | <i>Extend</i> merupakan kondisi jika salah satu <i>use case</i> telah dijalankan. |

2.9.2 Activity Diagram

Activity Diagram adalah rancangan aliran kegiatan dalam sebuah sistem untuk memperlihatkan urutan aktivitas proses pada sistem, dan Menggambarkan proses bisnis dan urutan aktivitas berdasarkan Use Case Diagram (Rizky, 2019). Adapun kode simbol yang digunakan dalam Activity Diagram yang akan dijelaskan pada Tabel 2.2

Tabel 2.2 Simbol Activity Diagram

Sumber : (Urva & Siregar, 2015)

| Gambar | Keterangan |
|---|---|
|  | <i>Start point</i> merupakan awal dari aktivitas pada aliran kerja. |
|  | <i>End point</i> merupakan akhir dari aktivitas pada aliran kerja. |
|  | <i>Activites</i> merupakan suatu proses atau kegiatan pada aliran kerja. |
|  | <i>Swimlane</i> pembagian <i>activity diagram</i> untuk menunjukkan interaksi pada masing masing kelas. |
|  | <i>Decision point</i> merupakan pilihan untuk pengambilan suatu kondisi, <i>true false</i> . |
|  | <i>State Transition</i> merupakan penghubung antar <i>Activities</i> untuk mengetahui aktivitas berikutnya. |

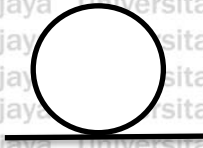



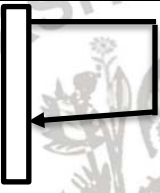

2.9.3 Sequence Diagram

Sequence Diagram adalah diagram yang menspesifikasi fungsi dari Use Case Diagram dan Activity Diagram yang telah dijalankan (Grgec & Mužar, 2007). Sequence diagram dapat digunakan untuk memvisualisasikan rangkaian pesan yang dikirim antara objek dan juga interaksi antara objek. Pesan yang dikirim dapat memberikan respon atau *return*. Adapun simbol-simbol yang digunakan dalam Sequence Diagram yang akan dijelaskan pada Tabel 2.3

Tabel 2.3 Simbol Sequence Diagram

Sumber : (Urva & Siregar, 2015)

| Gambar | Keterangan |
|--------|------------|
|--------|------------|

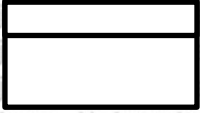
| | |
|---|--|
|  | <i>Entity Class</i> merupakan objek yang didefinisikan sebagai basis data atau kelas yang berhubungan langsung untuk mengoperasikan data ke basis data |
|  | <i>Boundary Class</i> merupakan objek yang dijadikan sebagai <i>interface</i> antara perilaku aktor dengan sistem |
|  | <i>Control Class</i> merupakan objek logika aplikasi. |
|  | <i>Message</i> merupakan pengirim pesan antar kelas. |
|  | <i>Recursive</i> merupakan pengiriman pesan yang dikirim untuk kelas itu sendiri. |
|  | <i>Lifetime</i> merupakan tempat awal dan akhir pesan dikeluarkan. |



2.9.4 Class Diagram

Class Diagram adalah struktur dari hubungan antara beberapa kelas dan mendefinisikan atribut, operasi dan metode pada masing-masing kelas (Aprianti & Maliha, 2016). Adapun kode simbol dalam *Class Diagram* yang akan dijelaskan pada Tabel 2.4

Tabel 2.4 Simbol *Class Diagram*

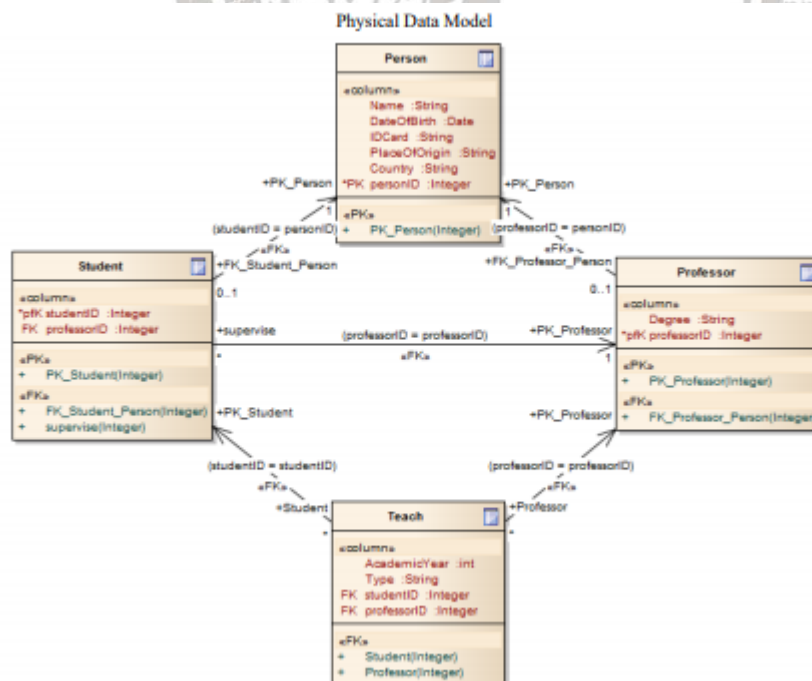
Sumber : (Hanindia & Swari, 2019)

| Gambar | Keterangan |
|---|---|
|  | <i>Class</i> merupakan yang terdiri dari atribut yang didefinisikan dalam kelas |

| | | |
|--|---|---|
| | | dan operasi yang terdapat dalam kelas. |
| |  | <i>Generalization</i> merupakan hubungan antar kelas <i>descendent</i> dan kelas <i>ancestor</i> untuk menurunkan suatu objek yang ada pada kelas <i>ancestor</i> . |
| |  | <i>Realization</i> adalah operasi yang dijalankan oleh suatu objek |

2.10 Physical Data Model

Physical data model (PDM) sebagai model perancangan basis data yang memvisualisasikan struktur data berdasarkan kelas atau tabel, untuk diimplementasikan pada basis data. Karen itu, setiap entitas yang berhubungan akan direpresentasikan sebagai tabel, dan atribut dari tabel akan direpresentasikan sebagai kolom pada basis data (Ribeiro et al., 2015). Tidak seperti pemodelan lainnya, *PDM* cenderung menspesifikasikan skema dan aspek – aspek yang dibutuhkan pada masing – masing tabel dari basis data, seperti tipe data pada setiap kolom beserta panjang dari tipe data tersebut, dan hubungan antara tabel direpresentasikan dengan *foreign key*. Gambar 2.2 akan menjelaskan contoh dari bentuk *physical data model*.

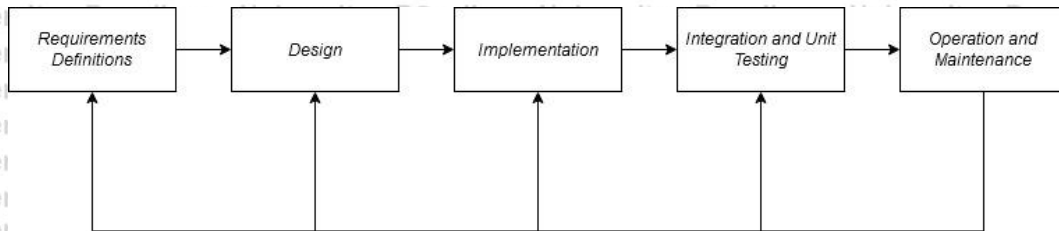


Gambar 2.2 Physical Data Model

Sumber : (Ribeiro et al., 2015)

2.11 Waterfall

Rick Sherman mendefinisikan metode *waterfall* sebagai :” Metodologi yang menggunakan pendekatan sekuensial atau linier dalam kebutuhan pengembangan perangkat lunak, ruang lingkup pengerjaan perangkat lunak dipecah menjadi urutan fase, dan dikelompokkan berdasarkan tingkatan” (Sherman, 2015). Dalam definisi tersebut terdapat poin penting yang dapat diambil, yaitu “fase”. Dalam metode *waterfall*, ruang lingkup pengembangan sistem informasi akan dikerjakan secara bertahap dan tidak dapat diulang kembali ke tahap sebelumnya atau mengerjakan terlebih dahulu tahap setelahnya. Rick Sherman juga mendefinisikan tahap – tahap yang dikerjakan pada metode *waterfall* adalah sebagai berikut.



Gambar 2.3 Metode Waterfall

Sumber : (Sherman, 2015)

2.11 Black Box Testing

Black Box Testing merupakan teknik pengujian perangkat lunak untuk melakukan pengecekan *output* yang dihasilkan setelah melakukan *input* berdasarkan masing – masing kebutuhan fungsional (Jaya, 2018). Metode *Black Box Testing* berfokus pada jalannya fungsi dan pemeriksaan *input* dan *output* pada perangkat lunak. Menurut (Mustaqbal et al., 2015), tujuan pengujian *Black Box* adalah untuk menemukan kesalahan dalam proses kerja dari masing – masing kebutuhan fungsional yang ada dalam sistem.

2.12 Usability Testing

Usability Testing merupakan metode pengujian yang digunakan untuk menilai kemudahan, dan kepuasan pengguna dengan menggunakan sistem informasi yang di uji (Henriyadi & Mulyati, 2016). Pengujian *Usability* bertujuan untuk melakukan penilaian terhadap aplikasi sebagai pengukur tingkat kepuasan pengguna. Menurut (Ika et al., 2015) Kuisisioner penilaian *usability* dibagi menjadi 4, yaitu *System Usability Scale*, *Quiosionnaire for User Interface Satisfaction*, *Software Usability Measurement Inventory* dan *Post-Study Usability Question*.

Tabel 2.5 melampirkan contoh dari bentuk pertanyaan dalam *system usability scale*. Responden akan melakukan penilaian mulai dari Sangat Tidak Setuju, Tidak Setuju, Netral, Setuju dan Sangat Setuju berdasarkan 10 pertanyaan *System Usability Scale*. Setiap pertanyaan memiliki skor yang berkisar antara 0 hingga 4. Pertanyaan yang bernomor ganjil, skor akan dikurangi 1. Sedangkan

pertanyaan yang bernomor angka genap, skor kontribusinya adalah 5 dikurangi skor. Masing – masing skor dari setiap responden akan dikalikan dengan 2.5 untuk mendapatkan nilai akhir dari masing – masing responden. Nilai keseluruhan *System Usability Scale* berkisar dari 0 hingga 100. Gambar 2.4 akan menunjukkan rumus perhitungan skor *System Usability Testing*.

Tabel 2.5 Contoh Kuisiner *System Usability Scale*

Sumber : (Ika et al., 2015)

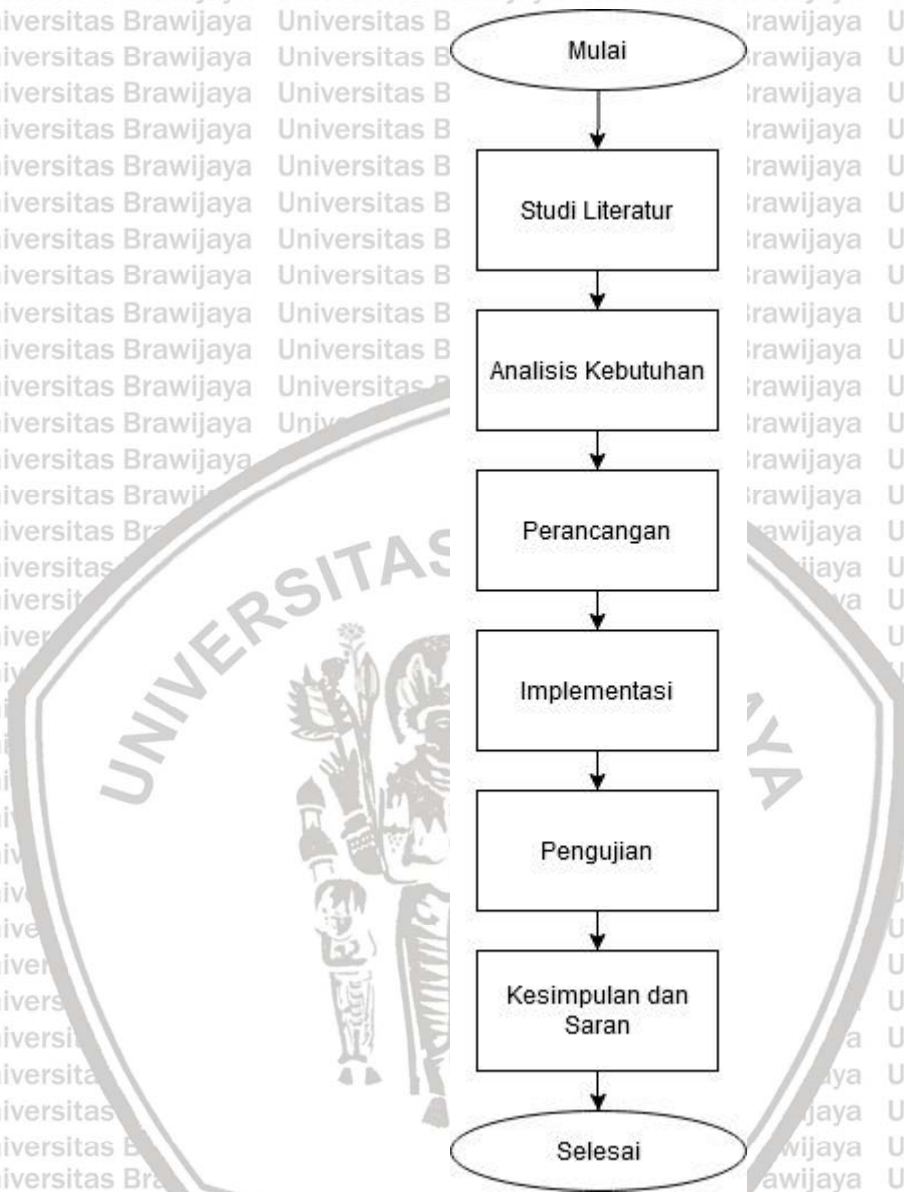
| Kode | Pertanyaan |
|------|--|
| R1 | Saya akan sering menggunakan aplikasi ini |
| R2 | Saya menilai aplikasi ini terlalu kompleks sehingga membuat banyak hal yang tidak perlu |
| R3 | Saya menilai aplikasi ini mudah digunakan |
| R4 | Saya membutuhkan bantuan teknis untuk menggunakan aplikasi ini |
| R5 | Saya menilai fitur yang disediakan pada aplikasi ini dirancang dan disiapkan dengan baik |
| R6 | Saya menilai aplikasi ini tidak konsisten dengan tujuan utama |
| R7 | Saya merasa kebanyakan orang akan tidak mudah menggunakan aplikasi ini dengan cepat |
| R8 | Saya menilai aplikasi ini sangat rumit untuk dipakai |
| R9 | Saya merasa sangat percaya diri menggunakan aplikasi ini |
| R10 | Saya merasa tampilan aplikasi ini bagus |

$$\begin{aligned}
 \text{Skor SUS} = & ((R1 - 1) + (5 - R2) + (R3 - 1) + \\
 & (5 - R4) + (R5 - 1) + (5 - R6) + \\
 & (R7 - 1) + (5 - R8) + (R9 - 1) + \\
 & (5 - R10)) * 2.5)
 \end{aligned}$$

Gambar 2.4 Rumus Perhitungan Skor *System Usability Scale*

Sumber : (Ika et al., 2015)

BAB 3 METODOLOGI



Gambar 3.1 Diagram Metodologi

3.1 Studi Literatur

Pada tahap studi literatur akan dilakukan pencarian dari berbagai sumber baik berupa buku, arsip, majalah, jurnal, dan artikel yang terkait dengan permasalahan dan teori - teori yang mendukung penelitian. Literatur yang dijadikan referensi dalam penelitian ini, merupakan literatur yang berkaitan dengan sistem informasi toko *online*. Pencarian literatur - literatur tersebut bertujuan untuk mengumpulkan informasi dan memahami teori - teori dari yang relevan dengan penelitian yang dilakukan.

3.2 Analisis Kebutuhan Sistem

Pada tahap analisis kebutuhan, akan dilakukan pengumpulan data dengan cara wawancara dari Narasumber. Narasumber yang akan diwawancarai adalah Saudari Ekadriani Fitria selaku pemilik dari Brand Nue Pins. Hasil dari analisis ini akan digunakan untuk mengidentifikasi kebutuhan fungsional dan non fungsional pada rancangan sistem. Kebutuhan sistem akan digambarkan dalam use case diagram untuk mengetahui tujuan aktor menggunakan sistem. Setelah itu, use case diagram akan dideskripsikan dalam bentuk use case scenario. Alur dari use case scenario, akan digambarkan dalam Activity Diagram.

3.3 Perancangan Sistem

Perancangan sistem dilakukan untuk mewujudkan kebutuhan yang telah dianalisa menjadi fungsi - fungsi yang akan berjalan pada sistem. Pada tahap perancangan, akan dilakukan perancangan arsitektur sistem yang dimodelkan dalam bentuk sequence diagram, class diagram, dan perancangan basis data dalam bentuk Entity Relation Diagram. Mekanisme perancangan akan dilakukan berdasarkan analisa kebutuhan yang telah didapat melalui hasil wawancara dengan pemilik Brand Nue Pins. Pembuatan diagram sequence, didasarkan pada kebutuhan fungsional yang telah dijabarkan pada tahap analisis kebutuhan. Tahap Perancangan juga dilakukan untuk memenuhi analisa kebutuhan yang telah didapat dari hasil wawancara dengan pemilik brand Nue Pins, untuk dijadikan rancangan dalam proses Implementasi Sistem Informasi.

3.4 Implementasi Sistem

Pada tahap implementasi, sistem informasi toko *online* Nue Pins akan diimplementasikan dalam bentuk *website* dengan menggunakan *framework Code Igniter* dan bahasa pemrograman PHP. Sistem Informasi Toko *Online* Nue Pins akan diintegrasikan dengan *API Midtrans*. *API Midtrans* akan digunakan sebagai media pembayaran. Perancangan interface dan kelas - kelas yang berkaitan dengan alur utama sistem, akan diimplementasikan dalam bentuk kode sumber. Tahap Implementasi dilakukan dengan tujuan untuk mewujudkan proses - proses yang telah dilakukan pada tahap analisis kebutuhan dan perancangan, menjadi sebuah sistem informasi yang dapat berjalan dalam memberikan Input dan Output kepada pengguna.

3.5 Pengujian Sistem

Pada tahap ini, pengujian sistem akan dilakukan dengan 2 metode, yaitu metode pengujian *blackbox* dan *usability*. Tujuan penggunaan *Blackbox Testing* adalah menguji sejauh mana sistem dapat melakukan proses dari inputan yang dimasukan oleh pengguna. Pengujian *Blackbox Testing* juga ditujukan untuk mengetahui apakah masing - masing fungsi yang telah diimplementasikan telah sesuai analisis kebutuhan fungsional yang telah didapatkan. Sedangkan tujuan pengujian menggunakan metode *usability testing* adalah untuk mengukur kualitas dari sistem informasi yang dibangun sehingga dapat menciptakan *user experience*

yang baik dan mengetahui kekurangan yang ada pada aplikasi. Pengujian sistem akan dilaksanakan kepada calon pemakai, yaitu calon pembeli produk *brand* Nue Pins yang dapat menggunakan Internet. Hasil dari pengujian *Black Box* akan digambarkan dalam bentuk tabel yang berisikan nama - nama fungsi yang telah dicantumkan pada kebutuhan fungsional. Sedangkan hasil pengujian *Usability* akan digambarkan dalam bentuk tabel *System Usability Scale*.

3.6 Kesimpulan dan Saran

Pada tahap ini, penarikan kesimpulan didasarkan pada tahap analisis kebutuhan sistem, perancangan yang telah dilakukan, implementasi yang telah dilakukan dan pengujian yang telah dilakukan. Penarikan kesimpulan juga diambil dari inti permasalahan dan solusi yang telah diwujudkan dalam penelitian. Tahap terakhir adalah penarikan saran. Penarikan saran, bertujuan untuk meningkatkan kualitas dari kekurangan - kekurangan yang belum diwujudkan pada penelitian ini sebagai landasan untuk pengembangan selanjutnya. Penarikan saran akan diambil dari hasil evaluasi yang diberikan oleh pemilik *brand* Nue Pins dan calon pengguna *website* toko *online* Nue Pins.



BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Sistem Informasi toko *online* Nue Pins merupakan sistem informasi berbasis *website* yang menggunakan *API Midtrans* sebagai media pembayaran. Sistem Informasi toko *online* Nue Pins dibuat untuk memindahkan operasional penjualan produk dari merek Nue Pins yang sebelumnya menggunakan *social media Instagram*, menjadi *website* yang dapat melakukan proses transaksi didalamnya. Sistem informasi toko *online* Nue Pins dapat digunakan oleh calon pembeli produk, dan calon pembeli dapat dikatakan sebagai *User*. *User* dapat melakukan registrasi akun pada *website* toko *online* Nue Pins dengan melakukan pengisian data pribadi seperti *Email*, Nama lengkap, dan *password*. Setelah *User* melakukan registrasi, *User* dapat memilih produk yang akan dibeli dengan cara memasukan ke keranjang belanja untuk melanjutkan ke proses transaksi. *User* dapat memasukan lebih dari satu produk ke keranjang belanja dan memilih produk mana yang terlebih dahulu untuk dibeli. Setelah *User* melakukan pemilihan barang yang akan dibeli, *User* diarahkan ke halaman pembayaran untuk melakukan pengisian data diri seperti alamat tujuan, nomor *handphone*, nama penerima, dan melakukan pemilihan metode pembayaran seperti melalui *bank transfer*, dan *minimarket* yang telah terdaftar oleh *Midtrans*. Setelah itu, *User* akan dikirimkan kode pembayaran, nominal harga yang akan dibayar dan cara pembayaran, pada halaman selanjutnya. Setelah *User* melakukan pembayaran, *User* dapat melihat status transaksi yang sedang berlangsung pada halaman pemesanan yang berada pada *website* toko *online* Nue Pins.

Pembayaran dapat diverifikasi dengan melihat status transaksi pada *website Midtrans*, oleh pemilik *brand* Nue Pins yang dapat dikatakan sebagai *Admin*. Setelah status transaksi adalah "*Settlement*" maka *Admin*, dapat melakukan pengiriman produk ke alamat yang telah dicantumkan oleh *User*. *Admin* juga akan menggunakan sistem informasi berbasis *website* khusus *Admin*, untuk melakukan penambahan produk baru, perubahan data produk, dan penghapusan produk. Pengisian data produk baru berupa nama produk, deskripsi produk, jumlah stok produk, kategori produk, harga produk, dan foto produk. Jumlah stok produk akan otomatis berkurang setelah *User* telah menyelesaikan pembayaran dan Jumlah stok produk dapat ditambahkan secara manual oleh *Admin* pada sistem informasi yang diperuntukan khusus *Admin*. Jika stok produk adalah 0, maka sistem akan otomatis menghilangkan informasi produk tersebut dari *website* toko *online* Nue Pins, sehingga tidak dapat dipilih oleh *User* untuk dimasukan ke keranjang belanja. *Admin* dapat melakukan perubahan status pemesanan yang dilakukan oleh *User* pada sistem informasi berbasis *website* khusus *Admin* setelah melakukan verifikasi pemesanan *User*.

4.2 Identifikasi Aktor

Berdasarkan hasil wawancara yang telah dilakukan oleh pemilik *brand* Nue Pins untuk pembentukan sistem informasi toko *online* berbasis *website*, terdapat dua aktor yang akan berinteraksi dengan sistem. Aktor tersebut merupakan calon pembeli sebagai *User* dan pemilik *brand* Nue Pins sebagai *Admin*. Pada tabel 4.1 akan dijelaskan perilaku yang dapat dilakukan oleh masing – masing aktor kepada sistem. Pendefinisian perilaku aktor juga bertujuan untuk mengetahui batasan dan hak yang dapat dilakukan oleh masing – masing aktor kepada sistem.

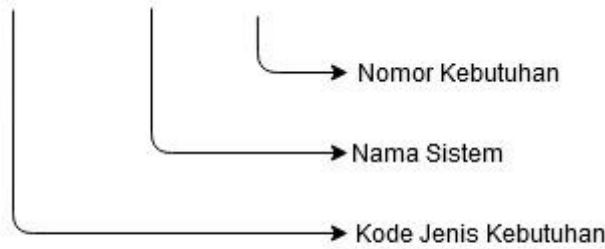
Tabel 4.1 Identifikasi Aktor

| Aktor | Deskripsi Aktor |
|-------|---|
| Admin | <i>Admin</i> merupakan pemilik dari <i>brand</i> Nue Pins selaku aktor yang memiliki hak akses <i>Create, Read, Update, dan Delete</i> . Terhadap data produk. Dan hak akses <i>Read, Update, dan Delete</i> terhadap data pemesanan. <i>Admin</i> dapat melakukan <i>login</i> untuk masuk kedalam sistem informasi khusus <i>Admin</i> . |
| User | <i>User</i> merupakan pengguna dari <i>website</i> toko <i>online</i> Nue Pins selaku aktor yang memiliki hak akses <i>Read</i> pada data produk, <i>Create</i> pada data pemesanan, dan <i>Read</i> pada data status pemesanan. <i>User</i> dapat melakukan registrasi akun dan dapat melakukan <i>login</i> untuk melakukan pembelian produk di <i>website</i> toko <i>online</i> Nue Pins. |

4.3 Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan sistem bertujuan untuk mengetahui kebutuhan apa saja yang dibutuhkan secara fungsional dan secara non fungsional pada Sistem Informasi toko *online* Nue Pins. Kebutuhan fungsional meliputi proses – proses yang berjalan untuk menyelesaikan proses bisnis pada sistem. Proses – proses tersebut dapat diartikan sebagai fitur – fitur yang berjalan pada sistem informasi. Kebutuhan non fungsional meliputi kepuasan pengguna dalam menggunakan sistem informasi. Untuk mengidentifikasi kebutuhan fungsional dan non fungsional akan digunakan penomoran kode untuk memberikan tanda pada masing – masing kebutuhan. Format penomoran kode dimulai dengan jenis kebutuhan, nama sistem, dan kode kebutuhan. Format penomoran kode akan dilampirkan pada gambar 4.1.

F/NF_ENUE_01



- F = Fungsional
- NF = Non Fungsional

Gambar 4.1 Penmoran Kebutuhan Sistem

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional didapatkan berdasarkan hasil wawancara dengan pemilik *brand* Nue Pins. Hasil wawancara terkait apa saja yang akan dilakukan oleh sistem, akan dijadikan sebagai fitur yang akan berjalan pada sistem informasi toko *online* Nue Pins. Penjabaran fitur – fitur yang akan dibuat, akan dilampirkan pada kebutuhan fungsional. Pada tabel 4.2 akan menunjukkan kebutuhan fungsional aplikasi berdasarkan masing – masing aktor yang telah diidentifikasi.

Tabel 4.2 Kebutuhan Fungsional

| No | Kode Kebutuhan | Nama Fungsi | Aktor | Deskripsi |
|----|----------------|-----------------|--------------|--|
| 1. | F_ENUE_01 | <i>Login</i> | <i>User</i> | Sistem mampu menyediakan fungsi <i>login</i> dengan menggunakan <i>email</i> dan <i>password</i> untuk masuk kedalam sistem informasi toko <i>online</i> Nue Pins pada <i>user</i> . |
| 2. | F_ENUE_02 | <i>Login</i> | <i>Admin</i> | Sistem mampu menyediakan fungsi <i>login</i> dengan menggunakan <i>email</i> dan <i>password</i> untuk masuk kedalam sistem informasi khusus <i>admin</i> . |
| 3. | F_ENUE_03 | <i>Register</i> | <i>User</i> | Sistem mampu menyediakan fungsi untuk mendaftarkan akun kepada <i>User</i> . |
| 4. | F_ENUE_04 | <i>Logout</i> | <i>User</i> | Sistem mampu menyediakan fungsi untuk mengeluarkan |

| | | | | |
|-----|-----------|-----------------------------|--------------|---|
| | | | | akun <i>User</i> dari sistem informasi toko online Nue Pins |
| 5. | F_ENUE_05 | <i>Logout</i> | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk mengeluarkan akun <i>Admin</i> dari sistem informasi khusus <i>Admin</i> . |
| 6. | F_ENUE_06 | Menampilkan produk | <i>User</i> | Sistem mampu menyediakan fungsi untuk menampilkan data produk kepada <i>User</i> berdasarkan kategori yang telah dipilih oleh <i>User</i> . |
| 7. | F_ENUE_07 | Keranjang Belanja | <i>User</i> | Sistem mampu menyediakan fungsi untuk menampung produk yang akan dipesan kepada <i>User</i> . |
| 8. | F_ENUE_08 | Menghapus Produk | <i>User</i> | Sistem mampu menyediakan fungsi untuk menghapus produk pada keranjang belanja <i>User</i> . |
| 9. | F_ENUE_09 | Menentukan jumlah pembelian | <i>User</i> | Sistem mampu menyediakan fungsi untuk mengatur jumlah pembelian pada masing – masing produk di keranjang belanja <i>User</i> . |
| 10. | F_ENUE_10 | Membuat pemesanan | <i>User</i> | Sistem mampu menyediakan fungsi untuk membuat pemesanan produk dengan melakukan pengisian data alamat tujuan, nama penerima, pemilihan metode pembayaran kepada <i>User</i> dan sistem mampu menampilkan kode pembayaran. |
| 11. | F_ENUE_11 | <i>Track Pemesanan</i> | <i>User</i> | Sistem mampu menyediakan fungsi untuk menampilkan data status produk yang telah dipesan oleh <i>User</i> . |
| 12. | F_ENUE_12 | Informasi produk | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk menampilkan data semua produk pada |

| | | | | |
|-----|-----------|-----------------------|--------------|---|
| | | | | pada sistem informasi khusus <i>Admin</i> . |
| 13. | F_ENUE_13 | Menambah data produk | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk penambahan data produk pada sistem informasi khusus <i>Admin</i> yang nantinya dapat ditampilkan ke sistem informasi toko <i>online</i> Nue Pins. |
| 14. | F_ENUE_14 | Mengubah data produk | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk mengubah data produk yang telah dibuat sebelumnya. |
| 15. | F_ENUE_15 | Menghapus data produk | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk menghapus data produk. |
| 16. | F_ENUE_16 | Informasi pemesanan | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk menampilkan data semua pemesanan yang telah dilakukan oleh <i>User</i> pada sistem informasi khusus <i>Admin</i> . |
| 17. | F_ENUE_17 | Verifikasi pemesanan | <i>Admin</i> | Sistem mampu menyediakan tombol fungsi untuk aktor admin dapat melakukan verifikasi pemesanan jika pembayaran telah berstatus " <i>Settlement</i> " pada informasi transaksi di <i>website Midtrans</i> . |
| 18. | F_ENUE_18 | Dashboard | <i>Admin</i> | Sistem mampu menyediakan fungsi untuk menampilkan informasi daftar pemasukan dalam hitungan bulan, informasi produk yang sering dipesan, dan daftar pemesanan terakhir. |

4.3.2 Kebutuhan Non Fungsional

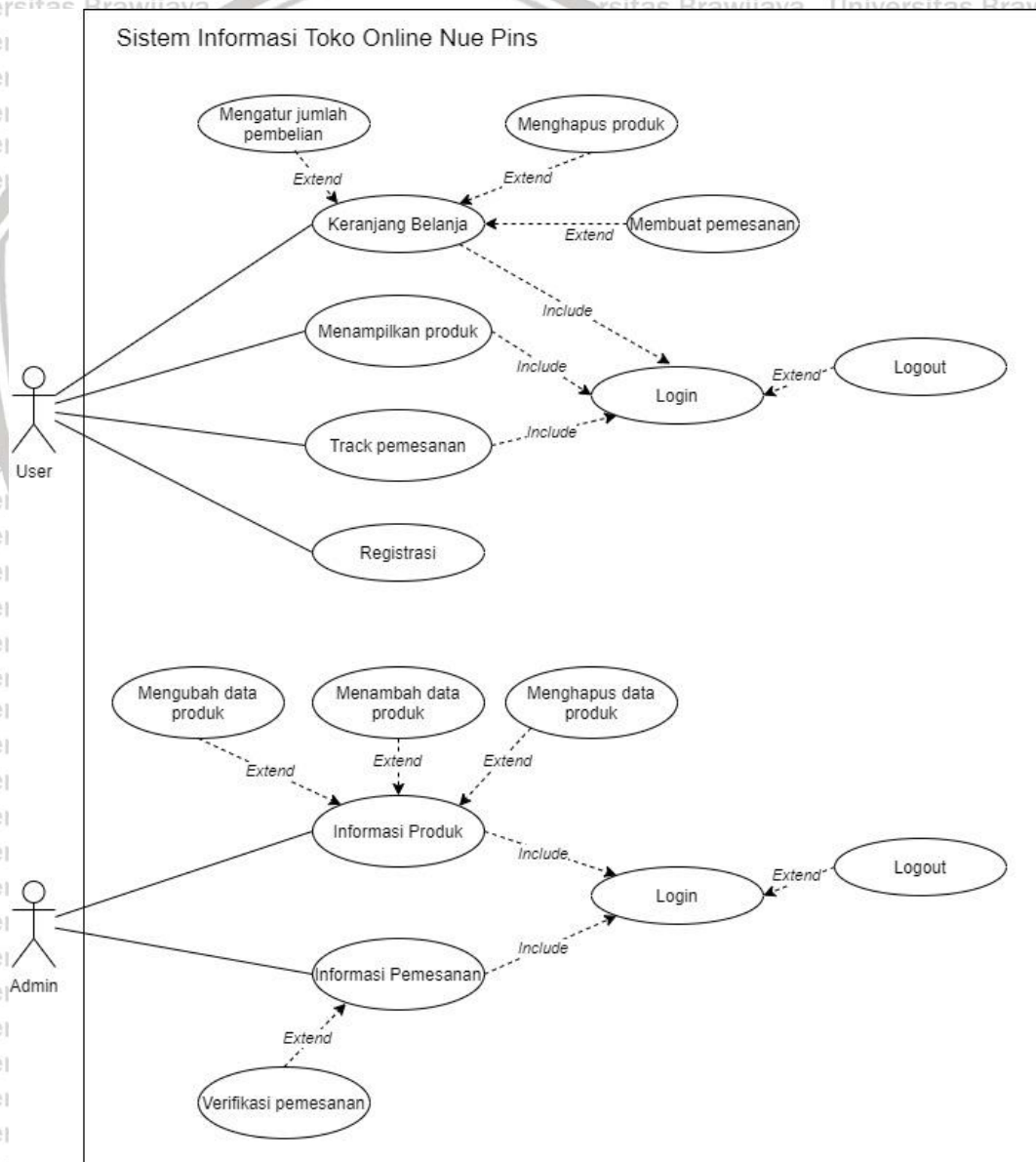
Kebutuhan non fungsional merupakan karakteristik pada sistem, seperti *platform* yang digunakan oleh sistem, komabilitas sistem, peforma sistem, dan akses untuk sistem. Kebutuhan non fungsional yang diperhatikan dalam

pengembangan sistem informasi toko online Nue Pins, adalah *Usability*. *Usability* merupakan kemudahan aktor dalam penggunaan sistem. Pada tabel 4.3 akan menunjukkan kebutuhan non fungsional pada sistem.

Tabel 4.3 Kebutuhan Non Fungsional

| No | Kode Kebutuhan | Nama Fungsi | Deskripsi |
|----|----------------|------------------|--|
| 1. | NF_ENUE_01 | <i>Usability</i> | Aktor mampu untuk mengoperasikan sistem, dan mempelajari langkah – langkah pada penggunaan sistem. |

4.4 Use Case Diagram



Gambar 4.2 Use Case Diagram

4.5 Use Case Scenario

Use case scenario akan menjelaskan interaksi antar aktor dengan sistem, termasuk pertukaran pesan dan tindakan yang dilakukan oleh aktor. Pada tahap ini, akan dijelaskan tujuan, aktor, *pre condition*, *post condition*, *main flow* dan *alternative flow* dari masing – masing *use case*.

4.5.1 Use Case Scenario Login User

Use case scenario login User menjelaskan Langkah dari aktor *User* yang akan masuk ke sistem informasi toko *online* Nue Pins melalui halaman *login*. *User* akan memasukan *email* dan *password* pada kolom yang telah disediakan. Jika *User* memasukan *email* dan *password* dengan benar, maka *User* akan diarahkan kehalaman utama sistem informasi *e-commerce* Nue Pins. Jika *User* memasukan *email* dan *password* yang salah, maka sistem akan memberikan notifikasi bahwa *email* atau *password* salah dan *User* akan tetap berada pada halaman *login*. Jika salah satu kolom dari *email* dan *password* tidak diisi, maka sistem akan memberikan notifikasi untuk melakukan pengisian pada semua kolom, dan *User* akan tetap berada pada halaman *login*. Tabel 4.4 akan akan menjelaskan *use case scenario login User*.

Tabel 4.4 Use Case Scenario Login User

| Login User | |
|-------------------------|---|
| Tujuan | Memasukan aktor kedalam sistem |
| Aktor | <i>User</i> |
| <i>Pre Condition</i> | Aktor telah berada di halaman <i>login</i> |
| <i>Post Condition</i> | <i>User</i> berhasil masuk ke halaman utama sistem. |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan halaman <i>login</i> 2. <i>User</i> memasukan <i>email</i> dan <i>password</i> 3. Sistem melakukan pengecekan <i>email</i> dan <i>password</i> 4. Sistem menampilkan halaman utama |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>email</i> dan <i>password</i> yang dimasukan salah, maka sistem akan memberikan notifikasi “<i>email</i> atau <i>password</i> salah” |

4.5.2 Use Case Scenario Login Admin

Use case scenario login Admin menjelaskan langkah dari aktor *admin* pada saat masuk ke aplikasi melalui halaman *login*. *Admin* akan melakukan pengisian *email* dan *password*. Jika *Admin* memasukan *email* dan *password* dengan benar, maka *Admin* akan diarahkan ke halaman utama sistem informasi khusus *Admin*. Jika

Admin memasukkan *email* atau *password* yang salah, maka sistem akan memberikan notifikasi bahwa *email* atau *password* salah, dan *Admin* akan tetap berada pada halaman *login*. Jika salah satu kolom dari *email* dan *password* tidak diisi, maka sistem akan memberikan notifikasi bahwa semua kolom harus diisi, dan *Admin* akan tetap berada pada halaman *login*. Tabel 4.5 akan akan menjelaskan *use case scenario login Admin*.

Tabel 4.5 Use Case Scenario Login Admin

| Login Admin | |
|-------------------------|--|
| Tujuan | Memasukan aktor kedalam sistem |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | Aktor telah berada di halaman <i>login</i> |
| <i>Post Condition</i> | <i>Admin</i> berhasil masuk ke sistem informasi <i>admin</i> . |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem informasi <i>admin</i> menampilkan halaman <i>login</i> 2. <i>Admin</i> memasukkan <i>email</i> dan <i>password</i> 3. Sistem melakukan pengecekan <i>email</i> dan <i>password</i> 4. Sistem menampilkan halaman utama sistem informasi <i>admin</i> |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>email</i> dan <i>password</i> yang dimasukan salah, maka sistem akan memberikan notifikasi "<i>email</i> atau <i>password</i> salah" |

4.5.3 Use Case Scenario Register

Use case scenario register menjelaskan langkah – langkah aktor *User* melakukan registrasi akun pada sistem informasi toko *online* Nue Pins. *User* akan melakukan pengisian terhadap semua kolom pada halaman *register*. Setelah selesai melakukan pengisian dengan benar, maka *User* akan mendapatkan *email* konfirmasi pada alamat *email* yang telah dicantumkan. Jika *User* tidak mengisi semua kolom, maka sistem akan memberikan notifikasi bahwa semua kolom harus diisi. Jika *User* mengisikan dua kolom *password* dengan nilai yang berbeda, maka sistem akan memberikan notifikasi bahwa *password* tidak sama dan harus diisikan sama. Jika *User* memasukkan format *email* yang salah pada kolom *email*, maka sistem akan memberikan notifikasi bahwa *email* tidak dapat digunakan. Tabel 4.6 akan akan menjelaskan *use case scenario register*.

Tabel 4.6 Use Case Scenario Register

| Register | |
|-----------------|---|
| Tujuan | Mendaftarkan akun <i>User</i> ke sistem |
| Aktor | <i>User</i> |

| | |
|-------------------------|--|
| <i>Pre Condition</i> | User telah berada pada halaman register |
| <i>Post Condition</i> | Sistem berhasil menyimpan data akun <i>User</i> kedalam <i>database</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan halaman <i>register</i> 2. <i>User</i> melakukan pengisian kolom nama lengkap, email, jenis kelamin, <i>password</i> dan konfirmasi <i>password</i>. 3. Sistem mengirimkan konfirmasi ke alamat <i>email User</i>. |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>User</i> mengisikan dua kolom <i>password</i> dengan isian yang berbeda, maka sistem akan memberikan notifikasi "Password tidak sama" 2. Jika <i>User</i> mengisikan kolom <i>email</i> dengan format yang salah, maka sistem akan memberikan notifikasi "email tidak dapat digunakan" |

4.5.4 Use Case Scenario Logout User

Use case scenario logout User menjelaskan Langkah dari aktor *User* saat mengeluarkan akun *User* dari sistem. *User* akan menekan gambar kepala yang terletak pada *navigation bar* bagian atas, dan menekan tombol *logout*. Tabel 4.7 akan menjelaskan *use case scenario logout User*.

Tabel 4.7 Use Case Scenario Logout User

| Log Out User | |
|-------------------------|--|
| Tujuan | Mengeluarkan aktor dari sistem |
| Aktor | <i>User</i> |
| <i>Pre Condition</i> | <i>User</i> telah masuk ke dalam sistem informasi <i>e-commerce</i> Nue Pins |
| <i>Post Condition</i> | Sistem berhasil mengeluarkan akun aktor dari sistem informasi <i>e-commerce</i> Nue Pins |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. <i>User</i> menekan tombol <i>drop down</i> yang bergambar kepala pada <i>navigation bar</i> bagian atas 2. <i>User</i> menekan tombol <i>logout</i> 3. Sistem mengeluarkan akun <i>User</i> dari sistem informasi <i>e-commerce</i> Nue Pins |
| <i>Alternative Flow</i> | - |

4.5.5 Use Case Scenario Logout Admin

Use case scenario logout Admin menjelaskan langkah – langkah yang dilakukan oleh aktor *Admin* pada saat mengeluarkan akun *Admin* dari sistem. *Admin* akan menekan tombol *logout* yang terletak pada bagian atas kanan pada sistem. Tabel 4.8 akan menjelaskan *use case scenario logout Admin*.

Tabel 4.8 Use case scenario logout Admin

| Log Out Admin | |
|-------------------------|---|
| Tujuan | Mengeluarkan aktor dari sistem |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | <i>Admin</i> telah masuk ke dalam sistem informasi khusus <i>Admin</i> |
| <i>Post Condition</i> | Sistem berhasil mengeluarkan akun aktor dari sistem khusus <i>Admin</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. <i>Admin</i> menekan tombol <i>logout</i> yang telah disediakan pada <i>navigation bar</i> bagian atas 2. Sistem mengeluarkan akun <i>Admin</i> dari sistem informasi khusus <i>Admin</i> |
| <i>Alternative Flow</i> | - |

4.5.6 Use Case Scenario Menampilkan Produk

Use scenario menampilkan produk menjelaskan langkah – langkah yang dilakukan *user* saat untuk memilih produk yang akan dibeli. *User* memilih kategori yang telah disediakan oleh sistem dan sistem akan menampilkan produk – produk berdasarkan kategori yang dipilih. Tabel 4.9 akan menjelaskan *use case scenario* menampilkan produk.

Tabel 4.9 Use Case Scenario Menampilkan Produk

| Menampilkan Produk | |
|---------------------------|---|
| Tujuan | Menampilkan Produk berdasarkan katgori yang dipilih oleh aktor |
| Aktor | <i>User</i> |
| <i>Pre Condition</i> | <i>User</i> telah berada pada halaman utama sistem informasi e-commerce Nue Pins |
| <i>Post Condition</i> | Sistem berhasil menampilkan produk berdasarkan kategori yang dipilih oleh <i>User</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Menampilkan halaman utama |

| | |
|-------------------------|---|
| | <ol style="list-style-type: none"> 2. <i>User</i> memilih kategori yang telah disediakan oleh sistem 3. Sistem menampilkan halaman produk berdasarkan kategori yang dipilih oleh <i>User</i>. |
| <i>Alternative Flow</i> | - |

4.5.7 Use Case Scenario Keranjang Belanja

Use case scenario Keranjang Belanja menjelaskan alur yang dilakukan *User* saat memilih produk untuk dimasukkan ke dalam keranjang belanja. *User* akan memilih produk untuk dimasukkan ke keranjang belanja. Selanjutnya sistem akan menambahkan produk yang dipilih *User* kedalam keranjang belanja. Jika *User* belum memilih produk, maka tidak ada data yang ditampilkan pada halaman keranjang belanja. Tabel 4.9 akan menjelaskan *use case scenario* keranjang belanja.

Tabel 4.9 Use Case Scenario keranjang belanja

| Keranjang Belanja | |
|-------------------------|--|
| Tujuan | Menampung produk yang telah dipilih oleh <i>User</i> |
| Aktor | <i>User</i> |
| <i>Pre Condition</i> | <i>User</i> telah berada pada halaman penampilan produk berdasarkan kategori yang telah dipilih |
| <i>Post Condition</i> | Sistem berhasil menambahkan produk kedalam keranjang belanja akun <i>User</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan data produk 2. <i>User</i> memilih produk 3. Sistem menambahkan produk yang telah dipilih oleh <i>User</i> kedalam keranjang belanja 4. <i>User</i> menekan tombol keranjang belanja 5. Sistem menampilkan halaman keranjang belanja |
| <i>Alternative Flow</i> | - |

4.5.8 Use Case Scenario Menghapus Produk

Use case scenario menghapus produk menjelaskan langkah – langkah aktor *User* saat menghapus produk dari keranjang belanja. Tabel 4.10 akan menjelaskan *use case scenario* menghapus produk.

Tabel 4.10 Use Case Scenario menghapus produk

| Menghapus Produk | |
|------------------|---|
| Tujuan | Menghapus produk dari keranjang belanja |
| Aktor | User |
| Pre Condition | User telah berada pada halaman keranjang belanja |
| Post Condition | Sistem berhasil menghapus produk dari keranjang belanja pada akun User |
| Main Flow | <ol style="list-style-type: none"> 1. Sistem mengambil data keranjang belanja dari database 2. Sistem menampilkan halaman keranjang belanja 3. User menekan tombol <i>delete</i> pada salah satu produk yang ada di keranjang belanja 4. Sistem menghapus produk dari keranjang belanja |
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika User belum memasukan data produk ke keranjang belanja, maka tidak ada data produk yang ditampilkan. |

4.5.9 Use Case Scenario Menentukan Jumlah Pembelian

Use case scenario menentukan jumlah pembelian menjelaskan langkah – langkah yang dilakukan oleh User saat menentukan jumlah pembelian pada masing – masing produk di keranjang belanja. Tabel 4.11 akan menjelaskan use case cenario menentukan jumlah pembelian.

Tabel 4.11 Use Case Scenario menentukan jumlah pembelian

| Menentukan jumlah pembelian | |
|-----------------------------|---|
| Tujuan | Mengatur jumlah pembelian yang ditentukan oleh aktor |
| Aktor | User |
| Pre Condition | User telah berada pada halaman keranjang belanja |
| Post Condition | Sistem berhasil melakukan <i>update</i> pada jumlah pembelian produk |
| Main Flow | <ol style="list-style-type: none"> 1. Sistem mengambil data keranjang belanja dari database 2. Sistem menampilkan halaman keranjang belanja 3. User menekan tombol tambah atau kurang pada masing – masing produk di keranjang belanja |

| | |
|-------------------------|--|
| | 4. Sistem melakukan <i>update</i> berdasarkan <i>input</i> yang ditentukan oleh <i>User</i> |
| <i>Alternative Flow</i> | 1. Jika <i>User</i> belum memasukan data produk ke keranjang belanja, maka tidak ada data yang ditampilkan |

4.5.10 Use Case Scenario Membuat Pemesanan

Use case scenario membuat pemesanan menjelaskan langkah – langka yang dilakukan *User* saat membuat pemesanan. *User* akan menentukan produk yang ingin dilanjutkan ke proses pemesanan. Selanjutnya sistem akan menampilkan halaman pengisian informasi pemesanan yang berisikan kolom nama penerima, alamat tujuan, nomor *handphone*, dan pemilihan metode pembayaran. *User* akan melakukan pengisian pada kolom yang tersedia. Jika *User* tidak mengisi semua kolom, maka sistem akan memberikan notifikasi bahwa semua kolom wajib diisi. Setelah *User* berhasil melakukan pengisian informasi pemesanan, maka *user* akan diarahkan ke halaman kode pembayaran dan sistem menampilkan kode pembayaran. Tabel 4.12 akan menjelaskan *use case scenario* membuat pemesanan.

Tabel 4.12 Use Case Scenario Membuat Pemesanan

| Membuat Pemesanan | |
|-------------------------|---|
| Tujuan | Menambahkan data pemesanan kedalam sistem |
| Aktor | <i>User</i> |
| <i>Pre Condition</i> | <i>User</i> telah berada pada halaman keranjang belanja |
| <i>Post Condition</i> | Sistem berhasil menambahkan data pemesanan kedalam <i>database</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan halaman keranjang belanja 2. <i>User</i> menekan tombol <i>checkout</i> 3. Sistem menampilkan halaman pengisian informasi pemesanan 4. <i>User</i> melakukan pengisian pada kolom yang telah disediakan, dan memilih metode pembayaran 5. Sistem menampilkan pemilihan metode pembayaran 6. <i>User</i> melakukan pemilihan metode pembayaran 7. Sistem menampilkan kode pembayaran 8. Sistem menyimpan data pemesanan <i>User</i> kedalam <i>database</i> |
| <i>Alternative Flow</i> | 1. Jika <i>User</i> tidak mengisi semua kolom, maka sistem akan menampilkan notifikasi “Semua kolom wajib |

| | |
|--|--|
| | diisi” dan akan tetap berada pada halaman pengisian informasi pemesanan. |
|--|--|

4.5.11 Use Case Scenario Track Pemesanan

Use case scenario track pemesanan menjelaskan langkah – langkah yang dilakukan aktor *User* saat melihat status pemesanan yang telah dibuat. Diawali dengan *User* menekan tombol “Pemesanan Saya” dan sistem akan melakukan pengecekan data terlebih dahulu berdasarkan akun *User*. Jika *User* belum melakukan pemesanan, maka tidak ada data yang ditampilkan pada halaman status pemesanan. Tabel 4.13 akan menjelaskan *use case scenario track* pemesanan.

Tabel 4.13 Use Case Scenario Track Pemesanan

| Track Pemesanan | |
|-------------------------|--|
| Tujuan | Menampilkan status pemesanan yang telah dibuat |
| Aktor | <i>User</i> |
| <i>Pre Condition</i> | <i>User</i> telah membuat pemesanan dan telah berada pada halaman status pemesanan |
| <i>Post Condition</i> | Sistem berhasil menampilkan data status pemesanan |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. <i>User</i> menekan tombol “Pemesanan Saya” yang telah disediakan oleh sistem 2. Sistem melakukan pengecekan data pemesanan berdasarkan akun <i>User</i>. 3. Sistem menampilkan halaman <i>track</i> pemesanan. |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>User</i> belum membuat pemesanan, maka tidak ada data yang ditampilkan. |

4.5.12 Use Case Scenario Informasi Produk

Use case scenario Informasi produk menjelaskan langkah – langkah aktor *admin* untuk melihat data produk pada sistem informasi khusus *Admin*. Diawali dengan *Admin* menekan tombol daftar produk dan sistem akan menampilkan data produk dari *database* yang diurutkan berdasarkan kategori produk. Tabel 4.14 akan menjelaskan *use case cenario* melihat data produk.

Tabel 4.14 Use Case Scenario Melihat Data Produk

| Melihat Data Produk | |
|---------------------|------------------------------|
| Tujuan | Menampilkan Informasi Produk |
| Aktor | <i>Admin</i> |

| | |
|-------------------------|---|
| <i>Pre Condition</i> | <i>Admin</i> berhasil masuk kedalam sistem informasi khusus <i>admin</i> |
| <i>Post Condition</i> | Sistem berhasil menampilkan data produk |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. <i>Admin</i> menekan tombol daftar produk 2. Sistem menampilkan data produk dari <i>database</i> yang diurutkan berdasarkan kategori produk |
| <i>Alternative Flow</i> | - |

4.5.13 Use Case Scenario Menambah Data Produk

Use case scenario menambah data produk menjelaskan langkah – langkah aktor *admin* saat melakukan penambahan data produk pada sistem informasi khusus *Admin*. Diawali dengan aktor *Admin* yang berada pada halaman daftar produk, lalu *Admin* akan menekan tombol tambah produk. Selanjutnya *Admin* akan melakukan pengisian data pada semua kolom. Jika semua kolom telah diisi maka sistem akan menyimpan data kedalam *database*. Jika terdapat kolom yang tidak diisi, maka sistem akan memberikan notifikasi bahwa semua kolom harus diisi. Tabel 4.15 akan menjelaskan *use case scenario* menambah data produk.

Tabel 4.15 *Use Case Scenario* Menambah Data Produk

| Menambah Data Produk | |
|-------------------------|---|
| Tujuan | Menambahkan data produk kedalam <i>database</i> |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | <i>Admin</i> telah berada pada halaman daftar produk |
| <i>Post Condition</i> | Sistem berhasil menambahkan data produk kedalam <i>database</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan halaman daftar produk 2. <i>Admin</i> menekan tombol tambah produk 3. Sistem menampilkan halaman pengisian data produk 4. <i>Admin</i> melakukan pengisian data pada kolom yang disediakan oleh sistem 5. Sistem menyimpan data kedalam <i>database</i> |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>Admin</i> tidak mengisi semua kolom, maka sistem akan memberikan notifikasi “Semua Kolom Harus Diisi” dan akan tetap berada pada halaman pengisian data produk |

4.5.14 Use Case Scenario Mengubah Data Produk

Use case scenario mengubah data produk menjelaskan langkah – langkah saat aktor *Admin* mengubah data produk yang sudah ada. Diawali dengan aktor *Admin* yang telah berada pada halaman daftar produk, lalu *Admin* menekan tombol *edit* pada salah satu produk. Selanjutnya *Admin* melakukan pengisian data pada kolom yang telah disediakan. Jika semua kolom telah terisi, maka sistem akan menyimpan data yang telah diperbarui kedalam *database*. Jika salah satu kolom tidak terisi, maka sistem akan menampilkan notifikasi bahwa semua kolom harus diisi. Tabel 4.16 akan menjelaskan *use case scenario* mengubah data produk.

Tabel 4.16 Use Case Scenario Mengubah Data Produk

| Mengubah Data Produk | |
|-------------------------|---|
| Tujuan | Mengubah data produk yang telah dibuat sebelumnya |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | <i>Admin</i> telah berada pada halaman daftar produk dan data produk telah tersedia |
| <i>Post Condition</i> | Sistem berhasil menyimpan data yang telah diubah kedalam <i>database</i> |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan halaman daftar produk 2. <i>Admin</i> menekan tombol <i>edit</i> pada salah satu produk di daftar produk 3. Sistem menampilkan halaman perubahan data produk 4. <i>Admin</i> melakukan pengisian data pada kolom yang disediakan 5. Sistem menyimpan data yang telah diubah kedalam <i>database</i> |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>Admin</i> tidak mengisi semua kolom, maka sistem akan memberikan notifikasi “Semua Kolom Harus Diisi” |

4.5.15 Use Case Scenario Menghapus Data Produk

Use case scenario menghapus data produk menjelaskan langkah – langkah aktor *admin* saat melakukan penghapusan data produk di sistem informasi khusus *Admin*. Diawali dengan aktor *Admin* yang telah berada pada halaman daftar produk, lalu *Admin* menekan tombol *delete* pada salah satu produk. Selanjutnya sistem akan menghapus produk yang telah dipilih oleh *Admin* dari *database*. Tabel 4.17 akan menjelaskan *use case scenario* menghapus data produk.

Tabel 4.17 Use Case Scenario Menghapus Data Produk

| Menghapus Data Produk | |
|-------------------------|--|
| Tujuan | Menghapus data produk yang telah dibuat sebelumnya |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | <i>Admin</i> telah berada pada halaman daftar produk dan data produk telah tersedia |
| <i>Post Condition</i> | Sistem berhasil menghapus data produk yang dipilih dari <i>database</i> . |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem menampilkan halaman daftar produk 2. <i>Admin</i> menekan tombol <i>delete</i> pada salah satu produk 3. Sistem menghapus produk yang dipilih oleh <i>Admin</i> dari <i>database</i> |
| <i>Alternative Flow</i> | - |

4.5.16 Use Case Scenario Informasi Pemesanan

Use case scenario informasi pemesanan menjelaskan langkah – langkah aktor *Admin* untuk melihat daftar pemesanan di sistem informasi khusus *Admin*. Diawali dengan aktor *Admin* menekan tombol daftar pemesanan dan sistem akan menampilkan data pemesanan dari *database*. Tabel 4.18 menjelaskan *use case scenario* melihat data pemesanan.

Tabel 4.18 Use Case Scenario Melihat Data Pemesanan

| Melihat Data Pemesanan | |
|-------------------------|---|
| Tujuan | Menampilkan Informasi Pemesanan |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | <i>Admin</i> berhasil masuk kedalam sistem informasi khusus <i>admin</i> |
| <i>Post Condition</i> | Sistem berhasil menampilkan data pemesanan |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. <i>Admin</i> menekan tombol daftar pemesanan 2. Sistem mengambil data pemesanan dari <i>database</i> yang diurutkan berdasarkan tanggal. 3. Sistem menampilkan halaman daftar pemesanan. |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika <i>User</i> belum membuat pemesanan, maka tidak ada data pemesanan yang ditampilkan. |

4.5.17 Use Case Scenario Verifikasi Pemesanan

Use case scenario verifikasi pemesanan menjelaskan langkah – langkah saat aktor *Admin* melakukan verifikasi pemesanan. Diawali dengan sistem menampilkan halaman daftar pemesanan dan sistem menampilkan tombol verifikasi pemesanan pada pemesanan yang telah dibayar oleh *user*. Jika pembayar belum dilakukan oleh *User*, maka tombol verifikasi pemesanan tidak ditampilkan oleh sistem. Selanjutnya *Admin* akan menekan tombol verifikasi pada salah satu produk yang telah tersedia tombol verifikasi. Sistem akan melakukan *update* pada status pemesanan *User*. Tabel 4.19 akan menjelaskan *use case scenario* verifikasi pemesanan.

Tabel 4.19 Use Case Scenario Verifikasi Pemesanan

| Verifikasi Pemesanan | |
|-------------------------|---|
| Tujuan | Melakukan verifikasi pemesanan yang telah dilakukan oleh <i>User</i> |
| Aktor | <i>Admin</i> |
| <i>Pre Condition</i> | <i>Admin</i> berada pada halaman daftar pemesanan |
| <i>Post Condition</i> | Sistem berhasil melakukan verifikasi pemesanan |
| <i>Main Flow</i> | <ol style="list-style-type: none"> 1. Sistem mengirim <i>request GET</i> pada <i>API Midtrans</i> untuk mendapatkan data status pembayaran 2. Sistem menampilkan halaman daftar pemesanan 3. <i>Admin</i> melakukan pengisian nomor resi pengiriman 4. <i>Admin</i> menekan tombol verifikasi pemesanan 5. Sistem melakukan <i>update</i> pada status pemesanan <i>user</i>. |
| <i>Alternative Flow</i> | <ol style="list-style-type: none"> 1. Jika pembayaran belum dilakukan oleh <i>User</i> atau belum berstatus <i>Settlement</i> pada <i>website midtrans</i>, maka tombol verifikasi pemesanan dinonaktifkan. |

4.5.18 Use Case Scenario Dashboard

Use case scenario dashboard menjelaskan Ingkah – langkah aktor *Admin* saat melakukan akses ke halaman *dashboard*, setelah melakukan *login*. Diawali dengan aktor *admin* telah berhasil melakukan *login*, lalu sistem akan menampilkan halaman *dashboard* kepada aktor *Admin*.

Tabel 4.20 Use Case Scenario Dashboard

| Dashboard | |
|-----------|---|
| Tujuan | Melakukan akses ke halaman <i>dashboard</i> dan mendapatkan informasi pada halaman <i>dashboard</i> |



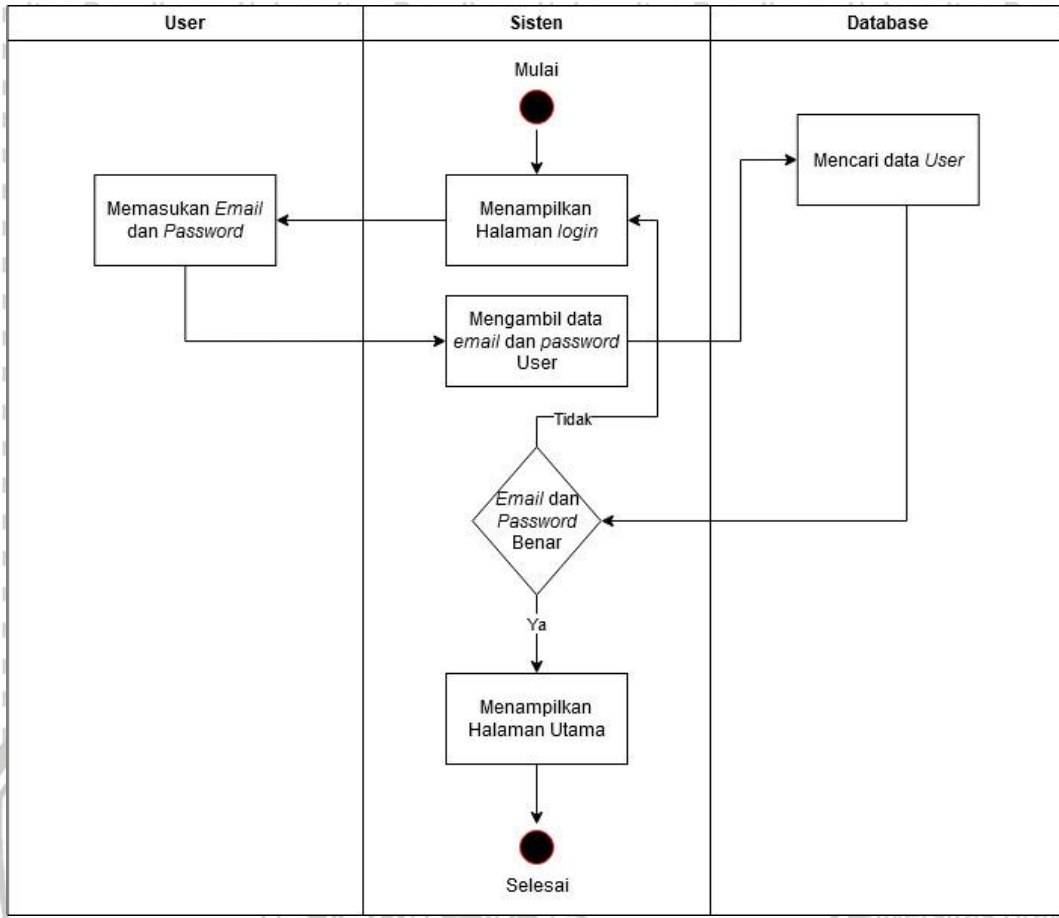
| | |
|------------------|---|
| Aktor | Admin |
| Pre Condition | Admin telah berhasil melakukan <i>login</i> |
| Post Condition | Sistem berhasil menampilkan informasi pada halaman <i>dashboard</i> |
| Main Flow | <ol style="list-style-type: none"> 1. Aktor berhasil melakukan <i>login</i> 2. Sistem menampilkan halaman <i>dashboard</i> sistem informasi Admin |
| Alternative Flow | - |

4.6 Activity Diagram

Pada tahap ini, akan digambarkan aktivitas - aktivitas yang dijalankan oleh aktor berdasarkan *main flow* dan *alternative flow* yang berada pada masing - masing *use scenario*. Tujuan pembuatan *activity diagram* adalah untuk memberikan detail dari perilaku aktor kepada sistem dan respon dari sistem kepada aktor.

4.6.1 Activity Diagram Login User

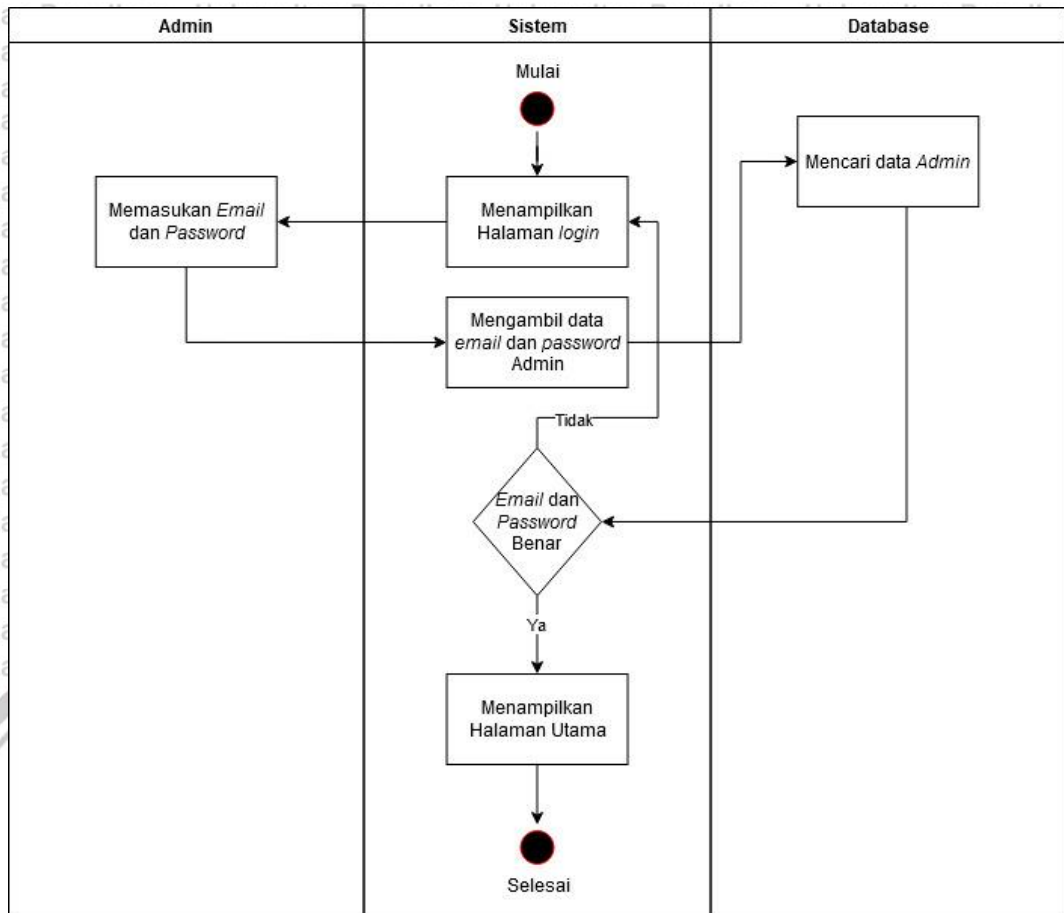
Activity diagram login user menjelaskan alur dari aktivitas aktor *User* saat melakukan *login* akun ke sistem. Dimulai dengan sistem menampilkan halaman *login*, lalu *User* akan memasukan *email* dan *password* kedalam kolom yang telah disediakan. Kemudian sistem akan melakukan pengecekan data *email* dan *password* yang telah dimasukan. Jika data *email* dan *password* tidak ditemukan di *database* maka *User* akan diberikan peringatan untuk memasukan *email* dan *password* dengan benar, dan *User* akan tetap berada pada halaman *login*. Jika *email* dan *password* benar, maka *User* akan diarahkan ke halaman utama sistem. Gambar 4.3 akan menjelaskan bentuk dari *activity diagram login user*.



Gambar 4.3 Activity Diagram Login User

4.6.2 Activity Diagram Login Admin

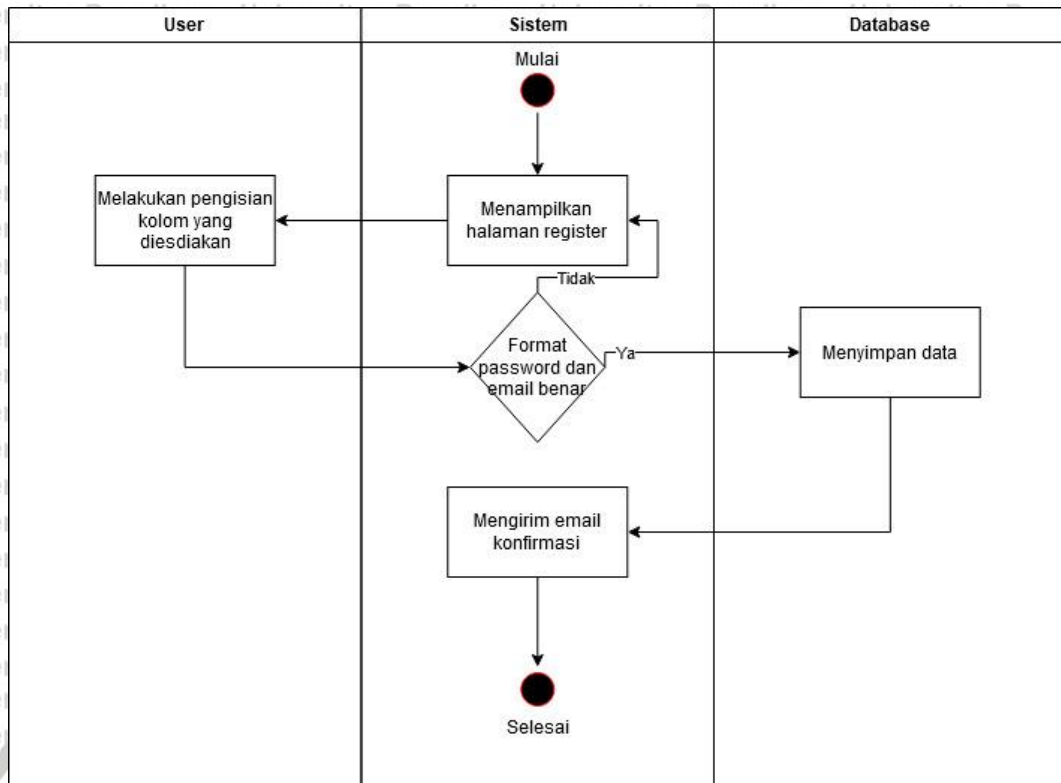
Activity diagram *login admin* menjelaskan alur dari aktivitas aktor *Admin* saat melakukan *login* akun kedalam sistem informasi *admin*. Dimulai dengan sistem menampilkan halaman *login*, lalu *admin* memasukkan *email* dan *password*. Kemudian sistem akan melakukan pengecekan terhadap data *email* dan *password*. Jika *email* dan *password* yang dimasukan adalah benar, maka *Admin* akan diarahkan ke halaman utama sistem informasi *Admin*. Jika *email* dan *password* yang dimasukan adalah salah, maka sistem akan memberikan peringatan untuk memasukan *email* dan *password* dengan benar, dan *Admin* akan tetap berada pada halaman *login*. Gambar 4.4 akan menjelaskan bentuk dari *activity diagram login Admin*.



Gambar 4.4 Activity Diagram Login Admin

4.6.3 Activity Diagram Register

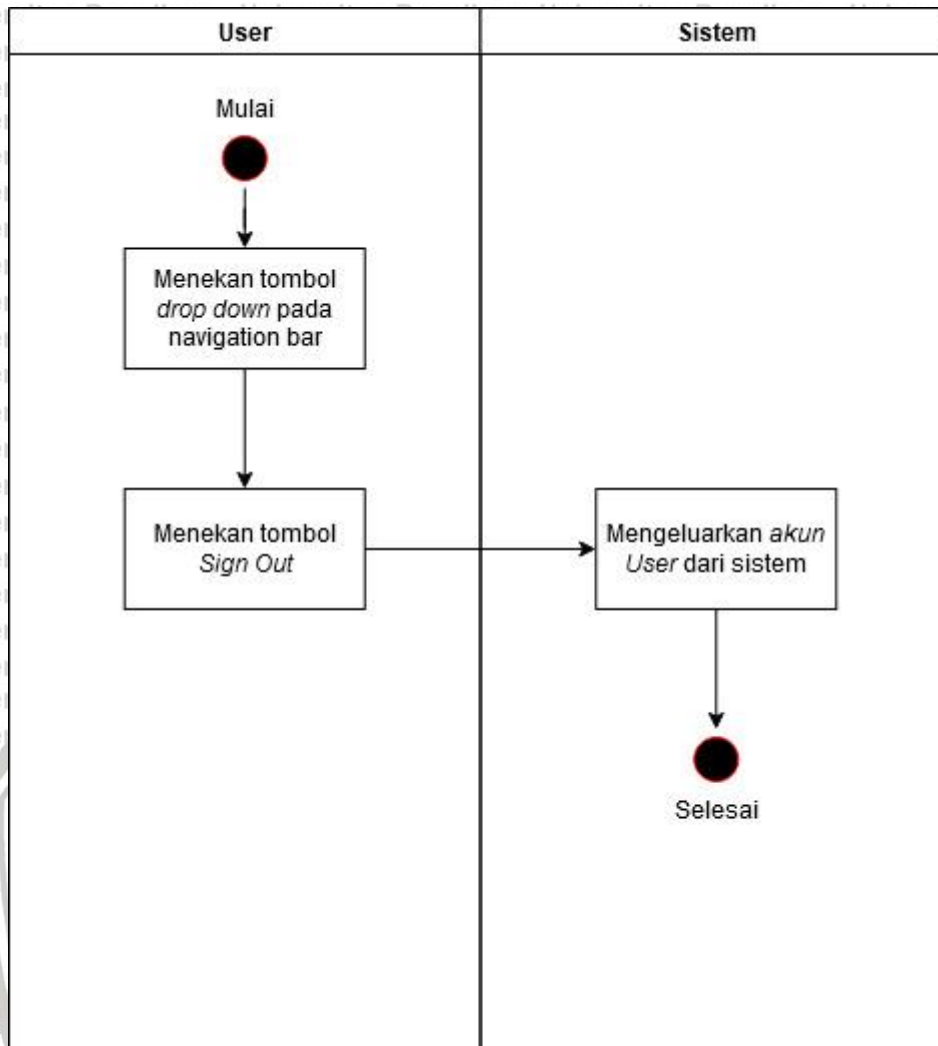
Activity diagram *register* menjelaskan alur dari aktivitas aktor *User* saat melakukan registrasi akun di sistem. Dimulai dengan sistem menampilkan halaman *register*, lalu *User* akan melakukan pengisian kolom yang disediakan. Setelah itu, sistem akan melakukan pengecekan terhadap *format* alamat *email* dan kecocokan dua kolom *password*. Jika *format email* dan kecocokan *password* tidak benar, maka sistem akan memberikan peringatan dan tetap mengarahkan *User* di halaman *register*. Jika *format email* dan kecocokan *password* benar, maka *User* akan diberikan *email* konfirmasi ke alamat *email* yang telah dimasukan. Gambar 4.5 akan menjelaskan bentuk dari *activity diagram register*.



Gambar 4.5 Activity Diagram Register

4.6.4 Activity Diagram Logout User

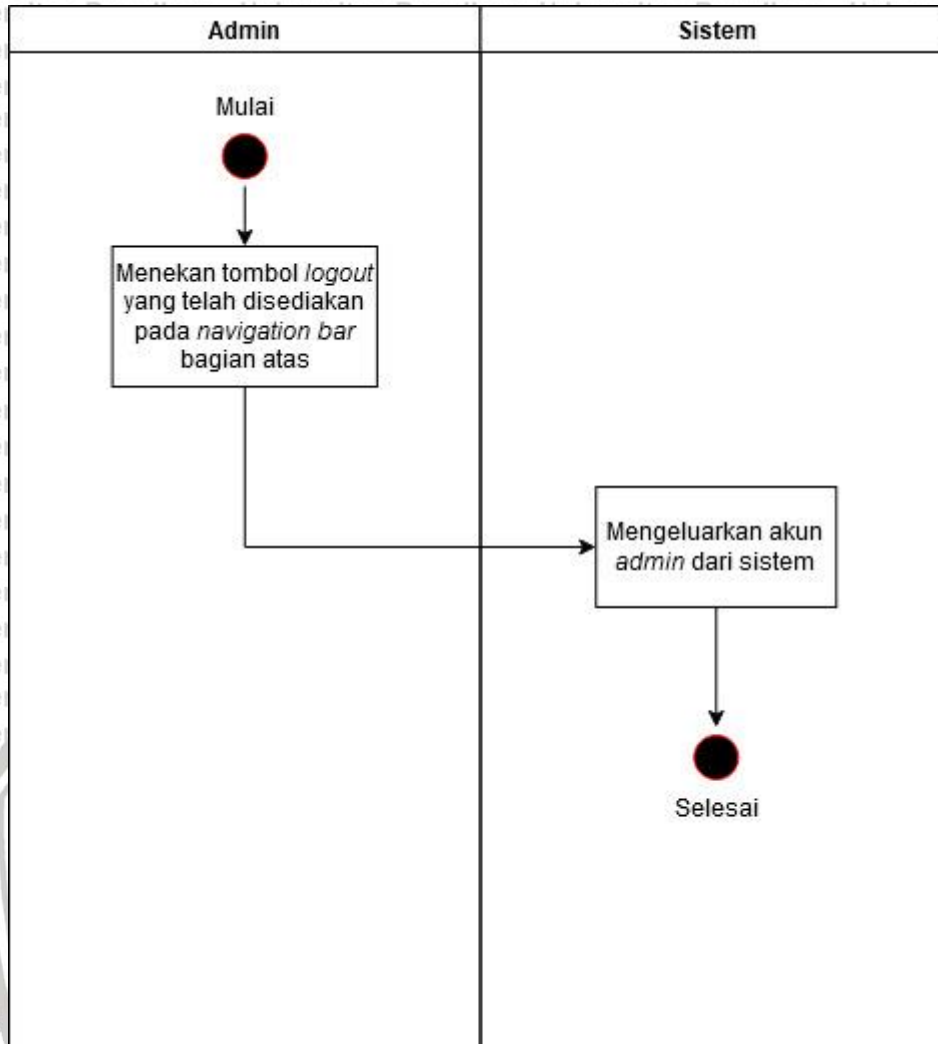
Activity diagram *logout User* menjelaskan alur dari aktivitas aktor *User* saat mengeluarkan akun dari sistem. Dimulai dengan *User* menekan tombol *drop down* pada bagian atas kanan, lalu *User* tombol *Sign Out*. Kemudian sistem akan mengeluarkan akun *User* dari sistem. Gambar 4.6 akan menjelaskan bentuk dari *activity diagram logout User*.



Gambar 4.6 Activity Diagram Logout User

4.6.5 Activity Diagram Logout Admin

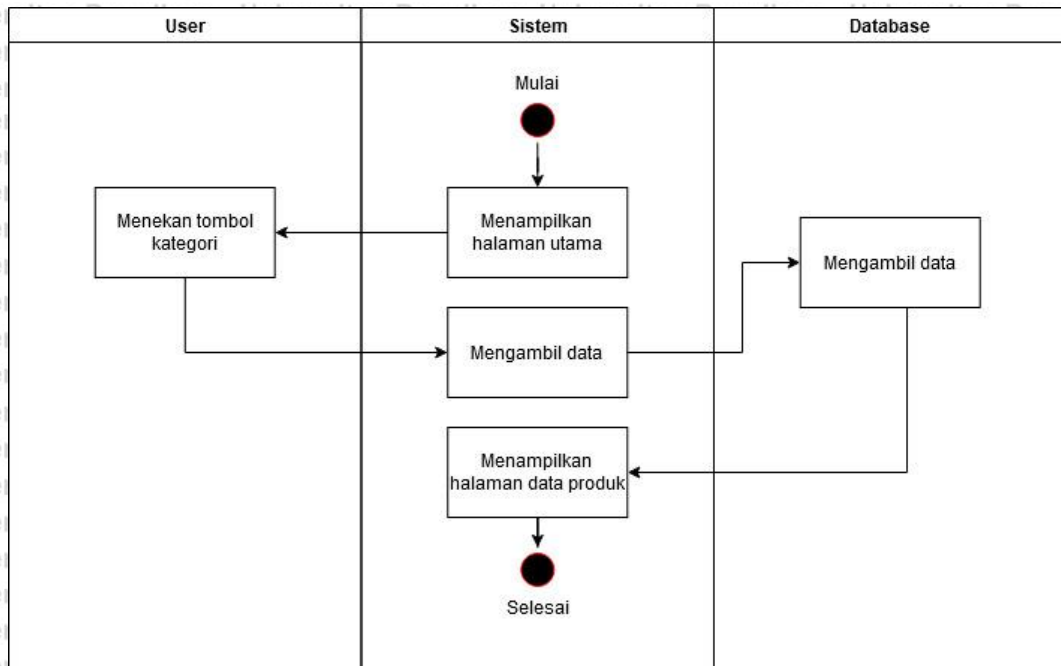
Activity diagram *logout Admin* menjelaskan alur dari aktivitas aktor *Admin* saat mengeluarkan akun dari sistem informasi khusus admin. Dimulai dengan *Admin* menekan tombol *logout* pada bagian atas kanan sistem informasi admin, lalu sistem mengeluarkan akun *Admin* dari sistem informasi admin. Gambar 4.7 akan menjelaskan bentuk dari *activity diagram logout Admin*.



Gambar 4.7 Activity Diagram Logout Admin

4.6.6 Activity Diagram Menampilkan Produk

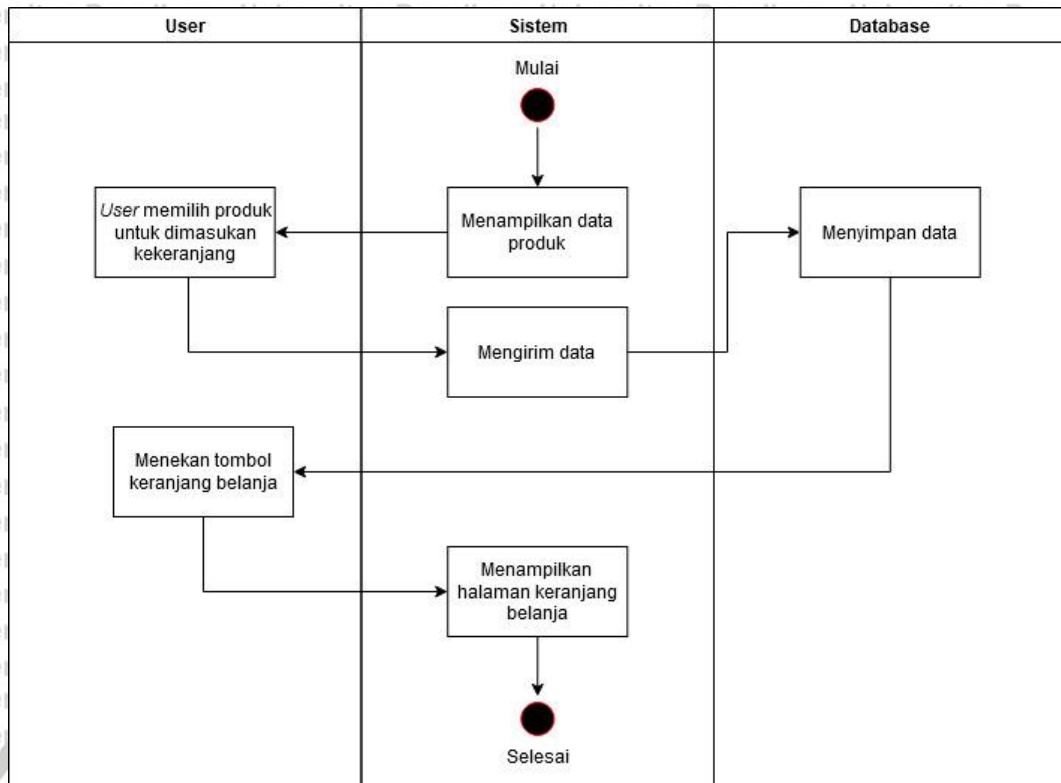
Activity diagram menampilkan produk menjelaskan alur dari aktivitas aktor *User* saat melakukan penelusuran produk berdasarkan pilihan kategori yang ditentukan. Dimulai dengan sistem menampilkan halaman utama sistem, lalu *User* menekan tombol salah satu dari kategori yang disediakan. Lalu sistem akan mengambil data dari *database* berdasarkan kategori yang dipilih oleh *User*, dan akan ditampilkan pada halaman data produk. Gambar 4.8 akan menjelaskan bentuk dari *activity* diagram menampilkan produk.



Gambar 4.8 Activity Diagram Menampilkan Produk

4.6.7 Activity Diagram Keranjang Belanja

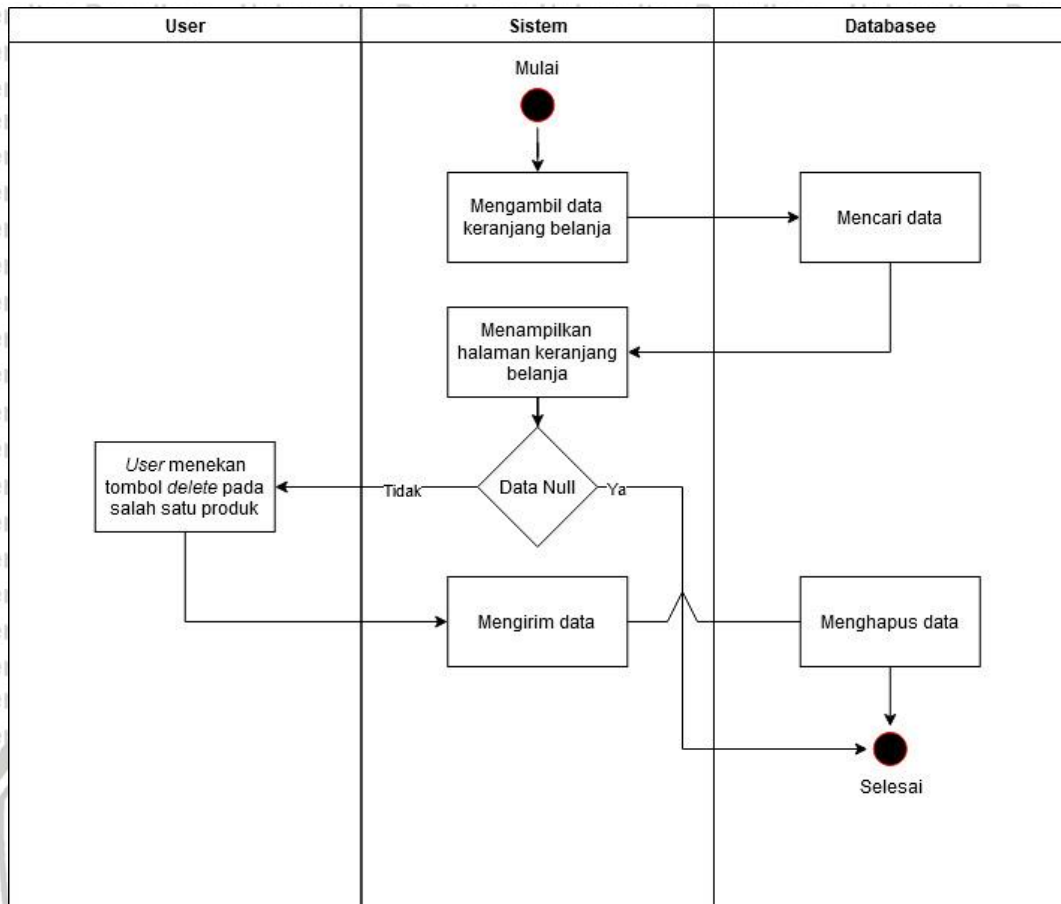
Activity diagram keranjang belanja menjelaskan alur dari aktivitas aktor *User* saat memasukan produk kedalam keranjang belanja. Dimulai dengan sistem menampilkan data produk berdasarkan kategori yang sebelumnya dipilih oleh *User*. Kemudian *User* akan memilih produk untuk dimasukan ke keranjang belanja. Lalu sistem akan mengirimkan data produk yang dipilih oleh *User* kedalam database. Kemudian *User* akan menekan tombol keranjang belanja, dan sistem akan menampilkan data produk yang telah dipilih oleh *User* di keranjang belanja. Gambar 4.9 akan menjelaskan bentuk dari activity diagram keranjang belanja.



Gambar 4.9 Activity Diagram Keranjang Belanja

4.6.8 Activity Diagram Menghapus Produk

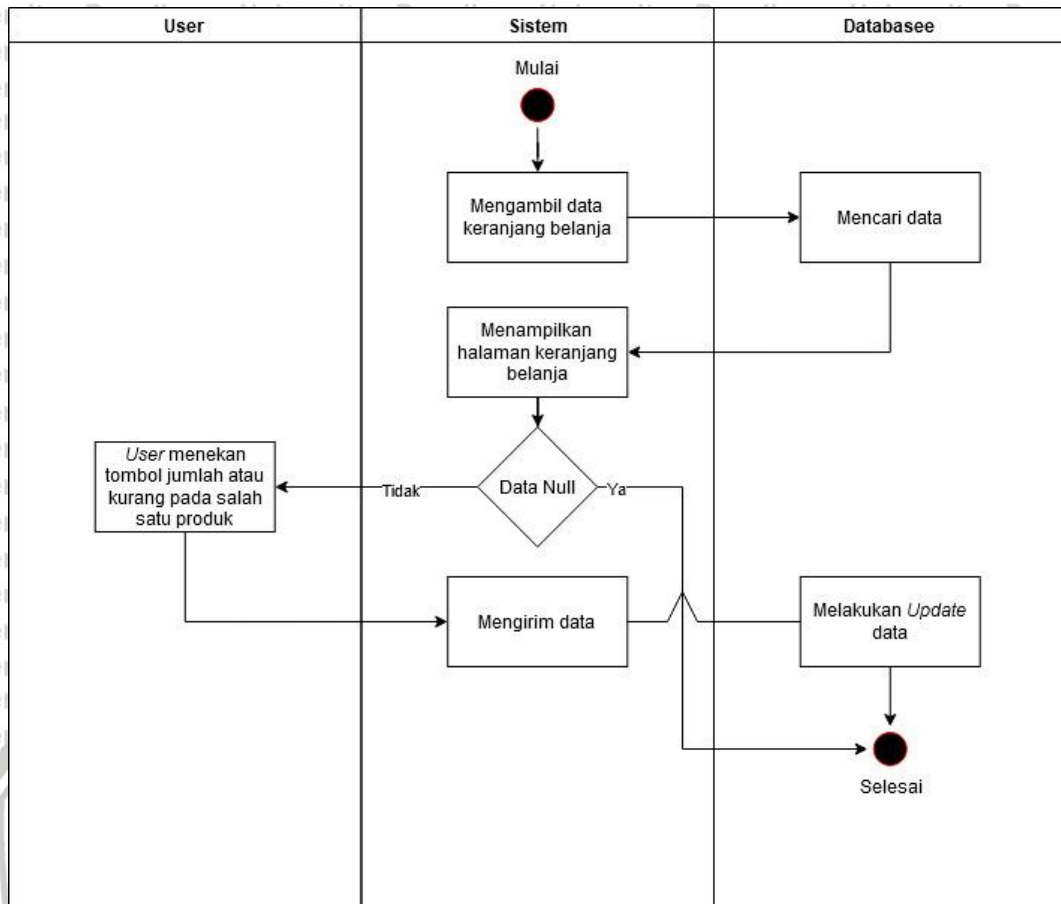
Activity diagram menghapus produk menjelaskan alur dari aktivitas aktor *User* saat melakukan penghapusan data produk dari keranjang belanja. Dimulai dengan sistem mengambil data keranjang belanja *User* dari *database* untuk ditampilkan ke halaman keranjang belanja. Lalu *User* menekan tombol *delete* pada salah satu data keranjang belanja. Kemudian sistem akan menghapus data produk yang telah dipilih dari *database*. Jika *User* belum memasukkan produk ke keranjang belanja, maka tidak ada data yang ditampilkan. Gambar 4.10 akan menunjukkan bentuk dari activity diagram menghapus produk.



Gambar 4.10 Activity Diagram Menghapus Produk

4.6.9 Activity Diagram Menentukan Jumlah Pembelian

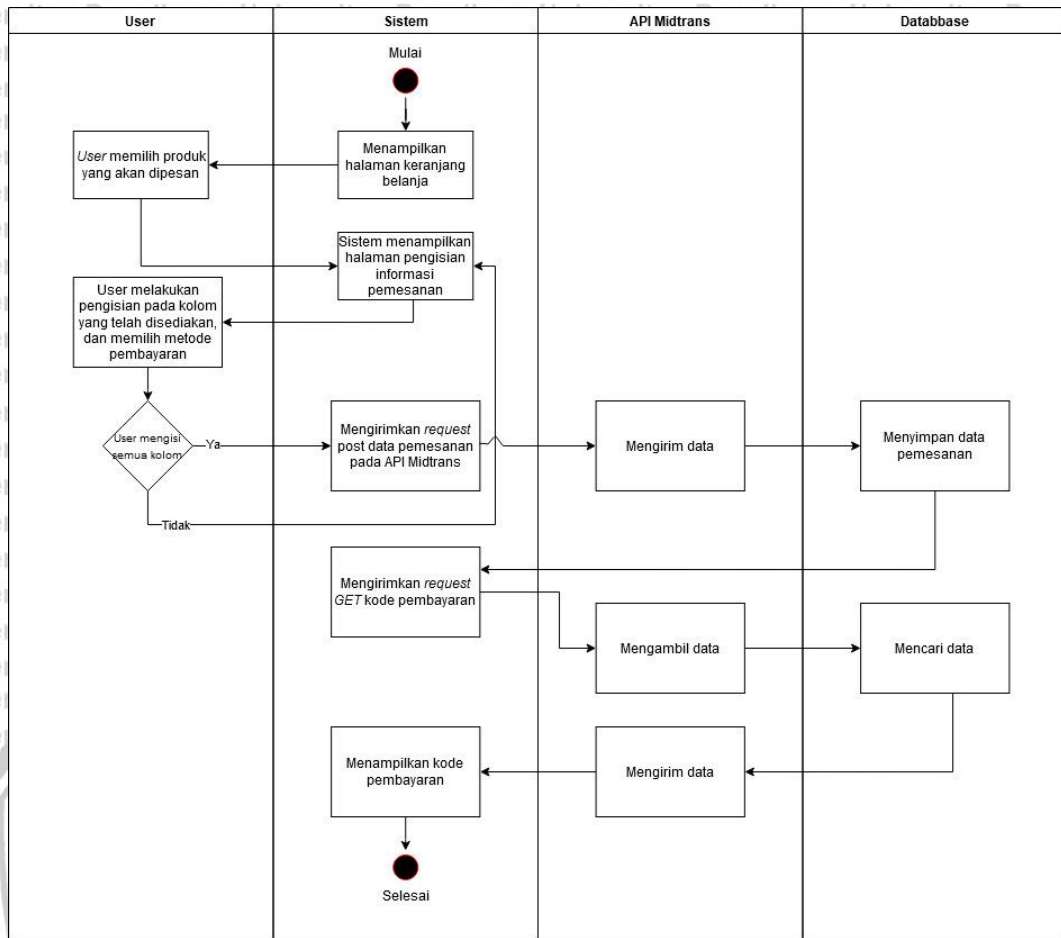
Activity diagram menentukan jumlah pembelian menjelaskan alur dari aktivitas aktor *User* saat menentukan jumlah pembelian pada salah satu produk di keranjang belanja. Dimulai dengan sistem mengambil data produk yang berada di keranjang belanja *User* untuk ditampilkan pada halaman keranjang belanja. Lalu *User* akan menekan tombol jumlah atau kurang pada salah satu produk di keranjang belanja. Kemudian sistem akan mengubah data jumlah pembelian di *database*. Jika *User* belum memasukan produk ke keranjang belanja, maka tidak ada data yang ditampilkan. Gambar 4.11 akan menjelaskan bentuk dari *activity diagram* menentukan jumlah pembelian.



Gambar 4.11 Activity Diagram Menentukan Jumlah Pembelian

4.6.10 Activity Diagram Membuat Pemesanan

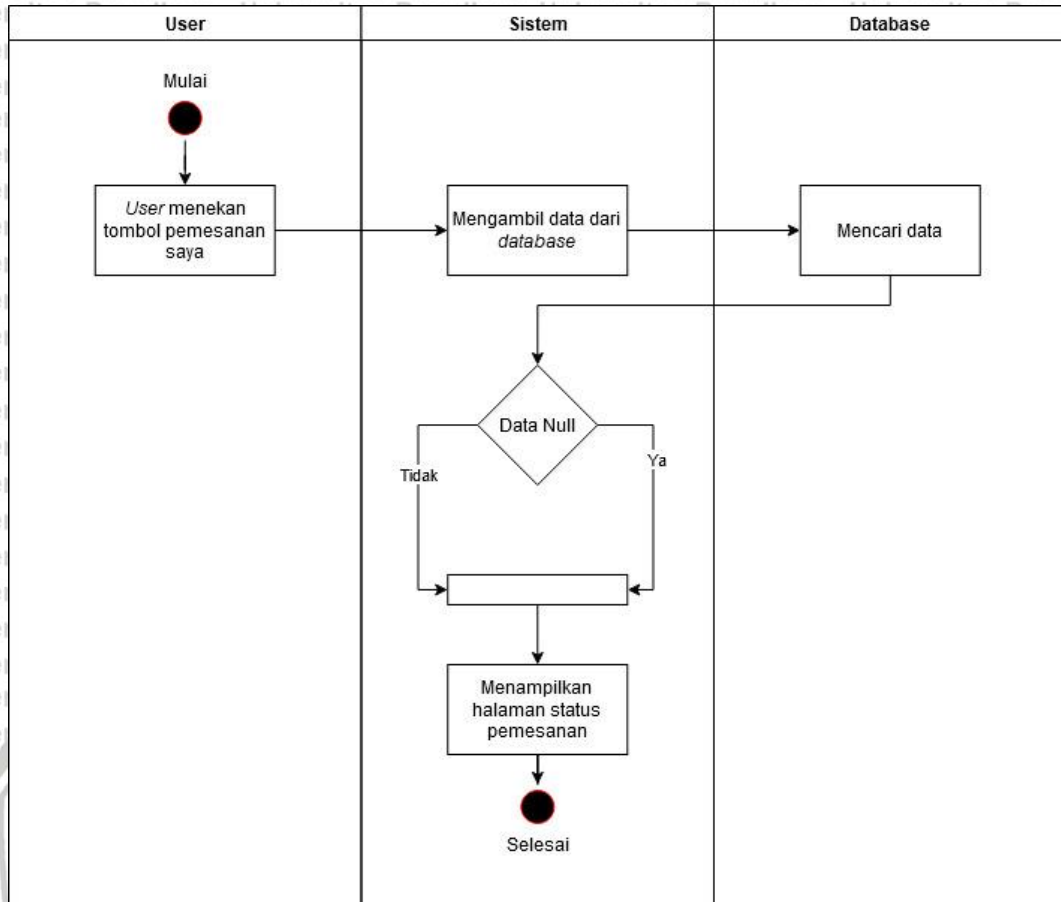
Activity diagram membuat pemesanan menjelaskan alur dari aktivitas aktor *User* saat membuat pemesanan. Dimulai dengan sistem menampilkan halaman keranjang belanja, lalu *User* memilih salah satu produk yang akan dipesan. Lalu sistem akan menampilkan halaman pengisian informasi pemesanan, yang terdapat beberapa kolom untuk diisi oleh *User*. Setelah melakukan pengisian informasi pemesanan, sistem akan melakukan pengecekan apakah semua kolom telah terisi. Jika tidak maka *User* akan diberikan peringatan untuk mengisi semua kolom dan akan tetap berada di halaman pengisian informasi pemesanan. Jika iya, maka sistem akan mengirimkan *request post* ke *API Midtrans* untuk menyimpan pemesanan yang telah dibuat oleh *User*, dan untuk membuat kode pembayaran berdasarkan pemesanan *User*. Kemudian sistem akan menampilkan halaman kode pembayaran dan menunjukan kode pembayaran kepada *User*. Gambar 4.12 akan menjelaskan bentuk dari activity diagram membuat pemesanan.



Gambar 4.12 Activity Diagram Membuat Pemesanan

4.6.11 Activity Diagram Track Pemesanan

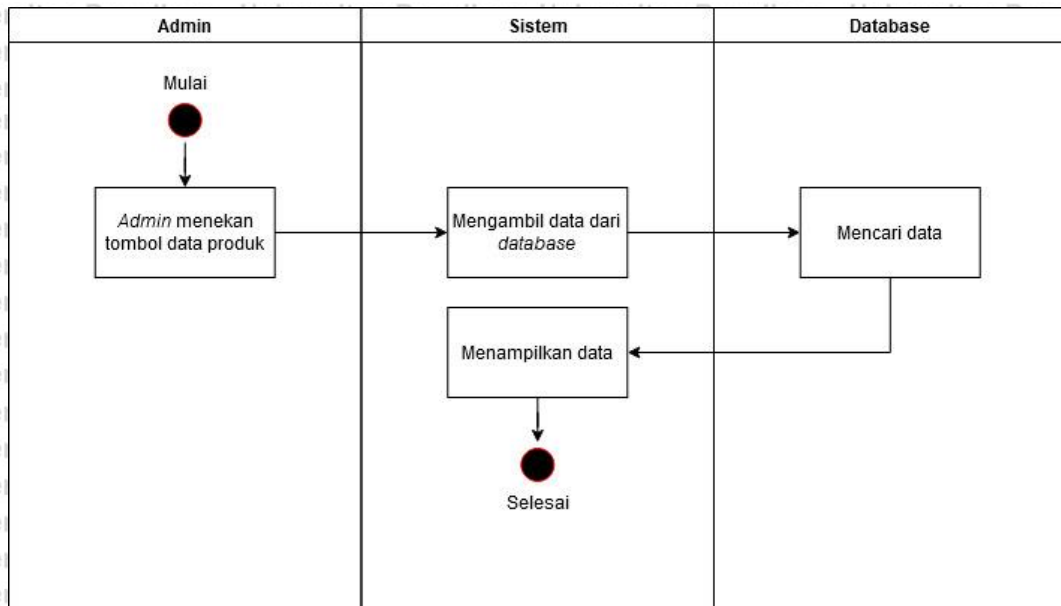
Activity diagram track pemesanan menjelaskan alur dari aktivitas aktor *User* saat melihat status pemesanan. Dimulai dengan *User* menekan tombol pemesanan saya yang telah disediakan, lalu sistem akan mengambil data status pemesanan *User* dari *database*. Jika *User* belum membuat pemesanan, maka tidak ada data yang ditampilkan pada halaman status pemesanan. Jika *User* telah membuat pemesanan, maka data status pemesanan akan ditampilkan di halaman status pemesanan. Gambar 4.13 akan menjelaskan bentuk dari *activity diagram track* pemesanan.



Gambar 4.13 Activity Diagram Track Pemesanan

4.6.12 Activity Diagram Informasi Produk

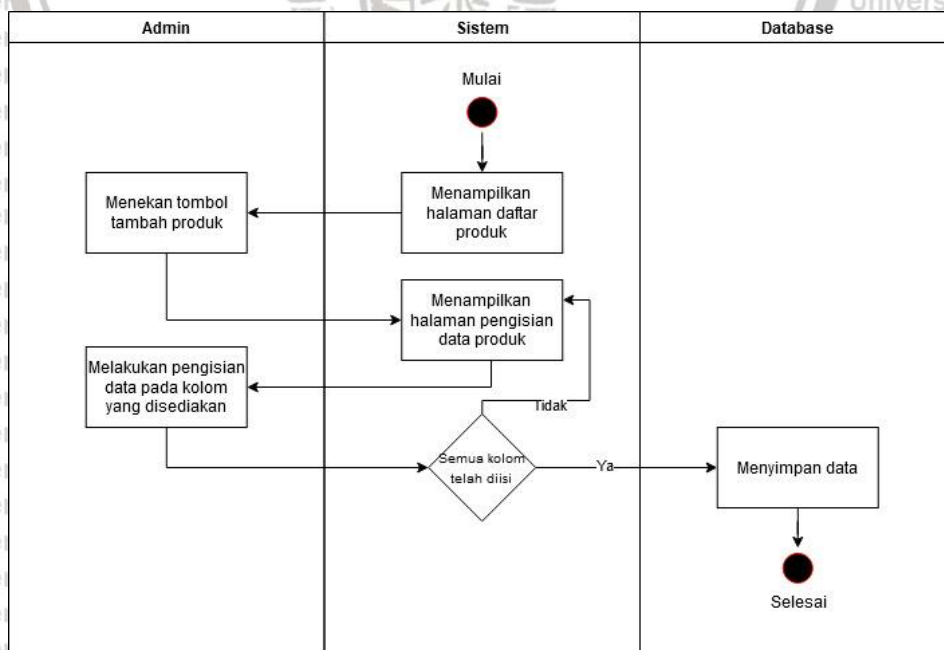
Activity diagram Informasi produk menjelaskan alur dari aktivitas aktor *Admin* saat melihat data produk pada sistem informasi khusus admin. Dimulai dengan *Admin* menekan tombol data produk, lalu sistem akan mengambil data produk dari *database* untuk ditampilkan di halaman daftar produk sistem khusus *admin*. Gambar 4.14 akan menjelaskan bentuk dari *activity* diagram melihat data produk.



Gambar 4.14 Activity Diagram Melihat Data Produk

4.6.13 Activity Diagram Menambah Data Produk

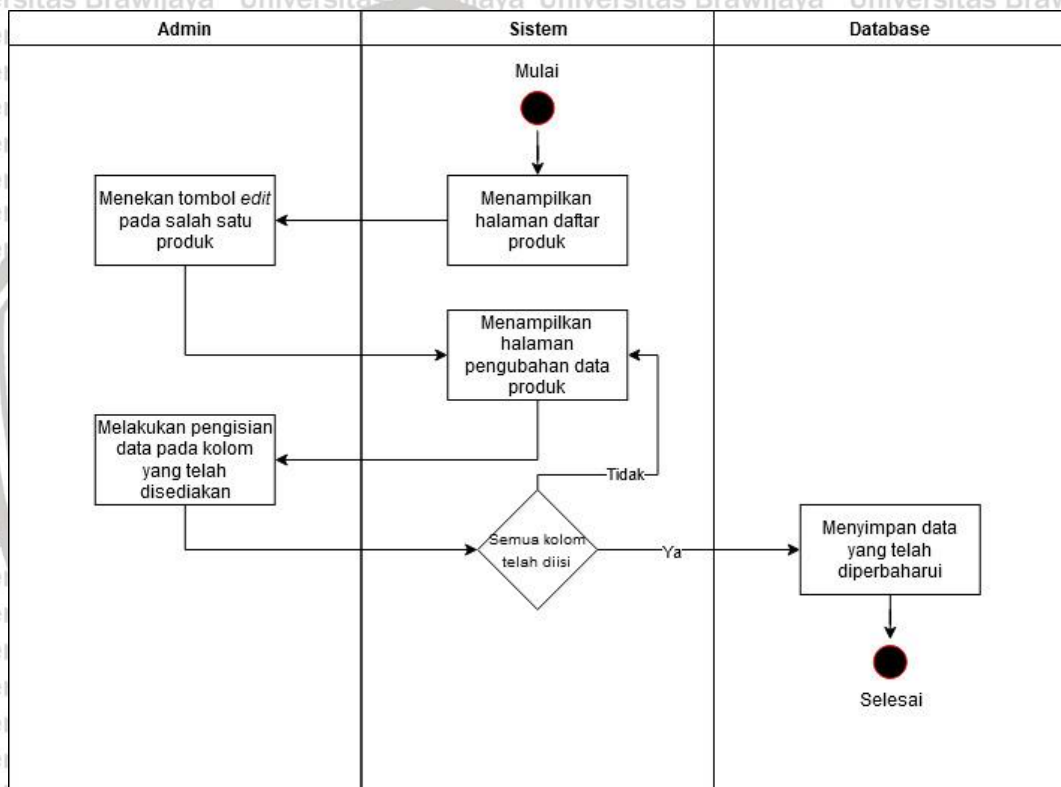
Activity diagram menambah data produk menjelaskan alur dari aktor *Admin* saat melakukan penambahan data produk. Dimulai dengan sistem menampilkan halaman daftar produk, lalu *Admin* menekan tombol tambah produk. Kemudian sistem akan menampilkan halaman pengisian data produk untuk dilakukan pengisian data produk oleh *Admin*. Jika semua kolom tidak diisi, maka sistem akan memberikan peringatan untuk mengisi semua kolom. Jika semua kolom telah diisi, maka data produk baru akan dimasukan kedalam *database*. Gambar 4.15 akan menjelaskan bentuk dari *activity diagram* menambah data produk.



Gambar 4.15 Activity Diagram Menambah Data Produk

4.6.14 Activity Diagram Mengubah Data Produk

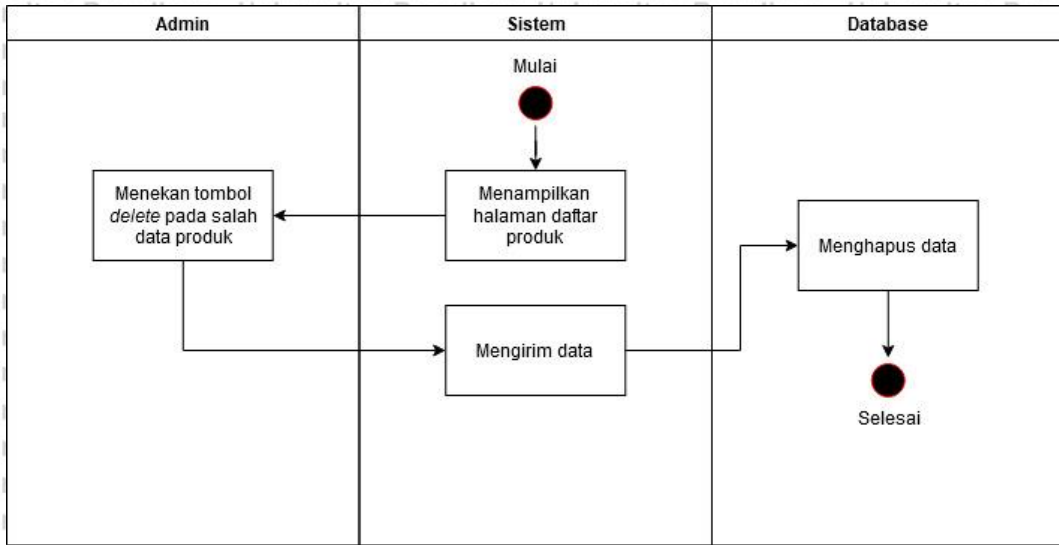
Activity diagram mengubah data produk menjelaskan alur dari aktivitas aktor *Admin* saat melakukan perubahan pada data produk yang sudah ada. Dimulai dengan sistem menampilkan halaman daftar produk, lalu *Admin* menekan tombol *edit* pada salah satu data produk. Kemudian sistem akan menampilkan halaman pengubahan data untuk dilakukan pengisian data baru oleh *Admin*. Jika semua kolom tidak diisi, maka sistem akan memberikan peringatan untuk mengisi semua kolom. Jika semua kolom telah diisi, maka data produk yang telah diubah akan dimasukkan kedalam *database*. Gambar 4.16 akan menjelaskan bentuk dari *activity* diagram mengubah data produk.



Gambar 4.16 Activity Diagram Mengubah Data Produk

4.6.15 Activity Diagram Menghapus Data Produk

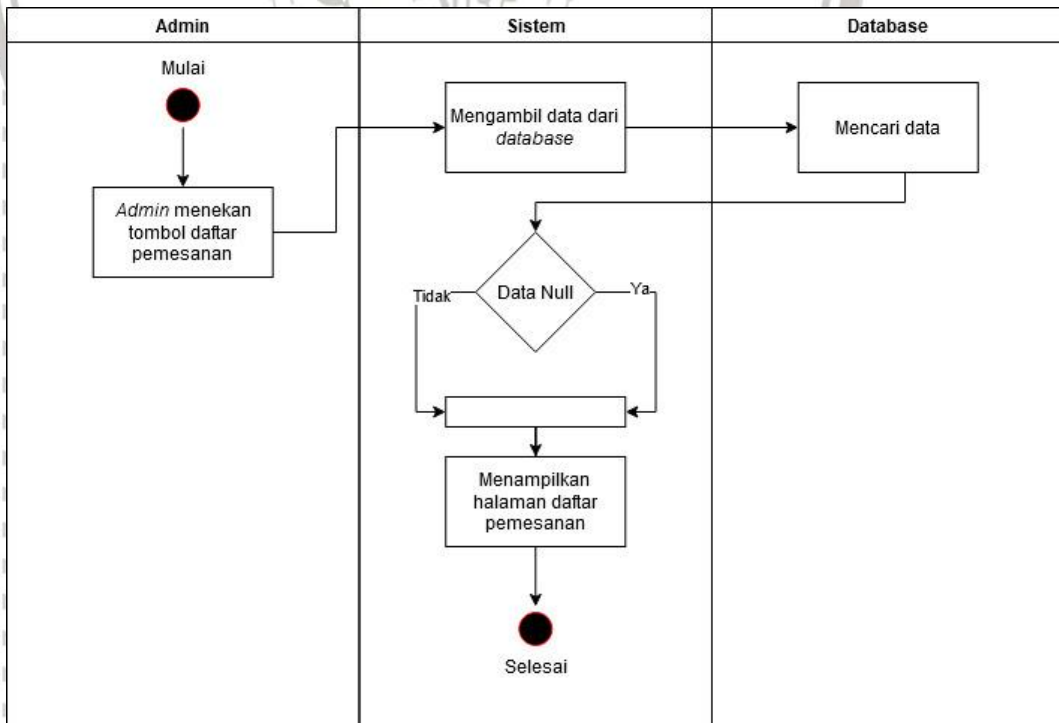
Activity diagram menghapus data produk menjelaskan alur dari aktivitas aktor *Admin* saat melakukan penghapusan data produk. Dimulai dengan sistem menampilkan halaman daftar produk, lalu *Admin* akan menekan tombol *delete* pada salah satu data produk. Kemudian sistem akan menghapus data produk data produk. Gambar 4.17 akan menjelaskan bentuk dari *activity* diagram menghapus data produk.



Gambar 4.17 Activity Diagram Menghapus Data Produk

4.6.16 Activity Diagram Informasi Pemesanan

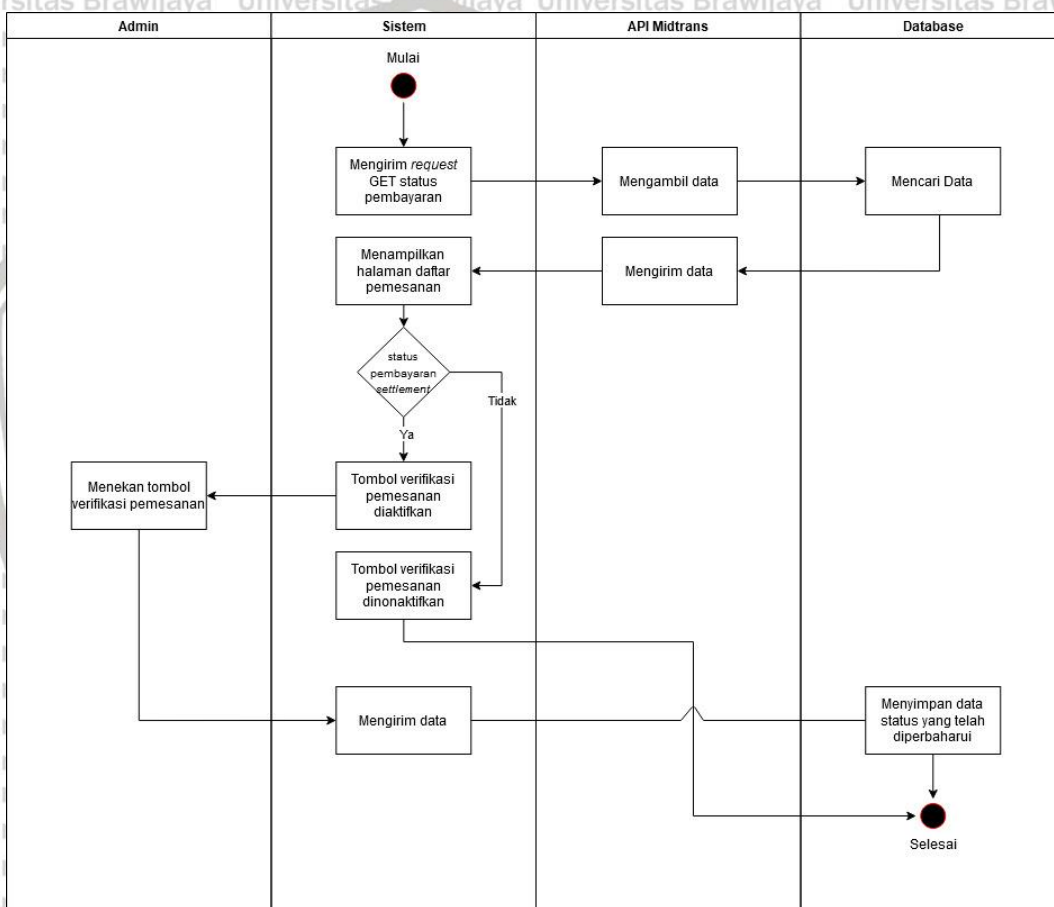
Activity diagram Informasi pemesanan menjelaskan alur dari aktivitas aktor *Admin* saat ingin melihat data pemesanan. Dimulai dengan *Admin* menekan tombol daftar pemesanan pada sistem informasi khusus admin, lalu sistem akan mengambil data dari *database* untuk ditampilkan pada halaman daftar pemesanan. Jika *User* belum membuat pemesanan, maka tidak ada data yang ditampilkan. Gambar 4.18 akan menjelaskan activity diagram melihat data pemesanan.



Gambar 4.18 Activity Diagram Melihat Data Pemesanan

4.6.17 Activity Diagram Verifikasi Pemesanan

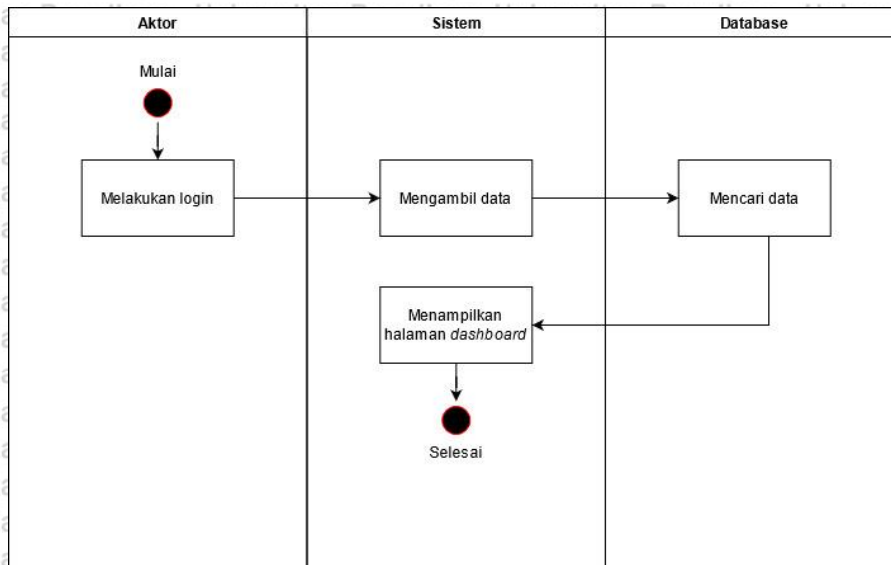
Activity diagram verifikasi pemesanan menjelaskan alur dari aktivitas aktor Admin saat melakukan verifikasi pemesanan. Dimulai dengan sistem mengirimkan request GET pada API Midtrans untuk mengambil data status pembayaran. Data status pembayaran digunakan untuk menyeleksi pemesanan yang akan diverifikasi. Jika status pemesanan adalah *settlement* maka tombol verifikasi pemesanan pada salah satu data pemesanan akan diaktifkan, Jika status pemesanan belum *settlement* atau masih berstatus *pending*, maka tombol verifikasi pemesanan akan dinonaktifkan. Gambar 4.19 akan menjelaskan bentuk dari activity diagram verifikasi pemesanan.



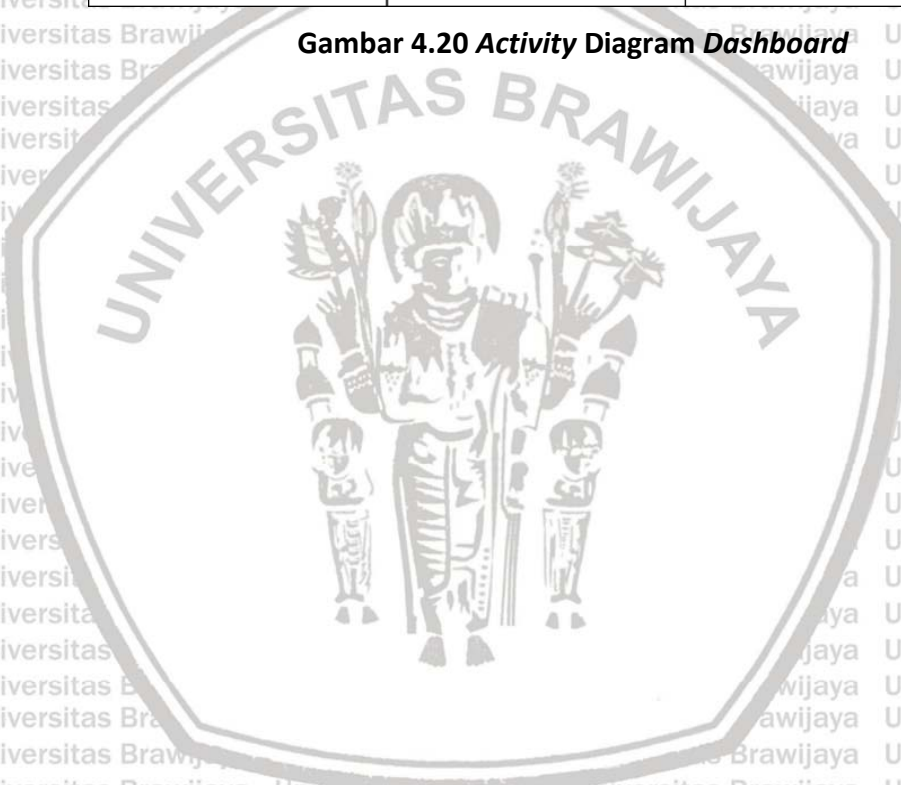
Gambar 4.19 Activity Diagram Verifikasi Pemesanan

4.6.18 Activity Diagram Dashboard

Activity diagram dashboard menjelaskan alur dari aktivitas aktor admin saat melakukan akses pada halaman dashboard sistem informasi Admin. Diawali dengan aktor Admin melakukan login, setelah login berhasil, sistem akan mengambil data dari database untuk ditampilkan pada halaman dashboard. Setelah itu, sistem akan menampilkan halaman dashboard pada aktor Admin.



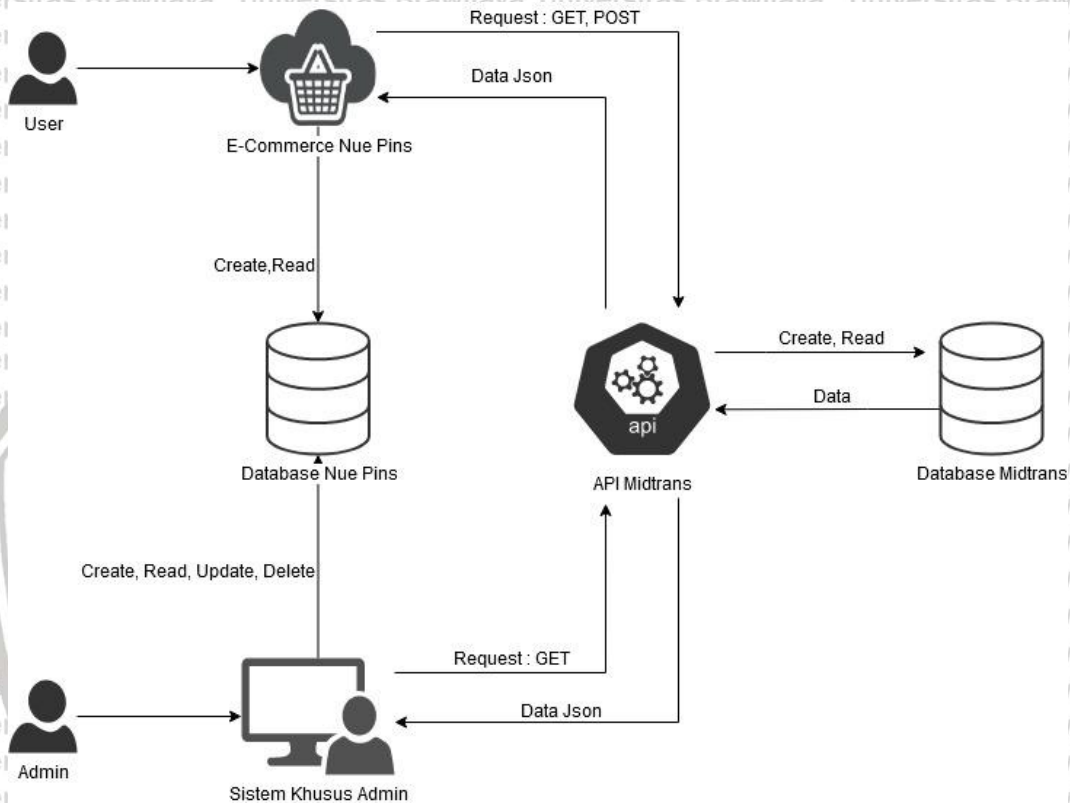
Gambar 4.20 Activity Diagram Dashboard



BAB 5 PERANCANGAN SISTEM

5.1 Perancangan Arsitektur Sistem

Arsitektur sistem yang akan dibuat terdiri dari 5 komponen perangkat yang saling berinteraksi, diantaranya *website* toko *online* Nue Pins, *website* khusus *Admin*, *API Midtrans*, *database Midtrans*, dan *database* Nue Pins. Gambar 5.1 akan menjelaskan struktur perancangan arsitektur sistem.



Gambar 5.1 Arsitektur Sistem

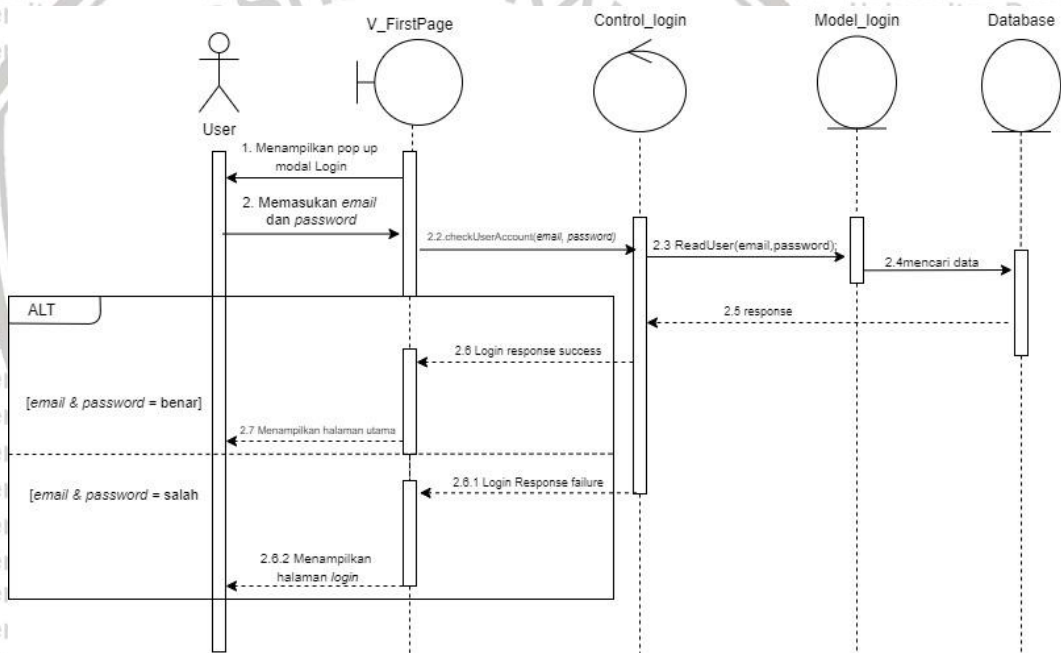
Gambar 5.1 menjelaskan alur interaksi dari komponen perangkat yang akan digunakan. Secara garis besar rancangan arsitektur yang dikembangkan antara dua aktor memiliki kesamaan. Namun terdapat beberapa perbedaan, diantaranya pada *request* ke *API Midtrans* dan hak akses pada *database* Nue Pins. *Request GET* dan *POST* yang dilakukan oleh komponen perangkat toko *online* Nue Pins, digunakan untuk mengirim data pemesanan dan mengambil data kode pembayaran. Sementara untuk sistem khusus *admin*, *request GET* dipergunakan untuk mengambil status pembayaran. Hak akses *create* dan *read* kepada *database* Nue Pins yang dilakukan oleh perangkat toko *online* Nue Pins, digunakan untuk membuat akun, membuat pemesanan, dan membaca data produk. Sementara hak akses *create, read, update, delete* yang dilakukan oleh perangkat sistem khusus *Admin*, dipergunakan untuk membuat, membaca, mengubah, menghapus data produk, menghapus, dan mengubah status pemesanan. *Database Midtrans*

merupakan *database* yang disediakan oleh perusahaan *midtrans* untuk menampung data pemesanan.

5.2 Perancangan *Sequence Diagram*

Perancangan *sequence diagram* menjelaskan hubungan antara aktor dengan sistem. Pada pemodelan *sequence diagram*, terdapat urutan dari proses – proses yang dilaksanakan maupun dari sebuah aktor ke sistem atau dari sistem itu sendiri. Terdapat komponen – komponen yang ada pada pemodelan *sequence diagram*, yaitu objek, yang menggambarkan kelas dan berfungsi untuk mendemonstrasikan bagaimana sebuah objek berperilaku terhadap sistem. *Activation boxes*, yang menggambarkan waktu yang diperlukan sebuah objek untuk menyelesaikan *task*. Aktor, yang menggambarkan pengguna yang berinteraksi dengan sistem, dan *lifeline* yang menggambarkan aktivitas dari sebuah objek. Alur interaksi objek pada *sequence diagram* didasarkan pada rancangan *activity diagram* yang telah dibuat sebelumnya.

5.2.1 *Sequence Diagram Login User*

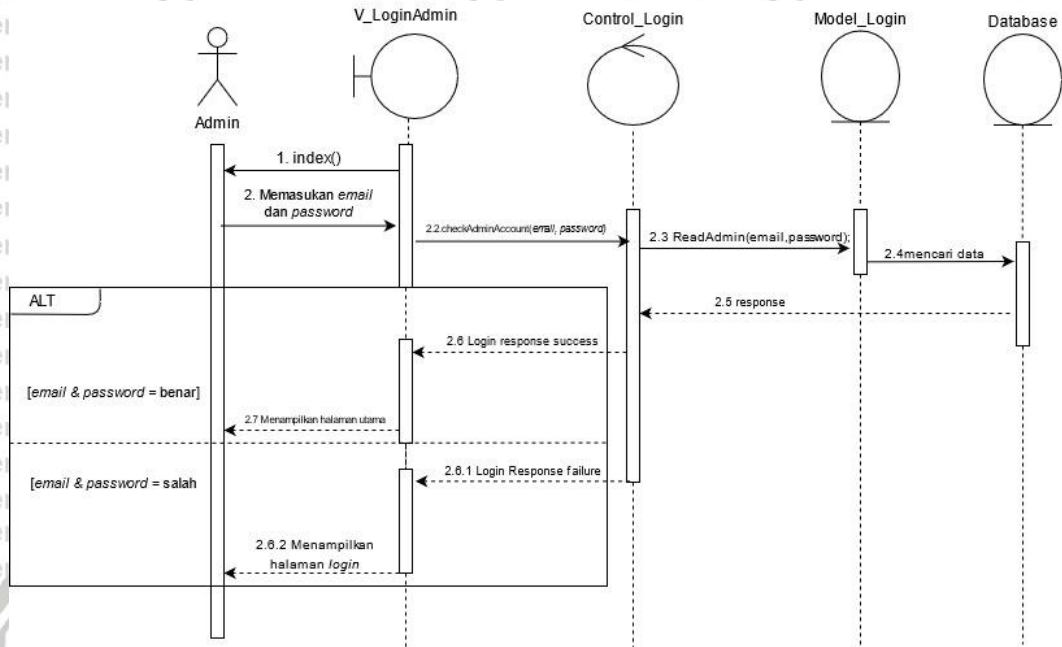


Gambar 5.2 *Sequence Diagram Login User*

Gambar 5.2 menunjukkan bentuk *sequence diagram* pada fungsi *login user*. Proses *login* diawali dengan aktor *User* memasukan *email* dan *password* pada *form input* yang disediakan pada halaman *login*. Kemudian sistem akan melakukan proses *checkUserAccount* yang berparameter *email* dan *password* yang telah dimasukan. Pada proses *checkUserAccount* terdapat arahan untuk melakukan pencarian data *User* dari *database*. Jika *email* dan *password* yang dimasukan oleh *User* terdapat kesamaan pada data akun *User* di *database*, maka sistem akan memberikan notifikasi *login* berhasil dan *User* akan diarahkan ke halaman utama.

Jika tidak, maka *User* akan diberikan notifikasi kesalahan dan tetap diarahkan ke halaman *Login*.

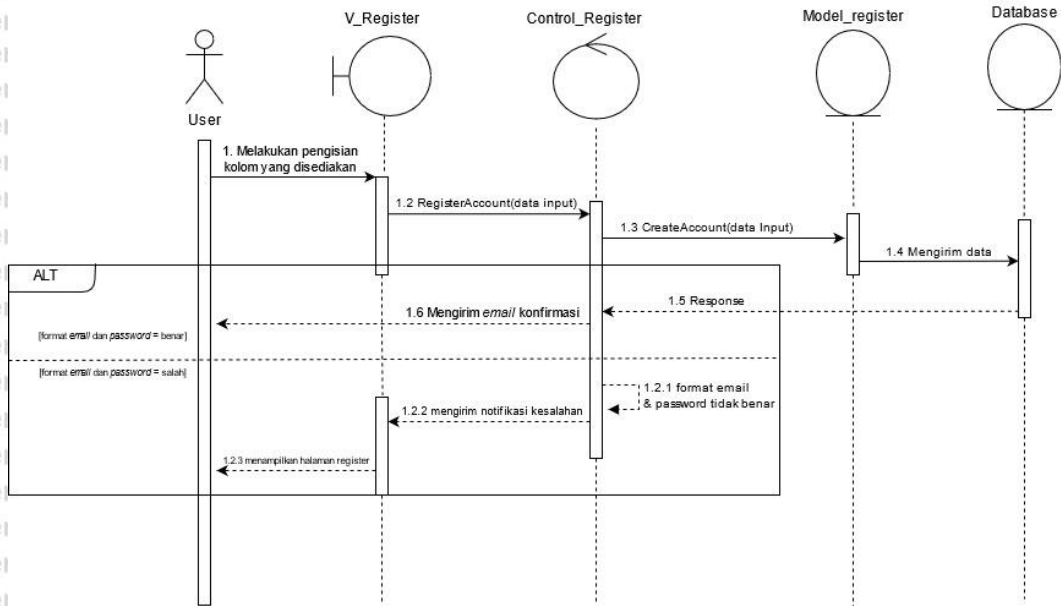
5.2.3 Sequence Diagram Login Admin



Gambar 5.3 Sequence Diagram Login Admin

Gambar 5.3 menunjukkan bentuk dari *sequence* diagram pada fungsi *login Admin*. Proses *login* untuk aktor *Admin*, diawali dengan *Admin* memasukkan *email* dan *password* pada *form input* yang disediakan pada halaman *login*. Kemudian sistem akan melakukan proses *checkAdminAccount* yang berparameter *email* dan *password* yang telah dimasukan. Pada proses *checkAdminAccount* terdapat arahan untuk melakukan pencarian data *Admin* dari *database*. Jika *email* dan *password* yang dimasukan oleh *Admin* terdapat kesamaan pada data akun *Admin* di *database*, maka sistem akan memberikan notifikasi *login* berhasil dan *Admin* akan diarahkan ke halaman utama sistem informasi *Admin*. Jika tidak, maka *Admin* akan diberikan notifikasi kesalahan dan tetap diarahkan ke halaman *Login*.

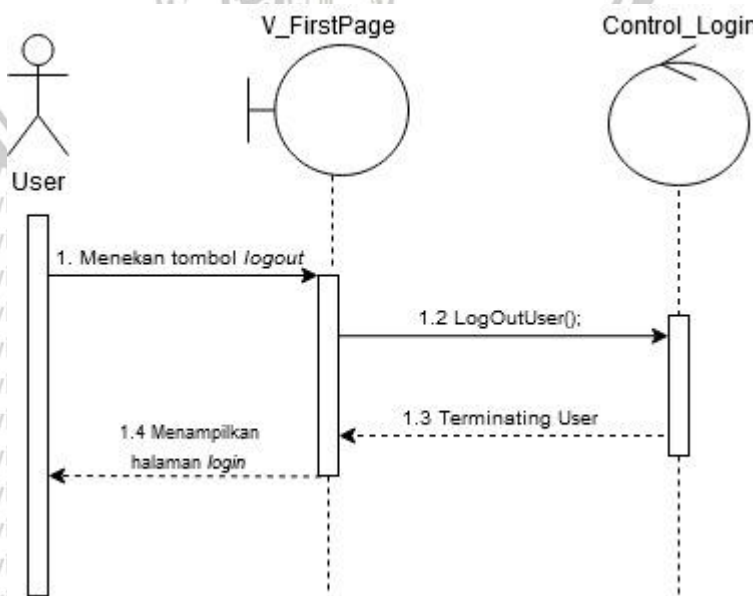
5.2.3 Sequence Diagram Register



Gambar 5.4 Sequence Diagram Register

Gambar 5.4 menunjukkan bentuk dari *sequence diagram* pada fungsi *register*. Proses registrasi akun *User*, diawali dengan aktor *User* mengisikan *form input* yang telah disediakan pada halaman register. Kemudian sistem akan melakukan proses *saveAccount* yang berparameter data yang telah diisi oleh *User* pada *form input*. Terdapat pengecekan terhadap pengisian *form input* yang disediakan, Jika *form input* telah diisi dengan benar, maka *User* akan dikirimkan konfirmasi ke *email* yang telah *diinput* oleh *User*. Jika tidak, maka *User* akan dikirimkan notifikasi kesalahan, dan akan diarahkan kembali pada halaman registrasi.

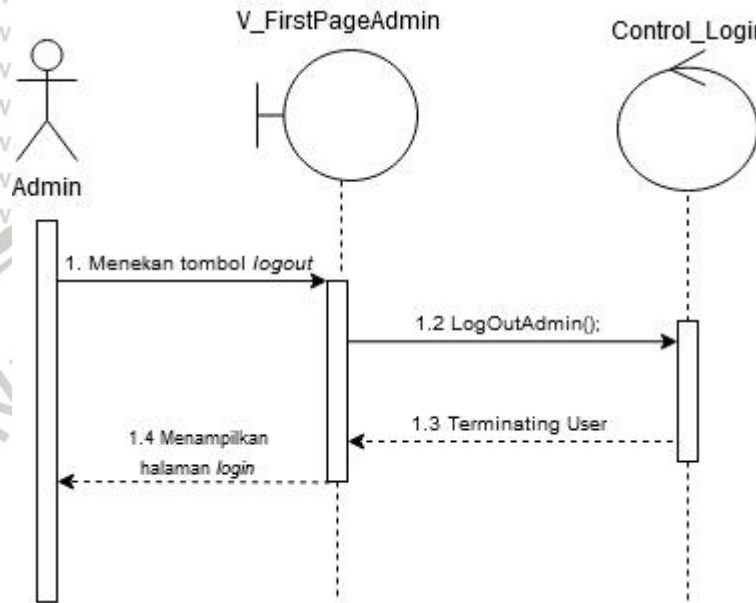
5.2.4 Sequence Diagram Logout User



Gambar 5.5 Sequence Diagram Logout User

Gambar 5.5 menunjukkan bentuk dari *sequence* diagram pada fungsi *logout* akun *User*. Proses *logout* akun *User*, diawali dengan aktor *User* menekan tombol *logout* yang telah disediakan oleh sistem. Kemudian sistem akan melakukan proses *LogginOutUser()*. Pada proses *LogginOutUser()*, terdapat arahan untuk melakukan terminasi dari akun *User*. Kemudian, sistem akan mengeluarkan akun *User*, dan mengarahkan *User* ke halaman *login*.

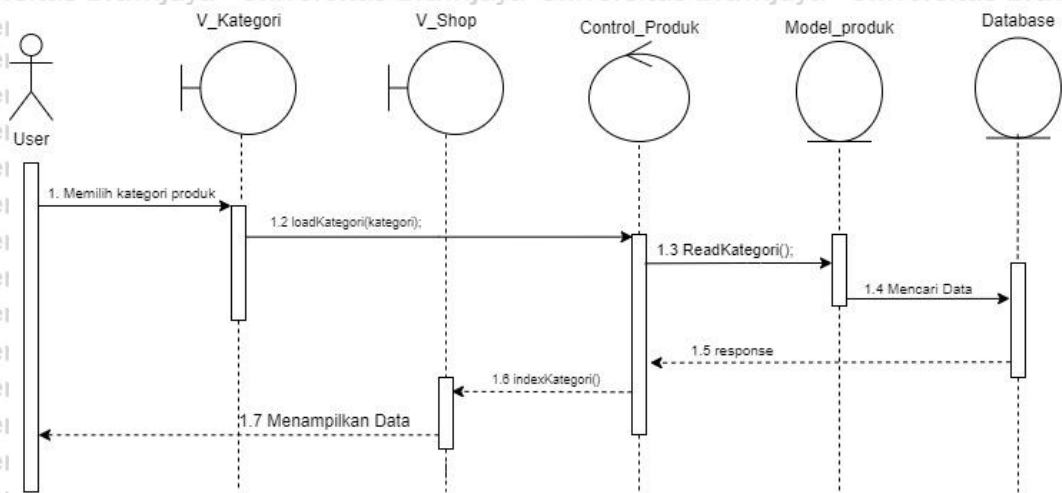
5.2.5 Sequence Diagram Logout Admin



Gambar 5.6 Sequence Diagram Logout Admin

Gambar 5.6 menunjukkan bentuk dari *sequence* diagram pada fungsi *logout* *Admin*. Proses *logout* pada akun aktor *Admin*, diawali dengan *Admin* menekan tombol *logout* yang telah disediakan pada sistem informasi khusus *Admin*. Kemudian sistem akan melakukan proses *LogginOutAdmin()*. Pada proses *LogginOutAdmin()* terdapat arahan untuk melakukan terminasi akun *Admin* dari sistem. Setelah akun *Admin* dikeluarkan dari sistem informasi khusus *Admin*, sistem akan mengarahkan *Admin* ke halaman *login*.

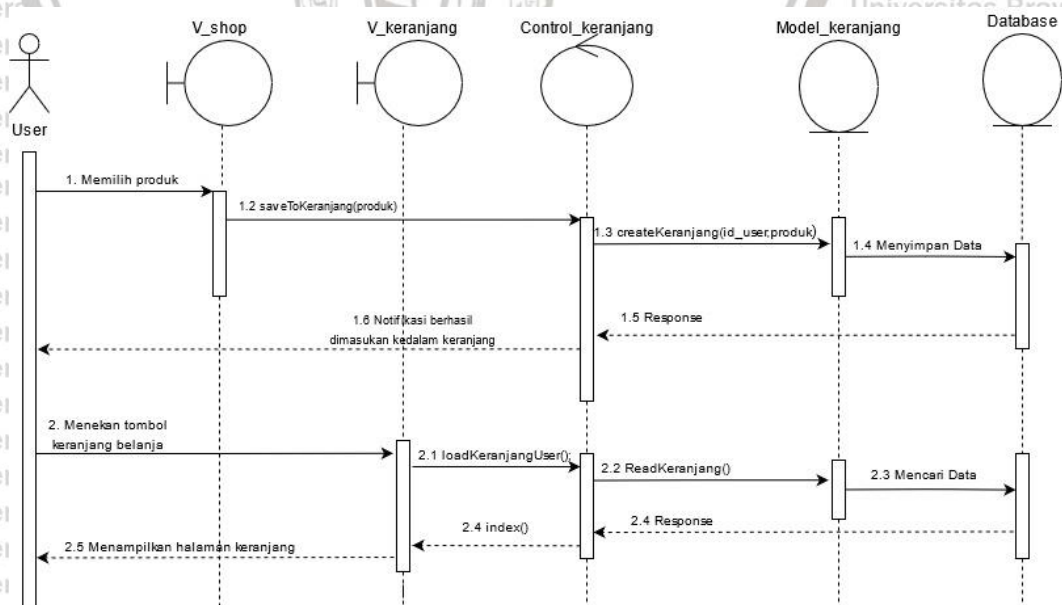
5.2.6 Sequence Diagram Menampilkan Produk



Gambar 5.7 Sequence Diagram Menampilkan Produk

Gambar 5.7 menunjukkan bentuk dari *sequence* diagram pada fungsi penampilan produk berdasarkan kategori yang dipilih oleh aktor *User*. Proses penampilan produk berdasarkan kategori, diawali dengan aktor *User* memilih kategori yang diinginkan. Kemudian sistem akan melakukan fungsi *loadDataProduk* yang berparameter kategori yang telah dipilih. Pada proses *loadDataProduk*, terdapat arahan untuk mengambil data kategori dari *database*. Data tersebut akan dikirimkan ke sistem, lalu sistem akan melakukan proses *setData* yang berparameter data dari *database*. Data tersebut akan dimasukkan pada masing – masing *variable* pada halaman utama. Lalu sistem akan menampilkan data produk dengan menggunakan fungsi *showData*.

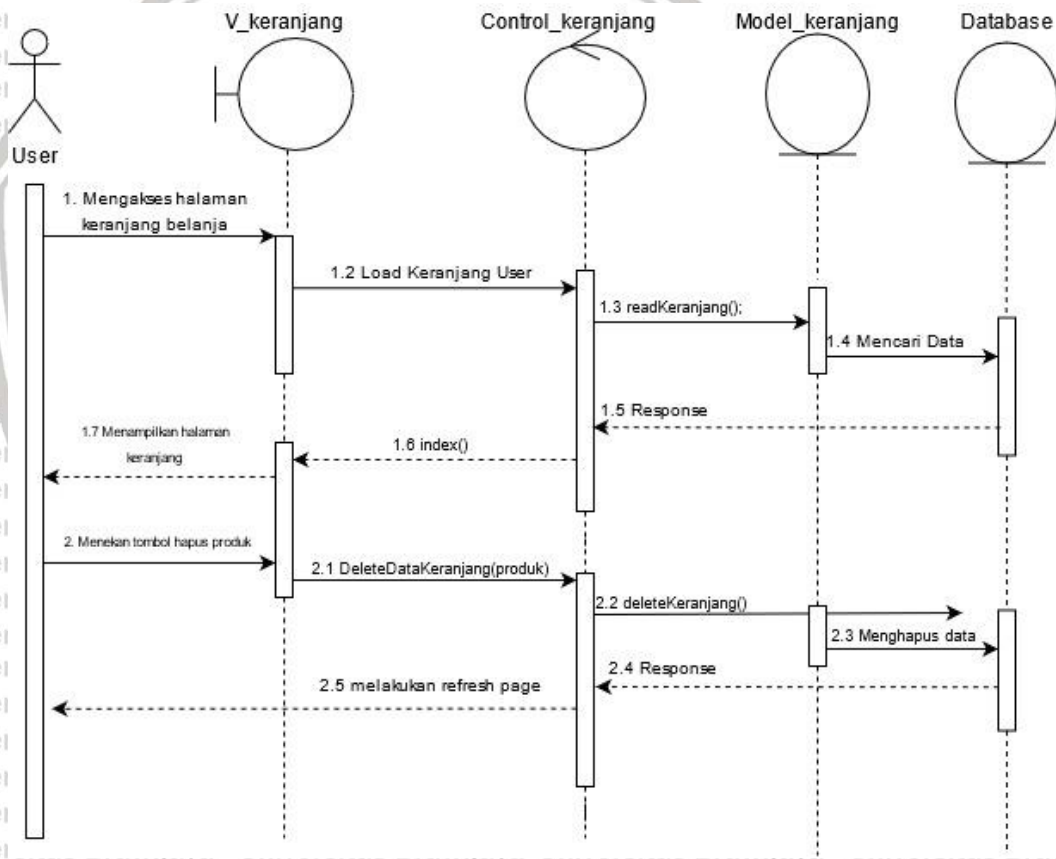
5.2.7 Sequence Diagram Keranjang Belanja



Gambar 5.8 Sequence Diagram Keranjang Belanja

Gambar 5.8 menunjukkan bentuk dari *sequence* diagram pada fungsi keranjang belanja *User*. Proses untuk memasukan produk ke dalam keranjang belanja, diawali dengan aktor *User* memilih produk terlebih dahulu. Kemudian sistem akan melakukan proses *saveToKeranjang* yang berparameter produk yang telah dipilih. Pada proses *saveToKeranjang*, terdapat arahan untuk menyimpan data kedalam *database*. Setelah data produk berhasil disimpan, lalu sistem akan memberikan notifikasi bahwa produk berhasil dimasukan ke keranjang. Proses selanjutnya, aktor *User* akan menekan tombol keranjang belanja dan sistem akan melakukan fungsi *loadKeranjangUser()*. Pada proses *loadKeranjangUser()*, terdapat arahan untuk mengambil data keranjang belanja berdasarkan akun *User*. Kemudian data akan dimasukan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*.

5.2.8 Sequence Diagram Menghapus Produk

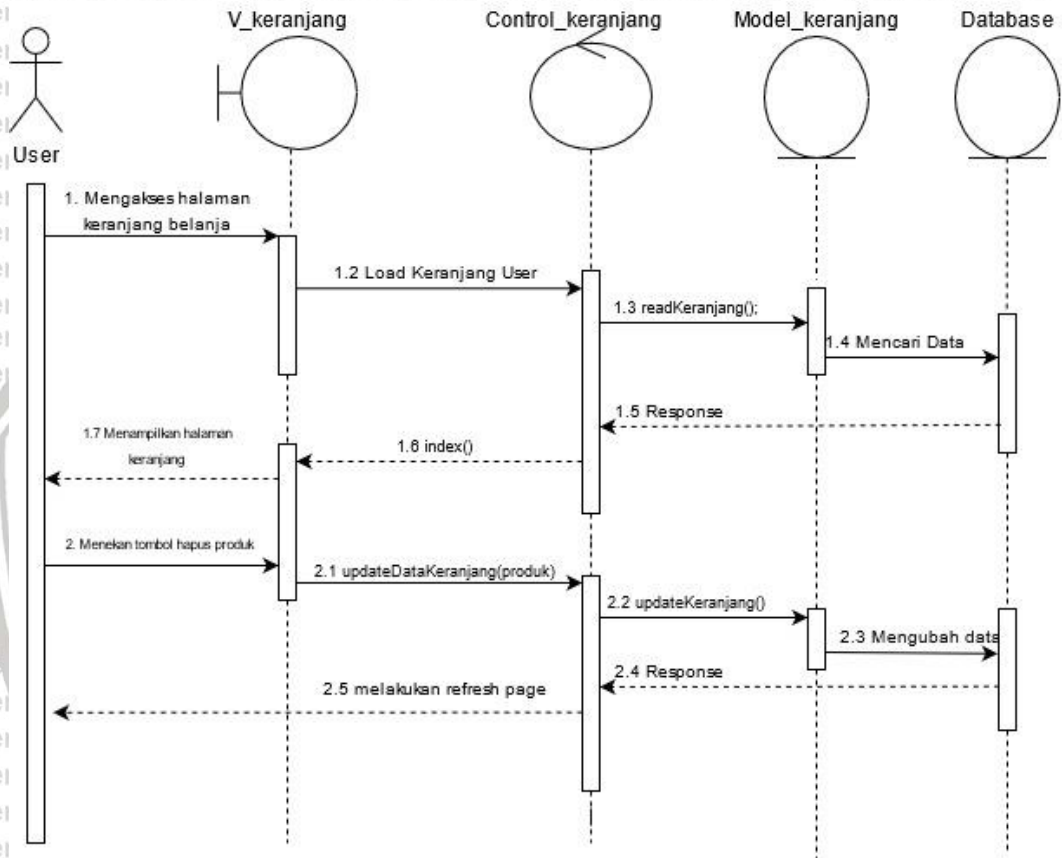


Gambar 5.9 Sequence Diagram Menghapus Produk

Gambar 5.9 menunjukkan bentuk *sequence* diagram pada fungsi penghapusan produk dari keranjang belanja *User*. Proses penghapusan produk pada keranjang belanja diawali dengan aktor *User* melakukan akses ke halaman keranjang belanja terlebih dahulu. Kemudian sistem akan melakukan proses *loadDataKeranjang*. Pada proses *loadDataKeranjang*, terdapat arahan untuk mengambil data keranjang belanja *User* dari *database*. Kemudian data akan dimasukan kedalam

variable dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Proses selanjutnya, *User* menekan tombol hapus produk. Lalu sistem akan melakukan proses *deleteDataKeranjang* berparameter produk yang telah dipilih. Pada proses *deleteDataKeranjang* terdapat arahan untuk melakukan hapus data dari *database*. Kemudian sistem akan melakukan proses *refreshPage* untuk memuat ulang halaman keranjang belanja.

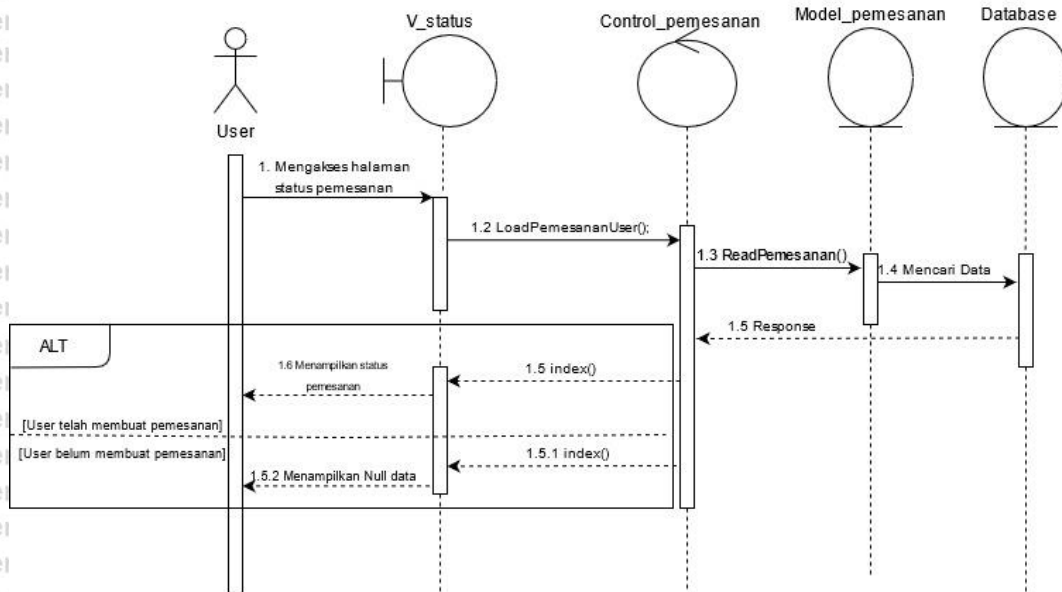
5.2.9 Sequence Diagram Menentukan Jumlah Pembelian



Gambar 5.10 Sequence Diagram Menentukan Jumlah Pembelian

Gambar 5.10 menjelaskan bentuk *sequence* Diagram pada fungsi untuk menentukan jumlah pembelian pada salah satu produk di keranjang belanja. Proses penentuan jumlah pembelian produk pada keranjang belanja diawali dengan aktor *User* melakukan akses ke halaman keranjang belanja terlebih dahulu. Kemudian sistem akan melakukan proses *loadDataKeranjang*. Pada proses *loadDataKeranjang*, terdapat arahan untuk mengambil data keranjang belanja *User* dari *database*. Kemudian data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Proses selanjutnya, *User* menekan tombol kurang atau tambah pada salah satu produk di keranjang belanja. Lalu sistem akan melakukan proses *updateDataKeranjang* berparameter jumlah, dan produk yang telah dipilih. Pada proses *updateDataKeranjang* terdapat arahan untuk melakukan

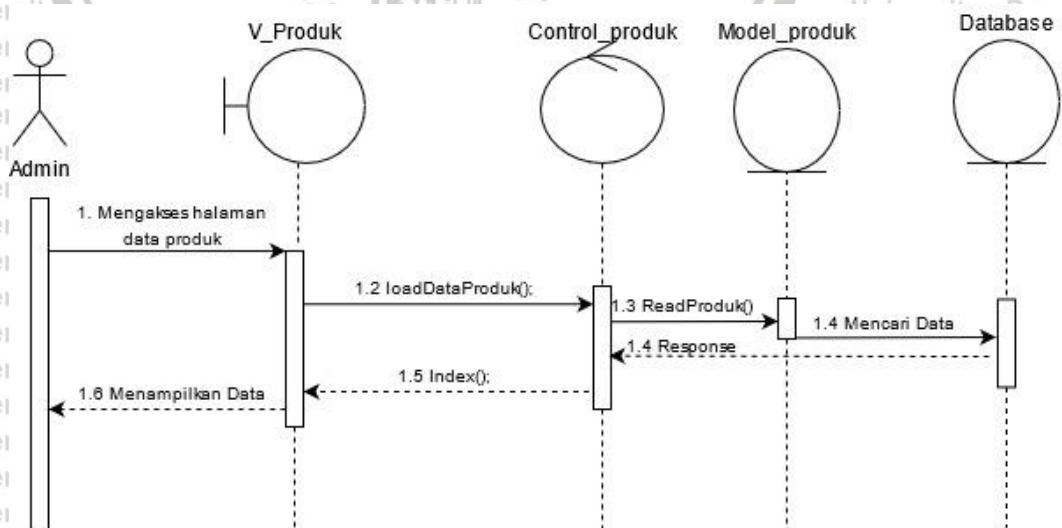
5.2.11 Sequence Diagram Track Pemesanan



Gambar 5.12 Sequence Diagram Track Pemesanan

Gambar 5.12 menunjukkan bentuk *sequence diagram track pemesanan* pada fungsi untuk melihat status pemesanan yang telah dibuat oleh *User*. Proses untuk melihat status pemesanan, diawali dengan aktor *User* mengakses halaman status pemesanan terlebih dahulu. Kemudian sistem akan melakukan proses *loadPemesanan()* berdasarkan akun *User*. Pada proses *loadPemesanan()*, terdapat arahan untuk mengambil data dari *database*. Kemudian data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Jika akun *User* belum membuat pemesanan, maka tidak ada data yang ditampilkan.

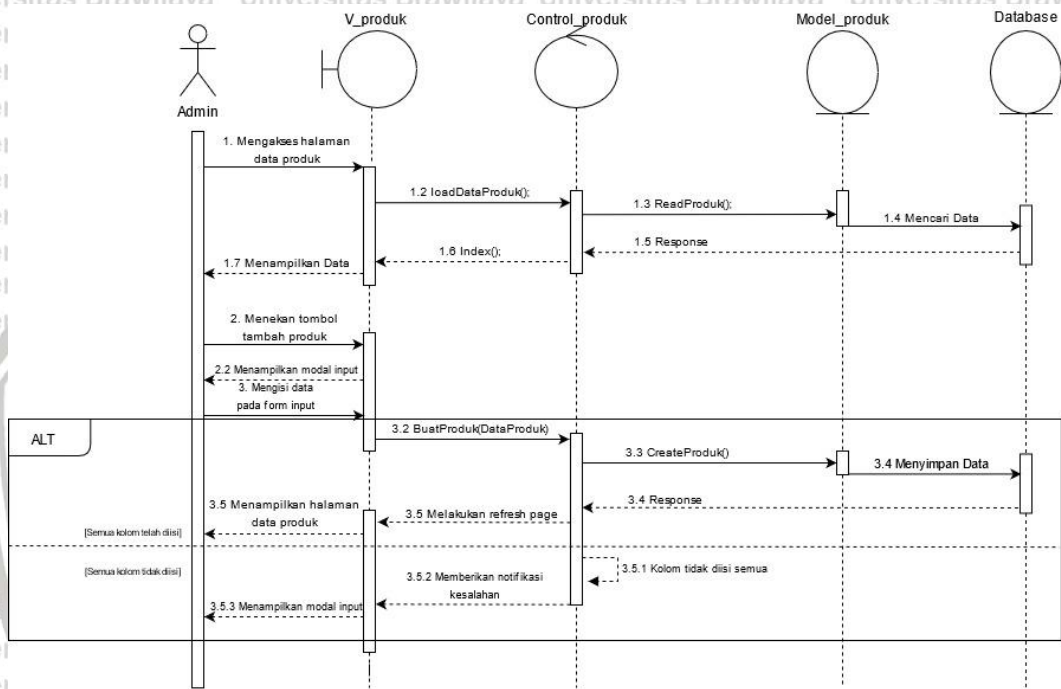
5.2.12 Sequence Diagram Informasi Produk



Gambar 5.13 Sequence Diagram Melihat Data Produk

Gambar 5.13 menunjukkan bentuk *sequence* diagram pada fungsi informasi produk di sistem informasi khusus *Admin*. Diawali dengan aktor *Admin* mengakses halaman data produk terlebih dahulu. Kemudian sistem akan melakukan proses *loadDataProduk()*. Pada proses *loadDataProduk()*, terdapat arahan untuk mengambil data dari *database*. Kemudian data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*.

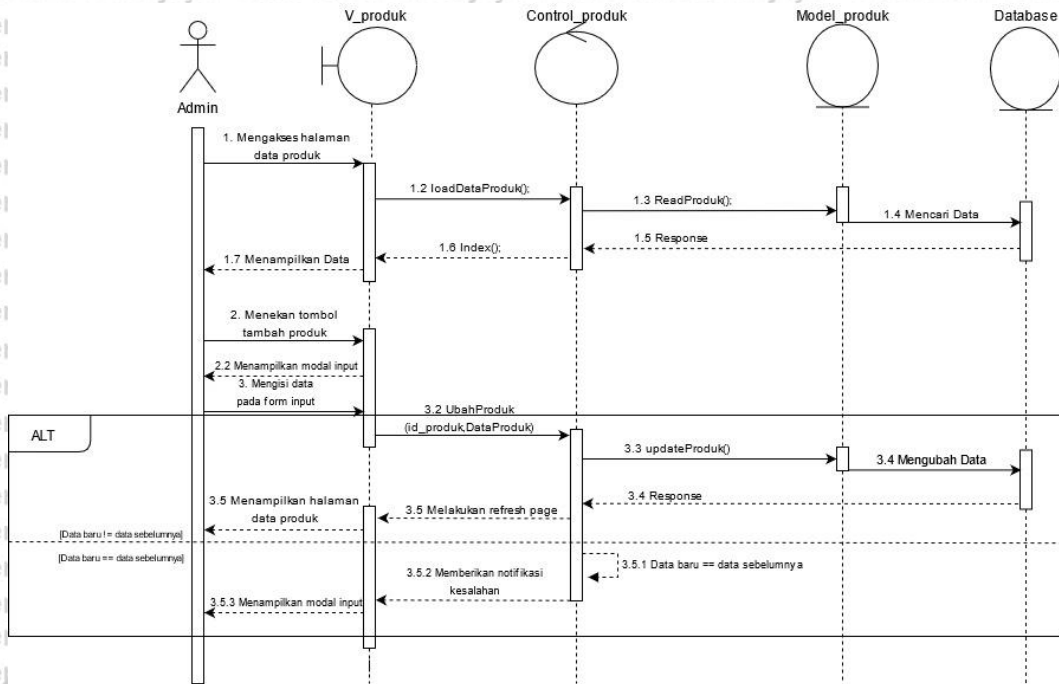
5.2.13 Sequence Diagram Menambah Data Produk



Gambar 5.14 Sequence Diagram Menambah Data Produk

Gambar 5.14 menunjukkan bentuk *sequence* diagram dari fungsi penambahan produk baru. Diawali dengan aktor *Admin* mengakses halaman data produk terlebih dahulu. Kemudian sistem akan melakukan proses *loadDataProduk()*. Pada proses *loadDataProduk()*, terdapat arahan untuk mengambil data dari *database*. Kemudian data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Proses selanjutnya *Admin* akan menekan tombol tambah produk. Kemudian sistem akan menampilkan *modal* yang berisikan *form input*. Setelah itu *Admin* akan melakukan pengisian pada *form input* yang telah disediakan. Kemudian sistem akan melakukan proses *buatProduk* yang berparameter data yang telah diinputkan pada *form*. Jika *form input* tidak diisi semua, maka sistem akan memberikan notifikasi kesalahan dan *modal* penambahan produk akan tetap ditampilkan. Jika *form input* telah diisikan semua, maka sistem akan menyimpan data kedalam *database*. Kemudian sistem akan melakukan proses *refreshPage()* untuk memuat ulang halaman data produk.

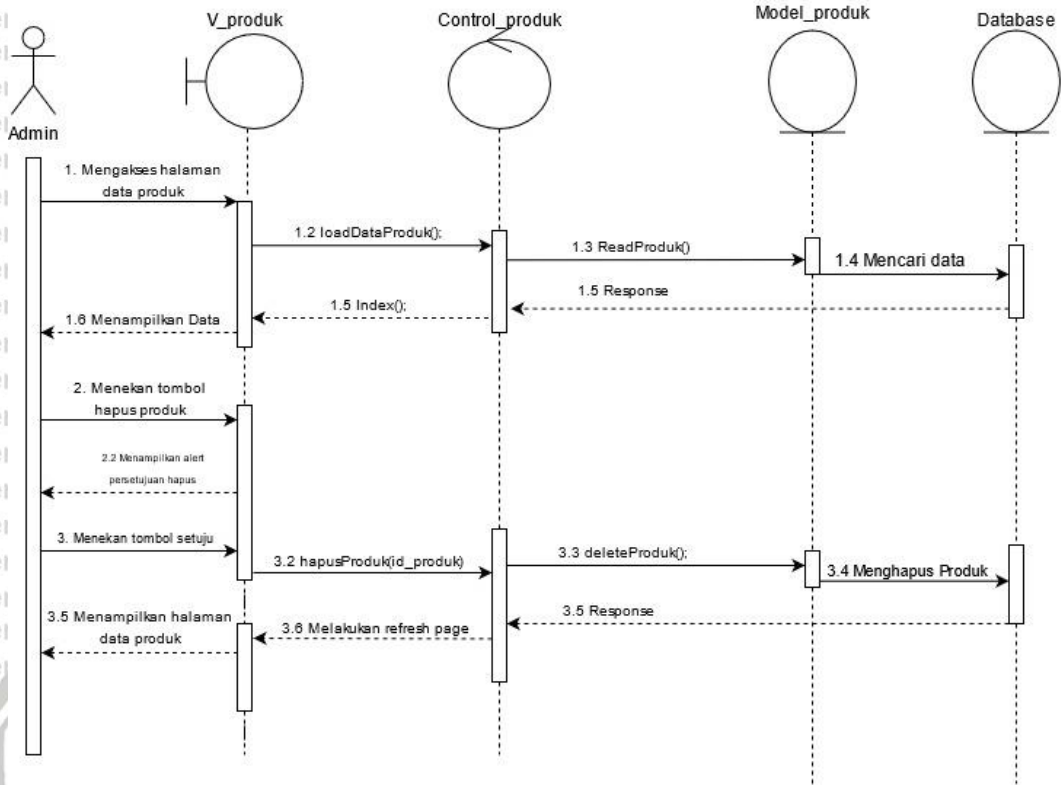
5.2.14 Sequence Diagram Mengubah Data Produk



Gambar 5.15 Sequence Diagram Mengubah Data Produk

Gambar 5.15 menunjukkan bentuk *sequence* diagram pada fungsi pengubahan data produk. Diawali dengan aktor *Admin* mengakses halaman data produk terlebih dahulu. Kemudian sistem akan melakukan proses *loadDataProduk()*. Pada proses *loadDataProduk()*, terdapat arahan untuk mengambil data dari *database*. Kemudian data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Proses selanjutnya *Admin* akan menekan tombol ubah pada salah satu produk. Kemudian sistem akan menampilkan *modal* yang berisikan *form input* yang berisi data sebelumnya. Setelah itu *Admin* akan melakukan perubahan data pada *form input*. Kemudian sistem akan melakukan proses *updateProduk* yang berparameter data yang telah diinputkan pada *form*. Jika data yang diisikan pada *form input* adalah data yang sama dengan data sebelumnya, maka sistem akan memberikan notifikasi kesalahan dan *modal* perubahan produk akan tetap ditampilkan. Jika *form input* telah diisikan semua, maka sistem akan mengubah data di *database*. Kemudian sistem akan melakukan proses *refreshPage()* untuk memuat ulang halaman data produk.

5.2.15 Sequence Diagram Menghapus Data Produk

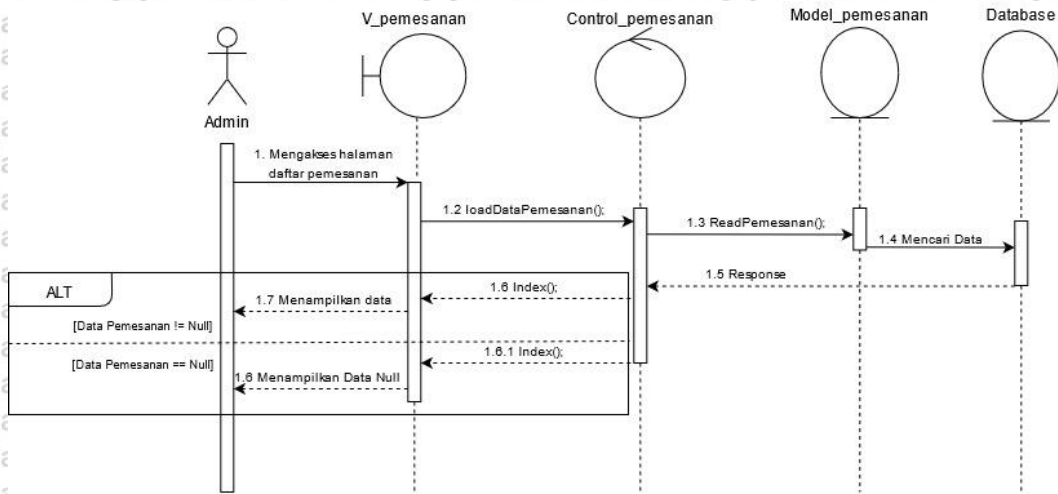


Gambar 5.16 Sequence Diagram Menghapus Data Produk

Gambar 5.16 menunjukkan bentuk *sequence* diagram pada fungsi penghapusan data produk. Diawali dengan aktor *Admin* mengakses halaman data produk terlebih dahulu. Kemudian sistem akan melakukan proses *loadDataProduk()*. Pada proses *loadDataProduk()*, terdapat arahan untuk mengambil data dari *database*. Kemudian data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Proses selanjutnya *Admin* akan menekan tombol hapus pada salah satu produk. Lalu sistem akan memberikan *alert* persetujuan untuk menghapus data produk. Setelah *Admin* menekan tombol setuju, maka sistem akan melakukan proses *deleteProduk* yang berparameter produk yang telah dipilih. Pada proses *deleteProduk*, terdapat arahan untuk menghapus data yang telah dipilih dari *database*. Setelah itu sistem akan melakukan proses *refreshPage()* untuk memuat ulang halaman daftar produk.



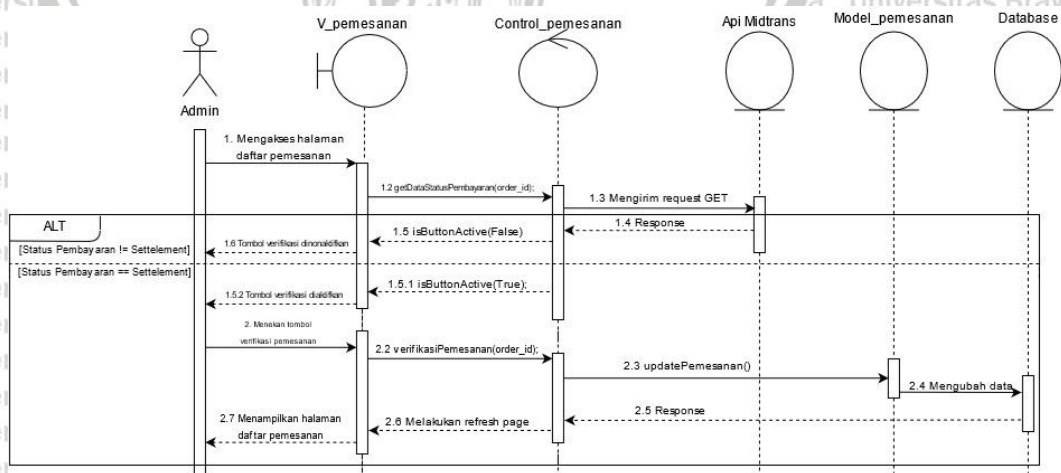
5.2.16 Sequence Diagram Informasi Pemesanan



Gambar 5.17 Sequence Diagram Melihat Data Pemesanan

Gambar 5.17 menunjukkan bentuk *sequence diagram* pada fungsi informasi data pemesanan. Diawali dengan aktor *Admin* mengakses halaman daftar pemesanan terlebih dahulu. Kemudian sistem akan melakukan proses *getDataPemesanan*. Pada proses *getDataPemesanan*, terdapat arahan untuk mengirimkan *request GET* pada *API Midtrans*. *Request GET* yang dikirimkan, digunakan untuk mengambil data pemesanan yang telah dibuat oleh *User*. Data yang dikirimkan oleh *API Midtrans* adalah data yang bertipe data *JSON*. Jika *User* telah membuat pemesanan, maka data akan dimasukkan kedalam *variable* dengan menggunakan proses *setData*. Setelah itu data akan ditampilkan dengan menggunakan fungsi *showData*. Jika *User* belum melakukan pemesanan, maka tidak ada data yang ditampilkan.

5.2.17 Sequence Diagram Verifikasi Pemesanan

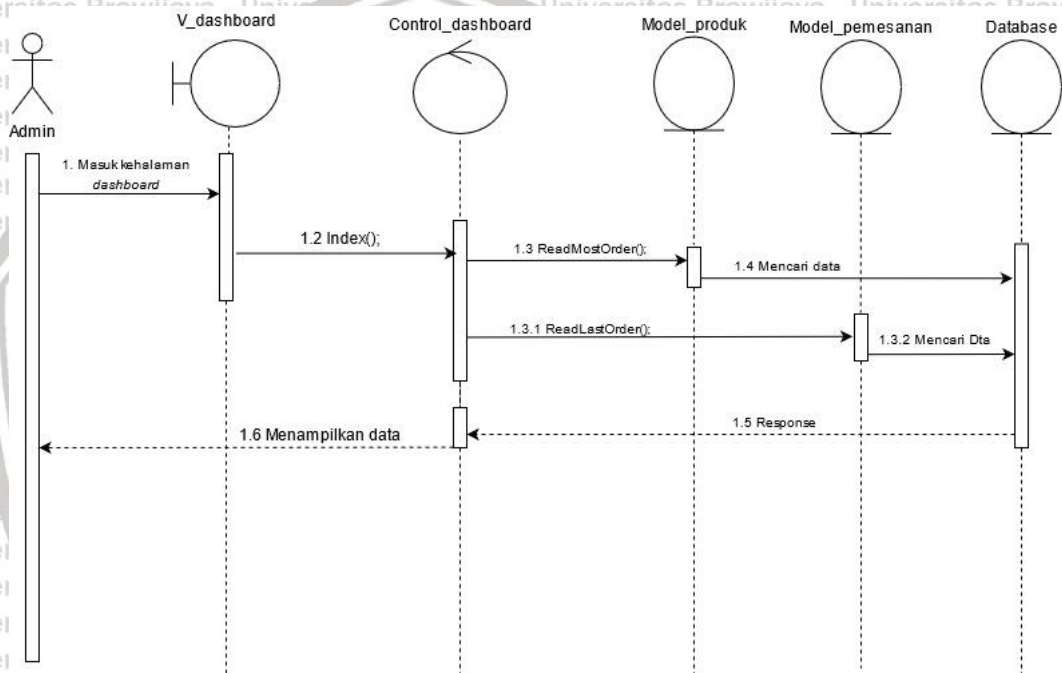


Gambar 5.18 Sequence Diagram Verifikasi Pemesanan

Gambar 5.18 menunjukkan bentuk *sequence diagram* pada fungsi verifikasi pemesanan. Proses verifikasi pemesanan diawali dengan aktor *Admin* mengakses

halaman daftar pemesanan terlebih dahulu. Setelah data status pemesanan diterima dari API *Midtrans*, maka terdapat kondisi dimana tombol verifikasi akan aktif atau tidak. Jika status pemesanan tidak sama dengan *settlement*, maka tombol verifikasi akan dinonaktifkan. Jika status pemesanan adalah *settlement*, maka tombol verifikasi akan diaktifkan, dan aktor *Admin* dapat menekan tombol tersebut. Setelah aktor *Admin* menekan tombol verifikasi pada salah satu produk, maka sistem akan melakukan proses *updateStatusPemesanan()*. Pada proses *updateStatusPemesanan()*, terdapat arahan untuk mengubah data status pemesanan *User*. Setelah *database* memberikan *response* bahwa data berhasil diubah, maka sistem akan melakukan proses *refreshPage()* untuk memuat ulang halaman daftar pemesanan.

5.2.18 Sequence Diagram Dashboard



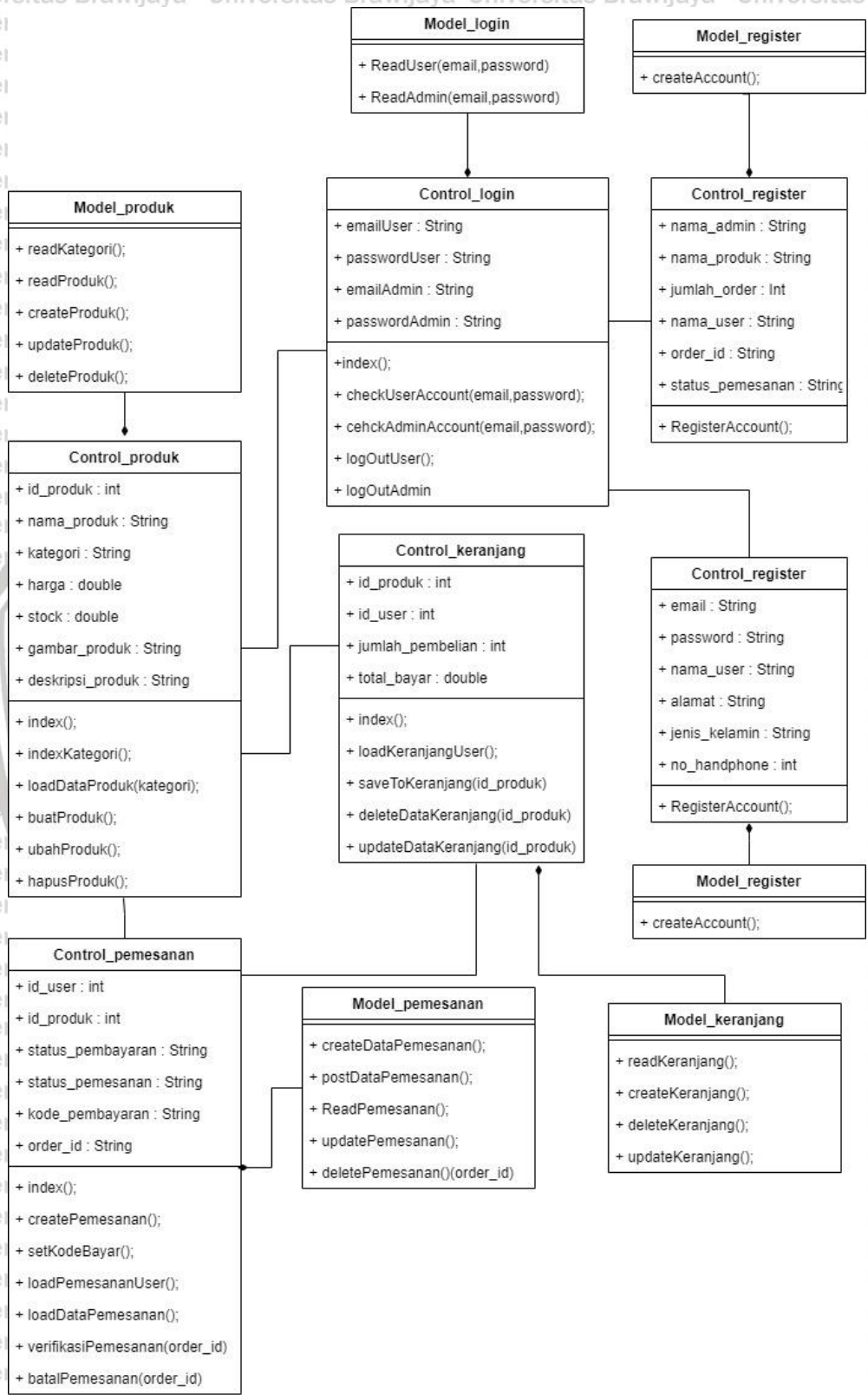
Gambar 5.19 Sequence Diagram Dashboard

Gambar 5.20 menunjukkan bentuk dari *sequence* diagram pada fungsi untuk menampilkan informasi di halaman *dashboard*. Diawali dengan aktor *Admin* melakukan akses ke halaman *dashboard* setelah melakukan *login*, lalu sistem akan menjalankan fungsi *index()* untuk mengambil data pada *database*. Setelah data diambil, sistem akan menampilkan data kepada aktor *Admin*.

5.3 Perancangan Class Diagram

Perancangan *class diagram* digunakan untuk menampilkan kelas-kelas yang ada pada sistem yang dapat memberikan sebuah gambaran relasi antar kelas – kelas yang ada pada sistem. Dalam perancangan *class diagram*, terdapat tiga komponen utama, yaitu Kelas, *Variable* atau atribut, dan *Method*. Setiap kelas mempunyai fungsinya masing – masing. Kelas yang berhubungan satu dengan lainnya memiliki relasi asosiasi. Sementara kelas yang merupakan turunan dari

kelas lain, memiliki relasi relasi agregasi. Gambar 5.11 akan melampirkan bentuk dari sistem informasi *e-commerce* Nue Pins.

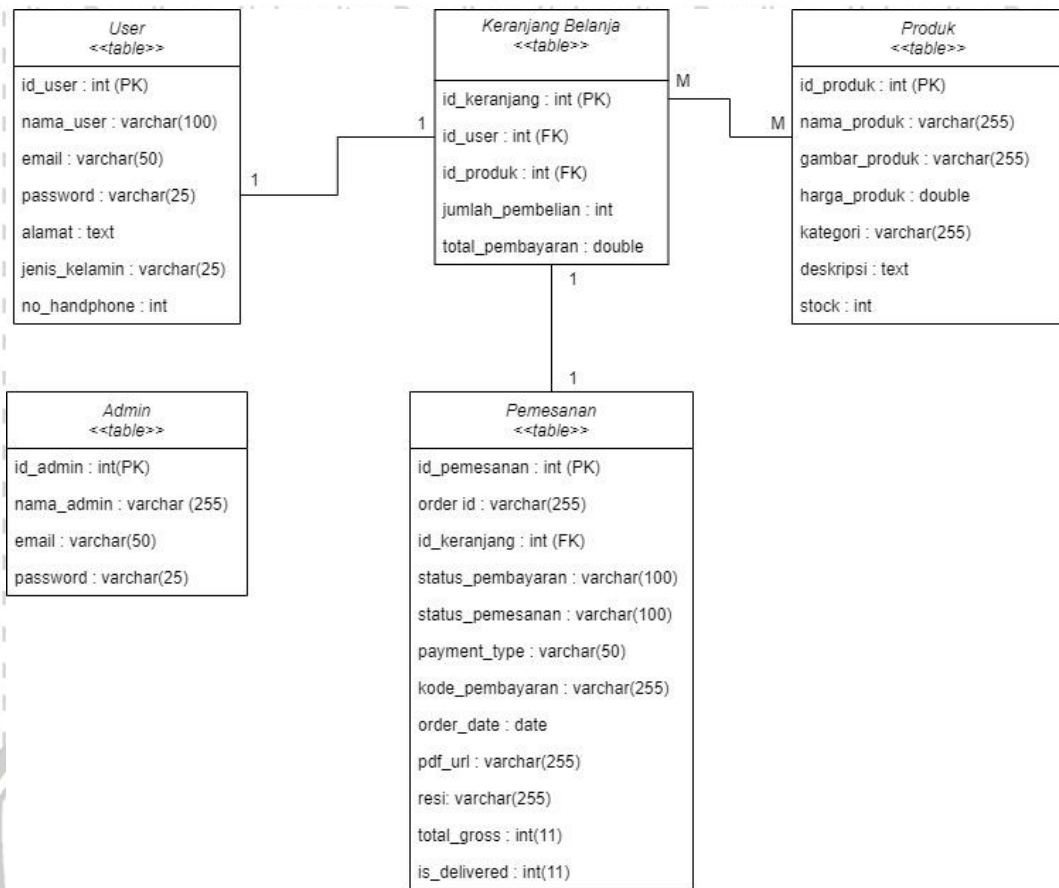


Gambar 5.20 Class Diagram Sistem Informasi Toko Online Nue Pins

Pada gambar 5.20 terdapat beberapa kelas yang berhubungan. Kelas yang merupakan agregasi dari kelas lain diantaranya adalah, kelas *Model_login* dengan kelas *control_login*, kelas *Model_register* dengan kelas *control_register*, kelas *Model_produk* dengan kelas *control_produk*, kelas *Model_keranjang* dengan kelas *control_keranjang*, kelas *Model_pemesanan* dengan kelas *control_pemesanan*, dan kelas *Model_dashboard* dengan kelas *control_dashboard*. Selanjutnya terdapat beberapa kelas yang memiliki hubungan asosiasi satu dengan lainnya, diantaranya adalah, kelas *control_login* dengan kelas *control_register* dan *control_produk*, kelas *control_pemesanan* dengan kelas *control_keranjang* dan kelas *control_produk*, kelas *control_keranjang* dengan kelas *control_produk* dan kelas *control_pemesanan*, dan kelas *control_dashboard* dengan kelas *control_login*.

5.4 Perancangan *Physical Data Model*

Physical Data Model digunakan untuk menggambarkan pemodelan dari *database* yang akan dibuat dalam bentuk relasi antar tabel. Pada *Physical data model*, hubungan antara tabel yang digunakan adalah hubungan secara kardinalitas, yaitu berapa banyaknya hubungan tabel satu sama lain. Untuk hubungan *one to one* akan ditandakan dengan angka 1 dan 1. Untuk hubungan *one to many* akan ditandain dengan 1 dan M. Lalu untuk hubungan *many to many* akan ditandai dengan M dan M. Gambar 5.21 akan menunjukkan bentuk dari *physical data model* yang dirancang sebagai pemodelan *database* sistem informasi Nue Pins.



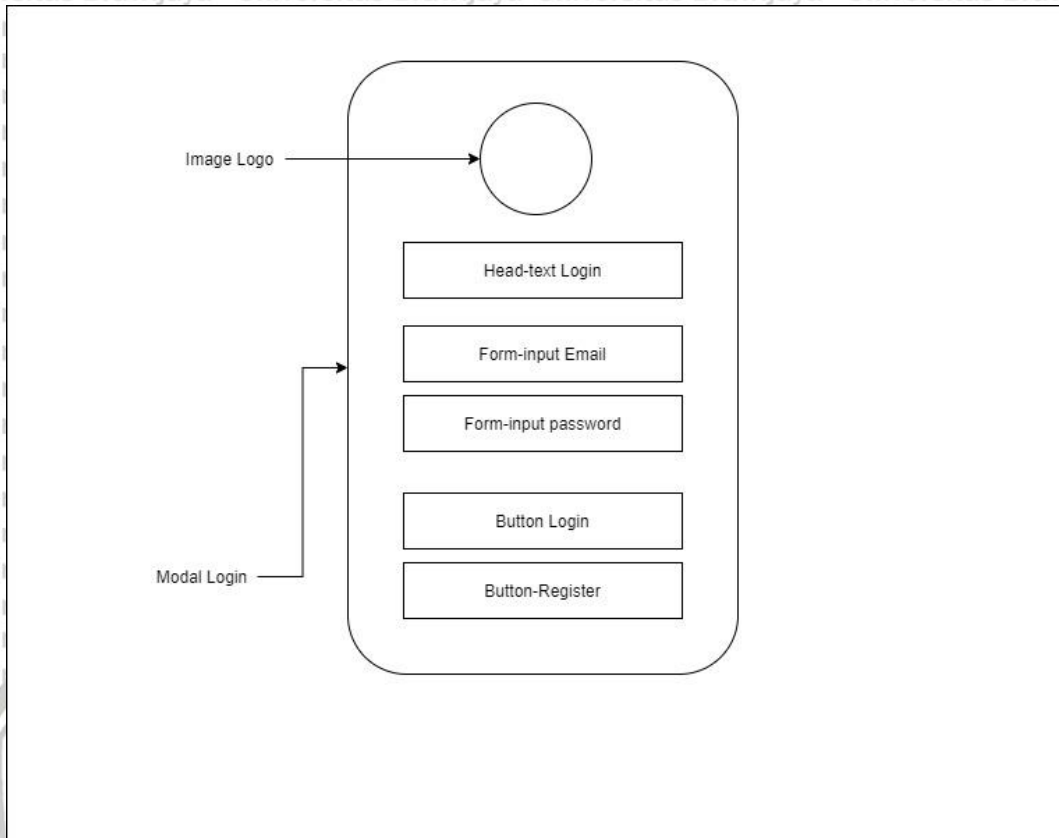
Gambar 5.21 Physical Data Model

Pada Gambar 5.21 terdapat 5 tabel yang saling berelasi satu sama lain, diantaranya tabel *User* berelasi dengan tabel keranjang belanja yang mempunyai relasi *one to one*, tabel *User* dengan tabel pemesanan yang mempunyai relasi *one to many*, tabel *Admin* dengan tabel produk dan tabel pemesanan memiliki relasi yang sama, yaitu *one to many*. Selanjutnya, tabel produk dengan tabel keranjang belanja dan pemesanan juga memiliki relasi yang sama, yaitu *many to many*.

5.5 Perancangan Antarmuka

Pada tahap perancangan antarmuka, akan dijadikan sebagai acuan dalam tahap implementasi sistem informasi *e-commerce* Nue Pins. Berikut adalah tahapan dari perancangan antarmuka sistem informasi *e-commerce* Nue Pins.

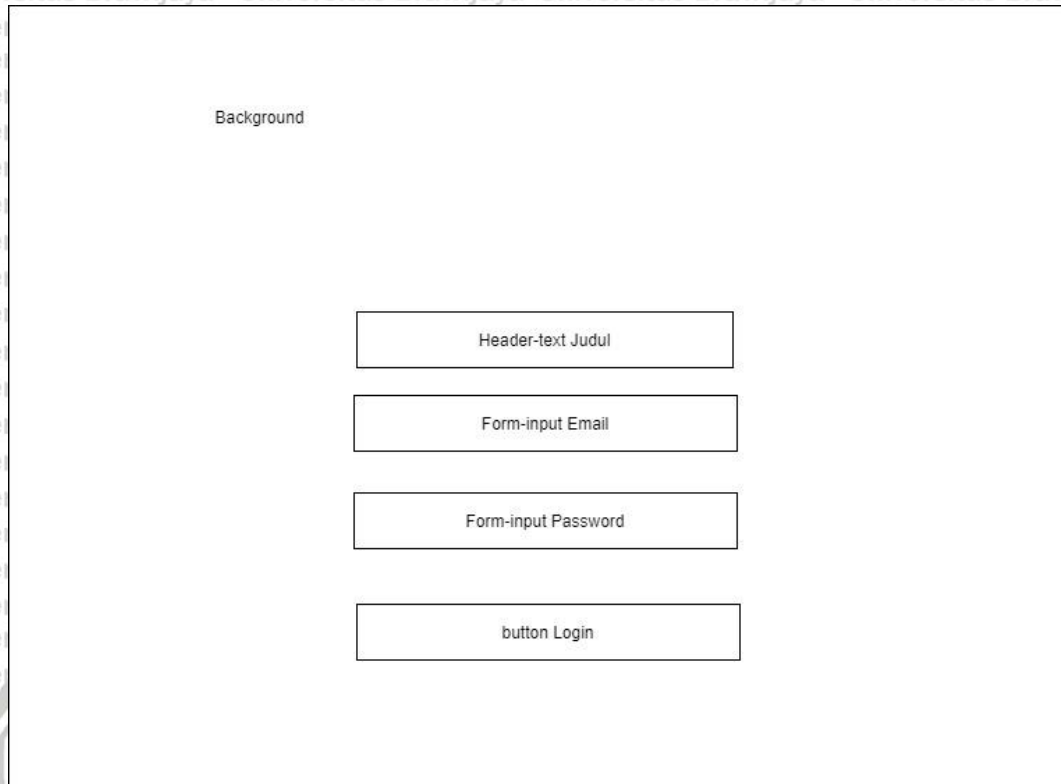
5.5.1 Perancangan Antarmuka Login User



Gambar 5.22 Perancangan Antarmuka Login User

Gambar 5.22 merupakan rancangan dari halaman *login* pada *website e-commerce* Nue Pins. Pada bagian kiri, terdapat blok lingkaran yang berisi gambar logo Nue Pins. Untuk bagian kanan, terdapat *card body* untuk menampung *header text Login*, *form input Email* dan *password*, dan *button login* dan *Register*.

5.5.2 Perancangan Antarmuka Login Admin

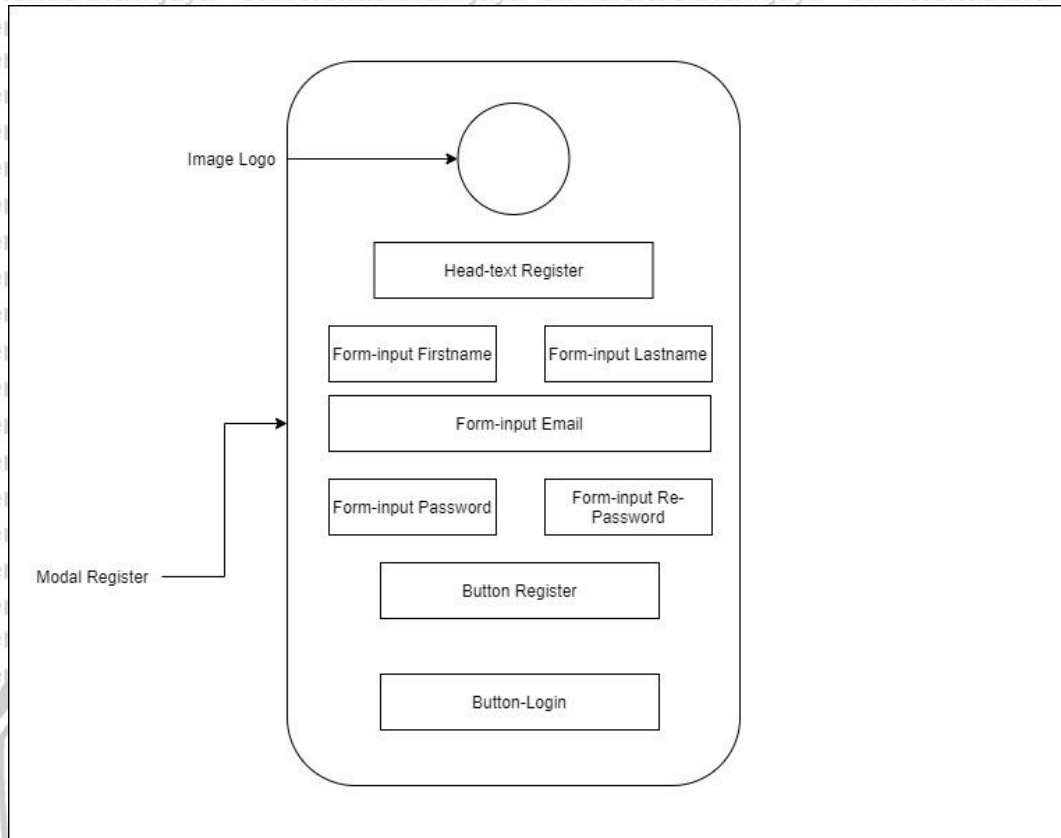


Gambar 5.23 Perancangan Antarmuka Login Admin

Gambar 5.23 merupakan rancangan dari halaman *login* pada sistem informasi khusus *Admin*. Terdapat *card body* pada bagian tengah, untuk menampung *header text Judul, form input email, form input password, dan button login*.



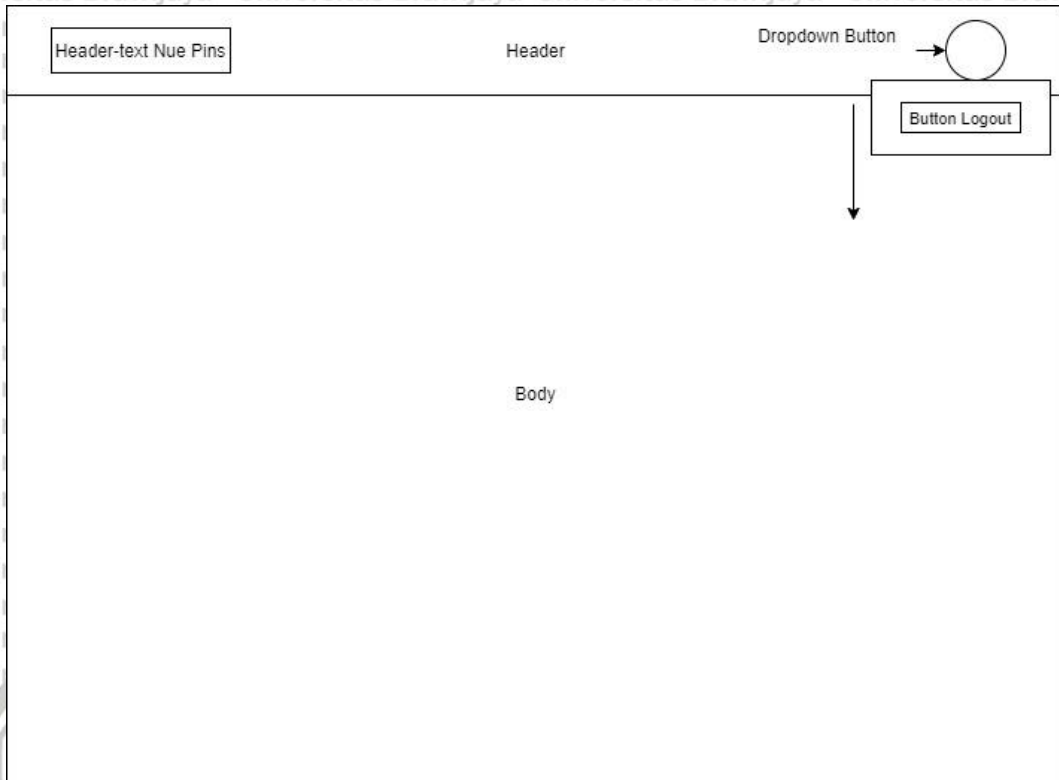
5.5.3 Perancangan Antarmuka Register



Gambar 5.24 Perancangan Antarmuka Register

Gambar 5.24 merupakan rancangan antarmuka dari halaman register pada website e-commerce Nue Pins. Terdapat *card body* untuk menampung beberapa element, diantaranya *header text register*, *form input firstname*, *lastname*, *email*, *password1*, dan *password2*. Untuk bagian bawah *card body*, terdapat tiga *button dropdown*, diantaranya *button dropdown* pemilihan bulan, tanggal, dan tahun. Kemudian pada bagian paling bawah, terdapat *button register*.

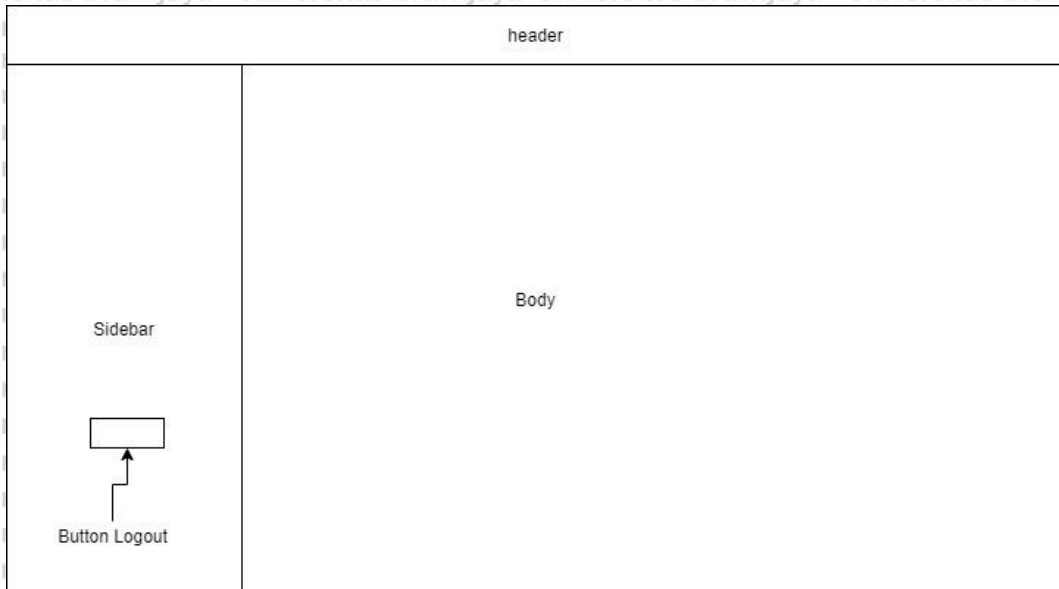
5.5.4 Perancangan Antarmuka Logout User



Gambar 5.25 Perancangan Antarmuka Logout User

Gambar 5.25 merupakan rancangan dari halaman untuk melakukan *logout*. Terdapat *header* pada bagian atas untuk menampung *header text* Nue Pins dan *button dropdown*. *Button* untuk melakukan *logout*, terdapat didalam *dropdown button* sebelah kanan *header*.

5.5.5 Perancangan Antarmuka Logout Admin

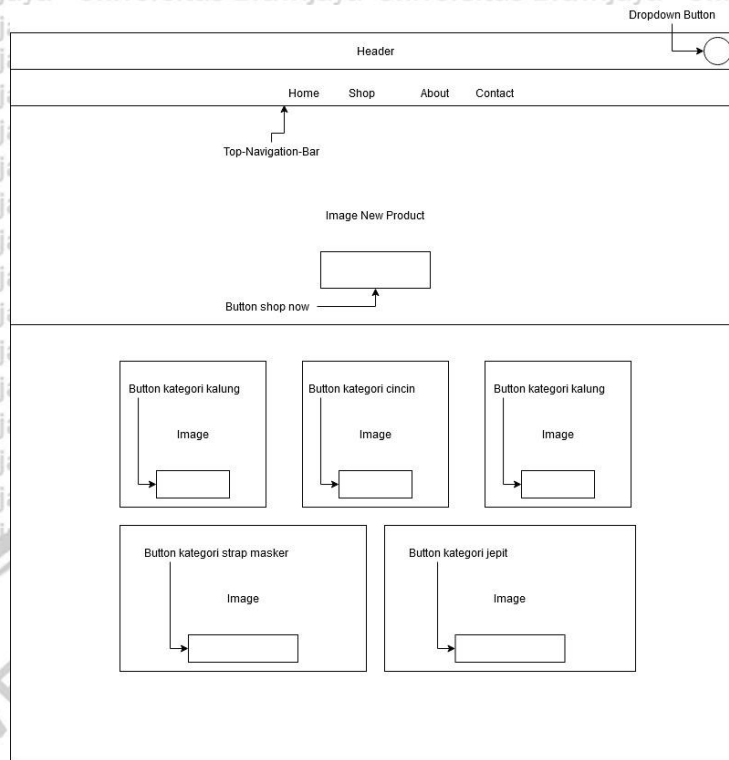


Gambar 5.26 Perancangan Antarmuka Logout Admin

Gambar 5.26 merupakan rancangan dari halaman *logout* pada sistem informasi khusus *Admin*. Tombol *logout* terletak pada bagian kanan *header*. Dan pada bagian kiri, terdapat *navigation sidebar* untuk memilih pengelolaan data produk atau data pemesanan.



5.5.6 Perancangan Antarmuka Menampilkan Produk



Gambar 5.27 Perancangan Antarmuka Menampilkan Produk (1)

Gambar 5.27 merupakan rancangan halaman utama *e-commerce* Nue Pins. Pada bagian atas terdapat header dan dibawahnya terdapat *top navigation bar* sebagai navigasi untuk mengakses halaman. Selanjutnya, terdapat gambar produk baru pada bagian bawah *top navigation bar*. Gambar produk baru tersebut dapat digeser ke arah kanan atau kiri untuk menampilkan gambar produk baru lainnya. Lalu, terdapat pemilihan kategori pada bagian bawah.

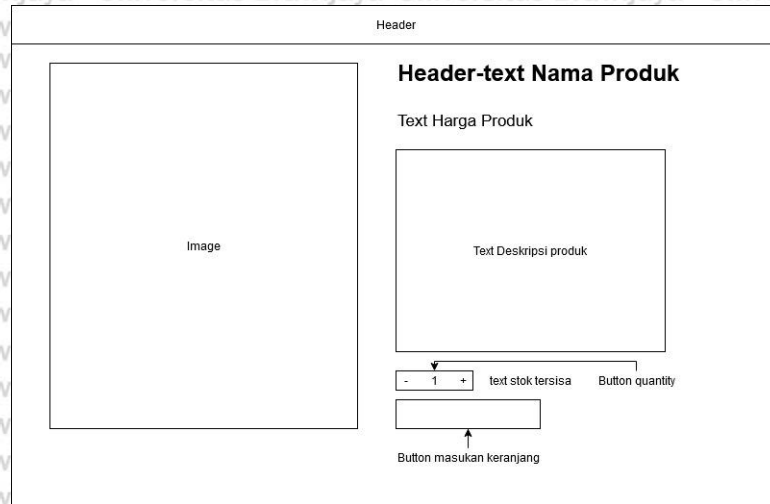




Gambar 5.28 Perancangan Antarmuka Menampilkan Produk (2)

Gambar 5.28 merupakan rancangan dari halaman untuk menampilkan produk berdasarkan kategori yang telah dipilih, pada *website e-commerce* Nue Pins. Pada bagian bawah *header*, terdapat gambar dari kategori yang dipilih. Lalu pada bagian bawah gambar kategori, terdapat tombol untuk melakukan pengurutan penampilan produk, dan disebelah kanan terdapat informasi jumlah penampilan produk. Lalu pada bagian bawah tombol pengurutan dan informasi jumlah penampilan produk, terdapat gambar produk yang ditampilkan sejumlah empat pada satu baris. Digambar tersebut terdapat tombol untuk menampilkan informasi produk secara *detail*, dan dibawah gambar produk, terdapat nama produk dan harga produk. Pada bagian bawah, terdapat pemilihan halaman.

5.5.7 Perancangan Antarmuka Keranjang Belanja



Gambar 5.29 Perancangan Antarmuka Keranjang Belanja (1)

Gambar 5.29 merupakan rancangan antarmuka dari halaman informasi *detail* produk pada *website e-commerce* Nue Pins. Pada bagian bawah kiri *header*, terdapat gambar produk yang berukuran besar. Lalu, disebelah kanan terdapat informasi produk yang akan ditampilkan seperti nama produk, harga produk, dan deskripsi produk. Lalu dibawah deskripsi produk terdapat tombol untuk memilih jumlah pembelian yang akan dimasukkan ke keranjang belanja, dan dibawahnya terdapat tombol untuk memasukkan produk ke keranjang belanja.

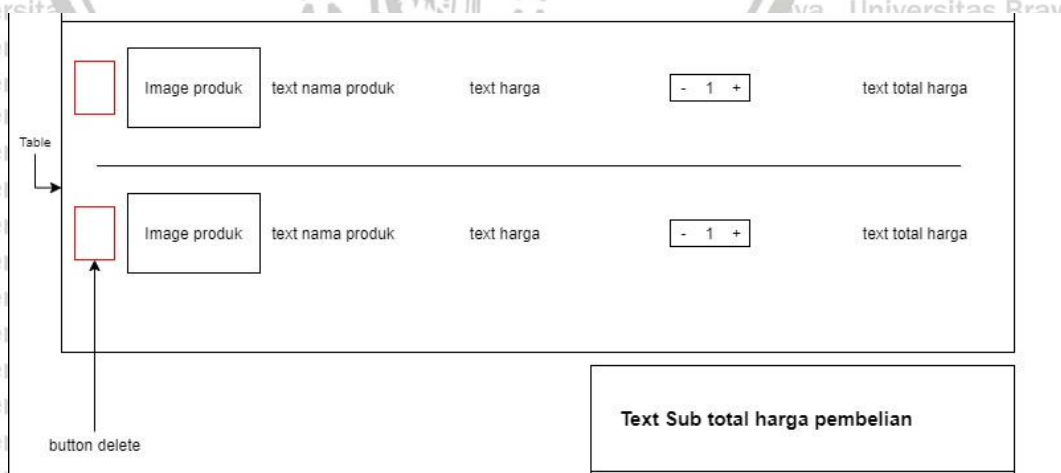




Gambar 5.30 Perancangan Antarmuka Keranjang Belanja (2)

Gambar 5.30 merupakan rancangan antarmuka dari halaman keranjang belanja pada *website e-commerce* Nue Pins. Keranjang belanja akan dibuat dengan menggunakan tabel yang diisi dengan data produk yang telah dipilih. Dalam tabel tersebut akan berisi data produk seperti gambar, nama produk, harga produk, tombol jumlah pembelian, total harga produk, dan tombol penghapusan produk dari keranjang belanja. Pada bagian bawah tabel, terdapat kotak yang berisikan informasi total harga pembelian dan tombol untuk melakukan *checkout*.

5.5.8 Perancangan Antarmuka Menghapus Produk

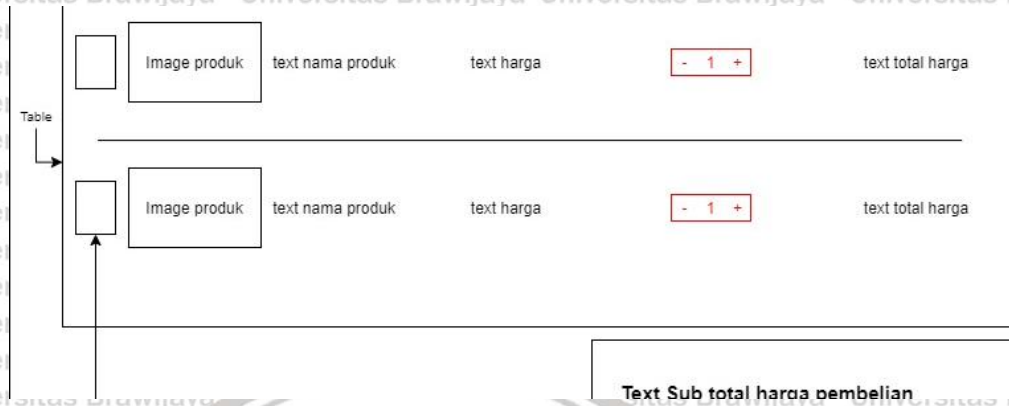


Gambar 5.31 Perancangan Antarmuka Menghapus Produk

Gambar 5.31 merupakan rancangan dari halaman keranjang belanja saat melakukan penghapusan produk. Tombol penghapusan produk, terletak pada

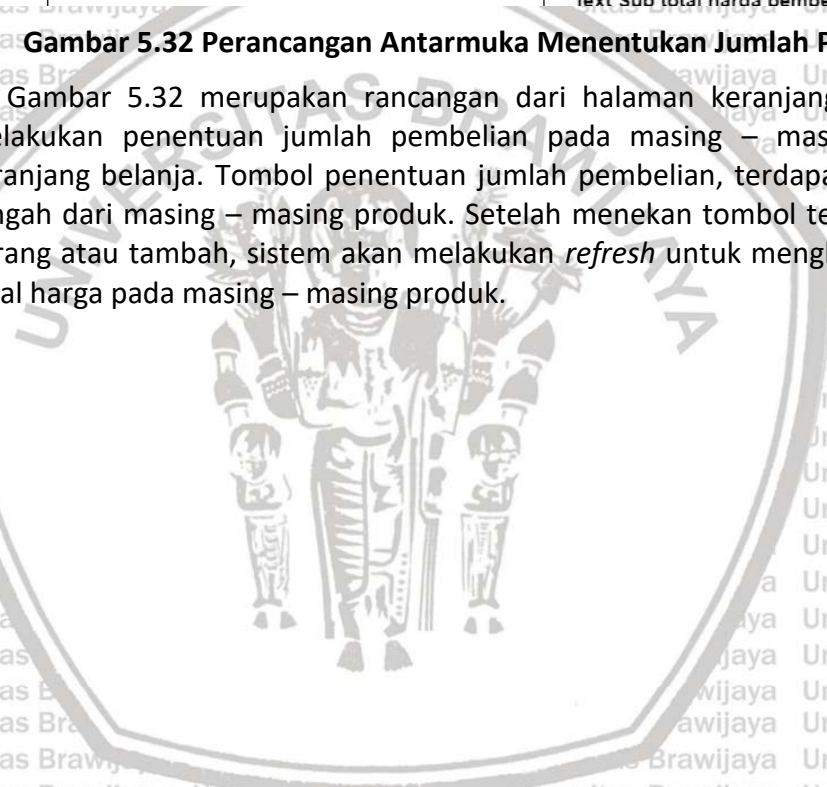
bagian kanan dari masing – masing produk. Konfirmasi penghapusan akan dimunculkan setelah menekan tombol penghapusan produk.

5.5.9 Perancangan Antarmuka Menentukan Jumlah Pembelian



Gambar 5.32 Perancangan Antarmuka Menentukan Jumlah Pembelian

Gambar 5.32 merupakan rancangan dari halaman keranjang belanja saat melakukan penentuan jumlah pembelian pada masing – masing barang di keranjang belanja. Tombol penentuan jumlah pembelian, terdapat pada bagian tengah dari masing – masing produk. Setelah menekan tombol tersebut, antara kurang atau tambah, sistem akan melakukan *refresh* untuk menghitung kembali total harga pada masing – masing produk.



5.5.10 Perancangan Antarmuka Membuat Pemesanan

The screenshot shows a checkout form with a 'Header' at the top. It is divided into two main sections:

- Text checkout details:** Contains input fields for 'Text firstname' and 'Text lastname' (both 'form-input'), a 'Text street address' ('form-input-textarea'), and three dropdown menus for 'text province', 'text city', and 'text district' (all labeled 'dropdowns button'). There is also a 'text postcode/ZIP' ('form-input') field.
- Text-informasi pemesanan:** Contains a table with 'text produk' and 'text total harga', a row for 'text nama produk x jumlah pembelian' and 'text total harga', and another row for 'text sub total' and 'text sub total harga'. Below this is a dropdown menu for 'dropdowns button metode pembayaran' and a 'Button buat pesanan' at the bottom.

Gambar 5.33 Perancangan Antarmuka Membuat Pemesanan (1)

Gambar 5.33 merupakan rancangan dari halaman *checkout* pada *website e-commerce* Nue Pins. Pada bagian kiri, terdapat pengisian informasi data pembeli. Pengisian data pembeli terdiri dari nama depan, nama belakang, nama jalan, pemilihan provinsi, pemilihan kota, pemilihan kecamatan, dan pengisian kode pos. Lalu, pada sebelah kanan terdapat informasi pembelian produk, dan terdapat tombol pemilihan metode pembayaran yang akan digunakan.

The screenshot shows a payment code display form with the following elements:

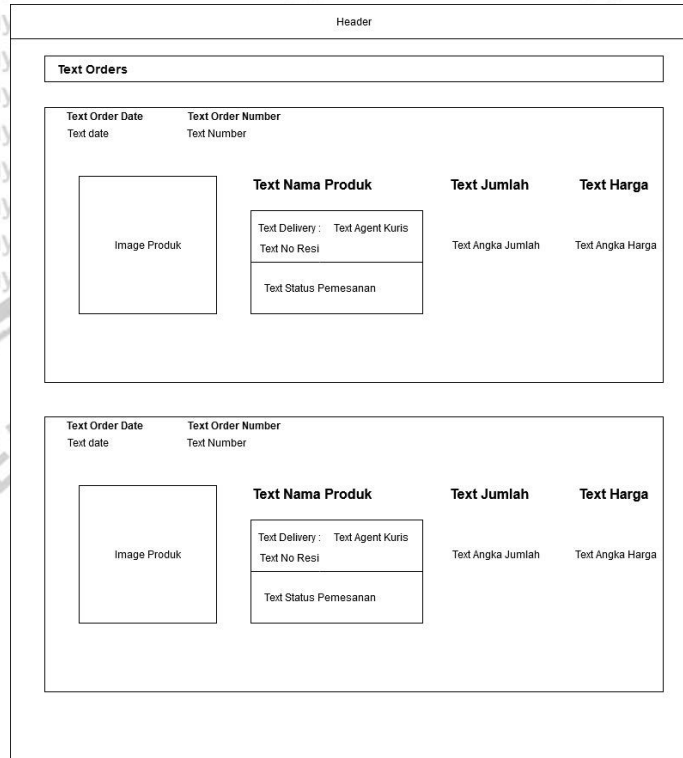
- A header 'text metode pembayaran'.
- A title 'Text Kode Pembayaran'.
- A row with 'text jumlah harus dibayar:' and 'text angka pembayaran'.
- A large rectangular box containing 'Text tata cara pembayaran'.
- A 'Button OK' at the bottom.

Gambar 5.34 Perancangan Antarmuka Membuat Pemesanan (2)

Gambar 5.34 merupakan rancangan dari halaman penampilan kode pembayaran pada *website e-commerce* Nue Pins. Pada bagian atas, terdapat *text*

yang menampilkan kode pembayaran yang telah dipilih. Lalu pada bagian bawahnya, terdapat *text* yang akan menampilkan kode pembayaran, informasi jumlah yang harus dibayar, dan tata cara pembayaran. Pada bagian bawah, terdapat *button* yang akan mengarahkan kehalaman utama *website e-commerce* Nue Pins.

5.5.11 Perancangan Antarmuka *Track Pemesanan*



Gambar 5.35 Perancangan Antarmuka *Track Pemesanan*

Gambar 5.35 merupakan rancangan dari halaman penampilan status pemesanan pada *website e-commerce* Nue Pins. Pada bagian bawah *header*, terdapat *text* Pemesanan, dan pada bagian bawahnya, terdapat kotak yang berisikan data pemesanan. Data pemesanan yang ditampilkan merupakan tanggal pemesanan, nomor pemesanan, gambar produk, nama produk, status pemesanan, nomor resi pengiriman, jumlah pembelian produk, dan jumlah harga yang dibayar.

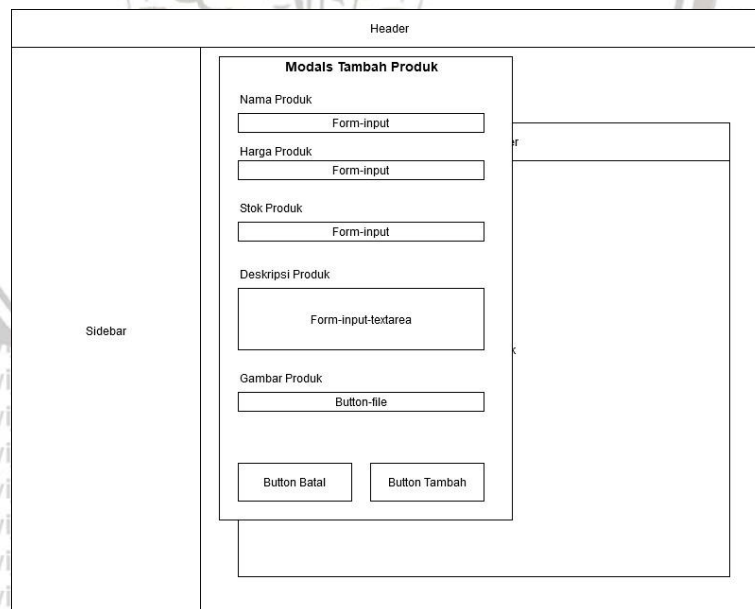
5.5.12 Perancangan Antarmuka Informasi Produk



Gambar 5.36 Perancangan Antarmuka Melihat Data Produk

Gambar 5.36 merupakan rancangan dari halaman informasi produk pada sistem informasi khusus *Admin*. Data produk akan ditampilkan dalam tabel yang akan diurutkan berdasarkan kategori produk. Lalu, pada bagian atas tabel, terdapat tombol untuk menambah data produk.

5.5.13 Perancangan Antarmuka Menambah Data Produk

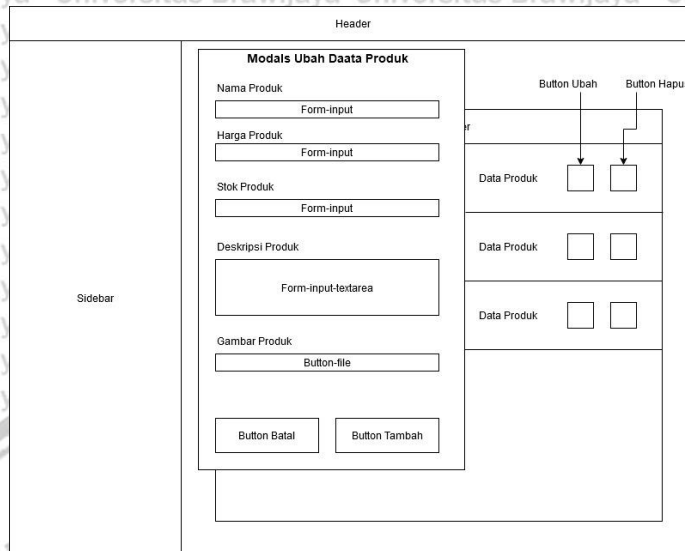


Gambar 5.37 Perancangan Antarmuka Menambah Data Produk

Gambar 5.37 merupakan rancangan dari halaman saat melakukan penambahan produk pada sistem informasi khusus *Admin*. Penambahan produk dilakukan dengan cara mengisi beberapa *form input*. *Form input* tersebut, terdiri

dari nama produk, harga produk, stok produk, deskripsi produk, dan gambar produk. *Form* pengisian akan ditampilkan dengan menggunakan *modals*.

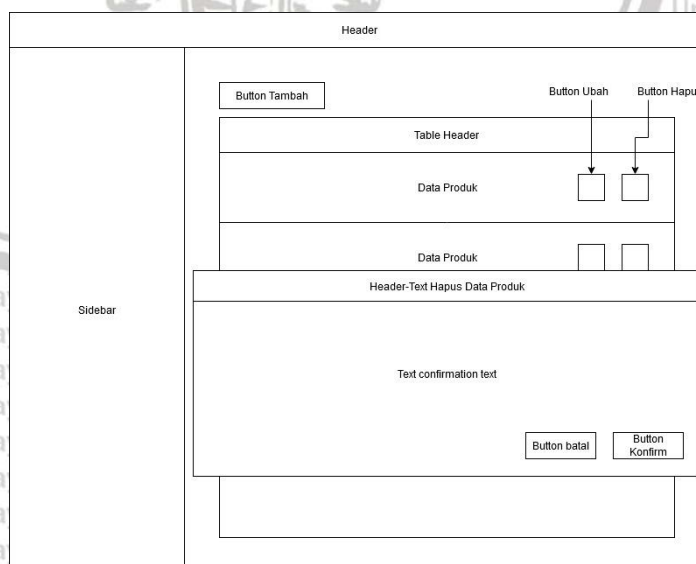
5.5.14 Perancangan Antarmuka Mengubah Data Produk



Gambar 5.38 Perancangan Antarmuka Mengubah Data Produk

Gambar 5.38 merupakan rancangan dari halaman pengubahan data produk pada sistem informasi khusus *Admin*. Tombol untuk mengubah data produk, terletak pada bagian kanan dari masing – masing produk didalam tabel. Pengubahan produk akan dilakukan dengan melakukan pengisian pada *form input* yang telah disediakan

5.5.15 Perancangan Antarmuka Menghapus Data Produk

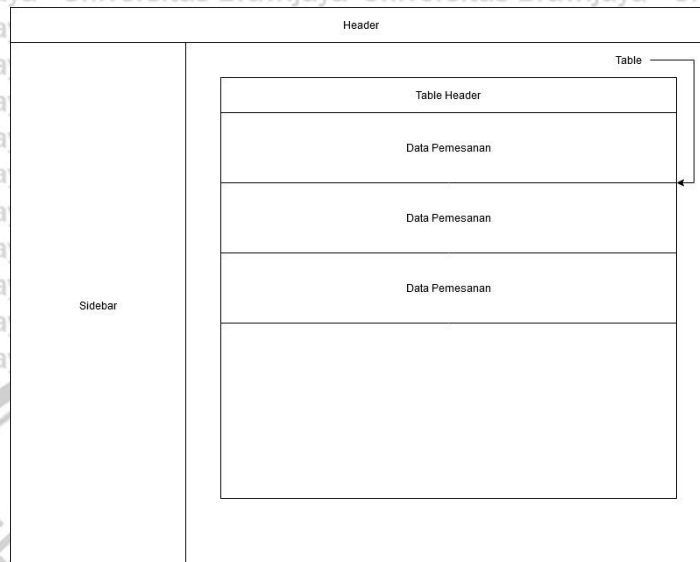


Gambar 5.39 Perancangan Antarmuka Menghapus Data Produk

Gambar 5.39 merupakan rancangan dari halaman penghapusan data produk pada sistem informasi khusus *Admin*. Tombol untuk menghapus data produk,

terletak pada sebelah kanan dari masing – masing produk didalam tabel. Setelah menekan tombol untuk menghapus data produk, akan ditampilkan *dialog* konfirmasi yang berisikan *text konfirmasi*, tombol batal, dan tombol *confirm*.

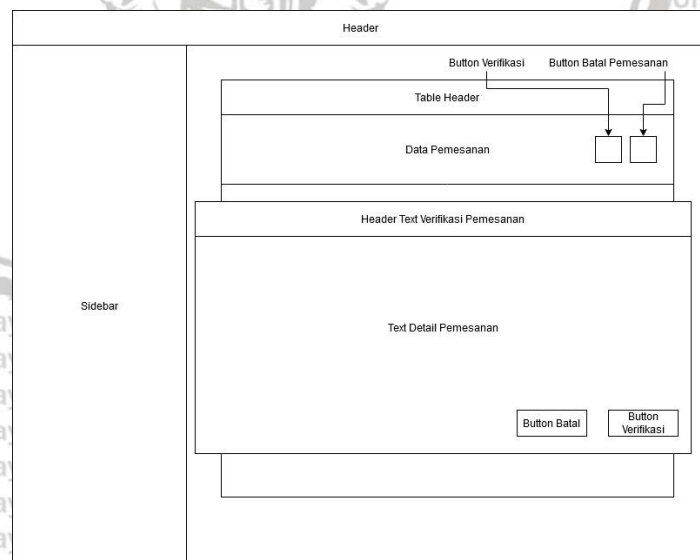
5.5.16 Perancangan Antarmuka Informasi Pemesanan



Gambar 5.40 Perancangan Antarmuka Melihat Data Pemesanan

Gambar 5.40 merupakan rancangan antarmuka dari halaman penampilan informasi pemesanan pada sistem informasi khusus *Admin*. Data pemesanan akan ditampilkan didalam tabel yang akan diurutkan berdasarkan tanggal pemesanan.

5.5.17 Perancangan Antarmuka Verifikasi Pemesanan

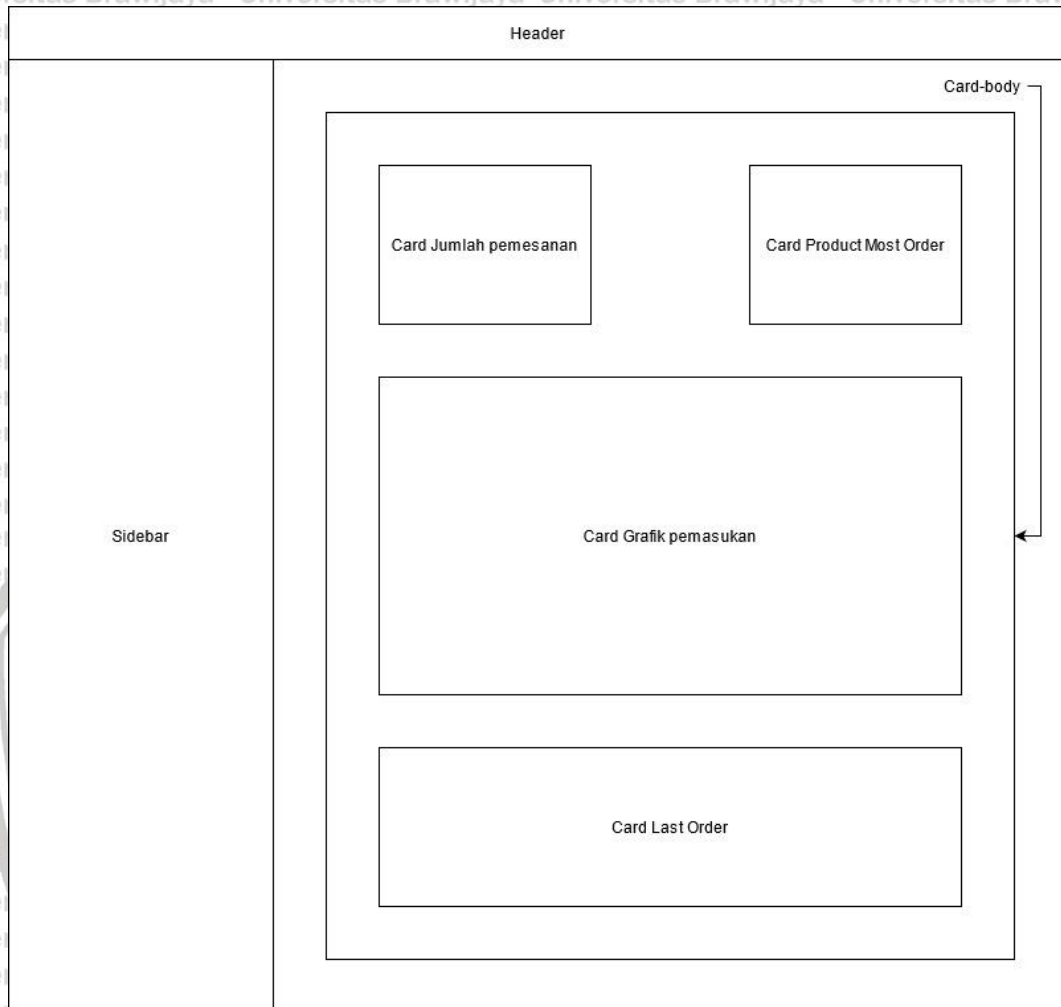


Gambar 5.41 Perancangan Antarmuka Verifikasi Pemesanan

Gambar 5.41 merupakan rancangan antarmuka dari halaman verifikasi pemesanan pada sistem informasi khusus *Admin*. Tombol verifikasi terletak pada

bagian kanan dari masing – masing data pemesanan didalam tabel. *Dialog* konfirmasi akan ditampilkan setelah menekan tombol verifikasi.

5.5.18 Perancangan Antarmuka Dashboard



Gambar 5.42 Perancangan Antarmuka Dashboard

Gambar 5.43 merupakan rancangan antarmuka dari halaman *dashboard* pada sistem informasi khusus *Admin*. Terdapat *card-body* yang menampung informasi dari data jumlah pemesanan, informasi dari data produk yang sering diorder, informasi grafik pemasukan, dan informasi order terakhir yang telah dilakukan oleh aktor *User*.

BAB 6 IMPLEMENTASI SISTEM

6.1 Spesifikasi Sistem

Spesifikasi sistem yang digunakan untuk implementasi sistem informasi toko *online* Nue Pins, menggunakan perangkat keras dan perangkat lunak. Perangkat keras digunakan sebagai alat untuk menjalankan perangkat lunak yang digunakan sebagai wadah untuk melakukan implementasi kode program. Sistem informasi toko *online* Nue Pins, dan sistem informasi khusus *Admin* dikembangkan dengan menggunakan bahasa pemrograman PHP. Data yang diperlukan dari *API Midtrans*, akan diambil dengan menggunakan format *JSON*.

6.1.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Perangkat keras yang digunakan untuk melakukan implementasi sistem, merupakan komputer dengan sistem operasi *windows 10*. Tabel 6.1 akan menjelaskan spesifikasi perangkat keras yang digunakan untuk mengimplementasikan kode program, mengimplementasikan basis data, dan menjalankan kode program.

Tabel 6.1 Spesifikasi Perangkat Keras

| | |
|----------------------|-----------------------------|
| Sistem Operasi | Windows 10 Pro |
| Versi Sistem Operasi | 1903 |
| Tipe Sistem | 64 Bit |
| Processor | Intel (R) Core(TM) i5-7200U |
| RAM | 8 GB |

Perangkat lunak yang digunakan untuk melakukan implementasi kode program, merupakan perangkat lunak yang dapat digunakan sebagai media pengembangan sistem berbasis *website*, diantaranya *text editor* dan pengelolaan basis data. Sedangkan perangkat lunak yang digunakan untuk menjalankan kode program, diantaranya *web browser* dan *web server* lokal. Tabel 6.2 akan menjelaskan spesifikasi perangkat lunak yang digunakan untuk mengimplementasikan kode program, basis data, dan menjalankan kode program.

Tabel 6.2 Spesifikasi Perangkat Lunak

| | | |
|------------------------|------------------------------|-------------|
| Text Editor | Microsoft Visual Studio Code | 1.56.2 |
| Pengelolaan Basis Data | Phpmyadmin | 5.0.2 |
| Web Server | Apache HTTP Server | 2.4.43.0 |
| Web Browser | Microsoft Edge | 90.0.818.66 |
| Bahasa Pemrograman | PHP | 7.4.8 |

6.2 Implementasi Basis Data

Basis data dari sistem informasi toko *online* Nue Pins, diimplementasikan dengan menggunakan pengelolaan basis data *phpmyadmin*. Basis data diimplementasikan berdasarkan *Physical Data Model* yang telah dirancang. Dalam bagian ini, akan ditampilkan implementasi basis data berdasarkan masing – masing tabel.

6.2.1 Implementasi Tabel User

Pada implementasi tabel *user*, terdapat kolom yang digunakan untuk menyimpan dan membaca data *user* dari sistem informasi toko *online* Nue Pins. Kolom *id_user* merupakan kolom yang digunakan sebagai *primary key* dan bertipe data *integer*. Kolom *id_user* mempunyai opsi *auto increment* untuk menambahkan isi kolom secara otomatis saat data baru dimasukan kedalam tabel *user*. Kolom *email*, dan *password* digunakan sebagai *username* untuk melakukan *login* pada sistem. Kolom lainnya pada tabel *user* digunakan untuk menyimpan informasi yang diperlukan dari aktor *User*. Tabel 6.3 akan menjelaskan *query* dari pembuatan tabel *user* dengan bahasa *SQL*.

Tabel 6.3 Implementasi Tabel User

| | | | | | | |
|----|------------------|--------------|--------|---------|-----------------|----|
| 1 | CREATE | TABLE | `user` | | | (|
| 2 | `id_user` | int(11) | NOT | NULL | AUTO_INCREMENT, | |
| 3 | `email` | varchar(50) | | NOT | NULL, | |
| 4 | `password` | varchar(255) | | NOT | NULL, | |
| 5 | `firstname` | varchar(100) | | NOT | NULL, | |
| 6 | `lastname` | varchar(100) | | NOT | NULL, | |
| 7 | `alamat` | text | | DEFAULT | NULL, | |
| 8 | `no_handphone` | varchar(15) | | DEFAULT | NULL, | |
| 9 | `is_emailActive` | int(11) | | NOT | NULL, | |
| 10 | PRIMARY | KEY | | | (`id_user`) |); |

6.2.2 Implementasi Tabel Admin

Pada implementasi tabel *admin*, terdapat kolom yang akan digunakan untuk menyimpan dan membaca informasi data *admin* pada sistem khusus *admin*. Kolom *id_admin* merupakan kolom yang digunakan sebagai *primary key*, bertipe data *integer* dan mempunyai opsi *auto increment* untuk menambahkan isi kolom secara otomatis saat data baru pada tabel *admin* ditambahkan. Kolom *email* dan *password* digunakan untuk melakukan *login* pada sistem khusus *admin*. Kolom lainnya pada tabel *admin* digunakan untuk menyimpan informasi data *Admin*. Tabel 6.4 akan menjelaskan *query* dari pembuatan tabel *admin* dengan bahasa *SQL*.

Tabel 6.4 Implementasi Tabel Admin

| | | | | | | |
|---|------------|---------|---------|------|-----------------|---|
| 1 | CREATE | TABLE | `admin` | | | (|
| 2 | `id_admin` | int(11) | NOT | NULL | AUTO INCREMENT, | |

| | | | | |
|---|-------------|--------------|-----|--------------|
| 3 | `firstname` | varchar(100) | NOT | NULL, |
| 4 | `lastname` | varchar(100) | NOT | NULL, |
| 5 | `email` | varchar(100) | NOT | NULL, |
| 6 | `password` | varchar(255) | NOT | NULL, |
| 7 | PRIMARY | KEY | | (`id_admin`) |

6.2.3 Implementasi Tabel produk

Pada implementasi tabel produk, terdapat kolom *id_produk* yang didefinisikan sebagai *primary key* dengan tipe data *integer*, dan mempunyai opsi *auto increment* untuk menambahkan isi kolom otomatis saat data baru telah dibuat pada tabel produk. Kolom gambar digunakan untuk menyimpan data *url* gambar produk, dengan tipe data *varchar*. Kolom lainnya digunakan untuk menyimpan informasi data produk seperti nama produk, harga produk, kategori produk, deskripsi produk, dan stok produk. Tabel 6.5 akan menjelaskan *query* dari pembuatan tabel produk dengan bahasa *SQL*.

Tabel 6.5 Implementasi Tabel Produk

| | | | | |
|---|-----------------|--------------|----------|-----------------|
| 1 | CREATE | TABLE | `produk` | (|
| 2 | `id_produk` | int(11) | NOT NULL | AUTO_INCREMENT, |
| 3 | `nama_produk` | varchar(255) | NOT | NULL, |
| 4 | `gambar_produk` | varchar(255) | NOT | NULL, |
| 5 | `harga_produk` | double | NOT | NULL, |
| 6 | `kategori` | varchar(255) | NOT | NULL, |
| 7 | `deskripsi` | text | NOT | NULL, |
| 8 | `stock` | int(11) | NOT | NULL, |
| 9 | PRIMARY | KEY | | (`id_produk`) |

6.2.4 Implementasi Tabel Keranjang Belanja

Pada implementasi tabel keranjang belanja, terdapat kolom *id_keranjang* yang didefinisikan sebagai *primary key* dengan tipe data *integer* dan mempunyai opsi *auto increment*. Tabel keranjang belanja juga memiliki kolom yang merupakan referensi dari tabel lain, diantaranya kolom *id_user* yang digunakan untuk mengetahui aktor *user* yang memasukan data ke keranjang belanja, dan kolom *id_produk* yang digunakan untuk memasukan data produk kedalam tabel keranjang belanja. Kolom yang merupakan referensi dari tabel lain, digunakan sebagai *foreign key* pada tabel keranjang belanja. Kolom jumlah pembelian digunakan untuk menyimpan data dari jumlah pembelian yang dipilih oleh aktor *user* saat ingin memasukan produk kedalam keranjang belanja. Kolom total pembayaran digunakan untuk menyimpan data dari hasil perkalian jumlah pembelian dengan harga produk. Tabel 6.6 akan menjelaskan *query* dari pembuatan tabel keranjang belanja dengan menggunakan bahasa *SQL*.

Tabel 6.6 Implementasi Tabel Keranjang Belanja

| | | | | |
|---|----------------|---------|---------------------|-----------------|
| 1 | CREATE | TABLE | `keranjang_belanja` | (|
| 2 | `id_keranjang` | int(11) | NOT NULL | AUTO_INCREMENT, |
| 3 | `id_user` | int(11) | NOT | NULL, |

| | | | | |
|----|--------------------|----------------------------|----------|-------------------|
| 4 | `id_produk` | int(11) | NOT | NULL, |
| 5 | `jumlah_pembelian` | int(11) | DEFAULT | NULL, |
| 6 | `total_pembayaran` | double | DEFAULT | NULL, |
| 7 | PRIMARY | KEY | | (`id_keranjang`), |
| 8 | KEY | `idx_user` | | (`id_user`), |
| 9 | KEY | `idx_produk` | | (`id_produk`), |
| 10 | CONSTRAINT | `keranjang_belanja_ibfk_1` | FOREIGN | KEY |
| | (`id_user`) | REFERENCES | `user` | (`id_user`), |
| 11 | CONSTRAINT | `keranjang_belanja_ibfk_2` | FOREIGN | KEY |
| | (`id_produk`) | REFERENCES | `produk` | (`id_produk`) |

6.2.5 Implementasi Tabel Pemesanan

Pada implementasi tabel pemesanan, terdapat kolom *id_pemesanan* yang didefinisikan sebagai *primary key* dengan tipe data integer dan mempunyai opsi *auto increment*. Tabel pemesanan juga mempunyai kolom *order_id* yang digunakan sebagai penyimpanan data *order_id* dari data *JSON* yang dibuat oleh *api midtrans*. Kolom *order_id* digunakan juga sebagai pengelompokan data pemesanan yang lebih dari satu produk. Kolom yang merupakan referensi dari kolom di tabel lain adalah kolom *id_user* dan *id_produk*. Kolom tersebut digunakan sebagai *foreign key* pada tabel pemesanan. Tabel pemesanan juga mempunyai kolom *pdf_url* untuk menyimpan *url pdf* tata cara pembayaran.

Tabel 6.7 Implementasi Tabel Pemesanan

| | | | | |
|----|---------------------|-----------------------|-------------|----------------------|
| 1 | CREATE | TABLE | `pemesanan` | (|
| 2 | `id_pemesanan` | int(11) | NOT | NULL AUTO_INCREMENT, |
| 3 | `order_id` | varchar(255) | NOT | NULL, |
| 4 | `id_produk` | int(11) | NOT | NULL, |
| 5 | `id_user` | int(11) | NOT | NULL, |
| 6 | `quantity` | int(11) | NOT | NULL, |
| 7 | `status_pembayaran` | varchar(255) | NOT | NULL, |
| 8 | `status_pemesanan` | varchar(255) | NOT | NULL, |
| 9 | `payment_type` | varchar(50) | NOT | NULL, |
| 10 | `kode_pembayaran` | varchar(255) | NOT | NULL, |
| 11 | `order_date` | date | NOT | NULL, |
| 12 | `pdf_url` | varchar(255) | NOT | NULL, |
| 13 | `resi` | varchar(255) | DEFAULT | NULL, |
| 14 | `total_gross` | int(11) | NOT | NULL, |
| 15 | `isDelivered` | int(11) | DEFAULT | NULL, |
| 16 | PRIMARY | KEY | | (`id_pemesanan`), |
| 17 | KEY | `idx_userpemesanan` | | (`id_user`), |
| 18 | KEY | `idx_produkpemesanan` | | (`id_produk`), |
| 19 | CONSTRAINT | `pemesanan_ibfk_1` | FOREIGN | KEY |
| | REFERENCES | `user` | | (`id_user`), |
| 20 | CONSTRAINT | `pemesanan_ibfk_2` | FOREIGN | KEY |
| | REFERENCES | `produk` | | (`id_produk`) |

6.3 Implementasi API Midtrans

Midtrans telah menyediakan *plugin* yang berisikan kelas – kelas yang dibutuhkan untuk kebutuhan integrasi, salah satunya *controller Snap* dan *libraries*

midtrans pada PHP. Untuk melakukan akses pada endpoint Midtrans, cukup dilakukan pengisian *server key* yang telah disediakan pada masing – masing akun Midtrans dialam konstruktor kelas Snap. *Server key* akan diinisasi sebagai *variable params* untuk dijadikan sebagai parameter pada *method config* di *libraries midtrans*. Tabel 6.8 akan menjelaskan kode sumber dari konstruktor *controller snap*.

Tabel 6.8 Impementasi Server Key API Midtrans

| No | Snap.php |
|----|--|
| 1 | public function __construct() |
| 2 | { |
| 3 | parent::__construct(); |
| 4 | \$params = array('server_key' => 'SB-Mid-server-hDlledrJ6TXf_qxY5nP3FeEk', 'production' => false); |
| 5 | \$this->load->library('midtrans'); |
| 6 | \$this->midtrans->config(\$params); |
| 7 | \$this->load->helper('url'); |
| 8 | \$this->load->model('model_login'); |
| 9 | \$this->load->model('model_produk'); |
| 10 | \$this->load->model('model_keranjang'); |
| 11 | \$this->load->model('model_pemesanan'); |
| 12 | \$this->load->library('session'); |
| | } |

Setelah *variable params* dikirimkan, *variable* tersebut akan dijadikan sebagai *serverkey* pada *function config*. *Variable server key* merupakan *variable static* yang digunakan untuk pemanggilan *api gateway* midrans salah satunya pada *function getSnapToken*. *Function getSnapToken* digunakan untuk membuat data transaksi sekaligus pembuatan kode pembayaran. Tabel 6.9 akan menampilkan kode sumber dari *function config* dan *getSnapToken* pada *libraries Midtrans*.

Tabel 6.9 Implementasi Library Midtrans

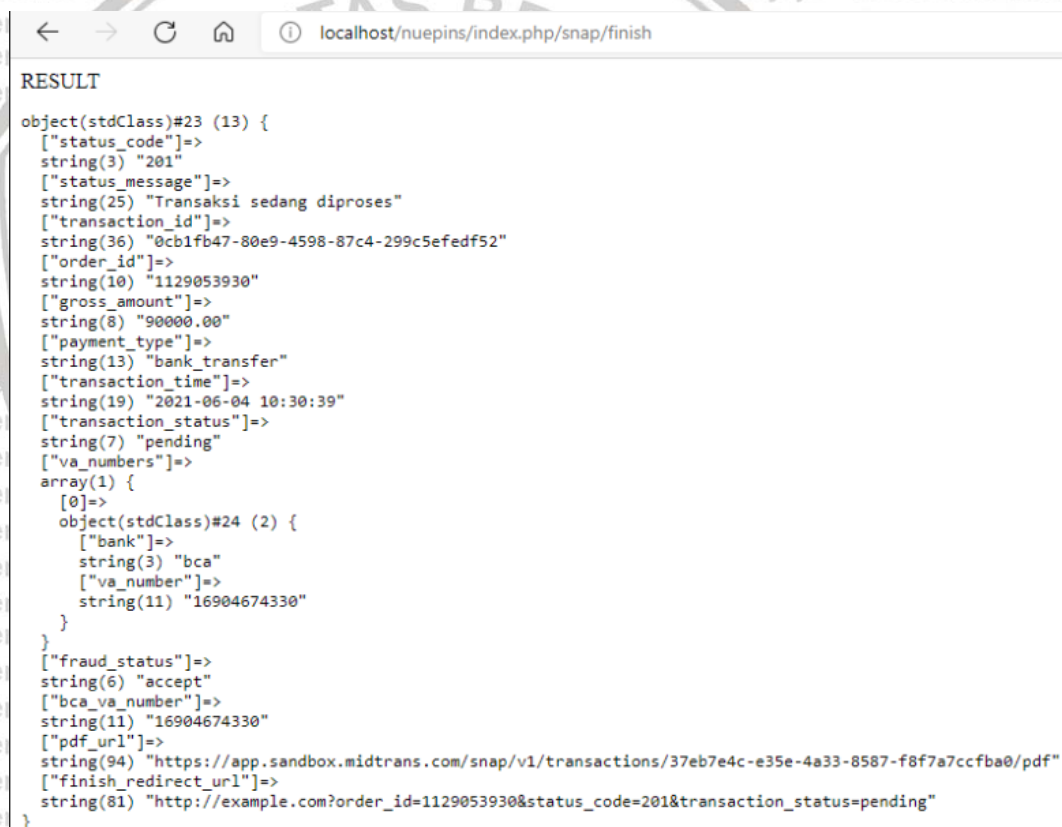
| No | Midtrans.php |
|----|--|
| 1 | public function config(\$params) |
| 2 | { |
| 3 | Midtrans::\$serverKey = \$params['server_key']; |
| 4 | Midtrans::\$isProduction = \$params['production']; |
| 5 | } |
| 6 | public static function getSnapToken(\$params) |
| 7 | { |

```

5     $result = Midtrans::post(
6         Midtrans::getSnapBaseUrl() . '/transactions',
7         Midtrans::$serverKey,
8         $params);
9     return $result->token;
}

```

Pengiriman data ke *API Midtrans* dikirimkan dengan menggunakan format *JSON*. Data yang dikirimkan merupakan data produk, data *user*, data alamat, data total pembayaran, data jumlah pembelian, dan data metode pembayaran. Format data *JSON* dapat dilihat pada gambar 6.1



```

RESULT
object(stdClass)#23 (13) {
  ["status_code"]=>
  string(3) "201"
  ["status_message"]=>
  string(25) "Transaksi sedang diproses"
  ["transaction_id"]=>
  string(36) "0cb1fb47-80e9-4598-87c4-299c5efedf52"
  ["order_id"]=>
  string(10) "1129053930"
  ["gross_amount"]=>
  string(8) "90000.00"
  ["payment_type"]=>
  string(13) "bank_transfer"
  ["transaction_time"]=>
  string(19) "2021-06-04 10:30:39"
  ["transaction_status"]=>
  string(7) "pending"
  ["va_numbers"]=>
  array(1) {
    [0]=>
    object(stdClass)#24 (2) {
      ["bank"]=>
      string(3) "bca"
      ["va_number"]=>
      string(11) "16904674330"
    }
  }
  ["fraud_status"]=>
  string(6) "accept"
  ["bca_va_number"]=>
  string(11) "16904674330"
  ["pdf_url"]=>
  string(94) "https://app.sandbox.midtrans.com/snap/v1/transactions/37eb7e4c-e35e-4a33-8587-f8f7a7ccfba0/pdf"
  ["finish_redirect_url"]=>
  string(81) "http://example.com?order_id=1129053930&status_code=201&transaction_status=pending"
}

```

Gambar 6.1 Result Data JSON API Midtrans

Gambar 6.1 Merupakan bentuk dari format data *JSON* yang dikirimkan ke *API Midtrans* setelah melakukan proses pemilihan metode pembayaran. Struktur dari data *json* tersebut merupakan objek yang berisikan data *array*.

6.4 Implementasi Kode Program

Kode program yang diimplementasikan pada pengembangan sistem informasi toko *online* Nue Pins, menggunakan bahasa pemrograman *PHP*. Implementasi kode program berisi baris fungsi dari kelas yang telah didefinisikan pada tahap perancangan. Pada tahap ini, akan ditampilkan kode sumber pada masing – masing kelas berdasarkan *function* yang telah dicantumkan pada perancangan *sequence diagram*.

6.4.1 Implementasi Login User

Implementasi kode program *login user* merupakan fungsi untuk aktor *User* melakukan *login* pada system. Terdapat *variable email* yang digunakan untuk melakukan pengecekan data *email* yang telah diinputkan oleh *user* ke *database*. Terdapat seleksi kondisi dimana jika *email* tidak terdaftar, maka *user* akan gagal *login*. Jika *email* terdaftar namun *password* yang dimasukan salah, maka *user* gagal *login*. Dan Jika *email* dan *password* yang diinputkan adalah benar, maka *email* user akan disimpan kedalam *session* dan *user* berhasil untuk *login*. Tabel 6.10 akan menampilkan kode sumber dari fungsi *login user*.

Tabel 6.10 Implementasi Kode Sumber Login User

| No | Control login.php |
|----|--|
| 1 | public function checkUserAccount(){ |
| 2 | \$email = \$this->input->post('emailLogin'); |
| 3 | \$password = \$this->input->post('pwdLogin'); |
| 4 | \$user = \$this->model_login->readUser(\$email); |
| 5 | if(\$user){ |
| 6 | if(\$user['is_emailActive'] == 1){ |
| 7 | if(password_verify(\$password, |
| 8 | \$user['password'])){ |
| 9 | \$data = ['email' => \$user['email']]; |
| 10 | \$this->session->set_userdata(\$data); |
| 11 | redirect(base_url()); |
| 12 | }else{ |
| 13 | \$this->session->set_flashdata('message', '<div class="alert |
| 14 | alert-danger" role="alert" style="text-align:center;"> |
| 15 | Incorrect Password</div>'); |
| 16 | redirect(base_url()); |
| 17 | } |
| 18 | }else{ |
| 19 | \$this->session->set_flashdata('message', '<div class="alert |
| 20 | alert-danger" role="alert" style="text-align:center;"> |
| 21 | Email has not been verified yet!</div>'); |
| 22 | redirect(base_url()); |
| 23 | echo "<script type='text/javascript'> |
| 24 | \$(document).ready(function(){ |
| 25 | \$('#myModal').modal('show'); |
| 26 | }); |
| 27 | }</script>"; |
| 28 | }else{ |
| 29 | \$this->session->set_flashdata('message', '<div class="alert |
| 30 | alert-danger" role="alert" style="text-align:center;"> |
| 31 | Email not registered</div>'); |
| 32 | redirect(base_url()); |


```
}
}
```

6.4.2 Implementasi Login Admin

Implementasi kode program *login admin*, merupakan fungsi untuk aktor *admin* melakukan *login* kedalam sistem khusus *admin*. Terdapat *variable email* dan *password* yang didapatkan dari hasil inputan aktor *admin*. *Variable* tersebut akan digunakan untuk proses pengecekan data akun *admin* di *database*. Terdapat seleksi kondisi, Jika *email* yang diinputkan salah, maka aktor *admin* akan gagal melakukan *login*. Jika *email* yang diinputkan benar, namun *password* yang diinputkan salah, maka aktor *admin* gagal melakukan *login*. Dan Jika *email* dan *password* yang diinputkan adalah benar, maka aktor *admin* berhasil melakukan *login*, dan data *email admin* akan disimpan kedalam *session*. Tabel 6.11 akan menampilkan kode sumber dari fungsi *login admin*.

Tabel 6.11 Implementasi Kode Sumber Login Admin

| No | Control_login.php |
|----|--|
| 1 | public function checkAdminAccount(){ |
| 2 | \$email = \$this->input->post('emailAdmin'); |
| 3 | \$password = \$this->input->post('pwdAdmin'); |
| 4 | \$admin = \$this->model_login->readAdmin(\$email); |
| 5 | if(\$admin){ |
| 6 | if(password_verify(\$password, |
| 7 | \$admin['password']))){ |
| 8 | \$data = ['emailadmin' => \$admin['email'] |
| 9 |]; |
| 10 | \$this->session->set_userdata(\$data); |
| 11 | redirect(base_url('admin/home')); |
| 12 | }else{ |
| 13 | \$this->session->set_flashdata('message', '<div class="alert |
| 14 | alert-danger" role="alert"> |
| 15 | Incorrect Password</div>'); |
| 16 | redirect(base_url('admin')); |
| 17 | }else{ |
| 18 | \$this->session->set_flashdata('message', '<div class="alert |
| 19 | alert-danger" role="alert"> |
| 20 | You are not admin, please click this to |
| 21 | leave</div>'); |
| 22 | redirect(base_url('admin')); |
| | } |
| | } |

6.4.3 Implementasi Register

Implementasi *Register* merupakan fungsi yang digunakan untuk aktor *user* melakukan registrasi akun kedalam sistem. Terdapat beberapa *variable* yang diambil dari hasil input *user* untuk dimasukan kedalam *database*. Setelah *user* berhasil melakukan pengisian, terdapat *variable isEmailActive* yang diberikan nilai 0. Nilai tersebut menandakan bahwa akun *user* berhasil terdaftar namun belum tervalidasi. Method *kirimEmail* akan dipanggil untuk mengirimkan *link* atau tautan

validasi ke *email* yang telah diinputkan oleh *user*. Setelah *User* mengakses *link* yang diberikan di *email*, maka akun *user* berhasil divalidasi dan dapat digunakan. *Variable isEmailActive* akan diubah menjadi 1. Tabel 6.12 akan menampilkan kode sumber dari fungsi *register*.

Tabel 6.12 Implementasi Kode Sumber Register

| No | Control_register.php |
|----|---|
| 1 | public function registerAccount(){ |
| 2 | \$this->form_validation->set_rules('email','Email', |
| 3 | 'is_unique[user.email],[|
| 4 | 'is_unique' => 'Email sudah terdaftar' |
| 5 |]); |
| 6 | if(\$this->form_validation->run() == false){ |
| 7 | \$this->load->view('v_register'); |
| 8 | }else{ |
| 9 | \$i = \$this->input; |
| 10 | \$email = \$i->post('email'); |
| 11 | \$data=array('firstname'=>htmlspecialchars(\$i- |
| 12 | >post('firstname'), |
| 13 | 'lastname' => htmlspecialchars(\$i->post('lastname'), |
| 14 | 'no_handphone'=>htmlspecialchars(\$i-> |
| 15 | post('no_handphone'), |
| 16 | 'password'=>password_hash(\$i- |
| 17 | >post('pwd'),PASSWORD_DEFAULT), |
| 18 | 'email' => |
| 19 | htmlspecialchars(\$email), |
| 20 | 'is_emailActive' => 0 |
| 21 |); |
| 22 | \$token = base64_encode(random_bytes(32)); |
| 23 | \$user_token = array ('email'=> \$email, |
| 24 | 'token' => \$token, |
| 25 | 'date_created' =>time() |
| 26 |); |
| 27 | \$this->model_register->createAccount(\$data); |
| 28 | \$this->model_register- |
| 29 | >createToken(\$user_token); |
| 30 | \$this-> kirimEmail(\$token, 'verify'); |
| 31 | \$headcount =[]; |
| 32 | \$data = array('nama' => '', |
| 33 | 'headcount' => \$headcount, |
| 34 | 'email' => \$email, |
| 35 | 'tittle' => '#Nue Experience |
| 36 | nuepins.com', |
| 37 | 'isi'=> |
| 38 | 'layoutuser/v_verify'); |
| 39 | \$this->load- |
| 40 | >view('layoutuser/wrapperuser',\$data,FALSE); |
| | } |

6.4.4 Implementasi Logout User

Pada implementasi kode sumber *logout User*, terdapat pemanggilan fungsi dari *php* untuk menghapus *session*. *Session* yang dihapus merupakan *session email*

yang dihasilkan setelah melakukan *login*. Tabel 6.13 akan menampilkan kode sumber dari fungsi *logout user*.

Tabel 6.13 Implementasi Kode Sumber Logout User

| No | Control_login.php |
|----|-------------------------------|
| 1 | public function logoutUser(){ |
| 2 | unset(\$_SESSION['email']); |
| 3 | redirect(base_url()); |
| | } |

6.4.5 Implementasi Logout Admin

Pada implementasi kode sumber *logout Admin*, terdapat pemanggilan fungsi untuk menghapus *session*. *Session* yang dihapus merupakan *session email* yang dihasilkan setelah melakukan *login*. Fungsi penghapusan *session* tersebut merupakan fungsi yang telah disediakan dari Bahasa pemrograman PHP. Tabel 6.14 akan menampilkan kode sumber dari fungsi *logout admin*.

Tabel 6.14 Implementasi Kode Sumber Logout Admin

| No | Control_login.php |
|----|----------------------------------|
| 1 | public function logoutAdmin(){ |
| 2 | unset(\$_SESSION['emailadmin']); |
| 3 | redirect(base_url('admin')); |
| | } |

6.4.6 Implementasi Menampilkan Produk

Pada implementasi kode sumber menampilkan produk, terdapat beberapa seleksi kondisi. Seleksi kondisi tersebut merupakan pemilihan kategori yang dipilih oleh *user*. Kategori yang dipilih oleh *user* akan disimpan kedalam *session* untuk digunakan sebagai penyimpanan histori akses. Pemanggilan data produk akan dilakukan berdasarkan kategori yang pilih oleh *user*. Tabel 6.15 akan menampilkan kode sumber dari fungsi menampilkan produk.

Tabel 6.15 Implementasi Kode Sumber Menampilkan Produk

| No | Control_produk.php |
|----|---|
| 1 | public function loadKategori(\$kategori) |
| 2 | { |
| 3 | if(\$this->session->has_userdata('email')){ |
| 4 | \$email = \$this->session->userdata('email'); |
| 5 | \$produk = \$this->model_produk- |
| 6 | >readKategori(\$kategori); |
| 7 | \$prd = \$this->model_produk->readProduk(); |
| 8 | \$firstname = \$this->model_login- |
| 9 | >readUser(\$email); |
| 10 | \$message = "Hai!". \$firstname['firstname']; |
| 11 | \$id_user = \$firstname['iduser']; |
| 12 | \$headcount = \$this->model_keranjang- |
| 13 | >readKeranjang(\$id_user); |
| 14 | \$data = array('nama' => \$message, |
| 15 | 'title' => \$title, |
| 16 | 'produk' => \$produk, |


```

17 'headcount' => $headcount,
18 'prd'=> $prd,
19 'tittle' => '#Nue Experience |
20 nuepins.com',
21 'isi' =>
22 'layoutuser/v_kategori');
23 $this->load-
24 >view('layoutuser/wrapperuser', $data, FALSE);
25 }else{
26 $produk = $this->model_produk-
27 >readKategori($kategori);
28 $prd = $this->model_produk->readProduk();
29 $headcount = [];
30 $data = array('nama' => '',
31 'title'=>$title,
32 'tittle' => '#Nue Experience |
33 nuepins.com',
34 'produk' => $produk,
35 'prd' => $prd,
36 'headcount'=> $headcount,
37 'isi' =>
38 'layoutuser/v_kategori');
39 $this->load-
40 >view('layoutuser/wrapperuser', $data, FALSE);
41 }
42 }
}

```

6.4.7 Implementasi Keranjang Belanja

Pada implementasi kode sumber keranjang belanja, terdapat seleksi kondisi untuk melakukan pengecekan apakah *user* sudah *login* atau belum. Jika *user* sudah *login*, maka data *email* user akan digunakan untuk mengambil data keranjang belanja dari *database*. Jika *user* belum melakukan *login*, maka *user* tidak bisa melakukan akses ke halaman keranjang belanja, dan akan diarahkan ke halaman utama. Tabel 6.16 akan menampilkan kode sumber dari fungsi keranjang belanja.

Tabel 6.16 Implementasi Kode Sumber Keranjang Belanja

| No | Control keranjang.php |
|----|--|
| 1 | public function loadKeranjangUser() { |
| 2 | if(\$this->session->has_userdata('email')) { |
| 3 | //Udah Login |
| 4 | \$email = \$this->session->userdata('email'); |
| 5 | \$firstname = \$this->model_login- |
| 6 | >readUser(\$email); |
| 7 | \$message = "Hai! ". \$firstname['firstname']; |
| 8 | \$id_user = \$firstname['id_user']; |
| 9 | \$keranjang = \$this->model_keranjang- |
| 10 | >readKeranjang(\$id_user); |
| 11 | \$headcount = \$this->model_keranjang- |
| 12 | >readKeranjang(\$id_user); |
| 13 | \$data = array('nama' => \$message, |
| 14 | 'keranjang' => \$keranjang, |
| 15 | 'headcount' => \$headcount, |

```

16 'tittle' => '#Nue Experience |
17 nuepins.com',
18 'layoutuser/v_keranjang');
19 $this->load-
20 >view('layoutuser/wrapperuser', $data, FALSE);
21 }else{
22 // Belom login
23 $headcount = [];
24 $data = array('nama' => '',
25 'headcount' => $headcount,
26 'tittle' => '#Nue Experience |
27 nuepins.com',
28 'isi' =>
29 'layoutuser/v_firstpage');
30 $this->load-
31 >view('layoutuser/wrapperuser', $data, FALSE);
32 }

```

6.4.8 Implementasi Menghapus Produk

Pada implementasi kode sumber menghapus produk, terdapat *variable id_keranjang* yang digunakan untuk melakukan penghapusan data dari *database*. Setelah data berhasil terhapus, maka *user* akan diarahkan ke halaman keranjang belanja. Tabel 6.17 akan menampilkan kode sumber dari fungsi menghapus produk.

Tabel 6.17 Implementasi Kode Sumber Menghapus produk

| No | Control_keranjang.php |
|----|--|
| 1 | public function deleteDataKeranjang(\$id_keranjang){ |
| 2 | \$data = array('id_keranjang' => \$id_keranjang); |
| 3 | \$this->model_keranjang->deleteKeranjang(\$data); |
| 4 | redirect(base_url('control_keranjang/loadKeranjangU |
| | ser'), 'refresh'); |
| | } |

6.4.9 Implementasi Menentukan Jumlah Pembelian

Pada implementasi menentukan jumlah pembelian, terdapat *variable id_keranjang* dan *jumlah_pembelian*. *id_keranjang* belanja digunakan untuk melakukan pemilihan data dari *database*, dan *variable jumlah_pembelian* merupakan data dari hasil inputan *user*, yang akan digunakan sebagai *update* data baru didalam tabel keranjang berdasarkan *id_keranjang*. Tabel 6.18 akan menampilkan kode sumber dari fungsi menentukan jumlah pembelian

Tabel 6.18 Implementasi Kode Sumber Menentukan Jumlah pembelian

| No | Control_keranjang.php |
|----|--|
| 1 | public function updateDataKeranjang(\$id_keranjang, \$jumlah_p |
| 2 | embelian){ |
| 3 | \$data = array('id_keranjang' => \$id_keranjang, |
| 4 | |


```

5     'jumlah_pembelian' => $jumlah_pembe
6     lian);
7     $this->model_keranjang->updateKeranjang($data);
8     redirect(base_url('control_keranjang/loadKeranjangU
    ser'),'refresh');
    }

```

6.4.10 Implementasi Membuat Pemesanan

Pada implementasi kode sumber membuat pemesanan, terdapat fungsi *finish()* yang merupakan fungsi dari *plugin API Midtrans*. Fungsi tersebut digunakan untuk mengirimkan data pembelian ke *endpoint API Midtrans*. Dalam fungsi *finish()* juga terdapat alur untuk melakukan pembuatan data pembelian kedalam *database Nue Pins*. Tabel 6.19 akan menampilkan kode sumber dari fungsi membuat pemesanan.

Tabel 6.19 Implementasi Kode Sumber Membuat Pemesanan

| No | Snap.php |
|----|---|
| 1 | public function finish() |
| 2 | { |
| 3 | \$result = json_decode(\$this->input- |
| 4 | >post('result_data')); |
| 5 | echo 'RESULT <pre>'; |
| 6 | var_dump(\$result); |
| 7 | echo '</pre>'; |
| 8 | |
| 9 | // Insert into DBDBDB |
| 10 | \$email = \$this->session->userdata('email'); |
| 11 | \$firstname = \$this->model_login->readUser(\$email); |
| 12 | \$id_user = \$firstname['id_user']; |
| 13 | \$keranjang = \$this->model_keranjang- |
| 14 | >readKeranjang(\$id_user); |
| 15 | |
| 16 | // if payment bank trf, echannel, store |
| 17 | if(\$result->payment_type == 'bank_transfer'){ |
| 18 | foreach(\$keranjang as \$keranjang){ |
| 19 | \$vanumb = ""; |
| 20 | \$bank = ""; |
| 21 | foreach(\$result->va_numbers as \$row){ |
| 22 | \$vanumb = \$row->va_number; |
| 23 | \$bank = \$row->bank; |
| 24 | } |
| 25 | \$data = array('order_id' => \$result- |
| 26 | >order_id, |
| 27 | 'id_produk' => \$keranjang- |
| 28 | >id_produk, |
| 29 | 'id_user' => \$id_user, |
| 30 | 'quantity' => \$keranjang- |
| 31 | >jumlah_pembelian, |
| 32 | 'status_pembayaran' => \$res |
| 33 | ult->transaction_status, |
| 34 | 'status_pemesanan' => \$resu |
| 35 | lt->status_message, |
| 36 | 'payment_type' => "Bank Tra |
| 37 | nsfer ". \$bank, |

```

38         'kode_pembayaran' => $vanum
39     b,
40         'order_date' => date("Y-m-
41     d"),
42         'pdf_url' => $result-
43     >pdf_url
44     );
45     $this->model_pemesanan-
46     >createPemesanan($data);
47     }
48     // Mandiri
49     }else if($result->payment_type == 'echannel'){
50     foreach($keranjang as $keranjang){
51     $data = array( 'order_id' => $result-
52     >order_id,
53     'id_produk' => $keranjang-
54     >id_produk,
55     'id_user' => $id_user,
56     'quantity' => $keranjang-
57     >jumlah_pembelian,
58     'status_pembayaran' => $res
59     ult->transaction_status,
60     'status_pemesanan' => $resu
61     lt->status_message,
62     'payment_type' => "Bank Tra
63     nsfer mandiri",
64     'kode_pembayaran' => $resul
65     t->bill_key,
66     'order_date' => date("Y-m-
67     d"),
68     'pdf_url' => $result-
69     >pdf_url
70     );
71     $this->model_pemesanan-
72     >createPemesanan($data);
73     }
74     // Indomart & Alfa
75     }else if($result->payment_type == 'cstore'){
76     foreach($keranjang as $keranjang){
77     $data = array( 'order_id' => $result-
78     >order_id,
79     'id_produk' => $keranjang-
80     >id_produk,
81     'id_user' => $id_user,
82     'quantity' => $keranjang-
83     >jumlah_pembelian,
84     'status_pembayaran' => $res
85     ult->transaction_status,
86     'status_pemesanan' => $resu
87     lt->status_message,
88     'payment_type' => "CSTORE",
89     'kode_pembayaran' => $resul
90     t->payment_code,
91     'order_date' => date("Y-m-
92     d"),
93     'pdf_url' => $result-
94     >pdf_url
95     );
96     }

```



```

97     $this->model_pemesanan-
98     >createPemesanan($data);
    }
}

```

6.4.11 Implementasi *Track Pemesanan*

Pada implementasi kode sumber *track pemesanan*, terdapat seleksi kondisi apakah *user* sudah melakukan *login* atau belum. Jika *user* sudah melakukan *login*, maka data *email* yang disimpan dalam *session* akan digunakan untuk mengambil data pemesanan berdasarkan *id_user*. Jika *user* belum melakukan *login*, *user* akan tetap diarahkan ke halaman *track pemesanan* namun, tidak terdapat data yang ditampilkan. Tabel 6.20 akan menampilkan kode sumber dari fungsi *track pemesanan*

Tabel 6.20 Implementasi Kode Sumber *Track Pemesanan*

| No | Control pemesanan.php |
|----|---|
| 1 | public function loadPemesananUser(){ |
| 2 | if(\$this->session->has_userdata('email')){ |
| 3 | \$email = \$this->session->userdata('email'); |
| 4 | \$firstname = \$this->model_login- |
| 5 | >readUser(\$email); |
| 6 | \$message = "Hai! ".\$firstname['firstname']; |
| 7 | \$id_user = \$firstname['id_user']; |
| 8 | \$headcount = \$this->model_keranjang- |
| 9 | >readKeranjang(\$id_user); |
| 10 | \$pemesanan = \$this->model_pemesanan- |
| 11 | >readPemesanan(\$id_user); |
| 12 | \$data = array('nama' => \$message, |
| 13 | 'headcount' => \$headcount, |
| 14 | 'pemesanan' => \$pemesanan, |
| 15 | 'title' => 'Checkout', |
| 16 | 'tittle' => '#Nue Experience |
| 17 | nuepins.com', |
| 18 | 'isi' => 'layoutuser/v_status |
| 19 | '); |
| 20 | \$this->load- |
| 21 | >view('layoutuser/wrapperuser', \$data, FALSE); |
| 22 | }else{ |
| 23 | \$headcount = []; |
| 24 | \$data = array('nama' => '', |
| 25 | 'headcount' => \$headcount, |
| 26 | 'tittle' => '#Nue Experience |
| 27 | nuepins.com', |
| 28 | 'isi' => 'layoutuser/v_status |
| 29 | '); |
| 30 | \$this->load- |
| 31 | >view('layoutuser/wrapperuser', \$data, FALSE); |



6.4.12 Implementasi Informasi Produk

Pada implementasi kode sumber informasi produk, terdapat pemanggilan fungsi dari *model_produk* yang akan mengambil semua data produk. Tabel 6.21 akan menampilkan kode sumber dari fungsi informasi produk.

Tabel 6.21 Implementasi Kode Sumber Informasi Produk

| No | Control_produk.php |
|----|--|
| 1 | public function loadDataProduk(){ |
| 2 | if(\$this->session->has_userdata('emailadmin')){ |
| 3 | \$email = \$this->session->userdata('emailadmin'); |
| 4 | \$firstname = \$this->model_login->readAdmin(\$email); |
| 5 | \$produk = \$this->model_produk->readProduk(); |
| 6 | \$prd = \$this->model_produk->readProduk(); |
| 7 | \$data = array('title' => 'Nuepins Administrator', |
| 8 | 'nama' => \$firstname['firstname'], |
| 9 | 'produk' => \$produk, |
| 10 | 'prd' => \$prd, |
| 11 | 'isi' => 'layoutadmin/v_produk'); |
| 12 | \$this->load- |
| 13 | ->view('layoutadmin/wrapper', \$data, FALSE); |
| 14 | }else{ |
| 15 | \$this->session- |
| 16 | ->set_flashdata('message', '<div class="alert alert- |
| 17 | danger" role="alert"> |
| 18 | You are not admin, please click |
| 19 | this to leave</div>'); |
| 20 | redirect(base_url('admin')); |
| | } |
| | } |

6.4.13 Implementasi Menambah Data Produk

Pada implementasi kode sumber menambah data produk, terdapat pemanggilan *library* yang disediakan oleh *framework code igniter* untuk menyimpan data gambar. Terdapat konfigurasi untuk opsi penyimpanan data gambar dan jika data gambar yang dimasukan tidak sesuai konfigurasi, maka penyimpanan data produk akan gagal. Jika pe data gambar yang dimasukan sesuai konfigurasi, maka penyimpanan data produk akan berhasil dan akan dilakukan pemanggilan fungsi dari *model_produk* untuk menyimpan data gambar yang telah dimasukan. Tabel 6.22 akan menampilkan kode sumber dari fungsi menambah data produk.

Tabel 6.22 Implementasi Kode Sumber Menambah Data Produk

| No | Control_produk.php |
|----|---|
| 1 | public function buatProduk(){ |
| 2 | \$config['upload_path'] = './assets/upload/'; |
| 3 | \$config['allowed_types'] = 'gif jpg png jpeg'; |
| 4 | \$config['max_size'] = '2400'; |
| 5 | \$config['max_width'] = '2024'; |
| 6 | \$config['max_height'] = '2024'; |
| 7 | \$this->load->library('upload', \$config); |


```

8         if(! $this->upload->do_upload('gambar_produk')){
9             if($this->session->has_userdata('emailadmin')){
10                $email = $this->session-
11                >userdata('emailadmin');
12                $firstname = $this->model_login-
13                >readAdmin($email);
14                $produk = $this->model_produk-
15                >readProduk();
16                $data = array('title'=>'Nuepins Adminis
17                trator',
18                'nama'=> $firstname['firstn
19                ame'],
20                'produk'=>$produk,
21                'isi' =>'layoutadmin/v_pro
22                duk');
23                $this->load-
24                >view('layoutadmin/wrapper',$data,FALSE);
25            }
26        }else{
27            $upload_gambar = array ('upload_data'=>$this-
28            >upload->data());
29            $config['image_library'] = 'gd2';
30            $config['source_image'] = './assets/upload'. $up
31            load_gambar['upload_data']['file_name'];
32            $config['new_image'] = './assets/upload';
33            $config['create_thumb'] = TRUE;
34            $config['maintain_ratio'] = TRUE;
35            $config['width'] = 250;
36            $config['height'] = 250;
37
38            $this->load->library('image_lib', $config);
39            $this->image_lib->resize();
40            $i = $this->input;
41            $harga = $i->post('harga_produk');
42            $numbers = preg_replace('/[^0-
43            9]/', '', $harga);
44            $data = array( 'nama_produk' => $i-
45            >post('nama_produk'),
46            'gambar_produk'=> $upload_
47            gambar['upload_data']['file_name'],
48            'harga_produk' => $numbers,
49            'kategori' => $i-
50            >post('kategori'),
51            'stock' => $i-
52            >post('stock'),
53            'deskripsi'=> $i-
54            >post('deskripsi'),
55            );
56            $this->model_produk->createProduk($data);
57            redirect(base_url('admin/produk'),'refresh');
58        }
59    }

```

6.4.14 Implementasi Mengubah Data Produk

Pada implementasi kode sumber mengubah data produk, terdapat *variable id_produk* yang digunakan untuk memilih data produk yang akan dilakukan


```

49     'nama_produk' => $i-
50     >post('nama_produk'),
51     'gambar_produk' => $upload_
52     gambar['upload_data']['file_name'],
53     'harga_produk' => $numbers,
54     'kategori' => $i-
55     >post('kategori'),
56     'stock' => $i-
57     >post('stock'),
58     'deskripsi' => $i-
59     >post('deskripsi'),
60     );
61     $this->model_produk->updateProduk($data);
62     redirect(base_url('admin/produk'),'refresh!');
63 }else{
64     $i = $this->input;
65     $harga = $i->post('harga_produk');
66     $numbers = preg_replace('/[^0-
67     9]/','',$harga);
68     $data = array(
69         'id_produk' => $id_produk,
70         'nama_produk' => $i-
71         >post('nama_produk'),
72         'harga_produk' => $numbers,
73         'kategori' => $i-
74         >post('kategori'),
75         'stock' => $i-
76         >post('stock'),
77         'deskripsi' => $i-
78         >post('deskripsi'),
79     );
80     $this->model_produk->updateProduk($data);
81     redirect(base_url('admin/produk'),'refresh!');
82 }

```

6.4.15 Implementasi Menghapus Data Produk

Pada implementasi kode sumber menghapus data produk, terdapat *variable id_produk* yang digunakan untuk memilih data produk yang akan dihapus. Terdapat pemanggilan fungsi dari *model_produk* untuk menghapus data produk berdasarkan *id_produk*. Tabel 6.24 akan menampilkan kode sumber dari fungsi menghapus data produk.

Tabel 6.24 Implementasi Kode Sumber Menghapus Data Produk

| No | Control produk.php |
|----|--|
| 1 | public function hapusProduk(\$id_produk){ |
| 2 | \$data = array('id_produk' => \$id_produk); |
| 3 | \$this->model_produk->deleteProduk(\$data); |
| 4 | redirect(base_url('admin/produk'),'refresh!'); |
| | } |


```

11 );
12 $this->model_pemesanan->updatePemesanan($data);
13 }
14 redirect(base_url('control_pemesanan/loadDataPemesa
15 nan'));
    }
    
```

6.4.18 Implementasi Dashboard

Pada implementasi kode sumber *dashboard*, terdapat beberapa *variable* yang digunakan untuk mengambil data jumlah harga pembelian berdasarkan masing – masing bulan. Dan pada implementasi kode sumber *dashboard* juga terdapat *variable – variable* yang diinisiasikan untuk mengambil data jumlah total pendapatan, jumlah order yang berhasil dilakukan, dan jumlah order yang masih *pending* atau menunggu. Tabel 6.27 akan menampilkan kode sumber dari fungsi *dashboard*.

Tabel 6.27 Implementasi Dashboard

| No | Control dashboard.php |
|----|--|
| 1 | public function index() |
| 2 | { |
| 3 | if(\$this->session->has_userdata('emailadmin')){ |
| 4 | \$email = \$this->session- |
| 5 | }>userdata('emailadmin'); |
| 6 | \$firstname = \$this->model_login- |
| 7 | }>readAdmin(\$email); |
| 8 | \$pemesanan = \$this->model_pemesanan- |
| 9 | }>readPemesananDashboard(); |
| 10 | \$pmsn = \$this->model_pemesanan- |
| 11 | }>readPemesananPending(); |
| 12 | \$gross_januari = \$this->model_pemesanan- |
| 13 | }>incomeJanuari(); |
| 14 | \$gross_februari = \$this->model_pemesanan- |
| 15 | }>incomeFebruari(); |
| 16 | \$gross_maret = \$this->model_pemesanan- |
| 17 | }>incomeMaret(); |
| 18 | \$gross_april = \$this->model_pemesanan- |
| 19 | }>incomeApril(); |
| 20 | \$gross_mei = \$this->model_pemesanan- |
| 21 | }>incomeMei(); |
| 22 | \$gross_juni = \$this->model_pemesanan- |
| 23 | }>incomeJuni(); |
| 24 | \$gross_juli = \$this->model_pemesanan- |
| 25 | }>incomeJuli(); |
| 26 | \$gross_agustus = \$this->model_pemesanan- |
| 27 | }>incomeAgustus(); |
| 28 | \$gross_september = \$this->model_pemesanan- |
| 29 | }>incomeSeptember(); |
| 30 | \$gross_oktober = \$this->model_pemesanan- |
| 31 | }>incomeOktober(); |
| 32 | \$gross_november = \$this->model_pemesanan- |
| 33 | }>incomeNovember(); |
| 34 | |

```

35     $gross_desember = $this->model_pemesanan-
36     >incomeDesember();
37     $data = array( 'title' => 'Nuepins Administrat
38     or',
39     'pemesanan' => $pemesanan,
40     'pmsn' => $pmsn,
41     'gross_januari' => $gross_janua
42     ri,
43     'gross_februari' => $gross_febr
44     uari,
45     'gross_maret' => $gross_maret,
46     'gross_april' => $gross_april,
47     'gross_mei' => $gross_mei,
48     'gross_juni' => $gross_juni,
49     'gross_juli' => $gross_juli,
50     'gross_agustus' => $gross_agust
51     us,
52     'gross_september' => $gross_sep
53     tember,
54     'gross_oktober' => $gross_oktob
55     er,
56     'gross_november' => $gross_nove
57     mber,
58     'gross_desember' => $gross_dese
59     mber,
60     'nama' => $firstname['firstname
61     '],
62     'isi' => 'layoutadmin/v_dashb
63     oard' );
64     $this->load-
65     >view('layoutadmin/wrapper', $data, FALSE);
66     }else{
67     $this->session->
68     >set_flashdata('message', '<div class="alert alert-
69     danger" role="alert">
70     You are not admin, please click <a href="home">
71     this </a>to leave</div>');
72     redirect(base_url('admin'));
73     }
74     }

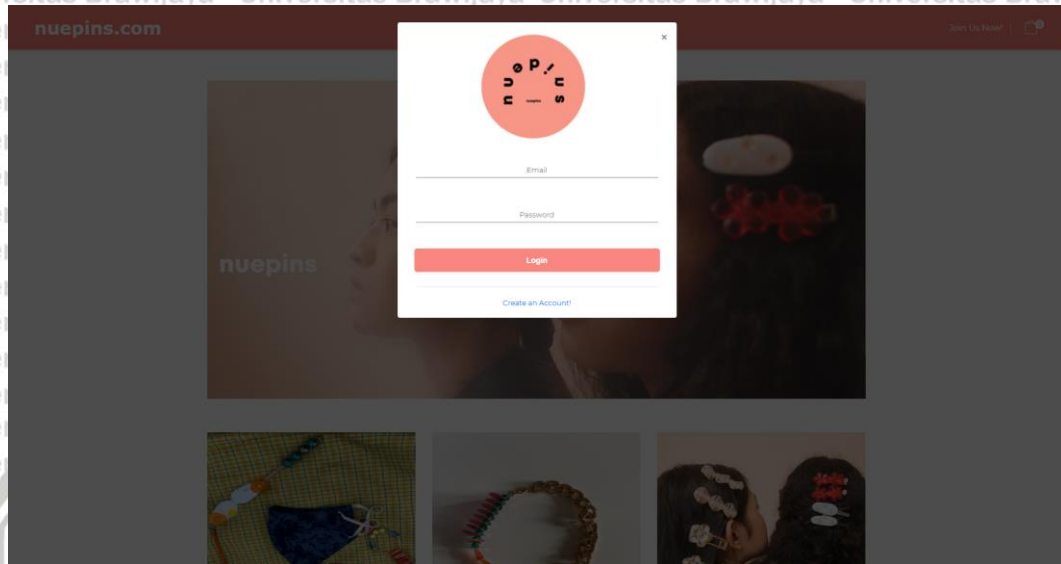
```

6.5 Implementasi Antarmuka

Implementasi antarmuka yang akan ditampilkan merupakan antarmuka dari masing – masing kebutuhan yang telah didefinisikan, diantaranya antarmuka *login user*, antarmuka *login admin*, antarmuka *register*, antarmuka *logout user* dan *admin*, antarmuka menampilkan produk, antar muka keranjang belanja, antarmuka menghapus produk, antarmuka menentukan jumlah pembelian, antarmuka membuat pemesanan, antarmuka *track pemesanan*, antarmuka informasi produk, antarmuka menambah, mengubah, dan menghapus data produk, antarmuka informasi pemesanan, dan antarmuka *dashboard*

6.5.1 Antarmuka Login User

Antarmuka *login user* merupakan tampilan saat *user* melakukan *login* untuk masuk ke dalam sistem. Pada tampilan *login user*, terdapat *modal* yang menampilkan inputan *email* dan *password*. Pada bagian bawah terdapat *link* untuk membuka *modal register* jika *user* belum mempunyai akun. Implementasi antarmuka *login user* akan ditampilkan pada gambar 6.2.



Gambar 6.2 Antarmuka Login User

6.5.2 Antarmuka Login Admin

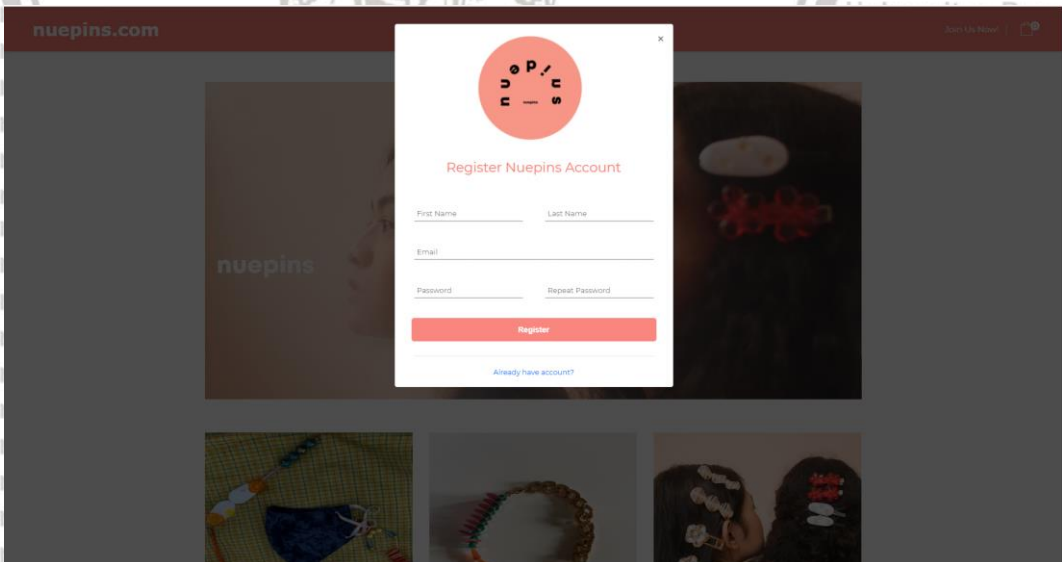
Antarmuka *login admin* merupakan tampilan saat *admin* melakukan *login* untuk masuk ke dalam sistem khusus *admin*. Pada tampilan *login admin*, terdapat inputan *email* dan *password*. Implementasi antarmuka *login admin* akan ditampilkan pada gambar 6.3.



Gambar 6.3 Antarmuka *Login Admin*

6.5.3 Antarmuka Register

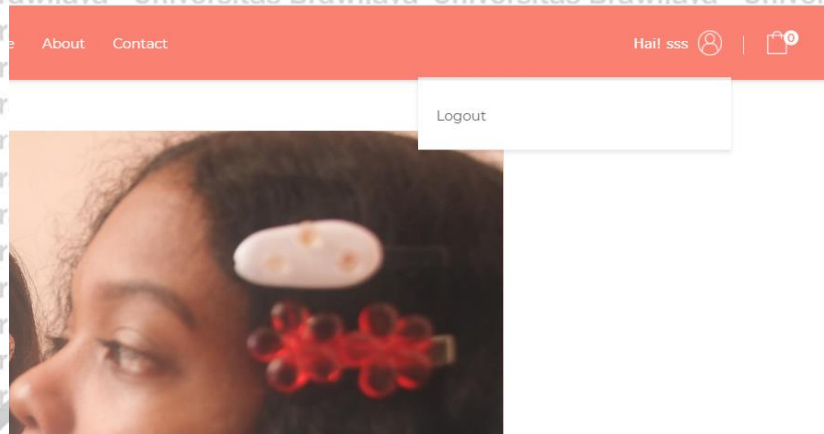
Antarmuka *register* merupakan tampilan saat aktor *user* melakukan *registrasi* untuk mendaftar akun. Pada tampilan *registrasi*, terdapat *modal* yang menampung inputan *firstname*, *lastname*, *email*, *phone number*, dan *password*. Pada bagian bawah, terdapat *link* untuk menampilkan *modal login* jika *user* sudah mempunyai *akun*. Antarmuka *register* akan ditampilkan pada gambar 6.4.



Gambar 6.4 Antarmuka *Register*

6.5.4 Antarmuka Logout User

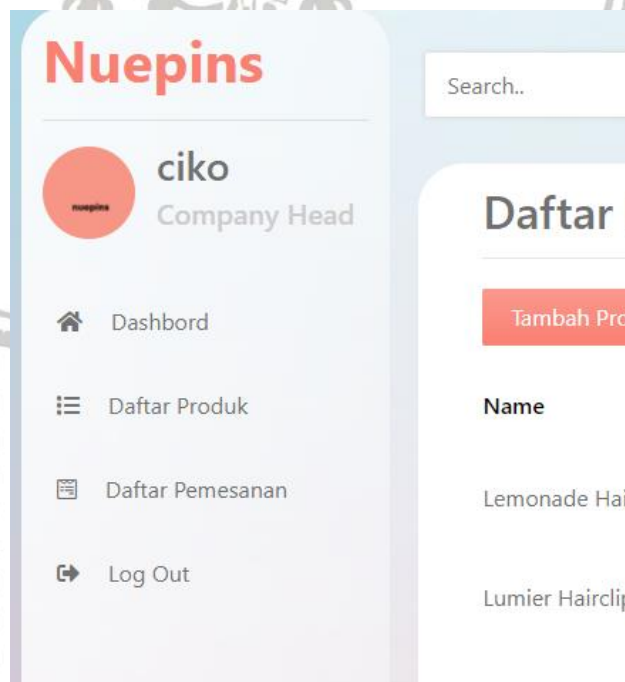
Antarmuka *logout user* merupakan tampilan saat aktor *user* melakukan *logout* atau mengeluarkan akun *user* dari sistem. Pada tampilan *logout user*, terdapat tombol *logout* yang dapat dibuka setelah *user* menekan gambar aktor disamping nama *user*. Antarmuka *logout user* akan ditampilkan pada gambar 6.5



Gambar 6.5 Antarmuka Logout User

6.5.5 Antarmuka Logout Admin

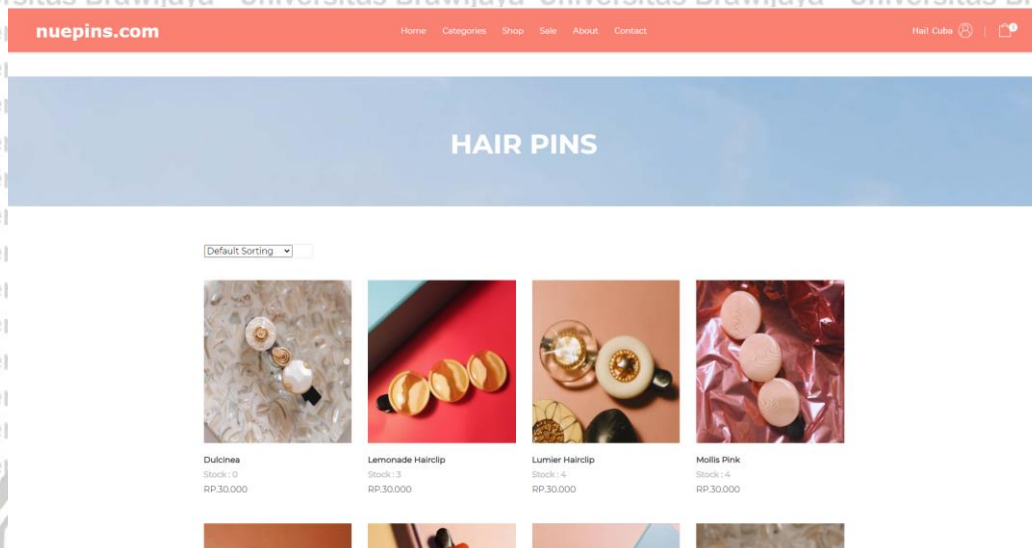
Antarmuka *logout admin* merupakan tampilan saat *admin* ingin mengeluarkan akun *admin* dari sistem khusus *admin*. Tombol *logout* terletak pada bagian *sidebar*. Antarmuka *logout admin* akan ditampilkan pada gambar 6.6



Gambar 6.6 Antarmuka Logout Admin

6.5.6 Antarmuka Menampilkan Produk

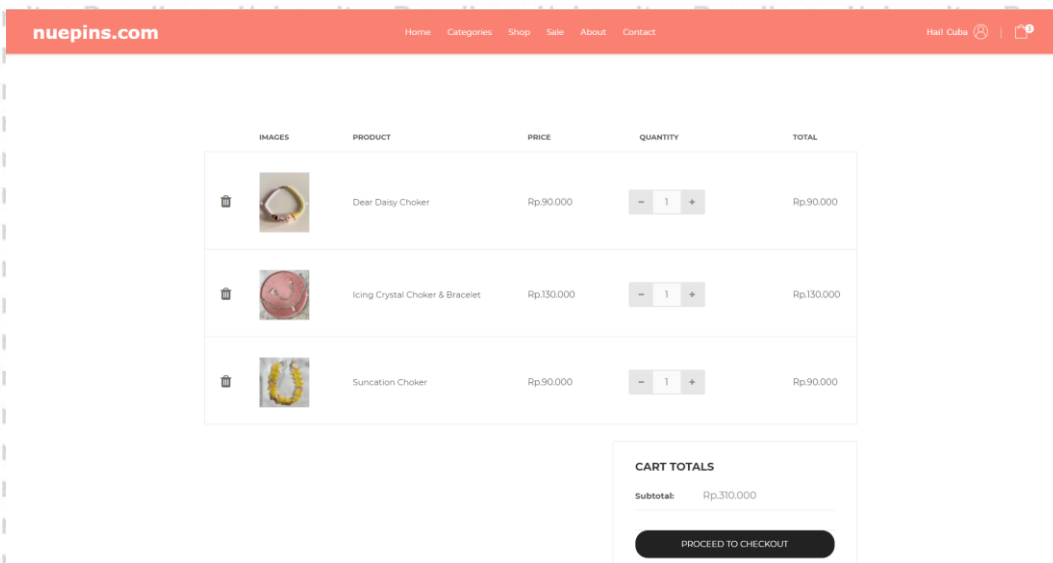
Antarmuka menampilkan produk merupakan tampilan saat aktor *user* melihat data produk dari Nue Pins berdasarkan kategori. Pada bagian atas terdapat judul dari kategori dan pada bagian bawah terdapat penampilan gambar produk, nama produk, dan stok produk. Antarmuka menampilkan produk akan ditampilkan pada gambar 6.7



Gambar 6.7 Antarmuka Menampilkan Produk

6.5.7 Antarmuka Keranjang Belanja

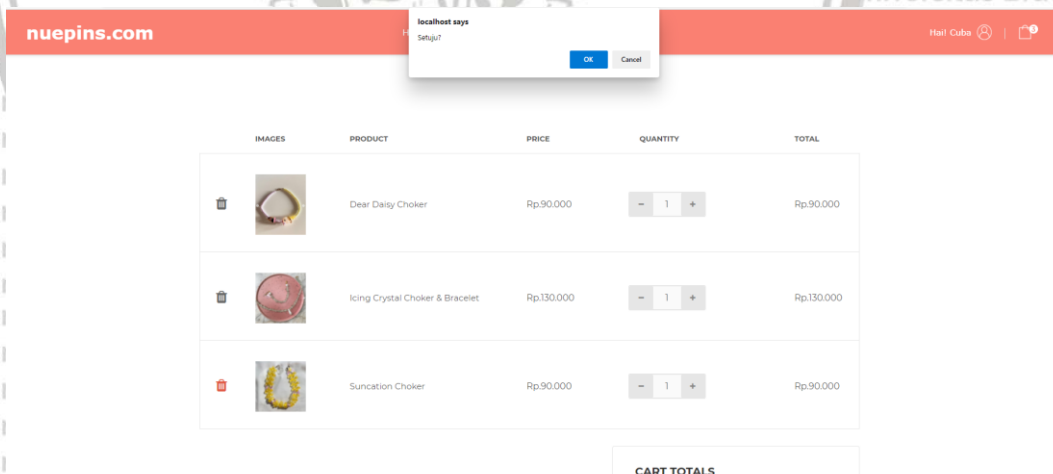
Antarmuka keranjang belanja merupakan tampilan saat aktor *user* melihat halaman keranjang belanja. Terdapat beberapa daftar produk yang telah dimasukkan kekeranjang belanja oleh *user*. Pada setiap baris data, terdapat data nama produk, gambar produk, harga produk, jumlah pembelian, dan jumlah total harga perproduk. Pada bagian bawah, terdapat informasi jumlah harga yang harus dibayar. Antarmuka keranjang belanja akan ditampilkan pada gambar 6.8



Gambar 6.8 Antarmuka Keranjang Belanja

6.5.8 Antarmuka Menghapus Produk

Antarmuka menghapus produk merupakan tampilan saat aktor *user* melakukan penghapusan data produk dari keranjang belanja. Terdapat tombol *delete* pada masing – masing baris data produk yang jika ditekan akan menjadi warna merah dan memberikan peringatan persetujuan untuk menghapus. Antarmuka menghapus produk akan ditampilkan pada gambar 6.9

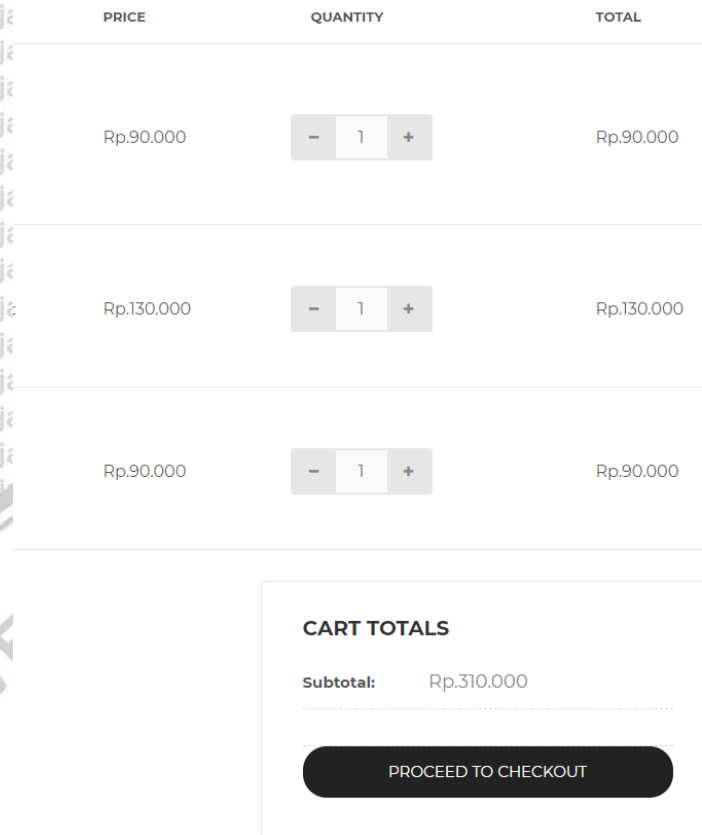


Gambar 6.9 Antarmuka Menghapus Produk

6.5.9 Antarmuka Menentukan Jumlah Pembelian

Antarmuka menentukan jumlah pembelian merupakan tampilan saat aktor *user* menentukan jumlah pembelian pada masing – masing baris data produk di keranjang belanja. Pada tampilan untuk menentukan jumlah pembelian, terdapat

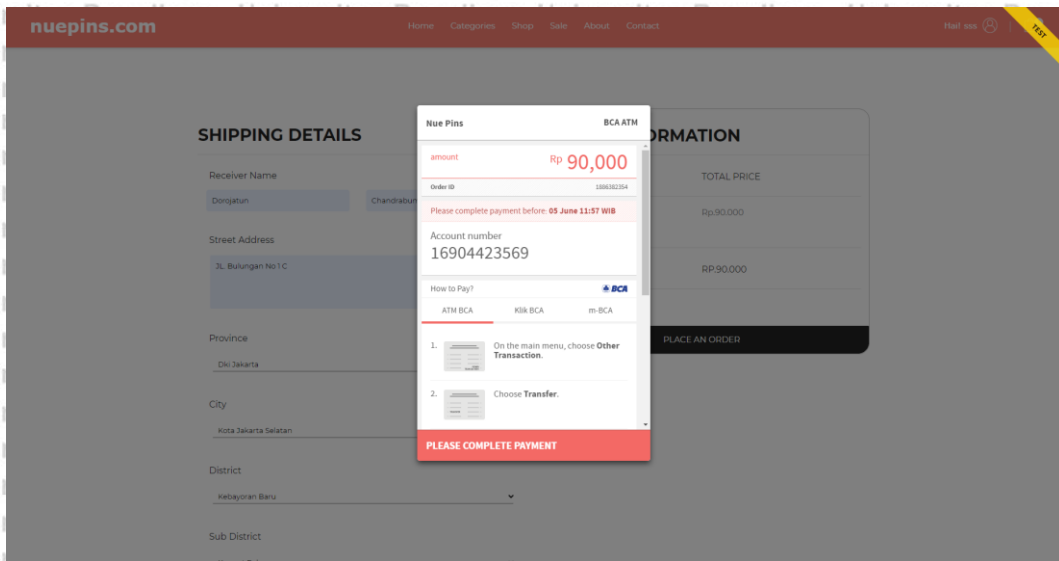
tombol tambah dan kurang untuk mengubah data jumlah pembelian. Antarmuka menentukan jumlah pembelian akan ditampilkan pada gambar 6.10



Gambar 6.10 Antarmuka Menentukan Jumlah pembelian

6.5.10 Antarmuka Membuat Pemesanan

Antarmuka membuat pemesanan merupakan tampilan saat aktor *user* melakukan *checkout* untuk membuat pemesanan. Pada bagian kanan halaman, terdapat pengisian data input penerima, diantaranya nama depan, nama belakang, nomor ponsel, nama jalan, pemilihan provinsi, kota, kecamatan dan kelurahan. Pada bagian kanan halaman, terdapat informasi nama produk, jumlah produk, dan total harga perproduk yang akan dibeli, dan subtotal harga pembayaran. Setelah *user* menekan tombol *checkout* maka sistem akan menampilkan *modal* untuk pemilihan metode pembayaran. Setelah memilih metode pembayaran, maka sistem akan menampilkan kode pembayaran. Antarmuka membuat pemesanan akan ditampilkan pada gambar 6.10





Gambar 6.11 Antarmuka Membuat Pemesanan

6.5.11 Antarmuka Track Pemesanan

Antarmuka *track* pemesanan merupakan tampilan saat aktor *user* melihat status pemesanan. Pada tampilan *track* pemesanan, terdapat kotak yang memisahkan data pemesanan *user* berdasarkan *order id*. Dalam kotak data pemesanan, terdapat tanggal pemesanan yang buat, *order id* nama produk, jumlah pembelian, total pembayaran, informasi status pemesanan, dan informasi pengiriman. Pada bagian atas kanan kotak data pemesanan, terdapat tombol untuk melakukan *download* cara pembayaran. Antarmuka *track* pemesanan akan ditampilkan pada gambar 6.12

MY ORDERS

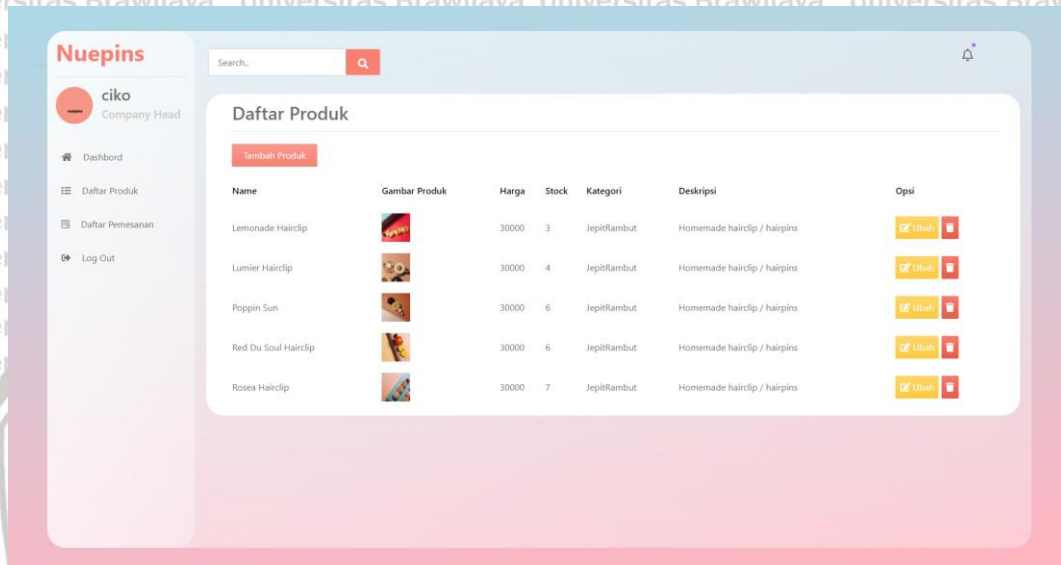
| Order Date | Order ID | | |
|---|---|----------------------------------|---------------|
| 2021-06-04 | 1129053930 | LEARN HOW TO PAY | |
| Payment Status: Ambil dari midtrans | | | |
|  | Dear Gloomy Choker | Buy Quantity | Total Payment |
| | Status Pemesanan : Transaksi sedang diproses | X 1 | RP.90.000 |
| | Pengiriman : Belum ada nomor resi | | |

| Order Date | Order ID | | |
|---|---|----------------------------------|---------------|
| 2021-06-03 | 318184220 | LEARN HOW TO PAY | |
| Payment Status: Ambil dari midtrans | | | |
|  | Lemonade Hairclip | Buy Quantity | Total Payment |
| | Status Pemesanan : Transaksi sedang diproses | X 2 | RP.60.000 |
| | Pengiriman : | | |

Gambar 6.12 Antarmuka Track Pemesanan

6.5.12 Antarmuka Informasi Produk

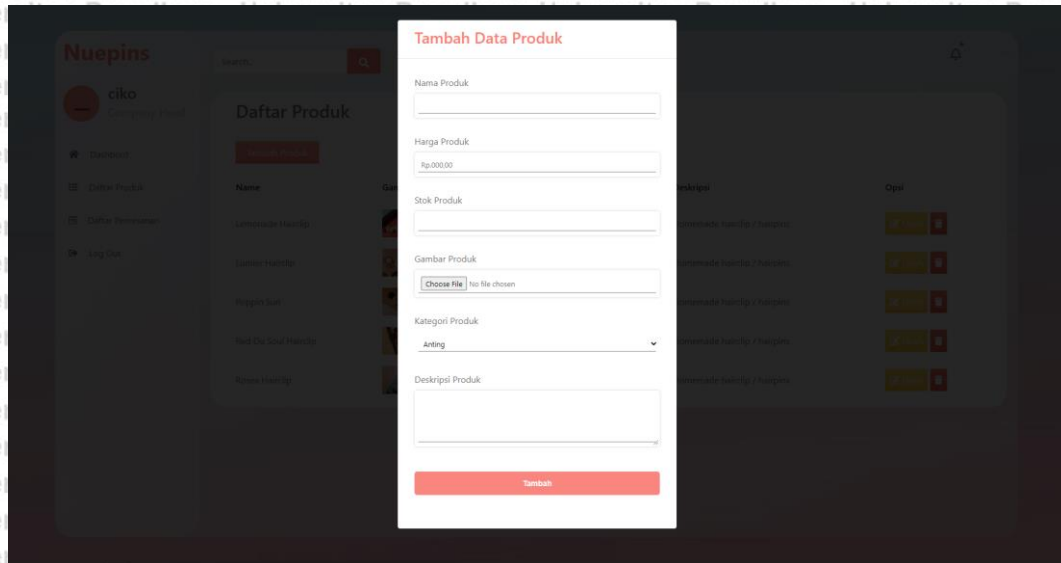
Antarmuka informasi produk merupakan tampilan saat aktor *admin* melakukan halaman data produk. Pada bagian atas terdapat tombol untuk menampilkan *modal* penambahan produk. Pada masing – masing baris data produk, terdapat penampilan data nama produk, gambar produk, stok produk, kategori, deskripsi produk, dan opsi untuk menghapus dan mengubah data produk. Antarmuka informasi produk akan ditampilkan pada gambar 6.13



Gambar 6.13 Antarmuka Informasi Produk

6.5.13 Antarmuka Menambah Data Produk

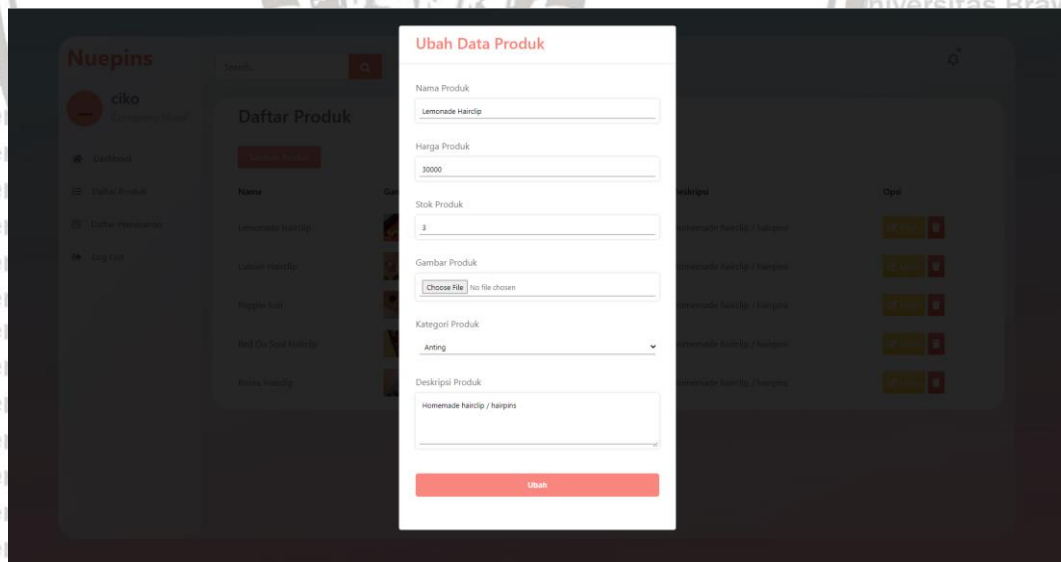
Antarmuka menambahkan data produk merupakan tampilan saat aktor *admin* melakukan penambahan data produk kedalam database. Pada tampilan untuk menambah data produk, terdapat inputan nama produk, harga produk, stok produk, gambar produk, kategori produk, dan deskripsi produk. Antarmuka menambah data produk akan ditampilkan pada gambar 6.14



Gambar 6.14 Antarmuka Menambah Data Produk

6.5.14 Antarmuka Mengubah Data Produk

Antarmuka mengubah data produk merupakan tampilan saat aktor *admin* melakukan pengubahan data produk. Terdapat inputan yang sama dengan antarmuka menambah data produk, tetapi inputan tersebut sudah terisi dengan data produk sebelum diubah. Antarmuka mengubah data produk akan ditampilkan pada gambar 6.15

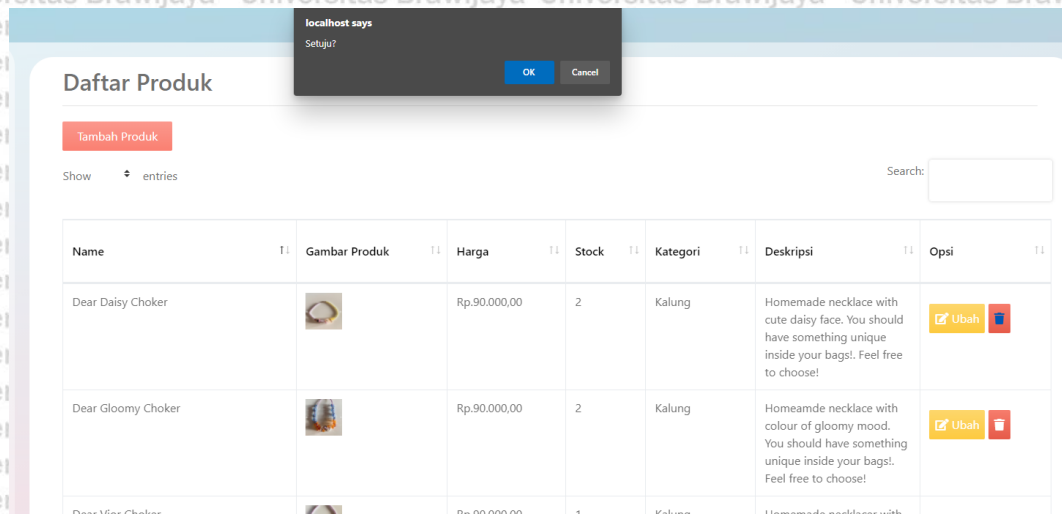


Gambar 6.15 Antarmuka Mengubah Data Produk

6.5.15 Antarmuka Menghapus Data Produk

Antarmuka menghapus data produk merupakan tampilan saat aktor *admin* melakukan penghapusan data produk. Pada tampilan untuk menghapus data

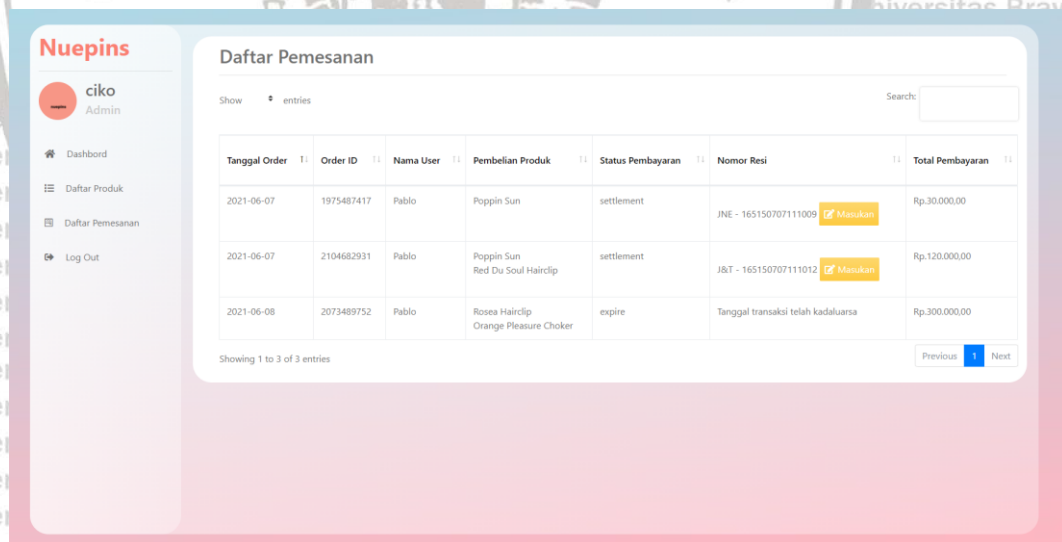
produk, terdapat tombol *delete* yang jika ditekan akan mengeluarkan peringatan. Antarmuka menghapus data produk akan ditampilkan pada gambar 6.16



Gambar 6.16 Menghapus Data Produk

6.5.16 Antarmuka Informasi Pemesanan

Antarmuka informasi pemesanan merupakan tampilan saat aktor *admin* melihat data pemesanan *user*. Pada tampilan informasi pemesanan terdapat tabel yang berisikan data pemesanan yang dikelompokkan berdasarkan *order_id*. Antarmuka informasi pemesanan akan ditampilkan pada gambar 6.17

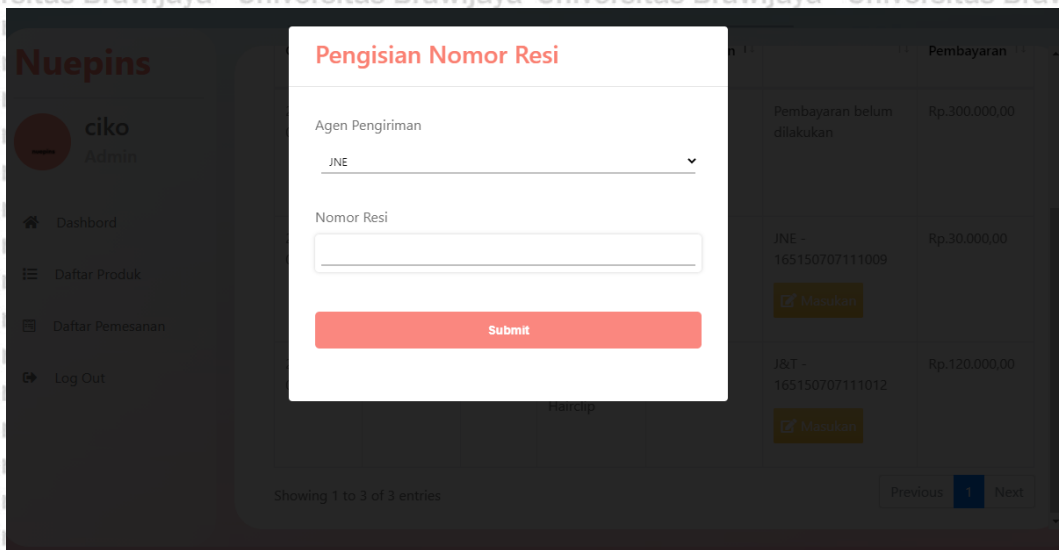


Gambar 6.17 Antarmuka Informasi Pemesanan

6.5.17 Antarmuka Verifikasi Pemesanan

Antarmuka verifikasi pemesanan merupakan tampilan saat aktor *admin* melakukan verifikasi jika *user* telah melakukan pembayaran. Pada tampilan verifikasi pemesanan, terdapat pengisian input nomor resi pengiriman dan pemilihan ekspidisi pengiriman. Tombol verifikasi pemesanan akan dimunculkan

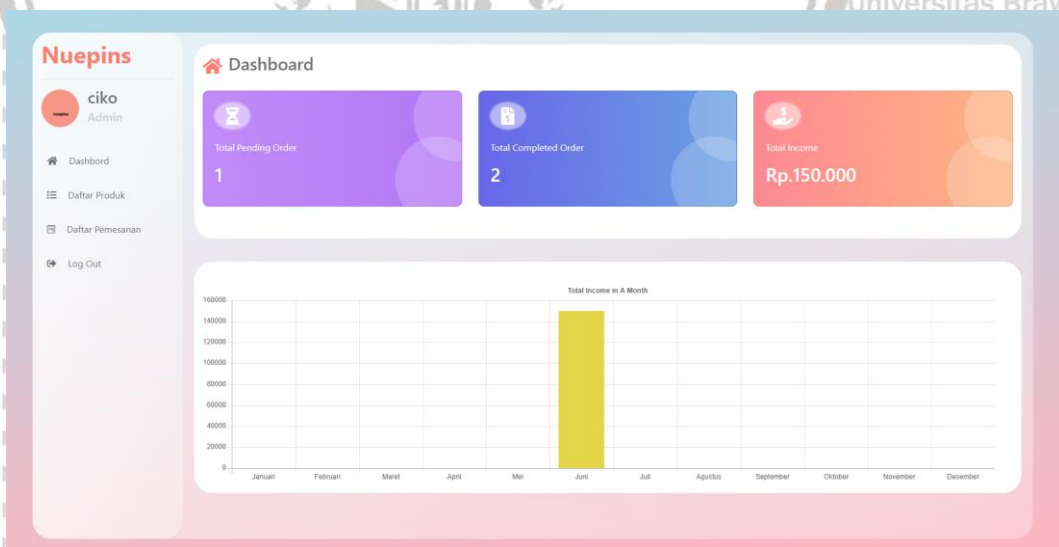
jika status pembayaran pada pemesanan adalah *settlement*. Antarmuka verifikasi pemesanan akan ditampilkan pada gambar 6.18



Gambar 6.18 Antarmuka Verifikasi Pemesanan

6.5.18 Antarmuka Dashboard

Antarmuka *dashboard* merupakan tampilan saat aktor *admin* melakukan akses pada halaman *dashboard* di sistem informasi khusus *admin*. Pada tampilan *dashboard*, terdapat 3 *card* pada bagian atas yang menampilkan informasi total pemasukan, total *order* yang telah selesai, dan total *order* yang masih berstatus *pending*. Pada bagian bawah, terdapat grafik pemasukan berdasarkan masing – masing bulan. Antarmuka *dashboard* akan ditampilkan pada gambar 6.19



Gambar 6.19 Antarmuka Dashboard

BAB 7 PENGUJIAN SISTEM

7.1 Pengujian *Blackbox*

Pengujian *blackbox* dilakukan untuk memeriksa setiap *input* pada fungsionalitas yang ada pada sistem. Pada pengujian *blackbox* akan dilakukan pengecekan terhadap fungsionalitas untuk memastikan kesesuaian terhadap kebutuhan sistem yang telah dirancang. Proses pengujian ini mengaju terhadap *use case scenario* yang telah dirancang pada bagian analisis kebutuhan sistem.

7.1.1 Pengujian *Blackbox Login User*

Pada pengujian *login user*, terdapat alur pengujian aktor *user* melakukan penginputan data *email* dan *password*. Setelah melakukan penginputan, aktor *user* menekan tombol *login*. Sistem akan melakukan pengecekan terhadap data *email* dan *password* yang telah diinputkan. Tabel 7.1 akan menjelaskan tahap dari hasil validasi pengujian *blackbox login user*.

Tabel 7.1 Pengujian *Blackbox Login User*

| | |
|-----------------------|--|
| Nama fungsi | Login User |
| Tujuan Pengujian | Memastikan <i>User</i> dapat melakukan <i>login</i> dan masuk kedalam sistem |
| Alur Pengujian | <ol style="list-style-type: none"> 1. User memasukkan data <i>email</i> dan <i>password</i> 2. User menekan tombol <i>login</i> |
| Input | <ol style="list-style-type: none"> 1. <i>Email</i> : pablo.breakbot@gmail.com 2. <i>Password</i> : 123pongka |
| Hasil yang diharapkan | <i>User</i> berhasil melakukan <i>login</i> dan sistem berhasil menyimpan data <i>email</i> ke <i>session</i> . |
| Hasil Pengujian | Sistem menampilkan halaman <i>utama</i> |
| Status | Valid |

7.1.2 Pengujian *Blackbox Login Admin*

Pada tahap pengujian *login admin*, terdapat alur pengujian aktor *admin* melakukan penginputan data *email* dan *password*. Setelah melakukan penginputan, aktor *admin* menekan tombol *login*. Sistem akan melakukan pengecekan terhadap hasil input yang telah dilakukan oleh *admin*. Tabel 7.2 akan menjelaskan tahap dari hasil validasi pengujian *blackbox login admin*.

Tabel 7.2 Pengujian *Blackbox Login Admin*

| | |
|-------------|-------------|
| Nama fungsi | Login Admin |
|-------------|-------------|

| | |
|-----------------------|--|
| Tujuan Pengujian | Memastikan Admin dapat melakukan login dan masuk kedalam sistem khusus admin |
| Alur Pengujian | <ol style="list-style-type: none"> Admin memasukkan data <i>email</i> dan <i>password</i> Admin menekan tombol <i>login</i> |
| Input | <ol style="list-style-type: none"> <i>Email</i> : nuepins.admn@gmail.com <i>Password</i> : nuepins2018 |
| Hasil yang diharapkan | Admin berhasil melakukan login dan sistem berhasil menyimpan data <i>email</i> ke <i>session</i> |
| Hasil Pengujian | Sistem menampilkan halaman utama sistem khusus admin |
| Status | Valid |

7.1.3 Pengujian Blackbox Register

Pada tahap pengujian *register*, terdapat alur pengujian aktor *user* melakukan penginputan data *firstname*, *lastname*, *email*, *phone number*, dan *password*. Sistem akan menyimpan data yang telah diinputkan oleh aktor *user* kedalam *database* dengan status *emailActive* adalah 0, dan sistem akan mengirimkan *email* aktivasi ke *email* yang telah diinputkan oleh user. Setelah *user* menekan *link* yang dikirimkan, status *emailActive* akan menjadi 1. Dan *user* dapat melakukan *login*. Tabel 7.3 akan menjelaskan tahap dari hasil validasi pengujian *blackbox register*

Tabel 7.3 Pengujian Blackbox Register

| | |
|------------------|---|
| Nama fungsi | Register |
| Tujuan Pengujian | Memastikan sistem mendaftarkan akun <i>User</i> pada sistem |
| Alur Pengujian | <ol style="list-style-type: none"> <i>User</i> memasukkan data <i>firstname</i> <i>User</i> memasukkan data <i>lastname</i> <i>User</i> memasukkan data <i>email</i> <i>User</i> memasukkan data <i>phone number</i> <i>User</i> memasukkan data <i>password</i> |
| Input | <ol style="list-style-type: none"> <i>Firstname</i> : Pablo <i>Lastname</i> : Cigar <i>Email</i> : pablo.breakbot@gmail.com <i>Phone number</i> : 081210438290 <i>Password</i> : 123ponka |

| | |
|-----------------------|---|
| Hasil yang diharapkan | User berhasil mendaftarkan melakukan registrasi dan sistem berhasil menyimpan data yang telah diinputkan oleh user |
| Hasil Pengujian | Sistem memberikan <i>link</i> validasi pada <i>email user</i> yang telah diinputkan, dan <i>user</i> dapat melakukan <i>login</i> . |
| Status | Valid |

7.1.4 Pengujian *Blackbox Logout User*

Pada tahap pengujian *logout user*, terdapat alur pengujian *User* menekan tombol pada gambar akun. Setelah itu, *User* melihat *dropdown* yang berisikan pilihan *My Order* dan *Logout*. *User* menekan tombol *logout* dan akun *user* berhasil dikeluarkan dari sistem. Tabel 7.4 akan menjelaskan tahap dari hasil validasi pengujian *logout user*.

Tabel 7.4 Pengujian *Blackbox Logout User*

| | |
|-----------------------|--|
| Nama fungsi | Logout User |
| Tujuan Pengujian | Memastikan sistem dapat mengeluarkan akun <i>User</i> dari sistem |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> menekan tombol pada gambar akun 2. <i>User</i> melihat <i>dropdown</i> yang berisikan <i>My Order</i> dan <i>Logout</i> 3. <i>User</i> menekan tombol <i>logout</i> |
| Hasil yang diharapkan | Akun <i>user</i> dapat dikeluarkan dari sistem |
| Hasil Pengujian | Sistem berhasil menghapus data <i>session email user</i> dan mengeluarkan akun <i>user</i> dari sistem |
| Status | Valid |

7.1.5 Pengujian *Blackbox Logout Admin*

Pada tahap pengujian *logout admin*, terdapat alur pengujian *admin* sedang berada pada halaman utama sistem khusus *admin*, lalu *admin* menekan tombol *logout*. Sistem akan menghapus data *session email admin* dan sistem menampilkan halaman *login* sistem khusus *admin*. Tabel 7.5 akan menjelaskan tahap dari hasil validasi pengujian *logout admin*.

Tabel 7.5 Pengujian *Blackbox Logout Admin*

| | |
|------------------|--|
| Nama fungsi | Logout Admin |
| Tujuan Pengujian | Memastikan sistem dapat mengeluarkan akun <i>admin</i> dari sistem khusus <i>admin</i> |

| | |
|-----------------------|--|
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> melihat halaman utama sistem khusus <i>admin</i> 2. <i>Admin</i> menekan tombol <i>logout</i> pada <i>sidebar</i> |
| Hasil yang diharapkan | Akun <i>admin</i> dapat dikeluarkan dari sistem khusus <i>admin</i> |
| Hasil Pengujian | Sistem berhasil menghapus data <i>session email admin</i> dan sistem menampilkan halaman <i>login</i> |
| Status | Valid |

7.1.6 Pengujian *Blackbox* Menampilkan Produk

Pada tahap pengujian menampilkan produk, terdapat alur pengujian *user* menekan tombol *catalog* pada header, lalu *user* memilih kategori. Tabel 7.6 akan menjelaskan tahap dari hasil validasi pengujian menampilkan produk.

Tabel 7.6 Pengujian *Blackbox* Menampilkan Data Produk

| | |
|-----------------------|---|
| Nama fungsi | Menampilkan Produk |
| Tujuan Pengujian | Memastikan sistem dapat menampilkan data produk berdasarkan kategori yang dipilih oleh <i>User</i> |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> menekan tombol <i>catalog</i> pada header 2. <i>User</i> menekan kategori yang dipilih |
| Input | <ol style="list-style-type: none"> 1. Kategori : <i>Necklace</i> 2. Kategori : <i>Hairclip</i> 3. Kategori : <i>Pins</i> 4. Kategori : <i>Rings</i> 5. Kategori : <i>Strap</i> |
| Hasil yang diharapkan | Sistem menampilkan data produk |
| Hasil Pengujian | Sistem berhasil menampilkan data produk |
| Status | Valid |

7.1.7 Pengujian *Blackbox* Keranjang Belanja

Pada tahap pengujian keranjang belanja, terdapat alur pengujian *User* melihat data produk pada halaman kategori, dan *User* menekan tombol *add to cart* pada salah satu produk. Lalu *User* menekan tombol keranjang belanja untuk melakukan akses kehalaman keranjang belanja. Tabel 7.7 akan menjelaskan tahap dari hasil validasi pengujian keranjang belanja.

Tabel 7.7 Pengujian *Blackbox* Keranjang Belanja

| | |
|-----------------------|--|
| Nama fungsi | Keranjang Belanja |
| Tujuan Pengujian | Memastikan <i>user</i> dapat menyimpan data kedalam keranjang belanja dan <i>user</i> dapat melihat keranjang belanja |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> melihat data produk 2. <i>User</i> menekan tombol <i>add to cart</i> pada salah satu produk 3. <i>User</i> menekan tombol keranjang belanja |
| Input | 1. Data Produk : id_produk |
| Hasil yang diharapkan | Sistem dapat menyimpan data produk kedalam keranjang belanja <i>user</i> dan <i>User</i> dapat melihat halaman keranjang belanja |
| Hasil Pengujian | Sistem berhasil menyimpan data produk ke dalam keranjang belanja <i>user</i> , dan sistem berhasil menampilkan halaman keranjang belanja kepada <i>user</i> . |
| Status | Valid |

7.1.8 Pengujian *Blackbox* Menghapus Produk

Pada tahap pengujian menghapus produk, terdapat alur pengujian *User* yang sedang berada pada halaman keranjang belanja, lalu *user* menekan tombol hapus pada salah satu produk. Lalu *User* menekan tombol setuju pada peringatan yang diberikan oleh sistem. Tabel 7.8 akan menjelaskan tahap dari hasil validasi pengujian menghapus produk.

Tabel 7.8 Pengujian *Blackbox* Menghapus Produk

| | |
|-----------------------|--|
| Nama fungsi | Menghapus Produk |
| Tujuan Pengujian | Memastikan <i>user</i> dapat menghapus produk dari keranjang belanja |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> melihat halaman keranjang belanja 2. <i>User</i> menekan tombol hapus pada salah satu produk di keranjang belanja 3. <i>User</i> menekan tombol setuju pada peringatan yang diberikan |
| Input | 1. Data produk : id_produk |
| Hasil yang diharapkan | Sistem dapat menghapus data produk dari keranjang belanja |

| | |
|-----------------|--|
| Hasil Pengujian | Sistem berhasil menghapus data produk dari keranjang belanja |
| Status | Valid |

7.1.9 Pengujian *Blackbox* Menentukan Jumlah Pembelian

Pada tahap pengujian menentukan jumlah pembelian, terdapat alur pengujian *User* yang sedang berada pada halaman keranjang belanja, lalu *User* menekan tombol tambah dan kurang pada salah satu data produk didalam keranjang belanja. Tabel 7.9 akan menjelaskan tahap dari hasil validasi pengujian menentukan jumlah pembelian.

Tabel 7.9 Pengujian *Blackbox* Menentukan Jumlah Pembelian

| | |
|-----------------------|--|
| Nama fungsi | Menentukan jumlah pembelian |
| Tujuan Pengujian | Memastikan <i>user</i> dapat menentukan jumlah pembelian pada masing – masing produk didalam keranjang belanja |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> melihat halaman keranjang belanja 2. <i>User</i> menekan tombol tambah pada salah satu produk dikeranjang belanja 3. <i>User</i> menekan tombol kurang pada salah satu produk dikeranjang belanja |
| Input | <ol style="list-style-type: none"> 1. Data produk : id_produk |
| Hasil yang diharapkan | Sistem dapat mengubah data jumlah pembelian pada produk yang dipilih oleh <i>user</i> didalam keranjang belanja |
| Hasil Pengujian | Sistem berhasil mengubah data jumlah pembelian pada produk yang dipilih oleh <i>user</i> didalam keranjang belanja |
| Status | Valid |

7.1.10 Pengujian *Blackbox* Membuat Pemesanan

Pada tahap pengujian membuat pemesanan, terdapat alur pengujian *User* yang sedang berada pada halaman keranjang belanja, lalu menekan tombol *checkout*. Setelah itu, *user* melakukan pengisian data penerima untuk proses pengiriman, lalu *user* menekan tombol *place an order*. Selanjutnya, *user* memilih metode pembayaran dan *user* mendapatkan kode pembayaran. Lalu, *User* melakukan pembayaran pada *website* pengujian pembayaran yang telah

disediakan oleh *midtrans*. Tabel 7.10 akan menjelaskan tahap dari hasil validasi pengujian membuat pemesanan

Tabel 7.10 Pengujian *Blackbox* Membuat Pemesanan

| | |
|-----------------------|--|
| Nama fungsi | Membuat Pemesanan |
| Tujuan Pengujian | Memastikan <i>user</i> dapat membuat pemesanan, mendapatkan kode pembayaran dan melakukan pembayaran |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> berada pada halaman keranjang belanja 2. <i>User</i> menekan tombol <i>checkout</i> pada keranjang belanja 3. <i>User</i> melakukan pengisian data penerima 4. <i>User</i> menekan tombol <i>Place an Order</i> 5. <i>User</i> memilih metode pembayaran 6. <i>User</i> mendapatkan kode pembayaran 7. <i>User</i> melakukan pembayaran |
| Input | <ol style="list-style-type: none"> 1. Keranjang : <i>id_keranjang</i> 2. Data Penerima : <i>firstname, lastname, phone number</i>, nama jalan, provinsi, kota, kecamatan, kelurahan, kode pos 3. Metode Pembayaran : <i>bank transfer BCA</i> 4. Kode Pembayaran : <i>16904109010</i> |
| Hasil yang diharapkan | Sistem dapat menyimpan data pemesanan <i>user</i> . <i>User</i> dapat melakukan pembayaran berdasarkan metode pembayaran yang dipilih |
| Hasil Pengujian | Sistem berhasil menyimpan data pemesanan <i>user</i> dan <i>user</i> dapat melakukan pembayaran berdasarkan metode pembayaran yang dipilih |
| Status | Valid |

7.1.11 Pengujian *Blackbox* Track Pemesanan

Pada tahap pengujian *track* pemesanan, terdapat alur pengujian *User* menekan gambar akun pada header, lalu *user* menekan tombol *my order* pada menu yang ditampilkan. Tabel 7.11 akan menjelaskan tahap dari validasi pengujian *track* pemesanan.

Tabel 7.11 Pengujian Blackbox Track Pemesanan

| | |
|-----------------------|---|
| Nama fungsi | Track Pemesanan |
| Tujuan Pengujian | Memastikan <i>user</i> dapat melihat status pemesanan yang telah dibuat |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>User</i> menekan gambar akun pada header 2. <i>User</i> menekan tombol <i>My Orders</i> pada menu dropdown |
| Input | 1. Data User : id_user |
| Hasil yang diharapkan | Sistem dapat menampilkan data pemesanan user dan menampilkan status pemesanan kepada <i>user</i> . |
| Hasil Pengujian | Sistem berhasil menampilkan data pemesanan <i>user</i> dan berhasil menampilkan status pemesanan kepada <i>user</i> . |
| Status | Valid |

7.1.12 Pengujian Blackbox Informasi Produk

Pada tahap pengujian informasi produk, terdapat alur pengujian *Admin* yang telah berada pada sistem informasi khusus admin, lalu menekan tombol daftar produk. Lalu sistem mengarahkan *admin* ke halaman daftar produk. Tabel 6.12 akan menjelaskan tahap dari hasil validasi pengujian informasi produk

Tabel 6.12 Pengujian Blackbox Informasi Produk

| | |
|-----------------------|---|
| Nama fungsi | Informasi Produk |
| Tujuan Pengujian | Memastikan <i>admin</i> dapat melihat data produk pada sistem khusus <i>admin</i> |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> berada pada sistem khusus <i>admin</i> 2. <i>Admin</i> menekan tombol daftar produk 3. <i>Admin</i> melihat data produk |
| Hasil yang diharapkan | Sistem dapat menampilkan data produk. |
| Hasil Pengujian | Sistem berhasil menampilkan data produk |
| Status | Valid |

7.1.13 Pengujian Blackbox Menambah Data Produk

Pada tahap pengujian menambah data produk, terdapat alur pengujian *admin* yang sedang berada pada halaman daftar produk, lalu *admin* menekan tombol tambah produk. Setelah itu *admin* melakukan penginputkan data yang diperlukan,

dan menekan tombol tambah. Tabel 7.13 akan menjelaskan tahap dari validasi pengujian menambah data produk.

Tabel 7.13 Pengujian *Blackbox* Menambah Data Produk

| | |
|-----------------------|--|
| Nama fungsi | Menambah Data Produk |
| Tujuan Pengujian | Memastikan <i>admin</i> dapat melakukan penambahan data produk |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> berada pada halaman daftar produk 2. <i>Admin</i> menekan tombol Tambah Produk 3. <i>Admin</i> melakukan pengisian data nama produk, harga produk, stok produk, gambar produk, kategori produk, dan deskripsi produk 4. <i>Admin</i> menekan tombol tambah |
| Input | <ol style="list-style-type: none"> 1. Nama Produk : Dulcinea 2. Harga Produk : 30000 3. Stok Produk : 3 4. Kategori : Hair Clip 5. Gambar produk : dulcineanue.png 6. Deskripsi Produk : Homemade hair clip / pin with beautiful gold and white color. You should have something unique inside your bags!. Feel free to choose |
| Hasil yang diharapkan | Sistem dapat menyimpan data produk yang telah diinputkan oleh <i>admin</i> |
| Hasil Pengujian | Sistem berhasil menyimpan data produk yang telah diinputkan |
| Status | Valid |

7.1.14 Pengujian *Blackbox* Mengubah Data Produk

Pada tahap pengujian mengubah data produk, terdapat alur pengujian *admin* yang telah berada pada halaman daftar produk, lalu *admin* menekan tombol ubah produk pada salah satu data produk. Setelah itu *admin* melakukan pengubahan data stok pada produk *Icing Crystal Choker & Bracelet*, dan menekan tombol ubah. Tabel 7.14 akan menjelaskan tahap dari hasil validasi pengujian mengubah data produk.

Tabel 7.14 Pengujian *Blackbox* Mengubah Data Produk

| | |
|-------------|----------------------|
| Nama fungsi | Mengubah Data Produk |
|-------------|----------------------|

| | |
|-----------------------|---|
| Tujuan Pengujian | Memastikan <i>admin</i> dapat melakukan perubahan data produk |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> berada pada halaman daftar produk 2. <i>Admin</i> menekan tombol ubah pada salah satu data produk 3. <i>Admin</i> melakukan perubahan data stok pada produk <i>Icing Crystal Choker & Bracelet</i> 4. <i>Admin</i> menekan tombol ubah |
| Input | <ol style="list-style-type: none"> 1. id_produk : 32 2. Stok : 3 |
| Hasil yang diharapkan | Sistem dapat mengubah data produk yang dipilih oleh admin |
| Hasil Pengujian | Sistem berhasil mengubah data produk |
| Status | Valid |

7.1.15 Pengujian *Blackbox* Menghapus Data Produk

Pada tahap pengujian, terdapat alur pengujian *admin* yang telah berada pada halaman daftar produk, lalu menekan tombol hapus pada produk *lemonade hairclip*. Setelah itu *admin* menekan tombol setuju yang merupakan peringatan dari sistem. Tabel 7.15 akan menjelaskan tahap dari hasil validasi pengujian menghapus data produk.

Tabel 7.15 Pengujian *Blackbox* Menghapus Data Produk

| | |
|-----------------------|---|
| Nama fungsi | Menghapus Data Produk |
| Tujuan Pengujian | Memastikan <i>admin</i> dapat menghapus salah satu data produk |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> berada pada halaman daftar produk 2. <i>Admin</i> menekan tombol hapus pada produk <i>Lemonade Hairclip</i> 3. <i>Admin</i> menekan tombol setuju |
| Input | 1. Id_produk : 16 |
| Hasil yang diharapkan | Sistem dapat menghapus produk yang dipilih oleh <i>admin</i> |
| Hasil Pengujian | Sistem berhasil menghapus data produk |
| Status | Valid |

7.1.16 Pengujian *Blackbox* Informasi Pemesanan

Pada tahap pengujian, terdapat alur pengujian *admin* yang sedang berada pada sistem khusus *admin*, lalu menekan tombol daftar pemesanan. Setelah itu *admin* telah berhasil masuk kehalaman daftar pemesanan. Tabel 7.16 akan menjelaskan tahap dari validasi pengujian informasi pemesanan.

Tabel 7.16 Pengujian *Blackbox* Informasi Pemesanan

| | |
|-----------------------|---|
| Nama fungsi | Informasi Pemesanan |
| Tujuan Pengujian | Memastikan <i>admin</i> dapat melihat data pemesanan pada sistem khusus <i>admin</i> |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> berada pada sistem khusus <i>admin</i> 2. <i>Admin</i> menekan tombol daftar pemesanan 3. <i>Admin</i> melihat data pemesanan |
| Hasil yang diharapkan | Sistem dapat menampilkan data pemesanan. |
| Hasil Pengujian | Sistem berhasil menampilkan data pemesanan |
| Status | Valid |

7.1.17 Pengujian *Blackbox* Verifikasi Pemesanan

Pada tahap pengujian verifikasi pemesanan, terdapat alur pengujian *admin* yang sedang berada pada halaman daftar pemesanan, lalu menekan tombol pengisian nomor resi. Setelah itu, *admin* melakukan pengisian nomor resi, pemilihan ekspedisi dan menekan tombol verifikasi. Tabel 7.17 akan menjelaskan tahap dari validasi pengujian verifikasi pemesanan.

Tabel 7.17 Pengujian *Blackbox* Verifikasi Pemesanan

| | |
|------------------|---|
| Nama fungsi | Verifikasi Pemesanan |
| Tujuan Pengujian | Memastikan <i>admin</i> dapat melakukan verifikasi dengan mengisi nomor resi pengiriman |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> berada pada halaman daftar pemesanan 2. <i>Admin</i> menekan tombol pengisian nomor resi pada salah satu data pemesanan 3. <i>Admin</i> melakukan pengisian nomor resi dan pemilihan ekspedisi 4. <i>Admin</i> menekan tombol verifikasi |
| Input | <ol style="list-style-type: none"> 1. Order_id : 21046829 2. Ekspedisi : J&T 3. Nomor Resi : 059928158 |

| | |
|-----------------------|--|
| Hasil yang diharapkan | Sistem dapat melakukan update pada data pemesanan |
| Hasil Pengujian | Sistem berhasil melakukan update pada data pemesanan |
| Status | Valid |

7.1.18 Pengujian *Blackbox Dashboard*

Pada pengujian *dashboard*, terdapat alur pengujian *admin* yang telah berhasil melakukan *login*, lalu *admin* akan diarahkan kehalaman *dashboard*. Tabel 7.18 akan menjelaskan tahap dari validasi pengujian *dashboard*.

Tabel 7.18 Pengujian *Blackbox Dashboard*

| | |
|-----------------------|---|
| Nama fungsi | Dashboard |
| Tujuan Pengujian | Memastikan <i>admin</i> dapat melakukan akses ke halaman dashboard |
| Alur Pengujian | <ol style="list-style-type: none"> 1. <i>Admin</i> telah berhasil melakukan <i>login</i> 2. <i>Admin</i> melihat halaman <i>dashboard</i> |
| Hasil yang diharapkan | Sistem dapat menampilkan halaman dashboard |
| Hasil Pengujian | Sistem berhasil menampilkan halaman dashboard |
| Status | Valid |

7.2 Pengujian *Usability*

Pengujian *usability* pada penelitian ini akan menggunakan selebaran *system usability scale* (SUS). Pengujian *system usability scale* merupakan pengujian yang berujuan untuk menguji apakah terdapat kekeliruan pada fitur yang dikembangkan, dan menguji apakah aktor dapat mengoprasikan sistem. Pengujian *system usability scale* dilaksanakan dengan menguji dua aktor yang akan menggunakan sistem, yaitu aktor *admin* sebagai pemilik Nue Pins, aktor *user* sebagai calon pelanggan. Daftar nama responden dan masing – masing perannya akan dijelaskan pada tabel 7.19

Tabel 7.19 Tabel Daftar Responden

| No | Nama Responden | Peran |
|----|-------------------|------------------|
| 1. | Ekadriani Fitria | Pemilik Nue Pins |
| 2. | Rara Fitrha Rahma | Calon Pelanggan |
| 3. | Mai Kusasi | Calon Pelanggan |

| | | |
|----|-----------------|-----------------|
| 4. | Tasha Paramitha | Calon Pelanggan |
| 5. | Anggun Lianti | Calon Pelanggan |

Pengujian ini dilakukan untuk mengetahui tingkat kemampuan sistem ketika digunakan oleh aktor. Pada daftar hasil pengujian, terdapat kolom P1 – P10. Kolom tersebut merupakan 10 pertanyaan dan akan diteruskan menjadi skor dari jawaban kolom P1 – P10. Lalu, terdapat kolom jabatan untuk menjelaskan peran dari responden. Setelah melakukan pengujian, maka skor dari penilaian pada masing – masing jawaban akan direkap kedalam tabel yang akan dijelaskan pada tabel 7.20. Penilaian masing – masing kolom berdasarkan parameter dari pengisian nilai, Sangat tidak setuju akan bernilai 1, Tidak setuju akan bernilai 2, Ragu – ragu akan bernilai 3, Setuju akan bernilai 4, Sangat setuju akan bernilai 5.

Tabel 7.20 Hasil Pengujian System Usability Scale

| Responden | Jabatan | Skor | | | | | | | | | |
|-----------|----------------|------|----|----|----|----|----|----|----|----|-----|
| | | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
| 1 | Pemilik | 5 | 1 | 4 | 1 | 5 | 1 | 2 | 1 | 5 | 1 |
| 2 | Calon Pengguna | 4 | 1 | 5 | 2 | 4 | 2 | 2 | 2 | 4 | 4 |
| 3 | Calon Pengguna | 4 | 2 | 5 | 2 | 4 | 2 | 3 | 2 | 5 | 4 |
| 4 | Calon Pengguna | 4 | 2 | 4 | 2 | 5 | 2 | 2 | 2 | 4 | 3 |
| 5 | Calon Pengguna | 4 | 2 | 4 | 1 | 4 | 3 | 3 | 2 | 5 | 4 |

Setelah melakukan pengumpulan nilai dari 5 responden, selanjutnya akan dilakukan perhitungan skor dari hasil pengujian *system usability scale* berdasarkan masing – masing responden, berdasarkan data yang telah didapatkan dari responden, untuk cara perhitungannya adalah sebagai berikut :

1. Untuk pertanyaan dengan nomor ganjil, skor yang didapatkan akan dikurangi 1
2. Untuk pertanyaan dengan nomor genap, nilai 5 akan dikurangi dari skor yang didapat
3. Skor SUS akan didapatkan dengan cara menambahkan semua nilai yang telah didapat pada masing – masing jawaban, lalu dikali 2,5

4. Setelah mendapatkan skor dari responden, selanjutnya akan dilakukan penjumlahan skor, dan setelah skor dijumlah akan dibagi berdasarkan jumlah responden untuk dicari rata – rata.

Hasil perhitungan akhir skor dari pengujian *system usability scale* akan ditampilkan pada tabel 7.21.

Tabel 7.21 Hasil Perhitungan System Usability Scale

| Skor hasil hitung | | | | | | | | | | Jumlah | Nilai Jumlah x 2.5 |
|------------------------------|----|----|----|----|----|----|----|----|-----|--------|-----------------------|
| P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | | |
| 4 | 4 | 3 | 4 | 4 | 4 | 1 | 4 | 3 | 4 | 35 | 87,5 |
| 3 | 4 | 4 | 3 | 3 | 3 | 1 | 3 | 3 | 1 | 28 | 70 |
| 3 | 3 | 4 | 3 | 4 | 3 | 2 | 3 | 4 | 1 | 30 | 75 |
| 3 | 3 | 3 | 3 | 4 | 3 | 1 | 3 | 3 | 2 | 25 | 62,5 |
| 3 | 3 | 2 | 4 | 3 | 3 | 2 | 3 | 4 | 2 | 29 | 72,5 |
| Skor rata rata (hasil Akhir) | | | | | | | | | | | 73,5 |

Hasil perhitungan skor hasil pengujian *system usability scale*, mendapatkan hasil rata – rata sebesar 73,5. Hasil skor tersebut menandakan bahwa sistem toko *online* Nue Pins yang telah dikembangkan dikategorikan sebagai *Good* dengan grade C. *Acceptability* dari grade C masuk kedalam kategori *acceptable* dan dapat diterima oleh pengguna.

BAB 8 PENUTUP

8.1 Kesimpulan

Berdasarkan hasil dari penelitian mengenai pengembangan sistem informasi toko *online* Nue Pins berbasis *web* dengan memanfaatkan *API Midtrans* sebagai media pembayaran, kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Proses analisis kebutuhan yang dilakukan dengan melakukan wawancara kepada pemilik *brand* atau merek Nue Pins, didapatkan 18 kebutuhan fungsional dan 1 kebutuhan non fungsional. Untuk identifikasi aktor, didapatkan 2 aktor, yaitu *admin* dan *user*. Perancangan sistem dirancang berdasarkan hasil dari proses analisis kebutuhan, yang terdiri dari perancangan arsitektur sistem, perancangan *sequence diagram*, perancangan *class diagram*, perancangan *physical data model*, dan perancangan antarmuka.
2. Implementasi sistem toko *online* Nue Pins, dilakukan berdasarkan hasil dari analisis kebutuhan dan perancangan sistem. Sistem menyediakan fitur sesuai dengan kebutuhan fungsional yang didefinisikan pada tahap analisis kebutuhan, dan melakukan penulisan alur kode program berdasarkan perancangan *sequence diagram*. Hasil implementasi mewujudkan 2 sistem informasi, yakni Sistem informasi toko *online* Nue Pins yang dapat melakukan pembelian produk dari *brand* Nue Pins dan melakukan pembayaran pada metode yang dilampirkan. Sistem khusus *admin*, yang dapat melakukan operasi pada produk dan pemesanan dari *brand* Nue Pins. Implementasi sistem dilakukan dengan menggunakan *frame work code igniter* dan memanfaatkan *API Midtrans* sebagai media pembayaran.
3. Pengujian sistem dilakukan dengan menggunakan metode pengujian *black box* dan *usability*. Hasil dari pengujian *blackbox* menunjukkan bahwa seluruh fitur yang disediakan dapat dijalankan. Hasil dari pengujian *usability* dengan menggunakan penilaian kuisisioner *system usability scale* melalui 5 responden, sistem informasi toko *online* Nue Pins dikategorikan pada *grade C*, yang termasuk dalam *range acceptability acceptable* dan dapat diterima oleh pengguna.

8.2 Saran

Hasil penelitian mengenai pengembangan sistem informasi toko *online* Nue Pins berbasis *web* dengan memanfaatkan *API Midtrans* sebagai media pembayaran, terdapat kekurangan yang nantinya dapat diperbaiki dalam penelitian berikutnya. Saran yang diberikan untuk penelitian berikutnya, yaitu penambahan fitur untuk menambahkan alamat *User*, dan dapat dipakai ketika proses *checkout* untuk membeli produk. Sehingga dapat memberikan kenyamanan dalam pemakaian alamat untuk tidak melakukan penulisan alamat secara berulang.

DAFTAR PUSTAKA

- Afuan Lasmedi. (2010). Pemanfaatan Framework Codeigniter Dalam Pengembangan Sistem Informasi Pendataan Laporan Kerja Praktek Mahasiswa Program Studi Teknik Informatika Unsoed (Codeigniter Framework Used in Information System Development for Student's Report Data Collection Prac. *Juita, 1*, 39–44.
- Aprianti, W., & Maliha, U. (2016). *Sistem Informasi Kepadatan Penduduk Kelurahan Atau Desa Studi Kasus Pada Kecamatan Bati-Bati*. 2(2013), 21–28.
- Eril. (2020, February 22). *Yuk Mengenal API dan Manfaatnya di Era Digital Saat Ini* - Qwords. <https://qwords.com/blog/apa-itu-api/>
- Grgec, M., & Mužar, R. (2007). Role of UML sequence diagram constructs in object lifecycle concept. *Journal of Information and Organizational Sciences*, 31(1), 63–74.
- Handayani, S., Anofrizen, & Jazman, M. (2016). Sistem Informasi E-Commerce Untuk Jaringan Penjualan Sepeda Motor Bekas Kabupaten Kampar (Studi Kasus Adira Finance). *Rekayasa Dan Manajemen Sistem Informasi*, 2(2), 106–111.
- Hanindia, M., & Swari, P. (2019). E-Learning Di Sma Muhammadiyah 1. *Jurnal Teknologi Informasi Dan Komputer*, 5, 1.
- Haviluddin. (2011). Memahami Penggunaan UML (Unified Modelling Language). *Memahami Penggunaan UML (Unified Modelling Language)*, 6(1), 1–15. <https://informatikamulawarman.files.wordpress.com/2011/10/01-jurnal-informatika-mulawarman-feb-2011.pdf>
- Henriyadi, H., & Mulyati, R. (2016). USABILITY TESTING Sistem Informasi: Studi kasus pada Aplikasi Repositori Publikasi Badan Penelitian dan Pengembangan Pertanian. *Jurnal Perpustakaan Pertanian*, 23(2), 54. <https://doi.org/10.21082/jpp.v23n2.2014.p54-63>
- Heriyanto, Y. (2018). Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car. *Jurnal Intra-Tech*, 2(2), 64–77.
- Ika, A., Nugroho, P. I., & Ferdiana, R. (2015). Pengujian Usability Website Menggunakan System Usability Scale. *JURNAL IPTEKKOM : Jurnal Ilmu Pengetahuan & Teknologi Informasi*, 17(1), 31. <https://doi.org/10.33164/iptekkom.17.1.2015.31-38>
- Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika Pengembangan IT (JPIT)*, 3(2), 45–46. <https://doi.org/10.30591/jpit.v3i1.647>
- Joshi, A. P., & Dr.Chetan R, D. (2015). *A Case Study : Performance Evolution Using JavaScript Object Notation Technology*. 4(2277), 4–5.

Khan, A. G. (2016). Electronic Commerce: A Study on Benefits and Challenges in an Emerging Economy. *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc*, 16(1).

Midtrans. (2016). *Midtrans | Solusi Payment Gateway Indonesia*.
<https://midtrans.com/>

Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). *PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*. 1(3), 31–36.

Putra, H. N. (2018). Implementasi Diagram UML (Unified Modelling Language) dalam Perancangan Aplikasi Data Pasien Rawat Inap pada Puskesmas Lubuk Buaya. *Jurnal & Penelitian Teknik Informatika*, 2(2), 69–77.
<https://doi.org/10.1016/j.mcna.2012.10.011>

Rachman Andi, R., Beny, & Fernando, E. (2017). Perancangan E-Commerce Berbasis Website Pada Toko Dunia Palembang. *Jurnal Ilmiah Processor*, 12(2), 1102–1117.

https://scholar.google.com/scholar?hl=id&as_sdt=0%2C5&q=Perancangan+E-Commerce+Berbasis+Website+Pada+Toko+Mirabella+Batik+Jambi+Andi&btnG=

Ribeiro, A., Silva, A., & da Silva, A. R. (2015). Data Modeling and Data Analytics: A Survey from a Big Data Perspective. *Journal of Software Engineering and Applications*, 08(12), 617–634. <https://doi.org/10.4236/jsea.2015.812058>

Rizky, M. (2019, November 22). *UML Diagram : Activity Diagram*.
<https://socs.binus.ac.id/2019/11/22/uml-diagram-activity-diagram/>

Sari, C. A. (2015). Perilaku Berbelanja Online Di Kalangan Mahasiswi Antropologi Universitas Airlangga. *Jurnal Antro Unair*, 4(2), 205–216.

Sherman, R. (2015). *Business Intelligence Guidebook: From Data Integration to Analytics* (S. Elliot (ed.)). MK. <https://doi.org/https://doi.org/10.1016/C2012-0-06937-2>

Sunardi, S., Riadi, I., & Raharja, P. A. (2019). Analisis Application Programming Interface Pada Mobile E-Voting Menggunakan Metode Test-Driven Development. *Techno (Jurnal Fakultas Teknik, Universitas Muhammadiyah Purwokerto)*, 20(2), 87. <https://doi.org/10.30595/techno.v20i2.4266>

Susandi, D., & Sukisno, S. (2017). Sistem Penjualan Berbasis E-Commerce Menggunakan Metode Objek Oriented pada Distro Dlapak Street Wear. *JSii (Jurnal Sistem Informasi)*, 4, 5–8. <https://doi.org/10.30656/jsii.v4i0.368>

Turban, E. et al. (2017). Electronic Commerce 2012: Managerial and Social Networks Perspectives, 7/e. New Jersey: Pearson Education. *Jurnal Informatika*, 8(1), 826–836.

Urva, G., & Siregar, H. F. (2015). *Pemodelan UML E- Marketing Minyak Goreng*. 9,

92-101.

Wira, D., Putra, T., & Andriani, R. (2019). *Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD*. 7(1).



LAMPIRAN HASIL WAWANCARA

Tempat & Tanggal : Jakarta, 09 – Februari – 2021

Narasumber : Ekadriani Fitria

Sebagai : Pemilik Merk/Brand Nue Pins

1. Bagaimana proses penjualan dan transaksi sejauh ini yang digunakan dalam penjualan produk?

Jawab : Se jauh ini, produk kami dijual dengan menggunakan *Instagram*, untuk prosesnya, pembeli menanyakan terlebih dahulu mengenai ketersediaan produk dan harga produk ke *contact* yang kami cantumkan di *bio Instagram*. Untuk proses transaksi, kami memberikan total harga dari pembelian produk, dan nomor rekening. Jika pembayaran sudah dilakukan, maka pembeli akan kami minta bukti transfer untuk dilanjutkan ke pengiriman. Proses penjualan kami juga dilakukan secara langsung dengan membuka gerai di pondok indah mall, plaza senayan, mall of Indonesia, grand Indonesia.

2. Adakah kendala yang dialami dalam proses transaksi dan penjualan sejauh ini?

Jawab : untuk kendala yang dialami, kalau proses transaksi lewat sosial media *instagram*, kami masih harus melakukan pengecekan secara manual dari bukti transfer yang dikirim dari pelanggan. Untuk proses penjualan, pelanggan terkadang menanyakan mengenai informasi barang dan kami terkadang lama untuk membalas.


3. Apakah Brand Nue Pins sudah mempunyai *website*?

Jawab : Tidak, belum punya

4. Jika akan dikembangkan sebuah *website*, apa saja fitur yang dibutuhkan didalamnya?

Jawab : Kami membutuhkan *website* yang didalamnya dapat dilakukan proses transaksi tanpa harus pelanggan memberikan bukti transfer, dan membantu pelanggan untuk melihat informasi dari produk kita. Kemudian, kami dapat melihat pemesanan yang dilakukan oleh pelanggan, dan memberikan nomor resi pengiriman kepada pelanggan melalui *website*.

Jakarta, 10-6-2021


Ekadriani Fitria
(Pemilik)

LAMPIRAN KUISIONER SYSTEM USABILITY SCALE

LEMBAR KUISIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI E-COMMERCE NUE PINS

Nama : *Ekadriani Fitria*

Pekerjaan : *PEMILIK NuePins*

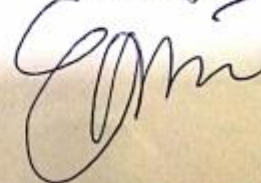
Jawablah pertanyaan berikut ini dengan memberi salah satu tanda *checklist* (✓) pada setiap pertanyaan di kolom jawaban yang tersedia.

| | | |
|---------------------------|-------------------|----------------|
| Keterangan : | | |
| STS : Sangat tidak setuju | TS: Tidak Setuju | R: Ragu – Ragu |
| S : Setuju | SS: Sangat Setuju | |

Pertanyaan :

| | STS | TS | R | S | SS |
|---|-----|----|---|---|----|
| 1. Saya berpikir akan menggunakan sistem ini lagi. | | | | | ✓ |
| 2. Saya merasa sistem ini sulit untuk digunakan | ✓ | | | | |
| 3. Saya merasa sistem ini mudah untuk digunakan | | | | ✓ | |
| 4. Saya membutuhkan bantuan dari orang lain untuk menggunakan sistem ini | ✓ | | | | |
| 5. Saya merasa fitur atau fungsi yang berjalan pada sistem ini berjalan dengan baik | | | | | ✓ |
| 6. Saya merasa banyak hal yang tidak perlu dalam sistem ini | ✓ | | | | |
| 7. Saya merasa orang lain tidak memahami penggunaan dari sistem ini | | ✓ | | | |
| 8. Saya merasa sistem ini membingungkan | ✓ | | | | |
| 9. Saya merasa tidak ada kendala dalam menggunakan sistem ini | | | | | ✓ |
| 10. Saya perlu membiasakan diri untuk menggunakan sistem ini | ✓ | | | | |

Jakarta, 10 Juli 2021



LEMBAR KUISIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI E-COMMERCE NUE PINS

Nama : Rara Fitrha Rahma

Pekerjaan : Pengacara

Jawablah pertanyaan berikut ini dengan memberi salah satu tanda checklist (✓) pada setiap pertanyaan di kolom jawaban yang tersedia.

| | | |
|---------------------------|-------------------|----------------|
| Keterangan : | | |
| STS : Sangat tidak setuju | TS: Tidak Setuju | R: Ragu - Ragu |
| S : Setuju | SS: Sangat Setuju | |

Pertanyaan :

| | STS | TS | R | S | SS |
|---|-----|----|---|---|----|
| 1. Saya berpikir akan menggunakan sistem ini lagi. | | | | ✓ | |
| 2. Saya merasa sistem ini sulit untuk digunakan | ✓ | | | | |
| 3. Saya merasa sistem ini mudah untuk digunakan | | | | | ✓ |
| 4. Saya membutuhkan bantuan dari orang lain untuk menggunakan sistem ini | | ✓ | | | |
| 5. Saya merasa fitur atau fungsi yang berjalan pada sistem ini berjalan dengan baik | | | | ✓ | |
| 6. Saya merasa banyak hal yang tidak perlu dalam sistem ini | | ✓ | | | |
| 7. Saya merasa orang lain tidak memahami penggunaan dari sistem ini | | ✓ | | | |
| 8. Saya merasa sistem ini membingungkan | | ✓ | | | |
| 9. Saya merasa tidak ada kendala dalam menggunakan sistem ini | | | | ✓ | |
| 10. Saya perlu membiasakan diri untuk menggunakan sistem ini | | | | ✓ | |

Jakarta, 10, Juni



LEMBAR KUISIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI E-COMMERCE NUE PINS

Nama: *Anggun Erantri*
Pekerjaan: *Arsitek Junior*

Jawablah pertanyaan berikut ini dengan memberi salah satu tanda *checklist* (✓) pada setiap pertanyaan di kolom jawaban yang tersedia.

| | | |
|--------------|-------------------|----------------|
| Keterangan : | TS: Tidak Setuju | R: Ragu - Ragu |
| S: Setuju | SS: Sangat Setuju | |

| Pertanyaan : | STS | TS | R | S | SS |
|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1. Saya berpikir akan menggunakan sistem ini lagi. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 2. Saya merasa sistem ini sulit untuk digunakan | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Saya merasa sistem ini mudah untuk digunakan | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4. Saya membutuhkan bantuan dari orang lain untuk menggunakan sistem ini | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Saya merasa fitur atau fungsi yang berjalan pada sistem ini berjalan dengan baik | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 6. Saya merasa banyak hal yang tidak perlu dalam sistem ini | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. Saya merasa orang lain tidak memahami penggunaan dari sistem ini | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. Saya merasa sistem ini membingungkan | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. Saya merasa tidak ada kendala dalam menggunakan sistem ini | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 10. Saya perlu membiasakan diri untuk menggunakan sistem ini | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Jakarta, 10 Juni 2021

LEMBAR KUISIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI E-COMMERCE NUE PINS

Nama : Tasha Paramitha
Pekerjaan : Pengajar

Jawablah pertanyaan berikut ini dengan memberi salah satu tanda checklist (✓) pada setiap pertanyaan di kolom jawaban yang tersedia.

| | | |
|---------------------------|-------------------|----------------|
| Keterangan : | TS: Tidak Setuju | R: Ragu – Ragu |
| STS : Sangat tidak setuju | SS: Sangat Setuju | |
| S : Setuju | | |

| Pertanyaan : | STS | TS | R | S | SS |
|---|-----|----|---|---|----|
| 1. Saya berpikir akan menggunakan sistem ini lagi. | | | | ✓ | |
| 2. Saya merasa sistem ini sulit untuk digunakan | | ✓ | | | |
| 3. Saya merasa sistem ini mudah untuk digunakan | | | | ✓ | |
| 4. Saya membutuhkan bantuan dari orang lain untuk menggunakan sistem ini | | ✓ | | | |
| 5. Saya merasa fitur atau fungsi yang berjalan pada sistem ini berjalan dengan baik | | | | | ✓ |
| 6. Saya merasa banyak hal yang tidak perlu dalam sistem ini | | ✓ | | | |
| 7. Saya merasa orang lain tidak memahami penggunaan dari sistem ini | | ✓ | | | |
| 8. Saya merasa sistem ini membingungkan | | ✓ | | | |
| 9. Saya merasa tidak ada kendala dalam menggunakan sistem ini | | | | ✓ | |
| 10. Saya perlu membiasakan diri untuk menggunakan sistem ini | | | ✓ | | |

Jakarta, Juni - 10 - 2021



LEMBAR KUISIONER SYSTEM USABILITY SCALE (SUS) SISTEM INFORMASI E-COMMERCE NUE PINS

Nama : Maisa Kusaci

Pekerjaan : wirasaha

Jawablah pertanyaan berikut ini dengan memberi salah satu tanda checklist (✓) pada setiap pertanyaan di kolom jawaban yang tersedia.

| | | |
|---------------------------|-------------------|----------------|
| Keterangan : | TS: Tidak Setuju | R: Ragu - Ragu |
| STS : Sangat tidak setuju | SS: Sangat Setuju | |
| S : Setuju | | |

| Pertanyaan : | STS | TS | R | S | SS |
|---|-----|----|---|---|----|
| 1. Saya berpikir akan menggunakan sistem ini lagi. | | | | ✓ | |
| 2. Saya merasa sistem ini sulit untuk digunakan | | ✓ | | | |
| 3. Saya merasa sistem ini mudah untuk digunakan | | | | | ✓ |
| 4. Saya membutuhkan bantuan dari orang lain untuk menggunakan sistem ini | | ✓ | | | |
| 5. Saya merasa fitur atau fungsi yang berjalan pada sistem ini berjalan dengan baik | | | | ✓ | |
| 6. Saya merasa banyak hal yang tidak perlu dalam sistem ini | | ✓ | | | |
| 7. Saya merasa orang lain tidak memahami penggunaan dari sistem ini | | | ✓ | | |
| 8. Saya merasa sistem ini membingungkan | | ✓ | | | |
| 9. Saya merasa tidak ada kendala dalam menggunakan sistem ini | | | | | ✓ |
| 10. Saya perlu membiasakan diri untuk menggunakan sistem ini | | | | ✓ | |

Jakarta, 10 Jun 2021

