**SW Technical Document**

**Pengembangan Aplikasi Pengelolaan Marketplace Multi-channel MAGER**

**Berbasis Website**

**Dibuat Oleh :**

|  |  |
| --- | --- |
| 12S19033 | San Antonio Limbong |
| 12S19050 | Yemima Febe Yanti Marpaung |
| 12S19052 | Mulyani Gabe Sayoni Simanjuntak |
| 12S19059 | Rut Yana Gultom |

**Untuk :**

**PT Tripasysfo Development**

**Jakarta**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Kerja Praktek 2022**  **Institut Teknologi Del** | | | |  |
| *No. Dokumen: SW-KP-22-111A* | | *Versi: 00.01* | *Tanggal : 24-06-22* | *Jumlah Halaman : 114* | |

**DAFTAR ISI**

[1 Introduction 9](#_Toc119326862)

[1.1 Purpose of Document 9](#_Toc119326863)

[1.2 Scope 9](#_Toc119326864)

[1.3 Definition,Acronym and Abbreviation 9](#_Toc119326865)

[1.4 Identification and Numbering 10](#_Toc119326866)

[1.5 Reference Documents 11](#_Toc119326867)

[1.6 Document Summary 12](#_Toc119326868)

[2 System Overview 13](#_Toc119326869)

[2.1 Current System Overview 13](#_Toc119326870)

[2.1.1 [BP-CS-01] Business Process Melihat Daftar Produk 13](#_Toc119326871)

[2.1.2 [BP-CS-02] Business Process Menambahkan Produk 14](#_Toc119326872)

[2.1.3 [BP-CS-03] Business Process Mengedit Produk 15](#_Toc119326873)

[2.1.4 [BP-CS-04] Business Process Menghapus Produk 16](#_Toc119326874)

[2.2 Target System Overview 17](#_Toc119326875)

[2.2.1 [BP-TS-01] Business Process Melihat Produk 18](#_Toc119326876)

[2.2.2 [BP-TS-02] Business Process Menambahkan Produk 18](#_Toc119326877)

[2.2.3 [BP-TS-03] Business Process Mengedit Produk 20](#_Toc119326878)

[2.2.4 [BP-TS-04] Business Process Menghapus Produk 21](#_Toc119326879)

[3 Software General Description 23](#_Toc119326880)

[3.1 Product Main Function 23](#_Toc119326881)

[3.2 Users Characteristics 24](#_Toc119326882)

[3.3 Constraints 24](#_Toc119326883)

[3.4 SW Environment 25](#_Toc119326884)

[3.4.1 Development Environment 25](#_Toc119326885)

[3.4.2 Operational Environment 25](#_Toc119326886)

[4 Requirement Definition 27](#_Toc119326887)

[4.1 External Interface 27](#_Toc119326888)

[4.1.1 User Interface 27](#_Toc119326889)

[4.1.2 Hardware Interface 28](#_Toc119326890)

[4.1.3 Software Interface 28](#_Toc119326891)

[4.1.4 Communication Interface 28](#_Toc119326892)

[4.2 Functional Description 28](#_Toc119326893)

[4.2.1 Use Case Diagram 29](#_Toc119326894)

[4.2.2 Use Case Scenario 30](#_Toc119326895)

[4.3 Data Requirement 34](#_Toc119326896)

[4.3.1 Entity Relationship Diagram 34](#_Toc119326897)

[4.4 Functional Requirement 1](#_Toc119326898)

[4.5 Non Functional Requirement 2](#_Toc119326899)

[4.6 Design Constraint 3](#_Toc119326900)

[4.6.1 Software Language 3](#_Toc119326901)

[4.6.2 Development Tools 4](#_Toc119326902)

[5 Design Description 5](#_Toc119326903)

[5.1 Data Description 5](#_Toc119326904)

[5.1.1 Domain/Type Definition 5](#_Toc119326905)

[5.1.2 Conceptual Data Model 12](#_Toc119326906)

[5.1.3 Physical Data Model 1](#_Toc119326907)

[5.1.4 Tables 1](#_Toc119326908)

[6 Detail Design Description 3](#_Toc119326909)

[6.1 Tabel Structure 3](#_Toc119326910)

[6.1.1 Tabel sme\_marketplace 3](#_Toc119326911)

[6.1.2 Tabel sme\_contact 5](#_Toc119326912)

[6.1.3 Tabel admin\_user 7](#_Toc119326913)

[6.1.4 Tabel sme\_bill 8](#_Toc119326914)

[6.1.5 Tabel sme\_packing\_info 12](#_Toc119326915)

[6.1.6 Tabel sme\_product\_transaction 13](#_Toc119326916)

[6.1.7 Tabel sme\_transaction 14](#_Toc119326917)

[6.1.8 Tabel sme\_product 16](#_Toc119326918)

[6.1.9 Tabel sme\_incurred\_fees 18](#_Toc119326919)

[6.1.10 Tabel sme\_product\_skus 19](#_Toc119326920)

[6.1.11 Tabel sme\_product\_image 21](#_Toc119326921)

[6.1.12 Tabel sme\_product\_skus\_image 22](#_Toc119326922)

[6.1.13 Tabel sme\_packed\_item 23](#_Toc119326923)

[6.2 Class Diagram 24](#_Toc119326924)

[6.3 Sequence Diagram 24](#_Toc119326925)

[6.3.1 Sequence Diagram Melihat Produk 24](#_Toc119326926)

[6.3.2 Sequence Diagram Menambahkan Produk 28](#_Toc119326927)

[6.3.3 Sequence Diagram Mengedit Produk 37](#_Toc119326928)

[6.3.4 Sequence Diagram Menghapus Produk 42](#_Toc119326929)

[6.4 Physical File 44](#_Toc119326930)

[6.5 Traceability 48](#_Toc119326931)

[6.5.1 Data 48](#_Toc119326932)

[6.5.2 Requirements 49](#_Toc119326933)

[7 Testing 51](#_Toc119326934)

[7.1 Test Preparation 51](#_Toc119326935)

[7.1.1 Procedural Preparation 51](#_Toc119326936)

[7.1.2 HW & Network Preparation 51](#_Toc119326937)

[7.1.3 SW Preparation 51](#_Toc119326938)

[7.2 Test Plan and Identification 51](#_Toc119326939)

[7.3 Test Script & Result 52](#_Toc119326940)

[7.3.1 Test Script Butir-Uji-1 52](#_Toc119326941)

[7.3.2 Test Script Butir-Uji-2 53](#_Toc119326942)

[7.3.3 Test Script Butir-Uji-3 54](#_Toc119326943)

[7.4 Test Summary Result & History 55](#_Toc119326944)

[8 SW Item Description & Installation 56](#_Toc119326945)

[8.1 SW Item & Location 56](#_Toc119326946)

[8.1.1 Client Side 56](#_Toc119326947)

[8.1.2 Server-side 57](#_Toc119326948)

[8.2 SW Installation 59](#_Toc119326949)

[8.3 Precondition 59](#_Toc119326950)

[8.4 Procedures (Working Instruction) 59](#_Toc119326951)

[8.5 Reporting 59](#_Toc119326952)

[LAMPIRAN 60](#_Toc119326953)

[Sejarah Versi 61](#_Toc119326954)

[Sejarah Perubahan 62](#_Toc119326955)

**DAFTAR TABEL**

[Tabel 1 Definisi 9](#_Toc119401122)

[Tabel 2 Akronim dan Singkatan 10](#_Toc119401123)

[Tabel 3 Identification and Numbering 10](#_Toc119401124)

[Tabel 4 Deskripsi/Fungsi 27](#_Toc119401125)

[Tabel 5 Hardware Interface 28](#_Toc119401126)

[Tabel 6 Software Interface 28](#_Toc119401127)

[Tabel 7 Use Case Scenario Melihat Produk 30](#_Toc119401128)

[Tabel 8 Use Case Scenario Menambahkan Produk 30](#_Toc119401129)

[Tabel 9 Use Case Scenario Mengedit Produk 32](#_Toc119401130)

[Tabel 10 Use Case Scenario Menghapus Produk 33](#_Toc119401131)

[Tabel 11 Functional Requirement 1](#_Toc119401132)

[Tabel 12 Non Functional Requirement 2](#_Toc119401133)

[Tabel 13 Domain/Type Defenition 5](#_Toc119401134)

[Tabel 14 Tables 1](#_Toc119401135)

[Tabel 15 Tabel sme\_marketplace 3](#_Toc119401136)

[Tabel 16 Tabel sme\_contact 5](#_Toc119401137)

[Tabel 17 Tabel admin\_user 7](#_Toc119401138)

[Tabel 18 Tabel sme\_bill 9](#_Toc119401139)

[Tabel 19 Tabel sme\_packing\_info 12](#_Toc119401140)

[Tabel 20 Tabel sme\_product\_transaction 14](#_Toc119401141)

[Tabel 21 Tabel sme\_transaction 15](#_Toc119401142)

[Tabel 22 Tabel sme\_product 16](#_Toc119401143)

[Tabel 23 sme\_incurred\_fees 18](#_Toc119401144)

[Tabel 24 sme\_product\_skus 19](#_Toc119401145)

[Tabel 25 sme\_product\_image 21](#_Toc119401146)

[Tabel 26 Tabel sme\_product\_skus\_image 22](#_Toc119401147)

[Tabel 27 Tabel sme\_packed\_item 23](#_Toc119401148)

[Tabel 28 Object Specification Melihat Produk 28](#_Toc119401149)

[Tabel 29 *Error Message* 29](#_Toc119401150)

[Tabel 30 Object Specification Menambahkan Produk 36](#_Toc119401151)

[Tabel 31 Error Message 37](#_Toc119401152)

[Tabel 32 Object Specification Mengedit Produk 41](#_Toc119401153)

[Tabel 33 Object Specification Menghapus Produk 43](#_Toc119401154)

[Tabel 34 Physical File 44](#_Toc119401155)

[Tabel 35 Data 48](#_Toc119401156)

[Tabel 36 Requirements 50](#_Toc119401157)

[Tabel 37 Test Script Butir-Uji-1 52](#_Toc119401158)

[Tabel 38 Test Script Butir-Uji-2 53](#_Toc119401159)

[Tabel 39 Test Script Butir-Uji-3 54](#_Toc119401160)

[Tabel 40 Test Summary Result & History 55](#_Toc119401161)

[Tabel 41 Client Side 56](#_Toc119401162)

[Tabel 42 Server-side 57](#_Toc119401163)

**DAFTAR GAMBAR**

[Gambar 1 [BP-CS-01] Current System BPMN Melihat Daftar Produk 13](#_Toc119401164)

[Gambar 2 [BP-CS-02] Current System BPMN Menambahkan Produk 14](#_Toc119401165)

[Gambar 3 [BP-CS-03] Current System BPMN Mengedit Produk 16](#_Toc119401166)

[Gambar 4 [BP-CS-04] Current System BPMN Menghapus Produk 17](#_Toc119401167)

[Gambar 5 [BP-TS-01] Target System BPMN Melihat Produk 18](#_Toc119401168)

[Gambar 6 [BP-TS-02] Target System BPMN Menambahkan Produk 19](#_Toc119401169)

[Gambar 7 [BP-TS-03] Target System BPMN Mengedit Produk 21](#_Toc119401170)

[Gambar 8 [BP-TS-04] Target System BPMN Menghapus Produk 22](#_Toc119401171)

[Gambar 9 Use Case Diagram 29](#_Toc119401172)

[Gambar 10 Entity Relational Diagram 1](#_Toc119401173)

[Gambar 11 Detail Entity Relational Diagram - Product 2](#_Toc119401174)

[Gambar 12 Conceptual Data Model 1](#_Toc119401175)

[Gambar 13 Detail Conceptual Data Model - Product 1](#_Toc119401176)

[Gambar 14 Physical Data Model 1](#_Toc119401177)

[Gambar 15 Detail Physical Data Model - Product 1](#_Toc119401178)

[Gambar 16 Struktur Tabel sme\_marketplace 5](#_Toc119401179)

[Gambar 17 Struktur Tabel sme\_contact 7](#_Toc119401180)

[Gambar 18 Struktur Tabel admin\_user 8](#_Toc119401181)

[Gambar 19 Struktur Tabel sme\_bill 12](#_Toc119401182)

[Gambar 20 Struktur Tabel sme\_packing\_info 13](#_Toc119401183)

[Gambar 21 Struktur Tabel sme\_product\_transaction 14](#_Toc119401184)

[Gambar 22 Struktur Tabel sme\_transaction 16](#_Toc119401185)

[Gambar 23 Struktur Tabel sme\_product 18](#_Toc119401186)

[Gambar 24 Struktur Tabel sme\_incurred\_fees 19](#_Toc119401187)

[Gambar 25 Struktur Tabel sme\_product\_skus 21](#_Toc119401188)

[Gambar 26 Struktur Tabel sme\_product\_image 22](#_Toc119401189)

[Gambar 27 Struktur Tabel sme\_product\_skus\_image 23](#_Toc119401190)

[Gambar 28 Struktur Tabel sme\_packed\_item 24](#_Toc119401191)

[Gambar 29 Class Diagram 24](#_Toc119401192)

[Gambar 30 Sequence Diagram Melihat Produk 25](#_Toc119401193)

[Gambar 31 Design Screen Layout Melihat Produk Semua 26](#_Toc119401194)

[Gambar 32 Design Screen Layout Melihat Produk Live 26](#_Toc119401195)

[Gambar 33 Design Screen Layout Melihat Produk Habis 27](#_Toc119401196)

[Gambar 34 Design Screen Layout Melihat Produk Diarsipkan 27](#_Toc119401197)

[Gambar 35 Detail Screen Layout Melihat Produk 28](#_Toc119401198)

[Gambar 36 Sequence Diagram Menambahkan Produk 29](#_Toc119401199)

[Gambar 37 Design Screen Layout Menambahkan Produk 31](#_Toc119401200)

[Gambar 38 Detail Screen Layout Memilih Marketplace 32](#_Toc119401201)

[Gambar 39 Detail Screen Layout Menambahkan Produk (Lazada) 34](#_Toc119401202)

[Gambar 40 Detail Screen Layout Menambahkan Produk (Tokopedia) 36](#_Toc119401203)

[Gambar 41 Sequence Diagram Mengedit Produk 37](#_Toc119401204)

[Gambar 42 Detail Screen Layout Mengedit Produk (Lazada) 39](#_Toc119401205)

[Gambar 43 Detail Screen Layout Mengedit Produk (Tokopedia) 41](#_Toc119401206)

[Gambar 44 Sequence Diagram Menghapus Produk 42](#_Toc119401207)

[Gambar 45 Detail Screen Layout Halaman Menghapus Produk 43](#_Toc119401208)

# **Introduction**

Pada bab ini dibahas mengenai tujuan penulisan dokumen, ruang lingkup dokumen,definisi, akronim, dan singkatan yang terdapat dalam dokumen, identifikasi dan aturan penomoran dokumen, dokumen rujukan, serta ringkasan dokumen. Berikut akan dijelaskan pada masing-masing sub bab tersebut.

## **Purpose of Document**

Dokumen ini ditulis oleh untuk para pengembang sistem dengan tujuan sebagai berikut:

1. Menjelaskan gambaran sistem, kebutuhan fungsional, kebutuhan antarmuka, dan kebutuhan lainnya.
2. Sebagai dokumen rujukan yang menjadi panduan bagi tim developer dalam mengembangkan sistem.

## **Scope**

Ruang lingkup dokumen ini meliputi penjelasan mengenai desain sistem yang akan dikembangkan dan penjelasan mengenai batasan pengembangan aplikasi serta fungsi-fungsi yang digunakan.

## **Definition,Acronym and Abbreviation**

Berikut adalah daftar definisi, akronim, dan singkatan yang digunakan dalam dokumen. Definisi yang digunakan pada dokumen ini dapat dilihat pada Tabel 1.

Tabel 1 Definisi

| **No** | **Definisi** | **Keterangan** |
| --- | --- | --- |
| 1. | *Developer* | Orang yang mengembangkan produk sesuai kebutuhan *client* |
| 2. | *Hardware* | Perangkat keras dan peranti yang mendukung sistem komputer |
| 3. | *Software* | Perangkat lunak yang digunakan untuk membangun sistem |
| 4. | *Supervisor* | Orang yang melakukan *review* terhadap dokumentasi proyek dan mengontrol pengerjaan proyek |
| 5. | *Deliverable* | Setoran |

Daftar akronim dan singkatan yang digunakan dalam dokumen ini diuraikan pada tabel 2.

Tabel 2 Akronim dan Singkatan

|  |  |  |
| --- | --- | --- |
| **No** | **Akronim dan Singkatan** | **Keterangan** |
| 1. | PiP | *Project Implementation Plan* adalah dokumen perencanaan pembangunan website Forum Mahasiswa IT Del yang menjadi langkah awal dalam pengerjaan proyek |
| 2 | ToR | *Term of Reference* adalah dokumen yang berisi batasan mengenai gambaran tujuan, ruang lingkup dan struktur sebuah proyek. |
| 3. | KP | Kerja Praktik |
| 4. | LA | *Log Activity* |
| 5. | ERD | *Entity Relationship Diagram* |
| 6. | IT DEL | Institut Teknologi Del |
| 7. | *Web App* | *Web Application* |

## **Identification and Numbering**

Artefak yang harus dikumpulkan sebagai bagian dari hasil pengerjaan Proyek Kerja Praktik di Institut Teknologi Del harus mengikuti kaidah penomoran yang dicantumkan dalam dokumen Standar Penomoran dan Tata Nama artefak Kerja Praktik dan Tugas Akhir yang akan ditunjukkan pada Tabel 4 yang ada di bawah ini.

Tabel 3 Identification and Numbering

| **No.** | **Kategori** | **Deskripsi Aturan yang Digunakan** |
| --- | --- | --- |
| 1. | Aturan penamaan dokumen dengan ketentuan | SW-KP-22-GG  Dimana GG adalah nomor kelompok. Maka dokumen ini dinamai dengan SW-KP-22-111A |
| 2. | Jenis *font* | 1. Judul dokumen adalah Times New Roman 22 pt, *font color* *bold-black.*  2. Judul bab adalah Arial 12 pt, *font color bold-black*  3. Judul sub bab adalah Arial 12 pt, *font color bold-black*  4. Isi bab adalah Times New Roman 12 pt dengan *font color black*  5. Judul tabel adalah Times New Roman 10 pt dengan *font color bold-black*  6. Isi tabel adalah Times New Roman 10 pt, *font color black*  7. Keterangan gambar adalah Times New Roman 10 pt dengan *font color bold-black*  8. Setiap kosa kata Bahasa Inggris menggunakan *font* Times New Roman 12 pt dengan *font italic.* |
| 3. | Penomoran dan Penamaan | Aturan penomoran dan penamaan bab serta sub bab:  Penomoran pada Bab: 1,2,3  Misalnya: **1 Pendahuluan**  Penomoran pada Sub bab: 1.1 , 1.2 dan seterusnya  Misalnya: **1.1 Maksud Penulisan Dokumen**  Penomoran pada tiap Sub-sub bab: 2.4.1 , 2.4.2 dan seterusnya  Misalnya: **2.4.1 Sumber Daya Manusia**  Aturan penomoran dan penamaan tabel dan gambar:  Tabel : **Tabel 1. Definisi**  Gambar: **Gambar 1. Struktur Organisasi** |

## **Reference Documents**

Dokumen yang menjadi rujukan dokumen ini, antara lain:

1. Std-KA-09, Standar Penomoran dan Tata Nama Artefak Kerja Praktek dan Tugas Akhir, terbitan tahun 2009
2. ToR-KP-22-111A, *Term of Reference* (ToR) adalah dokumen yang berisi topik, beserta deskripsi singkat mengenai proyek yang digunakan sebagai acuan dalam pelaksanaan Kerja Praktik.
3. PiP-KP-22-111A, *Project Implementation Plan* (PiP) adalah dokumen yang berisi perencanaan pekerjaan yang dilakukan sebelum mengembangkan aplikasi dan berisi rincian tugas yang akan dilakukan dalam pengerjaan proyek.
4. MoM-KP-22-111A, *Minutes of Meeting* (MoM) atau risalah rapat adalah dokumen yang berisi agenda rapat dan hasil yang dicapai dalam setiap rapat yang dilakukan selama pelaksanaan Kerja Praktik.
5. LA-KP-22-111A, *Log Activity* (LA) adalah laporan kegiatan yang dibuat per minggu, yang ditandatangani supervisor perusahaan.

## **Document Summary**

Dokumen teknis Pengembangan Aplikasi Pengelolaan *Marketplace* Multi-channel MAGER Berbasis *Website* terdiri dari 8 bab berisi tentang:

* + - 1. Bab 1 *Introduction*, yang menjelaskan tentang tujuan pembuatan dokumen, ruang lingkup dokumen, definisi, akronim dan singkatan yang digunakan dalam dokumen, aturan penomoran dokumen, dokumen rujukan, dan ringkasan dokumen.
      2. Bab 2 *System Overview*, yang menjelaskan gambaran proses bisnis dan prosedur dari sistem yang ada saat ini dan sistem yang akan dibangun.
      3. Bab 3 *Software General Description*, yang menjelaskan deskripsi umum perangkat lunak, fungsi-fungsi utama yang diberikan ke pengguna, karakteristik pengguna, batasan perangkat yang digunakan dan lingkungan dimana aplikasi akan dikembangkan dan dioperasikan.
      4. Bab 4 *Requirement Definition*, yang menjelaskan definisi kebutuhan – kebutuhan dalam pembangunan sistem, yaitu *external interface*, deskripsi fitur yang ada dalam sistem yang dibangun, kebutuhan data, kebutuhan fungsional, kebutuhan non-fungsional dan batasan perancangan sistem yang dibangun.
      5. Bab 5 *Design Description*, yang menjelaskan tentang perancangan basis data dari aplikasi yang dibangun, yaitu pendefinisian tipe data dan domain data, pemodelan data secara konseptual dan fisik, serta deskripsi dari tabel di basis data.
      6. Bab 6 *Detail Design Description*, yang menjelaskan tentang struktur tabel, *Class Diagram*, *Sequence Diagram, physical file* dan *traceability* dari sistem yang dibangun.
      7. Bab 7 *Testing*, yang menjelaskan tentang persiapan pengujian, perencanaan pengujian, identifikasi perencanaan pengujian, hasil pengujian dan sejarah pengujian yang dilakukan
      8. Bab 8 SW Item *Description & Installation*, yang menjelaskan semua berkas yang diperlukan untuk instalasi dan pengoperasian perangkat lunak.

# **System Overview**

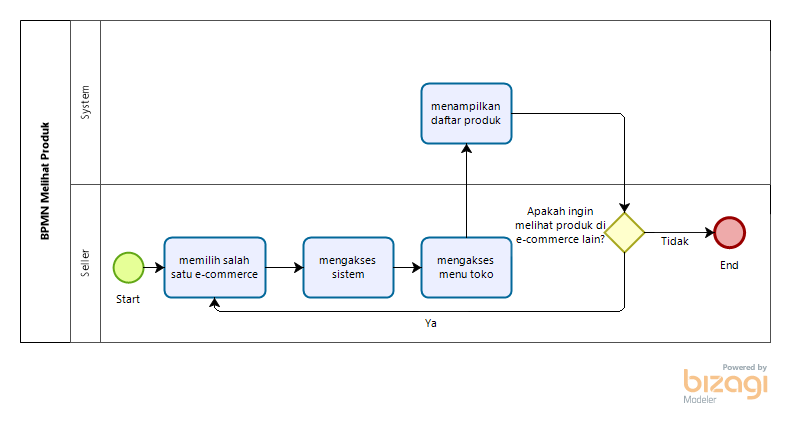
Pada bab ini menjelaskan deskripsi umum sistem saat ini dan sistem target. Bagian-bagian yang dijelaskan mencakup proses bisnis, prosedur dari setiap proses bisnis, dan waktu layanan.

## **Current System Overview**

Pada sub bab ini diuraikan sistem yang sudah ada dan sedang berlangsung, ditinjau dari proses bisnis, prosedur yang berlangsung dan efisiensi waktu.

### **[BP-CS-01] Business Process Melihat Daftar Produk**

Proses bisnis melihat daftar produk pada *current system* ini merupakan proses dimana *seller* melihat daftar produk yang tersedia di toko sesuai dengan marketplacenya masing-masing. Adapun prosesnya yakni sebagai berikut. *Seller memilih salah satu e-commerce kemudian mengakses e-commerce tersebut.* Selanjutnya *seller* memilih menu Toko. Terakhir, *system* akanmenampilkan detail dari produk *seller* pada *e-commerce* tersebut.



Gambar 1 [BP-CS-01] Current System BPMN Melihat Daftar Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Melihat Daftar Produk pada *current system* adalah sebagai berikut :

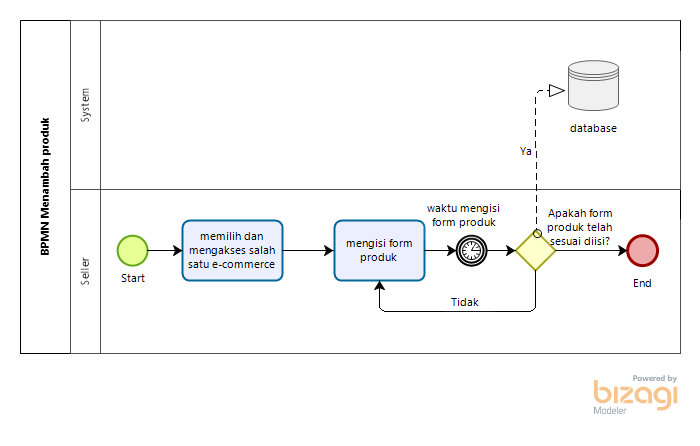
1. *Seller* memilih salah satu *e-commerce*
2. *Seller* mengakses system
3. *Seller* mengakses menu “Toko”
4. Daftar produk yang tersedia pada toko akan ditampilkan

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan melihat daftar produk pada *current system* yang berlaku saat ini adalah 2 - 4 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

### **[BP-CS-02] Business Process Menambahkan Produk**

Proses bisnis menambahkan produk pada *current system* ini merupakan proses dimana *seller* melakukan penambahan produk yang tersedia pada toko melalui akun toko sesuai dengan marketplace toko tersebut. Adapun prosesnya yakni sebagai berikut. Pertama, *seller* memilih salah satu *e-commerce* dan melakukan login. Berikutnya *seller* menekan tombol tambah produk. Terakhir, *seller* mengisi form sesuai dengan spesifikasi produk yang akan ditambahkan pada toko dan penambahan produk akan tersimpan pada *database*.



Gambar 2 [BP-CS-02] Current System BPMN Menambahkan Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Menambahkan Produk pada *current system* adalah sebagai berikut :

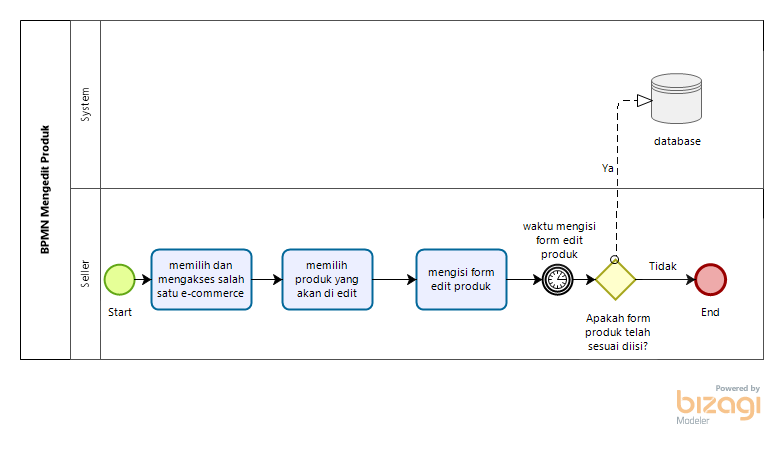
1. *Seller* memilih salah satu *e-commerce*
2. *Seller* mengakses system
3. *Seller* mengakses menu “Toko”
4. *Seller* menekan tombol tambah produk danmengisi form sesuai dengan spesifikasi produk yang akan ditambahkan pada toko.
5. Produk berhasil ditambahkan.

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan menambahkan produk pada *current system* yang berlaku saat ini adalah 6 - 10 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

### **[BP-CS-03] Business Process Mengedit Produk**

Proses bisnis mengedit produk pada *current system* ini merupakan proses dimana *seller* melakukan perubahan informasi produk di toko melalui akun toko sesuai dengan *marketplace* toko tersebut. Adapun prosesnya yakni sebagai berikut. Pertama, *seller* memilih dan mengakses *e-commerce.* Kemudian *seller* memilih produk yang akan di edit danmengisi form sesuai dengan spesifikasi produk yang akan diubah pada produk tersebut. Terakhir, perubahan pada produk yang di edit akan tersimpan pada *database.*



Gambar 3 [BP-CS-03] Current System BPMN Mengedit Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Mengedit Produk pada *current system* adalah sebagai berikut :

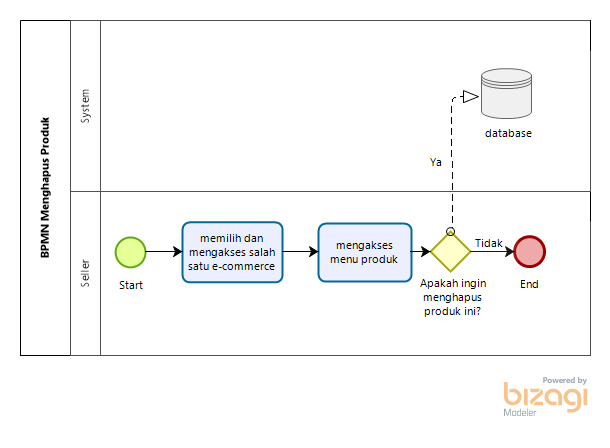
1. *Seller* memilih dan mengakses salah satu *e-commerce*.
2. *Seller* memilih produk yang akan diedit.
3. *Seller* mengisi form sesuai dengan spesifikasi produk yang akan diubah pada produk tersebut.
4. Produk berhasil diedit.

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan mengedit produk pada *current system* yang berlaku saat ini adalah 6 - 10 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

### **[BP-CS-04] Business Process Menghapus Produk**

Proses bisnis menghapus produk pada *current system* ini merupakan proses dimana *seller* melakukan penghapusan produk dari toko melalui akun toko sesuai dengan marketplace toko tersebut. Adapun prosesnya yakni sebagai berikut. Pertama, *seller* memilih dan mengakses *e-commerce* dengan masing-masing *marketplace*. Kemudian *seller* memilih menu Produk yang akan di hapus dan menekan tombol hapus. Dan perubahan yang dilakukan pada produk akan tersimpan pada *database.*



Gambar 4 [BP-CS-04] Current System BPMN Menghapus Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Mengedit Produk pada *current system* adalah sebagai berikut :

1. *Seller* memilih dan mengakses salah satu *e-commerce*.
2. *Seller* memilih produk yang akan dihapus.
3. *Seller* menekan tombol hapus.
4. Produk berhasil dihapus.

#### **Service Time**

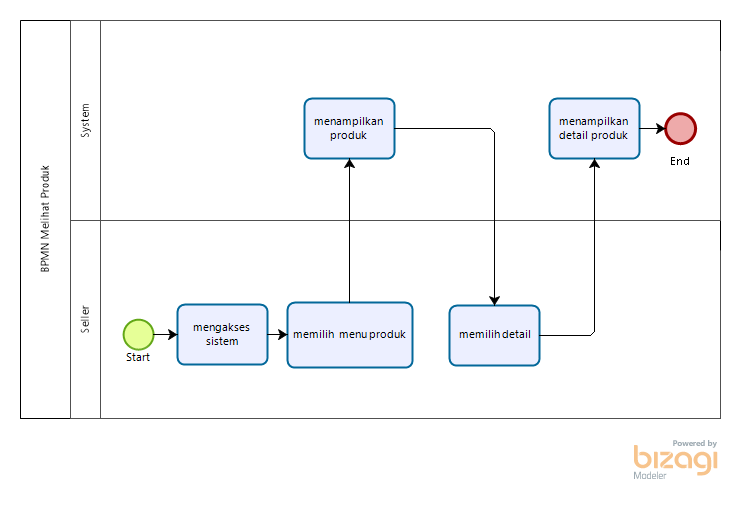
Waktu yang dibutuhkan untuk melakukan kegiatan mengedit produk pada *current system* yang berlaku saat ini adalah 2 - 4 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

## **Target System Overview**

Pada sub bab ini diuraikan sistem menjadi target dalam implementasi, ditinjau dari proses bisnis, prosedur yang ditargetkan dan efektivitas penggunaan waktunya.

### **[BP-TS-01] Business Process Melihat Produk**

Proses bisnis melihat produk pada *target system* ini merupakan proses dimana *seller* melihat data produk yang tersedia pada masing-masing toko pada setiap *marketplace* yang dimilikinya. Sehingga daftar maupun informasi produk yang tersedia pada masing-masing akun toko yang terintegrasi pada *marketplace* dapat dilihat secara bersamaan sehingga lebih efisien. Adapun prosesnya yakni sebagai berikut. Pertama, *seller* akan mengakses sistem dengan *login* ke sistem. Selanjutnya *seller* memilih menu produk lalu semua daftar produk yang dimiliki oleh *seller* akan ditampilkan oleh sistem.



Gambar 5 [BP-TS-01] Target System BPMN Melihat Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Melihat Produk pada *target system* adalah sebagai berikut :

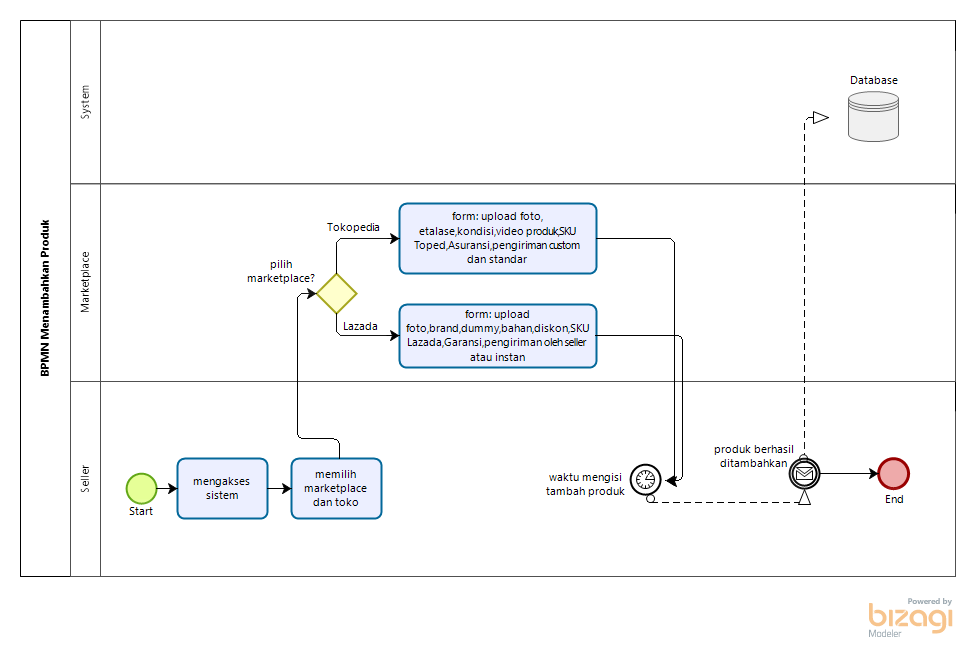
1. *Seller* melakukan *login* ke sistem.
2. *Seller* menekan tombol menu “Produk”
3. Daftar produk yang tersedia pada toko akan ditampilkan

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan melihat produk pada *target system* yang berlaku saat ini adalah 1 - 3 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

### **[BP-TS-02] Business Process Menambahkan Produk**

Proses bisnis menambah produk pada target system MAGER dimulai dengan proses dimana *seller* akan login ke sistem MAGER, kemudian pada menu produk *seller* akan memilih *marketplace* yang ingin di input produknya yaitu *marketplace* Lazada dan *marketplace* Tokopedia, setelah itu akan dipilih toko. Selanjutnya seller akan mengisi form tambah produk sesuai dengan marketplace yang dipilih, jika marketplace Lazada meliputi form upload foto, spesifikasi (brand, bahan, model, dummy), stok harga varian (tambah produk, harga dan stok, diskon, SKU Lazada), service (jenis garansi, garansi, kebijakan garansi), delivery (berat, panjang, lebar, tinggi, pre-order, opsi pengiriman oleh seller maupun instan), sedangkan pada marketplace Tokopedia meliputi form upload foto, informasi produk (nama, kategori, etalase) , detail produk (kondisi, deskripsi produk, video produk), varian dan harga (minimum pemesanan, harga satuan), pengelolaan (status produk, stok, SKU Tokopedia), berat dan pengiriman (berat, panjang, lebar, tinggi, asuransi pengiriman, layanan pengiriman custom maupun standar). Setelah form produk telah diisi maka akan disimpan ke *database* dan mendapatkan notifikasi bahwa form produk telah berhasil ditambah.



Gambar 6 [BP-TS-02] Target System BPMN Menambahkan Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Menambahkan Produk pada *target system* adalah sebagai berikut :

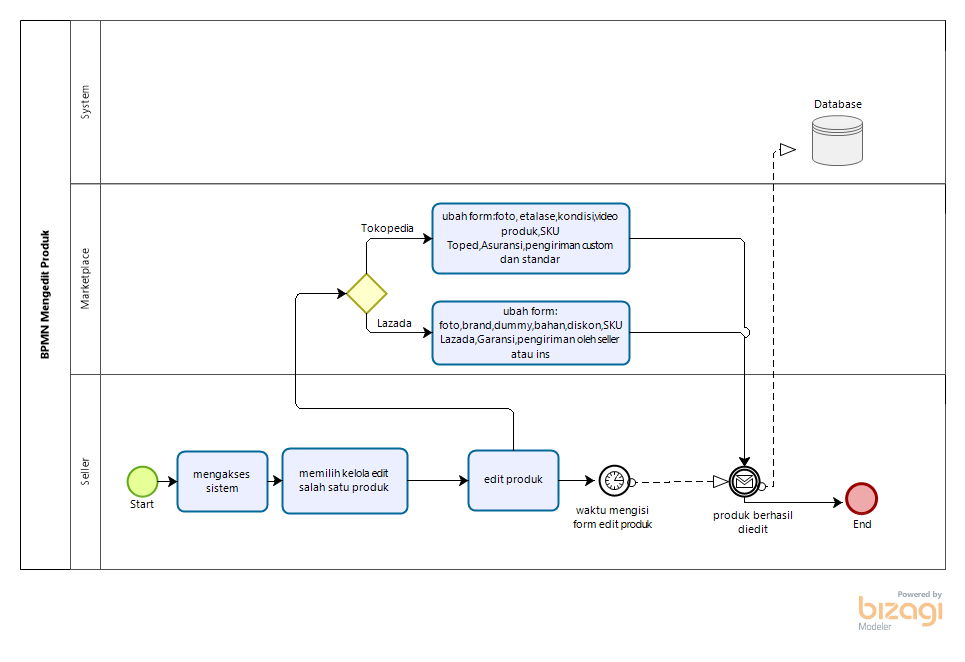
1. *Seller* melakukan *login* dengan sistem.
2. *Seller* memilih *marketplace* dan toko.
3. *Seller* mengisi form sesuai dengan field yang ditampilkan berdasarkan *marketplace* yang dipilih.
4. Produk berhasil ditambahkan.

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan menambahkan produk pada *target system* yang berlaku saat ini adalah 5 - 10 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

### **[BP-TS-03] Business Process Mengedit Produk**

Proses bisnis mengedit produk pada *target system* ini merupakan proses dimana *seller* melakukan perubahan informasi produk yang tersedia pada masing-masing toko sesuai dengan *marketplace* toko melalui sistem. Adapun prosesnya yakni sebagai berikut. Pertama, *Seller* melakukan *login* dengan sistem. Kemudian s*eller* memilih menu Produk diikuti dengan aksi berupa memilih produk serta menekan tombol edit. Tahapan berikutnya *seller* mengisi form sesuai dengan spesifikasi produk yang akan diubah pada produk tersebut.



Gambar 7 [BP-TS-03] Target System BPMN Mengedit Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Mengedit Produk pada *target system* adalah sebagai berikut :

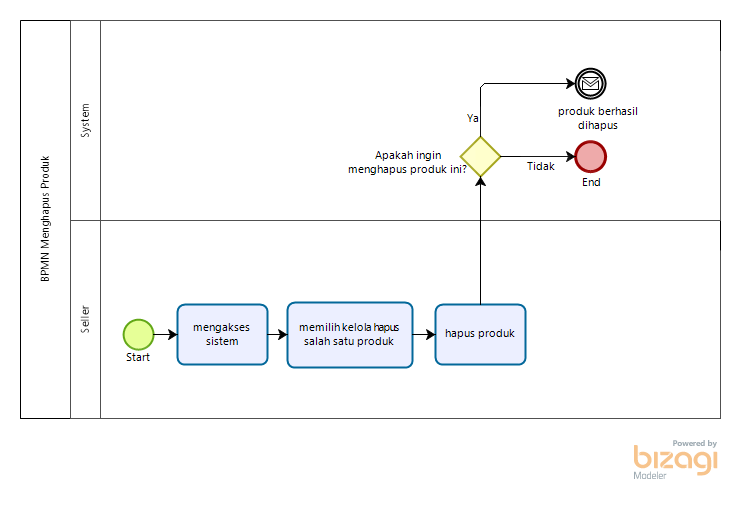
1. *Seller* melakukan *login* dengan sistem.
2. *Seller* memilih tombol kelola selanjutnya edit untuk mengupdate salah satu produk.
3. *Seller* mengedit form sesuai dengan field yang ditampilkan berdasarkan *marketplace* yang dipilih.
4. Produk berhasil diedit

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan mengedit produk pada *target system* yang berlaku saat ini adalah 1 - 3 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

### **[BP-TS-04] Business Process Menghapus Produk**

Proses bisnis menghapus produk pada *target system* ini merupakan proses dimana *seller* melakukan penghapusan produk. Adapun proses penghapusan produk ini dilakukan untuk semua produk, dimana tidak terpaku terhadap sebuah produk yang di marketplace tertentu dikarenakan sistem ini belum ada partisi/pembagian yang khusus pada setiap produk di toko maupun marketplace yang dimiliki. Untuk prosesnya yakni sebagai berikut. Pertama, seller akan melakukan login dengan sistem. Kemudian s*eller* memilih menu memilih kelola hapus, saat *seller* ingin menghapus produk maka akan ditampilkan pop-up validasi untuk menghapus produk. Jika *seller* memilih menghapus produk maka akan memberikan notifikasi bahwa produk telah berhasil dihapus, sedangkan jika *seller* memilih tidak maka proses akan selesai.



Gambar 8 [BP-TS-04] Target System BPMN Menghapus Produk

#### **Procedures**

Prosedur yang harus dilakukan dalam menjalankan Proses Bisnis Menghapus Produk pada *target system* adalah sebagai berikut :

1. *Seller* melakukan login dengan sistem.
2. *Seller* memilih tombol kelola selanjutnya hapus untuk menghapus salah satu produk.
3. *Seller* menekantombol hapus untukmemvalidasi penghapusan produk.
4. Produk berhasil dihapus.

#### **Service Time**

Waktu yang dibutuhkan untuk melakukan kegiatan menghapus produk pada *target system* yang berlaku saat ini adalah 1 - 3 menit untuk setiap akun yang dimiliki pada salah satu *marketplace.*

# **Software General Description**

Pada bab ini dibagi menjadi beberapa sub bab yang menjelaskan fungsi utama produk, karakteristik pengguna yang memanfaatkan sistem yang dibangun, batasan sistem, lingkungan perangkat lunak, ringkasan kebutuhan fungsional dan ringkasan kebutuhan non fungsional.

## **Product Main Function**

Pada sub bab ini akan dibahas mengenai fungsi dari Pengelolaan *Marketplace* Multi-channel MAGER yang akan dikembangkan dengan rincian sebagai berikut :

1. Fungsi Melihat Produk

Fungsi ini digunakan *seller* untuk melakukan Melihat produk dengan akun toko pada masing-masing *marketplace*. Sehingga output yang diharapkan adalah menampilkan daftar produk, *marketplace*, variasi, harga, stok, penjualan, status dan kelola produk yang terintegrasi dengan akun toko masing - masing *marketplace* seperti Lazada maupun Tokopedia.

1. Fungsi Menambahkan Produk

Fungsi ini digunakan *seller* untuk menambahkan produk di toko pada *marketplace*. Dalam penambahan produk terdapat beberapa komponen penting sehingga produk dapat dipublikasikan seperti Informasi Dasar, Spesifikasi atau Detail Produk, Variasi Produk, Harga serta Stok Produk, Pengiriman & Garansi Produk.

1. Fungsi Mengedit Produk

Fungsi ini digunakan *seller* untuk mengedit informasi produk yang telah dipublikasikan pada akun toko di *marketplace*. Dalam fungsi ini, *seller* dapat melakukan aksi berupa penyesuaian informasi produk seperti jumlah stok maupun harga produk, status produk berupa aktif maupun tidak aktif, serta beragam perubahan lainnya sesuai dengan kondisi sesungguhnya berdasarkan penyesuaian yang dilakukan oleh *seller.*

1. Fungsi Menghapus Produk

Fungsi ini digunakan *seller* untuk menghapus produk yang sebelumnya dibuat pada toko melalui *marketplace*. Fungsi menghapus produk meliputi semua status produk yang terdapat pada akun toko baik produk yang aktif maupun produk yang tidak aktif.

## **Users Characteristics**

Pengguna dari Pengelolaan Marketplace Multi-channel MAGER memiliki satu *role* meliputi:

**Seller**

Deskripsi Pengguna : *Seller* merupakan pengguna yang telah terdaftar di *website*

*marketplace* Multi-channel MAGER. *Seller* pada *marketplace* Multi-channel MAGER memiliki wewenang dalam melihat daftar produk, menambahkan produk, mengedit produk serta menghapus produk.

Wewenang : *Seller*

Prasyarat : Pengguna sudah login dan sedang mengakses *website* MAGER

Deskripsi Tugas : 1. Melihat daftar produk

2. Menambahkan produk

3. Mengedit produk

4. Menghapus produk

## **Constraints**

Pada proses Pengembangan Website Marketplace Multi-channel MAGER, sistem memiliki batasan khusus, antara lain:

1. *Seller* harus melakukan Registrasi terlebih dahulu di halaman Registrasi dengan mengisikan *username, password* dan mencantumkan *email* aktif, kemudian melakukan verifikasi melalui *email* aktif tersebut untuk dapat *login* kedalam sistem dengan *username* dan *password* yang didaftarkan.
2. *Seller* harus mendaftarkan (*link*) toko terlebih dahulu kedalam sistem dengan melakukan autentikasi toko sesuai dengan layanan *marketplace* yang digunakan.
3. *Seller* perlu mendaftarkan (*link*) ulang toko kedalam sistem secara berkala dalam durasi 30 hari . Hal ini sesuai dengan kebijakan layanan *marketplace* Tokopedia maupun Lazadayang digunakan untuk kredensial pemanggilan antarmuka API.
4. Reliabilitas data pada sistem sangat tergantung dengan penyediaan API dari pihak *marketplace* Lazada maupun Tokopedia. Apabila pada proses interaksi dengan *marketplace* penyedia API sedang mengalami kendala maka secara langsung akan mempengaruhi stabilitas data pada sistem.

## **SW Environment**

Pada sub bab ini dijelaskan mengenai lingkungan perangkat lunak dan perangkat keras yang digunakan oleh Tim *Developer* dalam Pengembangan Website *Marketplace* Multi-channel MAGER.

### **Development Environment**

Spesifikasi lingkungan pengembangan yang digunakan oleh tim *developer* dalam Pengembangan Marketplace Multi-channel MAGER adalah sebagai berikut:

1. Sistem Operasi : Windows
2. *Processor* : Intel(R) Core(™) i5-7200U CPU @ 2.50GHz
3. RAM : 8GB
4. DBMS : MySQL
5. *Framework* : Laravel
6. *Text Editor* : Visual Studio Code
7. *Design* : Figma
8. *Data Modeling* : app.diagrams.net, Visual Paradigm

### **Operational Environment**

*Operational Environment* menjelaskan spesifikasi aplikasi yang dibutuhkan dalam pengoperasian Pengembangan Marketplace Multi-channel MAGER. Semua kebutuhan ini berguna agar sistem dapat berjalan (beroperasi) dengan baik. Untuk pengoperasian sistem ini dibutuhkan piranti sebagai berikut:

Spesifikasi perangkat lunak lingkungan operasional aplikasi antara lain:

1. Server
2. Operating System : Windows 10
3. Server Tools : Digital Ocean Virtual Private Server
4. Server Specification : Ubuntu Server 18.04
5. DBMS : PGAdmin 4.24
6. Client

a. Operating System : Windows

b. Browser : Chrome v.84 dan Mozilla Firefox v.72

1. Spesifikasi perangkat keras lingkungan operasional aplikasi antara lain:
2. Processor : Intel(R) Core(™) i5-7200U CPU @ 2.50GHz
3. RAM : 8.00 GB
4. Perangkat keras lainnya : Keyboard, Mouse, Monitor

# **Requirement Definition**

Pada bab ini akan diuraikan deskripsi *interface* yang dibutuhkan untuk mengoperasikan sistem yang akan dibangun, yaitu *external interface* yang terdiri dari *user interface* untuk menghubungkan pengguna dengan perangkat lunak, *hardware interface* untuk mengoperasikan perangkat keras, *software interface* untuk mengoperasikan perangkat lunak dan *communication interface* untuk menangani komunikasi perangkat lunak. Pada bab ini juga terdapat uraian mengenai kebutuhan fungsional dan non fungsional serta kebutuhan data.

## **External Interface**

Kebutuhan eksternal merupakan kebutuhan yang dibutuhkan untuk mengoperasikan *website* yang dibangun. Kebutuhan pengguna dapat ditinjau dari antarmuka terdiri dari:

1. *User Interface*
2. *Hardware Interface*
3. *Software Interface*
4. *Communication Interface*

### **User Interface**

Kebutuhan antarmuka pengguna *(user interface)* dalam penggunaan *website* Pengelolaan Marketplace Multi-channel MAGER yang menjadi penghubung antara sistem dan pengguna diantaranya adalah sebagai berikut:

Tabel 4 Deskripsi/Fungsi

|  |  |  |
| --- | --- | --- |
| **No** | ***User Interface*** | **Deskripsi/Fungsi** |
| 1 | *Keyboard* | *Keyboard interface* digunakan untuk memasukkan data yang diperlukan ke dalam aplikasi. |
| 2 | *Mouse* | *Mouse interface* digunakan untuk membantu pemasukan data (sebagai pointer kursor di layar monitor). |
| 3 | *Monitor* | *Monitor interface* digunakan untuk melihat tampilan output berdasarkan proses yang dilakukan |

### **Hardware Interface**

Kebutuhan antarmuka perangkat keras (*hardware interface*) adalah kebutuhan yang digunakan oleh pengguna untuk berinteraksi dengan *website* Pengelolaan Marketplace Multi-channel MAGER. Adapun kebutuhan perangkat keras bagi pengguna dapat dilihat pada tabel berikut:

Tabel 5 Hardware Interface

|  |  |  |
| --- | --- | --- |
| **No** | **Hardware** | **Spesifikasi** |
| 1 | *Keyboard* | QWERTY |
| 2 | *Mouse* | *Blus Mouse* |
| 3 | *Monitor* | *LCD, LED, resolusi minimal 1280 x 800* |

### **Software Interface**

Kebutuhan antarmuka perangkat lunak (*software interface*) merupakan kebutuhan antarmuka terhadap perangkat lunak untuk membantu berjalannya sistem yang akan dibangun. Beberapa kebutuhan antarmuka perangkat lunak yang diperlukan adalah:

Tabel 6 Software Interface

|  |  |  |
| --- | --- | --- |
| **No** | **Software** | **Spesifikasi** |
| 1 | *Web Browser* | *Mozilla Firefox, Google Chrome, Microsoft Edge* |
| 2 | Sistem Operasi | *Windows, Linux, iOS* |

### **Communication Interface**

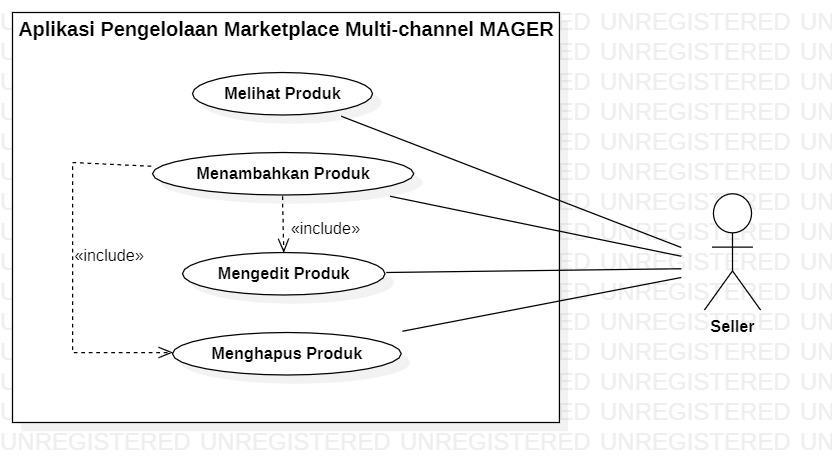
Kebutuhan antarmuka komunikasi (*Communication Interface*) yang digunakan untuk berinteraksi dengan Pengelolaan Marketplace Multi-channel MAGER adalah jaringan internet sebagai penghubung pengguna dengan sistem yang telah dibangun.

## **Functional Description**

Pada sub bab ini dijelaskan deskripsi fungsional yang terdapat pada Pengelolaan Marketplace Multi-channel MAGER berbasis website, yang digambarkan melalui *use case diagram* dan *use case scenario.*

### **Use Case Diagram**

*Use case diagram* merupakan jenis kebutuhan yang pendefinisian layanan yang harus disediakan sistem, mencakup bagaimana sistem harus bereaksi terhadap *input* tertentu dan bagaimana perilaku sistem pada situasi tertentu. Kebutuhan fungsional Pengelolaan Marketplace Multi-channel MAGER Berbasis Website dijelaskan pada *use case* diagram berikut.

****

Gambar 9 Use Case Diagram

### **Use Case Scenario**

Pada sub bab ini dijelaskan *use case scenario* dari setiap fungsi yang terdapat pada Pengelolaan Marketplace Multi-channel MAGER Berbasis Website.

Tabel 7 Use Case Scenario Melihat Produk

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case ID Number* | UC01 | | |
| *Use Case Name* | Melihat Produk | | |
| *Use Case Description* | *Use case* berikut menjelaskan tentang proses Melihat Produk yang dilakukan oleh *seller* terhadap Marketplace Multi-channel MAGER Berbasis Website. | | |
| *Primary Actor* | *Seller* | | |
| *Precondition* | *Seller* telah login kedalam website Marketplace Manager MAGER. | | |
| *Primary Flow of Events* | *User Action* | *System Response* | |
| 1. Memilih menu produk |  |  |
|  | 1. Menampilkan daftar produk yang tersedia di akun toko pada setiap *marketplace* |  |
| 1. Memilih detail produk |  |  |
|  | 1. Menampilkan detail dari produk yang dipilih |  |
| *Post Condition* | Berhasil melakukan Melihat produk | | |

Tabel 8 Use Case Scenario Menambahkan Produk

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case ID Number* | UC02 | | |
| *Use Case Name* | Menambahkan Produk | | |
| *Use Case Description* | *Use case* berikut menjelaskan tentang proses Menambahkan Produk yang dilakukan oleh *seller* terhadap Marketplace Multi-channel MAGER Berbasis Website. | | |
| *Primary Actor* | *Seller* | | |
| *Precondition* | *Seller* telah login kedalam website Marketplace Manager MAGER. | | |
| *Primary Flow of Events* | *User Action* | *System Response* | |
| 1. Memilih menu produk |  |  |
|  | 1. Menampilkan daftar produk yang tersedia di akun toko pada setiap *marketplace* |  |
| 1. Menekan *button* ‘Tambah Produk’ |  |  |
|  | 1. Menampilkan halaman pilih *marketplace* dan pilih toko |  |
| 1. Memilih *Marketplace* atau Toko |  |  |
|  | 1. Menampilkan komponen *form* penambahan produk. Misalkan jika memilih *marketplace* Lazada, maka akan ditampilkan *form upload photo,* nama produk, kategori, deskripsi produk, brand, bahan, model, *dummy*, harga, stok, seller SKU, jenis garansi, garansi, berat, panjang, lebar, tinggi. |  |
| 1. Mengisi form produk yang tersedia |  |  |
| 1. Menekan *button* ‘Simpan Perubahan’ |  |  |
| *Alternative Flow of Events* | 7a. Pengguna tidak mengisi form penambahan produk dengan benar. Misalkan tidak mengisi salah satu form yang wajib diisi |  | |
|  | 7b. Menampilkan *pop-up* modal ‘Ada Masalah’ sebagai indikasi Penambahan Produk gagal dilakukan | |
| *Post Condition* | Berhasil melakukan penambahan produk | | |

Tabel 9 Use Case Scenario Mengedit Produk

|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case ID Number* | UC03 | | |
| *Use Case Name* | Mengedit Produk | | |
| *Use Case Description* | *Use case* berikut menjelaskan tentang proses Mengedit Produk yang dilakukan oleh *seller* terhadap Marketplace Multi-channel MAGER Berbasis Website. | | |
| *Primary Actor* | *Seller* | | |
| *Precondition* | *Seller* telah login kedalam website Marketplace Manager MAGER. | | |
| *Primary Flow of Events* | *User Action* | *System Response* | |
| 1. Memilih menu produk |  |  |
|  | 1. Menampilkan daftar produk yang tersedia di akun toko pada setiap *marketplace* |  |
| 1. Memilih *button* kelola pada produk yang ingin diedit |  |  |
|  | 1. Menampilkan tampilan ‘Edit’ dan ‘Hapus’ |  |
| 1. Memilih *button* ‘Edit’ |  |  |
|  | 1. Menampilkan form edit produk. Misalkan jika memilih *marketplace* Lazada, maka akan ditampilkan *form edit upload photo,* nama produk, kategori, deskripsi produk, brand, bahan, model, *dummy*, harga, stok, seller SKU, jenis garansi, garansi, berat, panjang, lebar, tinggi. |  |
| 1. Mengedit form produk yang akan diubah sesuai dengan kebutuhan. |  |  |
| 1. Menekan *button* ‘Simpan Perubahan’ |  |  |
| *Alternative Flow of Events* | 7a. Pengguna tidak mengisi form edit produk dengan benar. Misalkan seller menghapus salah satu form produk yang wajib untuk diisi. |  | |
|  | 7b. Menampilkan pop-up modal ‘Ada Masalah’ sebagai indikasi Penambahan Produk gagal dilakukan | |
| *Post Condition* | Berhasil melakukan edit produk | | |

Tabel 10 Use Case Scenario Menghapus Produk

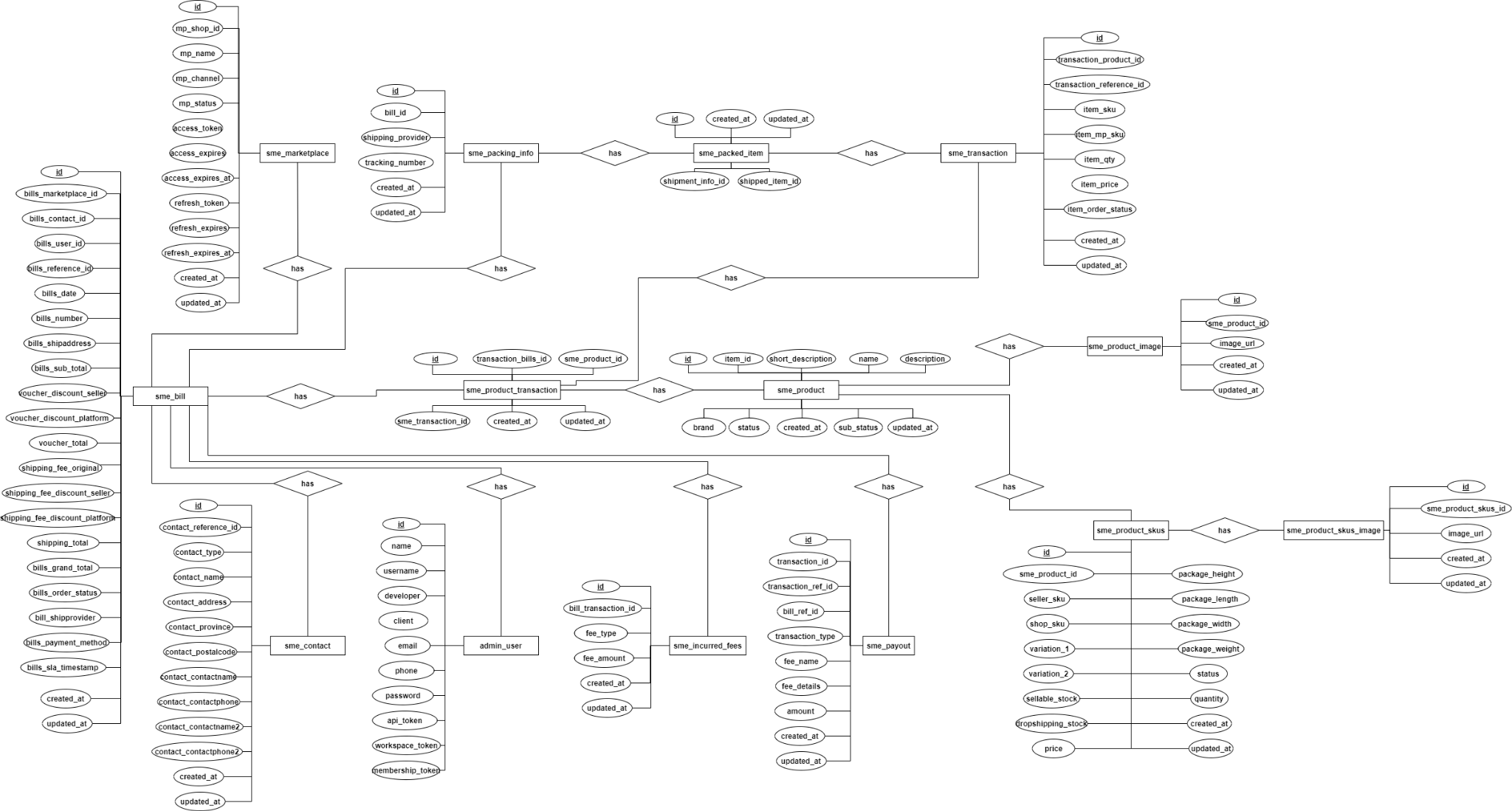
|  |  |  |  |
| --- | --- | --- | --- |
| *Use Case ID Number* | UC04 | | |
| *Use Case Name* | Menghapus Produk | | |
| *Use Case Description* | *Use case* berikut menjelaskan tentang proses Menghapus Produk yang dilakukan oleh *seller* terhadap Marketplace Multi-channel MAGER Berbasis Website. | | |
| *Primary Actor* | *Seller* | | |
| *Precondition* | *Seller* telah login kedalam website Marketplace Manager MAGER. | | |
| *Primary Flow of Events* | *User Action* | *System Response* | |
| 1. Memilih menu produk |  |  |
|  | 1. Menampilkan daftar produk yang tersedia di akun toko pada setiap *marketplace* |  |
| 1. Memilih *button* kelola pada produk yang ingin dihapus |  |  |
|  | 1. Menampilkan tampilan ‘Edit’ dan ‘Hapus’ |  |
| 1. Memilih button ‘Hapus’ |  |  |
|  | 1. Menampilkan validasi penghapusan produk. |  |
| 1. Menekan button ‘Ok’ |  |  |
| *Post Condition* | Berhasil melakukan penghapusan produk | | |

## **Data Requirement**

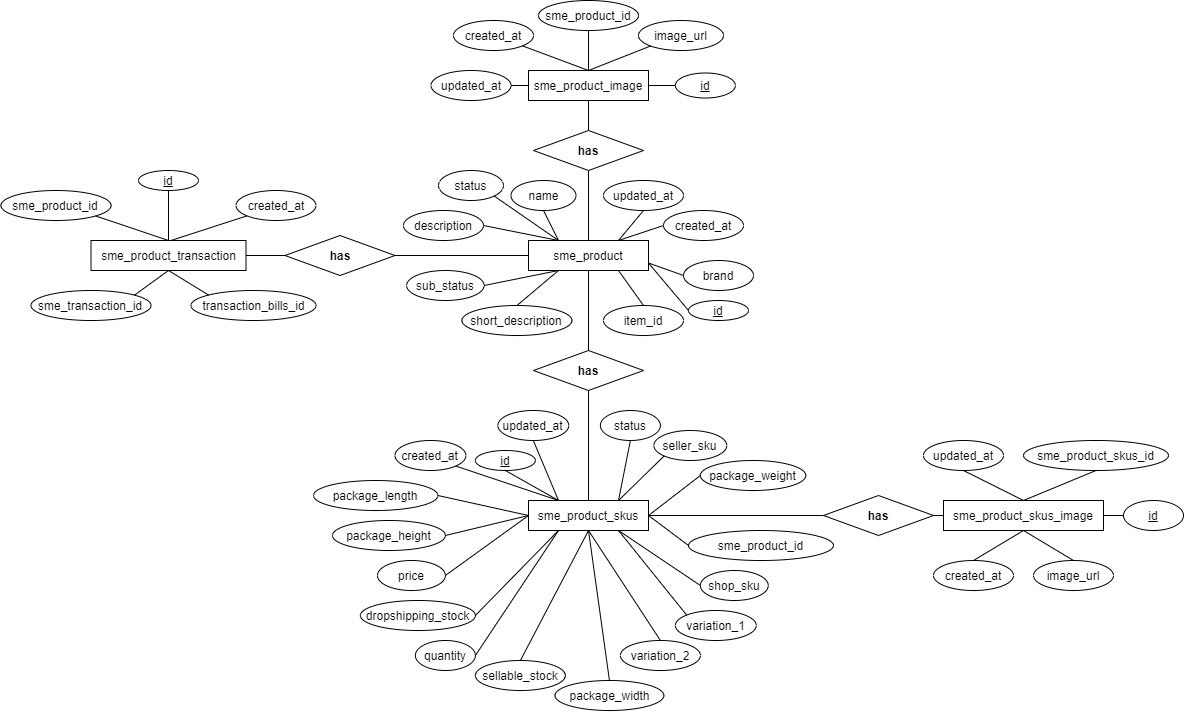
Pada sub bab ini dijelaskan mengenai data yang dibutuhkan dalam mengelola aplikasi.

### **Entity Relationship Diagram**

*Entity Relationship Diagram* (ERD) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki hubungan antar relasi. Hubungan tersebut dapat dilihat pada gambar di bawah ini.



Gambar 10 Entity Relational Diagram

****

Gambar 11 Detail Entity Relational Diagram - Product

Pada gambar ERD diatas, menjelaskan tentang relasi antar entitas dengan entitas lainnya, yaitu:

* + - 1. Entitas sme\_bill berelasi dengan entitas sme\_contact (kardinalitas one-to-one), dimana satu sme\_bill hanya memiliki satu sme\_contact.
      2. Entitas sme\_bill berelasi dengan entitas sme\_marketplace (kardinalitas one-to-one), dimana satu sme\_bill hanya memiliki satu sme\_marketplace.
      3. Entitas sme\_bill berelasi dengan entitas admin\_user (kardinalitas one-to-one), dimana satu sme\_bill hanya memiliki satu admin\_user.
      4. Entitas sme\_bill berelasi dengan entitas sme\_product\_transaction (kardinalitas many-to-one), dimana banyak sme\_bill hanya memiliki satu sme\_product\_transaction.
      5. Entitas sme\_bill berelasi dengan entitas sme\_packing\_info (kardinalitas many-to-one), dimana banyak sme\_bill hanya memiliki satu sme\_packing\_info.
      6. Entitas sme\_bill berelasi dengan entitas sme\_incurred\_fees (kardinalitas many-to-one), dimana banyak sme\_bill hanya memiliki satu sme\_incurred\_fees.
      7. Entitas sme\_bill berelasi dengan entitas sme\_payout (kardinalitas many-to-one), dimana banyak sme\_bill hanya memiliki satu sme\_payout.
      8. Entitas sme\_product\_transaction berelasi dengan entitas sme\_product (kardinalitas one-to-many), dimana satu sme\_product\_transaction memiliki banyak sme\_product.
      9. Entitas sme\_product\_transaction berelasi dengan entitas sme\_transaction (kardinalitas one-to-one), dimana satu sme\_product\_transaction hanya memiliki satu sme\_transaction.
      10. Entitas sme\_packing\_info berelasi dengan entitas sme\_packed\_item (kardinalitas one-to-one), dimana satu sme\_packing\_info hanya memiliki satu sme\_packed\_item.
      11. Entitas sme\_packed\_item berelasi dengan entitas sme\_transaction (kardinalitas one-to-one), dimana satu sme\_packed\_item hanya memiliki satu sme\_transaction.
      12. Entitas sme\_product berelasi dengan entitas sme\_product\_image (kardinalitas one-to-one), dimana satu sme\_product hanya memiliki satu sme\_product\_image.
      13. Entitas sme\_product berelasi dengan entitas sme\_product\_skus (kardinalitas one-to-one), dimana satu sme\_product hanya memiliki satu sme\_product\_skus.
      14. Entitas sme\_product\_skus berelasi dengan entitas sme\_product\_skus\_image (kardinalitas one-to-one), dimana satu sme\_product\_skus hanya memiliki satu sme\_product\_skus\_image.
      15. Entitas sme\_product\_skus berelasi dengan entitas sme\_product\_image (kardinalitas one-to-one), dimana satu sme\_product\_skus hanya memiliki satu sme\_product\_image.

## **Functional Requirement**

Pada sub bab ini dijelaskan *functional requirement* yang akan digunakan dalam Pengelolaan Marketplace Multi-channel MAGER Berbasis *Website*.

Tabel 11 Functional Requirement

| **Features-ID** | **Feature Name** | **Deskripsi** |
| --- | --- | --- |
| FR-01 | Melihat Produk | Fungsi ini digunakan *seller* untuk melakukan Melihat produk dengan akun toko pada masing-masing *marketplace*. Sehingga *output* yang diharapkan adalah menampilkan daftar produk, *marketplace*, variasi, harga, stok, penjualan, status dan kelola produk yang terintegrasi dengan akun toko masing - masing marketplace seperti Lazada maupun Tokopedia. |
| FR-02 | Menambahkan Produk | Fungsi ini digunakan *seller* untuk menambahkan produk di toko pada *marketplace*. Dalam penambahan produk terdapat beberapa komponen penting sehingga produk dapat dipublikasikan seperti Informasi Dasar, Spesifikasi atau Detail Produk, Variasi Produk, Harga serta Stok Produk, Pengiriman & Garansi Produk. |
| FR-03 | Mengedit Produk | Fungsi ini digunakan *seller* untuk mengedit informasi produk yang telah dipublikasikan pada akun toko di *marketplace*. Dalam fungsi ini, *seller* dapat melakukan aksi berupa penyesuaian informasi produk seperti jumlah stok maupun harga produk, status produk berupa aktif maupun tidak aktif, serta beragam perubahan lainnya sesuai dengan kondisi sesungguhnya berdasarkan penyesuaian yang dilakukan oleh *seller.* |
| FR-04 | Menghapus Produk | Fungsi ini digunakan *seller* untuk menghapus produk yang sebelumnya dibuat pada toko melalui *marketplace*. Fungsi menghapus produk meliputi semua status produk yang terdapat pada akun toko baik produk yang aktif maupun produk yang tidak aktif. |

## **Non Functional Requirement**

Pada sub bab ini dijelaskan *non-functional requirement* yang akan digunakan dalam Pengembangan Pengelolaan Marketplace Multi-channel MAGER Berbasis *Website*.

Tabel 12 Non Functional Requirement

| ***Fitur ID*** | ***Parameter*** | ***Spesifikasi*** |
| --- | --- | --- |
| NFR-01 | *Availability* | Aplikasi ini harus dapat beroperasi terus menerus selama 7 hari per minggu tanpa henti, karena aplikasi ini akan bersifat *web-based* dan akan diakses oleh pengguna yang membutuhkan dari berbagai tempat dan waktu yang berbeda-beda. |
| NFR-02 | *Reliability* | Sistem tetap beroperasi secara terus menerus dalam menjalankan fungsinya selama sebulan tanpa mengalami kerusakan. |
| NFR-03 | *Ergonomy* | Sistem memiliki tampilan yang sederhana sehingga membuat *user* nyaman menggunakannya. |
| NFR-04 | *Portability* | Sistem dapat digunakan di *environment* mana saja*.* |
| NFR-05 | *Memory* | Minimum *memory* adalah 128MB |
| NFR-06 | *Respon Time* | Sistem merespon perintah Maksimal 1 menit. |
| NFR-07 | *Safety* | Sistem harus mampu menjaga kerahasiaan setiap data pengguna yang tersimpan dalam *database.* |
| NFR-08 | *Security* | Aplikasi ini menjamin keamanan informasi dari setiap pengguna dengan menggunakan teknik *hashing* pada *password* setiap pengguna. |
| NFR-09 | *Bahasa Komunikasi* | Menggunakan Bahasa Indonesia. |
| NFR-10 | *Halaman Utama* | Setiap layar harus mengandung logo Aplikasi Pengelolaan Marketplace Multi-channel MAGER Berbasis Web. |

## **Design Constraint**

Pada sub bab ini dijelaskan desain batasan dalam pengembangan Pengelolaan Marketplace Multi-channel MAGER berbasis *website.*

### **Software Language**

Bahasa pemrograman yang digunakan adalah sebagai berikut:

1. PHP : *Hypertext Preprocessor* adalah bahasa pemrograman *server side scripting* yang bersifat *open source* yang digunakan sebagai bahasa pengembangan aplikasi.
2. HTML: *HyperText Markup Language* merupakan bahasa interpretasi yang digunakan dalam pendefinisian struktur dan informasi sistem
3. CSS: *Cascading Style Sheet* bahasa yang digunakan untuk menangani tampilan dan format sistem.
4. *Javascript* untuk membantu membuat halaman *web* interaktif
5. MySQL merupakan perangkat lunak sistem manajemen basis data yang berguna untuk menangani interaksi dengan basis data.

### **Development Tools**

*Tools* yang digunakan dalam pengembangan sistem ini adalah sebagai berikut:

1. HeidiSQL sebagai aplikasi untuk pengelolaan, pengembangan dan administrasi database.
2. Framework Laravel sebagai kerangka kerja *backend* pengembangan *website* yang digunakan.
3. VueJS sebagai kerangka kerja *frontend* pengembangan *website* yang digunakan.

# **Design Description**

Pada bab ini akan dijelaskan mengenai deskripsi desain pada Pengelolaan Marketplace Multi-channel MAGER Berbasis *Website.*

## **Data Description**

Sub bab ini menjelaskan mengenai data, diantaranya adalah nama domain, konsep model

data, dan *physical data model*.

### **Domain/Type Definition**

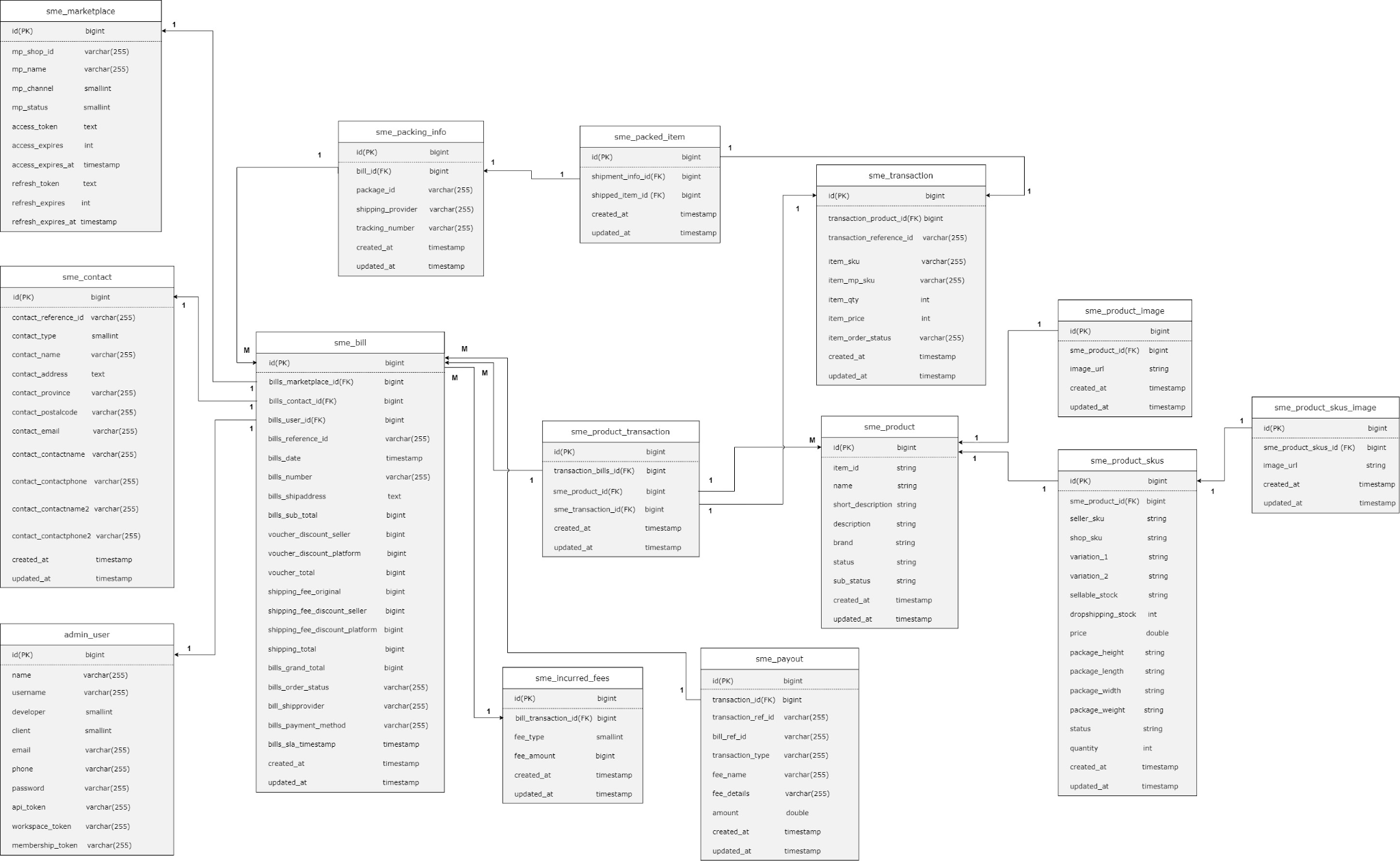
Domain dari setiap tabel yang terdapat pada website informasi Pengelolaan Marketplace Multi-channel MAGER Berbasis *Website* dapatdilihat pada tabel dibawah ini.

Tabel 13 Domain/Type Defenition

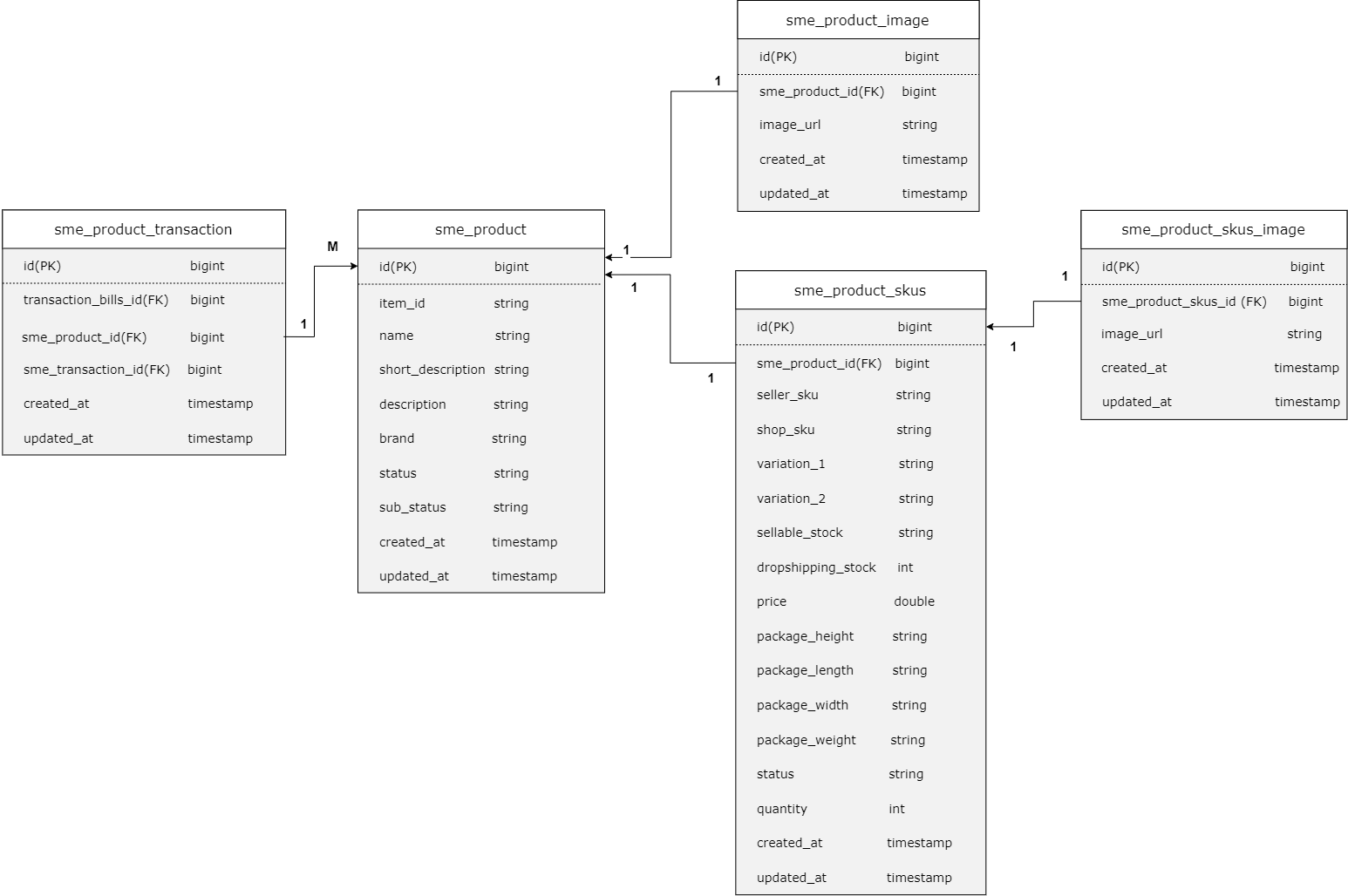
| **Table name** | **Domain name** | **Power Designer Type** |
| --- | --- | --- |
| sme\_marketplace | id | bigint |
| mp\_shop\_id | varchar(255) |
| mp\_name | varchar(255) |
| mp\_channel | smallint |
| mp\_status | smallint |
| access\_token | text |
| access\_expires | int |
| access\_expires\_at | timestamp |
| refresh\_token | text |
| refresh\_expires | int |
| refresh\_expires\_at | timestamp |
| sme\_contact | id | bigint |
| contact\_reference\_id | varchar(255) |
| contact\_type | smallint |
| contact\_name | varchar(255) |
| contact\_address | text |
| contact\_province | varchar(255) |
| contact\_postalcode | varchar(255) |
| contact\_email | varchar(255) |
| contact\_contactname | varchar(255) |
| contact\_contactphone | varchar(255) |
| contact\_contactname2 | varchar(255) |
| contact\_contactphone2 | varchar(255) |
| created\_at | timestamp |
| updated\_at | timestamp |
| admin\_user | id | bigint |
| name | varchar(255) |
| username | varchar(255) |
| developer | smallint |
| client | smallint |
| email | varchar(255) |
| phone | varchar(255) |
| password | varchar(255) |
| api\_token | varchar(255) |
| workspace\_token | varchar(255) |
| membership\_token | varchar(255) |
| sme\_bill | id | bigint |
| bills\_marketplace\_id | bigint |
| bills\_contact\_id | bigint |
| bills\_user\_id | bigint |
| bills\_reference\_id | varchar(255) |
| bills\_date | timestamp |
| bills\_number | varchar(255) |
| bills\_shipaddress | text |
| bills\_sub\_total | bigint |
| voucher\_discount\_seller | bigint |
| voucher\_discount\_platform | bigint |
| voucher\_total | bigint |
| shipping\_fee\_original | bigint |
| shipping\_fee\_discount\_seller | bigint |
| shipping\_fee\_discount\_platform | bigint |
| shipping\_total | bigint |
| bills\_grand\_total | bigint |
| bills\_order\_status | varchar(255) |
| bill\_shipprovider | varchar(255) |
| bills\_payment\_method | varchar(255) |
| bills\_sla\_timestamp | timestamp |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_packing\_info | id | bigint |
| bill\_id | bigint |
| package\_id | varchar(255) |
| shipping\_provider | varchar(255) |
| tracking\_number | varchar(255) |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_product\_transaction | id | bigint |
| transaction\_bills\_id | bigint |
| sme\_product\_id | bigint |
| sme\_transaction\_id | bigint |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_incurred\_fees | id | bigint |
| bill\_transaction\_id | bigint |
| fee\_type | smallint |
| fee\_amount | smallint |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_payout | id | bigint |
| transaction\_id | bigint |
| transaction\_ref\_id | varchar(255) |
| bill\_ref\_id | varchar(255) |
| transaction\_type | varchar(255) |
| fee\_name | varchar(255) |
| fee\_details | varchar(255) |
| amount | double |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_transaction | id | bigint |
| transaction\_product\_id | bigint |
| transaction\_reference\_id | varchar(255) |
| item\_sku | varchar(255) |
| item\_mp\_sku | varchar(255) |
| item\_qty | int |
| item\_price | int |
| item\_order\_status | varchar(255) |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_product | id | bigint |
| item\_id | string |
| name | string |
| short\_description | string |
| description | string |
| brand | string |
| status | string |
| sub\_status | string |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_packed\_item | id | bigint |
| shipment\_info\_id | bigint |
| shipped\_item\_id | bigint |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_product\_image | id | bigint |
| sme\_product\_id | bigint |
| image\_url | string |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_product\_skus | id | bigint |
| sme\_product\_id | bigint |
| seller\_sku | string |
| shop\_sku | string |
| variation\_1 | string |
| variation\_2 | string |
| sellable\_stock | string |
| dropshipping\_stock | int |
| price | double |
| package\_height | string |
| package\_length | string |
| package\_width | string |
| package\_weight | string |
| status | string |
| quantity | int |
| created\_at | timestamp |
| updated\_at | timestamp |
| sme\_product\_skus\_image | id | bigint |
| sme\_product\_skus\_id | bigint |
| image\_url | string |
| created\_at | timestamp |
| updated\_at | timestamp |

### **Conceptual Data Model**

Pada *conceptual data model* dijelaskan hubungan antar data dalam Pengelolaan Marketplace Multi-channel MAGER berbasis *website* yang akan dibangun dan hubungan yang terjadi antara tabel pada basis data. Hubungan tersebut dapat dilihat pada gambar di bawah ini.



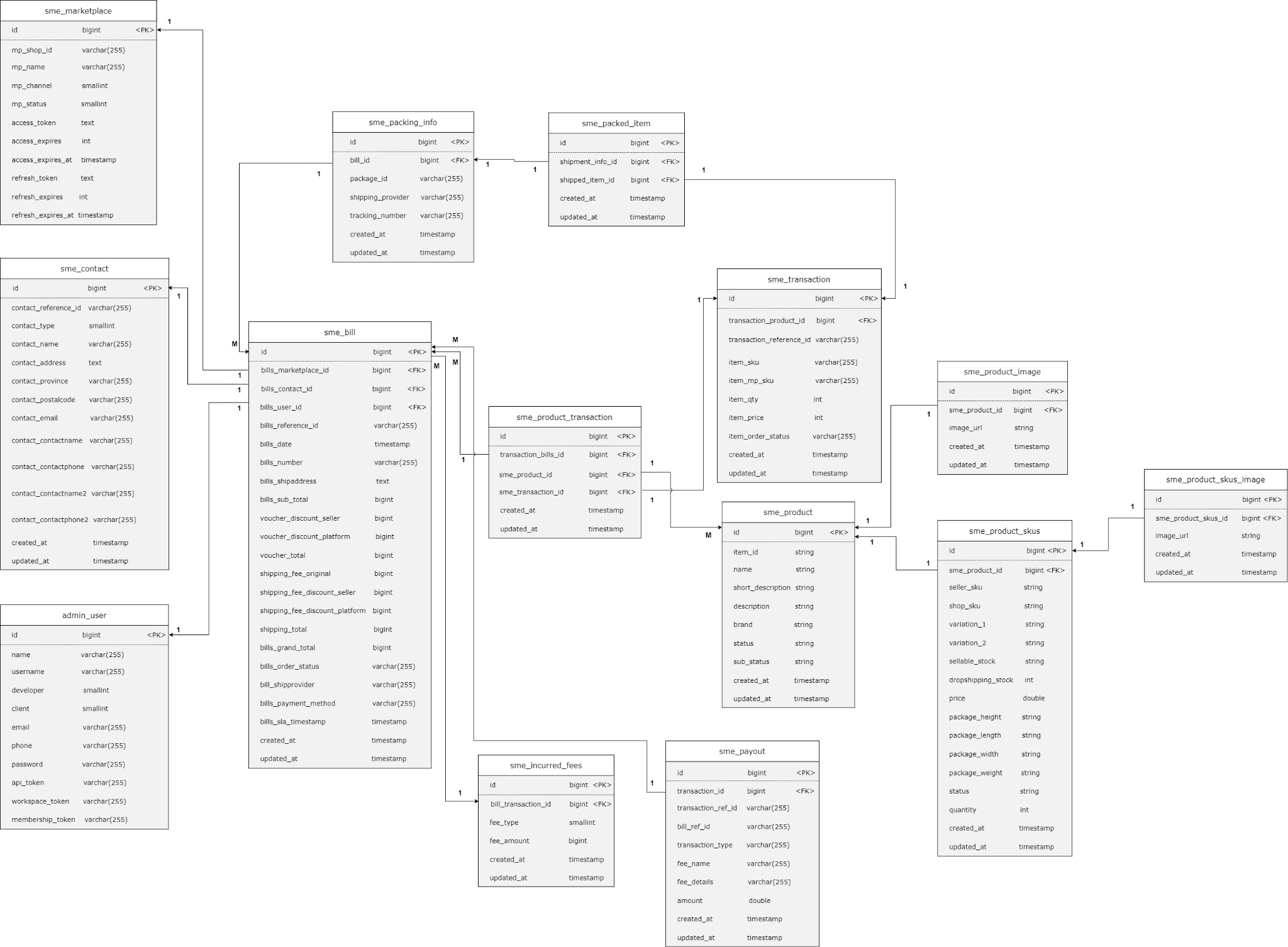
Gambar 12 Conceptual Data Model



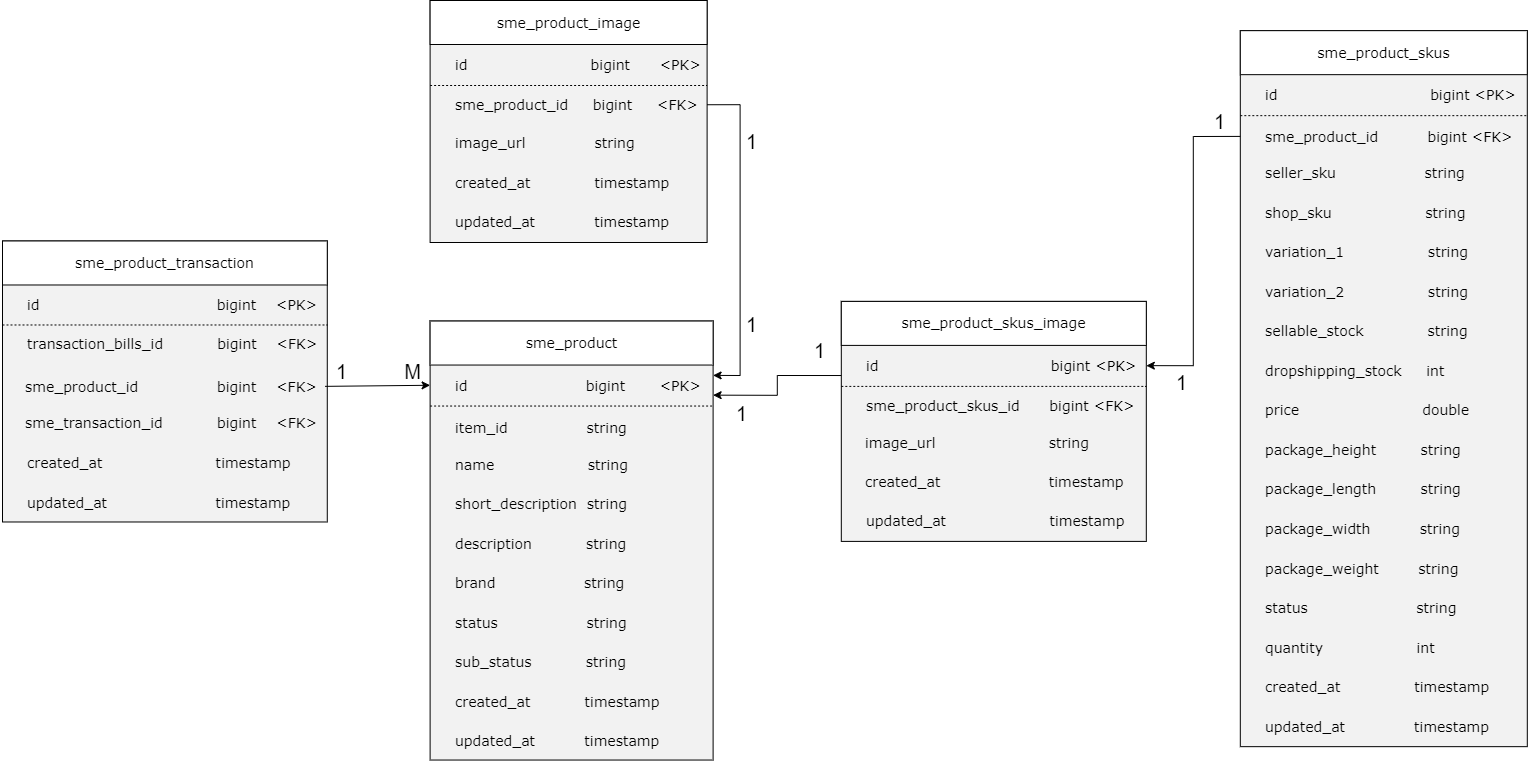
Gambar 13 Detail Conceptual Data Model - Product

### **Physical Data Model**

Pada *physical data model* digambarkan tabel master dan tabel lain yang saling berhubungan, tetapi berbeda dengan *conceptual* data model. Pada setiap tabel model data ini ditambahkan tipe setiap atribut yang ada. Hubungan tersebut dapat dilihat pada gambar di bawah ini.



Gambar 14 Physical Data Model



Gambar 15 Detail Physical Data Model - Product

### **Tables**

Deskripsi dari tabel-tabel yang ada pada basis data dapat dilihat pada tabel di bawah ini.

Tabel 14 Tables

| ***Nama Tabel*** | ***Primary key*** | ***Deskripsi isi*** |
| --- | --- | --- |
| sme\_marketplace | id | Tabel ini berfungsi untuk menyimpan informasi marketplace pemilik toko |
| sme\_contact | id | Tabel ini berfungsi untuk menyimpan informasi kontak customer |
| admin\_user | id | Tabel ini berfungsi untuk menyimpan user system |
| sme\_bill | id | Tabel ini berfungsi untuk menyimpan data pemesanan |
| sme\_packing\_info | id | Tabel ini berfungsi untuk menyimpan informasi packing paket pembeli |
| sme\_product\_transaction | id | Tabel ini berfungsi untuk menyimpan informasi transaksi produk |
| sme\_incurred\_fees | id | Tabel ini berfungsi untuk menyimpan informasi mengenai biaya pengeluaran. |
| sme\_payout | id | Tabel ini berfungsi untuk menyimpan informasi mengenai pembayaran. |
| sme\_transaction | id | Tabel ini berfungsi untuk menyimpan informasi transaksi |
| sme\_product | id | Tabel ini berfungsi untuk menyimpan informasi mengenai rincian produk. |
| sme\_product\_image | id | Tabel ini berfungsi menyimpan informasi mengenai gambar produk. |
| sme\_product\_skus | id | Tabel ini berfungsi untuk menyimpan informasi mengenai kode unik yang menjadi penanda produk. |
| sme\_product\_skus\_image | id | Tabel ini berfungsi untuk menyimpan informasi mengenai kode unik yang menjadi penanda gambar produk. |

# **Detail Design Description**

Pada sub bab ini dijelaskan *sequence diagram* dari Pengembangan Pengelolaan Marketplace Multi-channel MAGER Berbasis *Website*.

## **Tabel Structure**

Pada bagian ini akan dijelaskan spesifikasi dari seluruh tabel yang saling berelasi yang digunakan untuk Pengembangan Pengelolaan Marketplace Multi-channel MAGER Berbasis *Website*.

### **Tabel sme\_marketplace**

Identifikasi/Nama : sme\_marketplace

Deskripsi Isi : id, mp\_shop\_id,option\_id, mp\_name, mp\_channel, mp\_status,

access\_token,access\_expires,access\_expires\_at,refresh\_token,

refresh\_expires\_refresh\_expires\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_bill

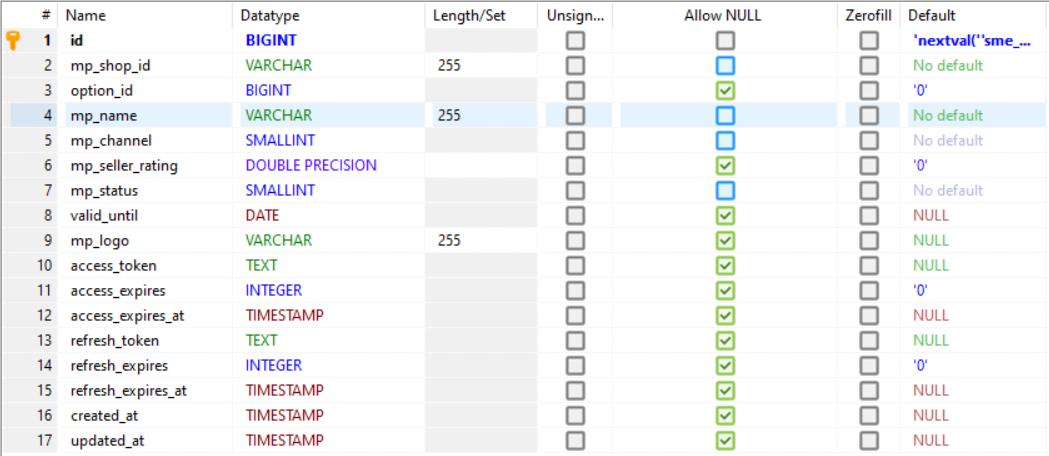
Tabel 15 Tabel sme\_marketplace

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_marketplace | BIGINT | N | - | *primary key* |
| mp\_shop\_id | berisi id dari setiap shop di marketplace | VARCHAR | N | - |  |
| option\_id | berisi id option | BIGINT | Y | - |  |
| mp\_name | berisi nama dari marketplace | VARCHAR | N | - |  |
| mp\_channel | berisi channel marketplace | SMALLINT | N | - |  |
| mp\_seller\_rating | berisi rating dari setiap seller marketplace | DOUBLE PRECISION | Y | - |  |
| mp\_status | berisi status marketplace | SMALLINT | N | - |  |
| valid\_until | berisi informasi valid | DATE | Y | - |  |
| mp\_logo | berisi logo marketplace | VARCHAR | Y | - |  |
| access\_token | berisi akses token | TEXT | Y | - |  |
| access\_expires | berisi informasi expires access | INTEGER | Y | - |  |
| access\_expires\_at | berisi informasi expires at access | TIMESTAMP | Y | - |  |
| refresh\_token | berisi informasi refresh token | TEXT | Y | - |  |
| refresh\_expires | berisi informasi refresh expires | INTEGER | Y | - |  |
| refresh\_expires\_at | berisi informasi refresh expires at | TIMESTAMP | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_marketplace" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_marketplace\_id\_seq''::regclass)',  "mp\_shop\_id" VARCHAR(255) NOT NULL,  "option\_id" BIGINT NULL DEFAULT '0',  "mp\_name" VARCHAR(255) NOT NULL,  "mp\_channel" SMALLINT NOT NULL,  "mp\_seller\_rating" DOUBLE PRECISION NULL DEFAULT '0',  "mp\_status" SMALLINT NOT NULL,  "valid\_until" DATE NULL DEFAULT NULL,  "mp\_logo" VARCHAR(255) NULL DEFAULT NULL,  "access\_token" TEXT NULL DEFAULT NULL,  "access\_expires" INTEGER NULL DEFAULT '0',  "access\_expires\_at" TIMESTAMP NULL DEFAULT NULL,  "refresh\_token" TEXT NULL DEFAULT NULL,  "refresh\_expires" INTEGER NULL DEFAULT '0',  "refresh\_expires\_at" TIMESTAMP NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 16 Struktur Tabel sme\_marketplace

### **Tabel sme\_contact**

Identifikasi/Nama : sme\_contact

Deskripsi Isi : id, contact\_reference\_id, contact\_type, contact\_name,

contact\_address\_contact\_province\_contact\_postalcode, contact\_email,

contact\_contacname, contact\_contactphone,contact\_contactname2,

contact\_contactphone2

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_bill

Tabel 16 Tabel sme\_contact

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_contact | BIGINT | N | - | *primary key* |
| contact\_type | berisi type contact dari user | SMALLINT | N | - |  |
| contact\_name | berisi contact name dari user | TEXT | N | - |  |
| contact\_address | berisi contact address dari user | TEXT | N | - |  |
| contact\_province | berisi province contact dari user | TEXT | Y | - |  |
| contact\_postalcode | berisi contact postalcode dari user | TEXT | Y | - |  |
| contact\_email | berisi informasi contact email dari user | TEXT | Y | - |  |
| contact\_contactname | berisi informasi contact name | TEXT | Y | - |  |
| contact\_contactphone | berisi informasi contact phone | TEXT | Y | - |  |
| contact\_contactname2 | berisi informasi contact name 2 | TEXT | Y | - |  |
| contact\_contactphone2 | berisi informasi contact phone 2 | TEXT | Y | - |  |
| contact\_reference\_id | berisi informasi reference id | VARCHAR | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_contact" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_contact\_id\_seq''::regclass)',  "contact\_type" SMALLINT NOT NULL,  "contact\_name" TEXT NOT NULL,  "contact\_address" TEXT NOT NULL,  "contact\_province" TEXT NULL DEFAULT NULL,  "contact\_postalcode" TEXT NULL DEFAULT NULL,  "contact\_email" TEXT NULL DEFAULT NULL,  "contact\_contactname" TEXT NULL DEFAULT NULL,  "contact\_contactphone" TEXT NULL DEFAULT NULL,  "contact\_contactname2" TEXT NULL DEFAULT NULL,  "contact\_contactphone2" TEXT NULL DEFAULT NULL,  "contact\_reference\_id" VARCHAR(255) NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 17 Struktur Tabel sme\_contact

### **Tabel admin\_user**

Identifikasi/Nama : admin\_user

Deskripsi Isi : id, name, username, developer, client, email, phone, password,

api\_token, workspace\_token, membership\_token

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_bill

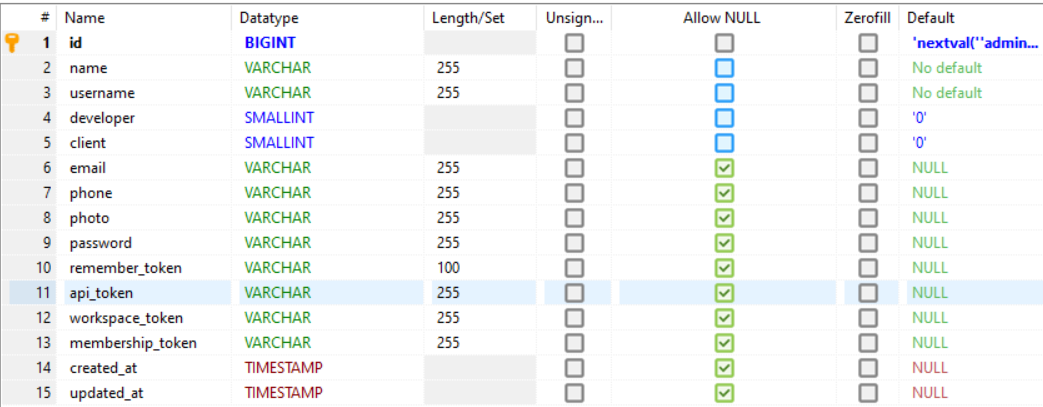
Tabel 17 Tabel admin\_user

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada admin\_user | BIGINT | N | - | *primary key* |
| *name* | berisi nama dari user | VARCHAR | N | - |  |
| *username* | berisi username dari user | VARCHAR | N | - |  |
| *developer* | informasi developer | SMALLINT | N | - |  |
| *client* | informasi client | SMALLINT | N | - |  |
| *email* | email dari user | VARCHAR | Y | - |  |
| *phone* | phone dari user | VARCHAR | Y | - |  |
| *photo* | photo dari user | VARCHAR | Y | - |  |
| *password* | password dari user | VARCHAR | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "admin\_user" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''admin\_user\_id\_seq''::regclass)',  "name" VARCHAR(255) NOT NULL,  "username" VARCHAR(255) NOT NULL,  "developer" SMALLINT NOT NULL DEFAULT '0',  "client" SMALLINT NOT NULL DEFAULT '0',  "email" VARCHAR(255) NULL DEFAULT NULL,  "phone" VARCHAR(255) NULL DEFAULT NULL,  "photo" VARCHAR(255) NULL DEFAULT NULL,  "password" VARCHAR(255) NULL DEFAULT NULL,  "remember\_token" VARCHAR(100) NULL DEFAULT NULL,  "api\_token" VARCHAR(255) NULL DEFAULT NULL,  "workspace\_token" VARCHAR(255) NULL DEFAULT NULL,  "membership\_token" VARCHAR(255) NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:

****

Gambar 18 Struktur Tabel admin\_user

### **Tabel sme\_bill**

Identifikasi/Nama : sme\_bill

Deskripsi Isi : id, bill\_marketplace\_id, bills\_contact\_id, bill\_user\_id, bills\_reference\_id,

bilss\_date, bills\_number, bills\_shipaddress, bill\_sub\_total, voucher\_discount\_seller, voucher\_discount\_platform\_voucher\_total, shipping\_fee\_original, shipping\_fee\_discount\_seller, shippung\_total, bills\_grand\_total, bills\_order\_status, bill\_shipprovider, bills\_payment\_method, bills\_sla\_timestamp, created\_at, updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_contact, sme\_marketplace, admin\_user,

sme\_product\_transaction, sme\_packing\_info, sme\_incurred\_fees,

sme\_payout

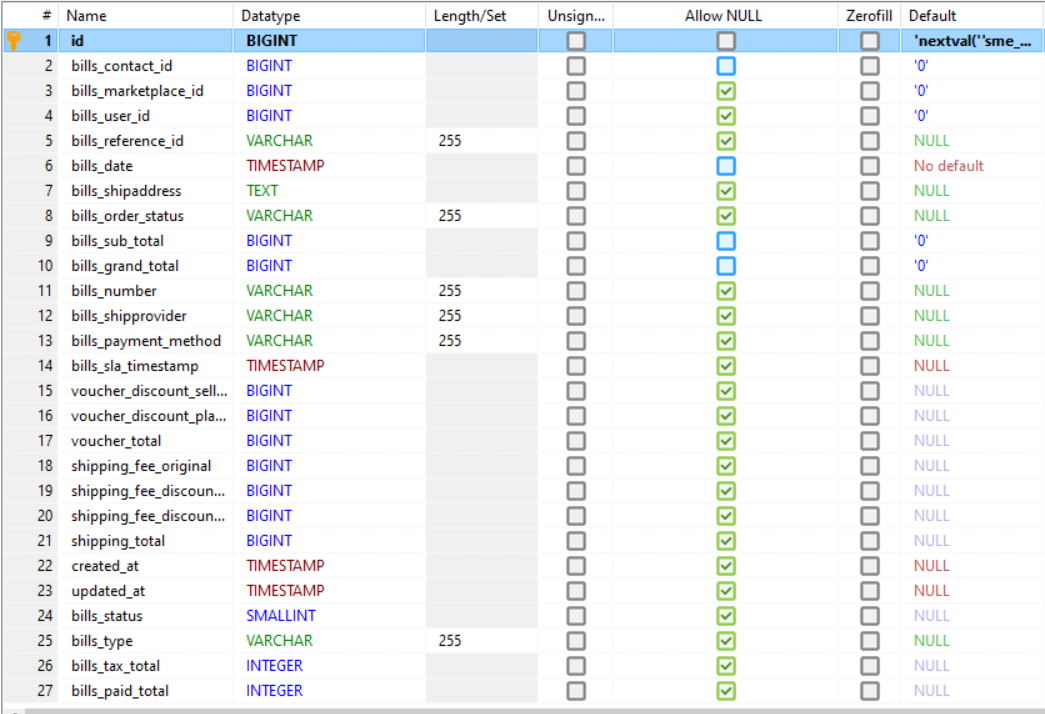
Tabel 18 Tabel sme\_bill

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_bill | BIGINT | N | - | *primary key* |
| bills\_contact\_id | berisi informasi id kontak pembayaran | BIGINT | N | - |  |
| bill\_marketplace\_id | berisi informasi id pembayaran marketplace | BIGINT | Y | - |  |
| bill\_user\_id | berisi id dari pembayaran user | BIGINT | Y | - |  |
| bills\_reference\_id | berisi id dari reference | VARCHAR | Y | - |  |
| bilss\_date | berisi informasi waktu pembayaran | TIMESTAMP | N | - |  |
| bills\_shipaddress | berisi informasi pembayaran | TEXT | Y | - |  |
| bills\_order\_status | berisi informasi status pembayaran pesanan | VARCHAR | Y | - |  |
| bills\_sub\_total | berisi informasi total pembayaran | BIGINT | N | - |  |
| bills\_grand\_total | berisi informasi grand total | BIGINT | N | - |  |
| bills\_number | berisi informasi nomor pembayaran | VARCHAR | Y | - |  |
| bills\_shipprovider | berisi informasi penyedia pembayaran | VARCHAR | Y | - |  |
| bills\_payment\_method | berisi informasi metode pembayaran | VARCHAR | Y | - |  |
| bills\_sla\_timestamp | berisi informasi bills sla timestamp | TIMESTAMP | Y | - |  |
| voucher\_discount\_platform | berisi informasi platform voucher discount | BIGINT | Y | - |  |
| voucher\_discount\_seller | berisi informasi voucher discount seller | BIGINT | Y | - |  |
| voucher\_total | berisi informasi total voucher | BIGINT | Y | - |  |
| shipping\_discount\_original | berisi informasi discount original | BIGINT | Y | - |  |
| shipping\_fee\_discount\_seller | berisi informasi biaya diskon pengiriman seller | BIGINT | Y | - |  |
| shipping\_total | berisi total pengiriman | BIGINT | Y | - |  |
| bills\_status | berisi informasi status pembayaran | SMALLINT | Y | - |  |
| bill\_type | berisi type pembayaran | VARCHAR | Y | - |  |
| bills\_tax\_total | berisi total pajak pembayaran | INTEGER | Y | - |  |
| bills\_paid\_total | berisi total pembayaran | INTEGER | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_bill" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_bill\_id\_seq''::regclass)',  "bills\_contact\_id" BIGINT NOT NULL DEFAULT '0',  "bills\_marketplace\_id" BIGINT NULL DEFAULT '0',  "bills\_user\_id" BIGINT NULL DEFAULT '0',  "bills\_reference\_id" VARCHAR(255) NULL DEFAULT NULL,  "bills\_date" TIMESTAMP NOT NULL,  "bills\_shipaddress" TEXT NULL DEFAULT NULL,  "bills\_order\_status" VARCHAR(255) NULL DEFAULT NULL,  "bills\_sub\_total" BIGINT NOT NULL DEFAULT '0',  "bills\_grand\_total" BIGINT NOT NULL DEFAULT '0',  "bills\_number" VARCHAR(255) NULL DEFAULT NULL,  "bills\_shipprovider" VARCHAR(255) NULL DEFAULT NULL,  "bills\_payment\_method" VARCHAR(255) NULL DEFAULT NULL,  "bills\_sla\_timestamp" TIMESTAMP NULL DEFAULT NULL,  "voucher\_discount\_seller" BIGINT NULL DEFAULT NULL,  "voucher\_discount\_platform" BIGINT NULL DEFAULT NULL,  "voucher\_total" BIGINT NULL DEFAULT NULL,  "shipping\_fee\_original" BIGINT NULL DEFAULT NULL,  "shipping\_fee\_discount\_seller" BIGINT NULL DEFAULT NULL,  "shipping\_fee\_discount\_platform" BIGINT NULL DEFAULT NULL,  "shipping\_total" BIGINT NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  "bills\_status" SMALLINT NULL DEFAULT NULL,  "bills\_type" VARCHAR(255) NULL DEFAULT NULL,  "bills\_tax\_total" INTEGER NULL DEFAULT NULL,  "bills\_paid\_total" INTEGER NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 19 Struktur Tabel sme\_bill

### **Tabel sme\_packing\_info**

Identifikasi/Nama : sme\_packing\_info

Deskripsi Isi : id, bill\_id, package\_id, shipping\_prodivder, tracking\_number, created\_at,

updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_bill, sme\_packed\_item

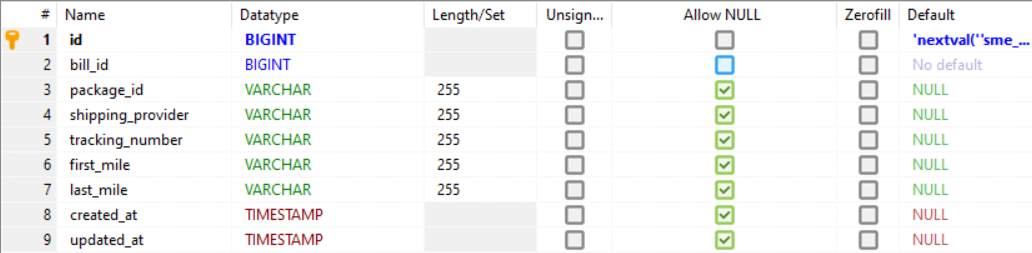
Tabel 19 Tabel sme\_packing\_info

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_packing\_info | BIGINT | N | - | *primary key* |
| bill\_id | berisi informasi id pembayaran | BIGINT | N | - |  |
| package\_id | berisi informasi mengenai id pengemasan | VARCHAR | Y | - |  |
| shipping\_provider | berisi informasi penyedia | VARCHAR | Y | - |  |
| tracking\_number | berisi informasi nomor tracking | VARCHAR | Y | - |  |
| first\_mile | berisi informasi first mile | VARCHAR | Y | - |  |
| last\_mile | berisi informasi first mile | VARCHAR | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_packing\_info" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_packing\_info\_id\_seq''::regclass)',  "bill\_id" BIGINT NOT NULL,  "package\_id" VARCHAR(255) NULL DEFAULT NULL,  "shipping\_provider" VARCHAR(255) NULL DEFAULT NULL,  "tracking\_number" VARCHAR(255) NULL DEFAULT NULL,  "first\_mile" VARCHAR(255) NULL DEFAULT NULL,  "last\_mile" VARCHAR(255) NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 20 Struktur Tabel sme\_packing\_info

### **Tabel sme\_product\_transaction**

Identifikasi/Nama : sme\_product\_transaction

Deskripsi Isi : id, transaction\_bills\_id, sme\_product\_id, sme\_transaction\_id, created\_at,

updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_bill, sme\_product, sme\_transaction,

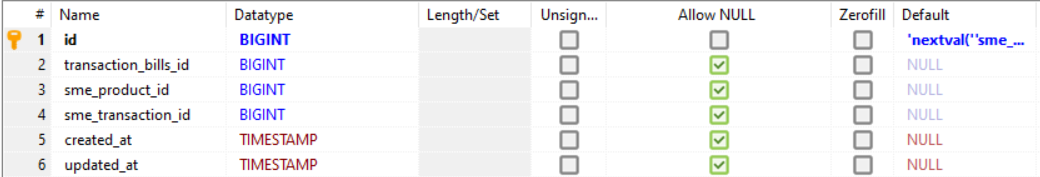
Tabel 20 Tabel sme\_product\_transaction

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_product\_transaction | BIGINT | N | - | *primary key* |
| transaction\_bills\_id | berisi informasi id transaksi pembayaran | BIGINT | Y |  |  |
| sme\_product\_id | berisi informasi id product sme | BIGINT | Y |  |  |
| sme\_transaction\_id | berisi informasi id transaction sme | BIGINT | Y |  |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_product\_transaction" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_product\_transaction\_id\_seq''::regclass)',  "transaction\_bills\_id" BIGINT NULL DEFAULT NULL,  "sme\_product\_id" BIGINT NULL DEFAULT NULL,  "sme\_transaction\_id" BIGINT NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 21 Struktur Tabel sme\_product\_transaction

### **Tabel sme\_transaction**

Identifikasi/Nama : sme\_transaction

Deskripsi Isi : id, transaction\_product\_id, transaction\_reference\_id, item\_sku,

item\_mp\_sku, item\_qty, item\_price, item\_order\_status, created\_at,

updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_product\_transaction, sme\_transaction

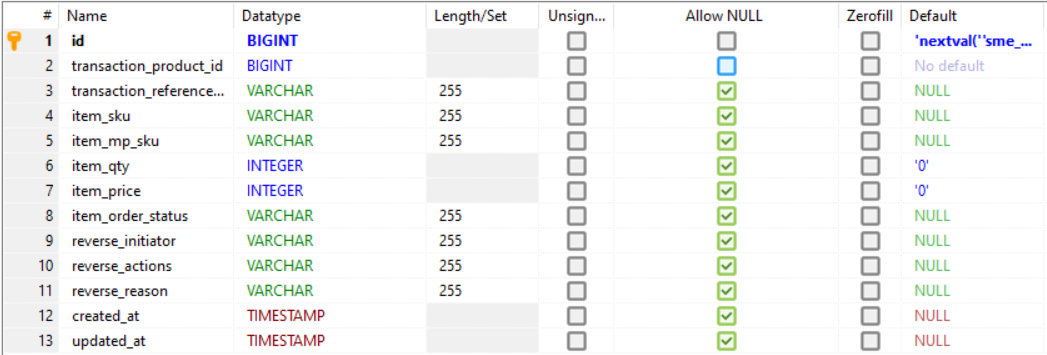
Tabel 21 Tabel sme\_transaction

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_transaction | BIGINT | N | - | *primary key* |
| transaction\_product\_id | berisi informasi transaksi produk | BIGINT | N | - |  |
| transaction\_reference\_id | berisi informasi id referensi transaksi | VARCHAR | Y | - |  |
| item\_sku | berisi informasi item sku | VARCHAR | Y | - |  |
| item\_mp\_sku | berisi informasi mp sku | VARCHAR | Y | - |  |
| item\_qty | berisi informasi jumlah item | INTEGER | Y | - |  |
| item\_price | berisi informasi harga item | INTEGER | Y | - |  |
| item\_order\_status | berisi informasi status order item | VARCHAR | Y | - |  |
| reverse\_initiator | berisi informasi reverse initiator | VARCHAR | Y | - |  |
| reverse\_actions | berisi informasi reverse actions | VARCHAR | Y | - |  |
| reverse\_reason | berisi informasi reverse reason | VARCHAR | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_transaction" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_transaction\_id\_seq''::regclass)',  "transaction\_product\_id" BIGINT NOT NULL,  "transaction\_reference\_id" VARCHAR(255) NULL DEFAULT NULL,  "item\_sku" VARCHAR(255) NULL DEFAULT NULL,  "item\_mp\_sku" VARCHAR(255) NULL DEFAULT NULL,  "item\_qty" INTEGER NULL DEFAULT '0',  "item\_price" INTEGER NULL DEFAULT '0',  "item\_order\_status" VARCHAR(255) NULL DEFAULT NULL,  "reverse\_initiator" VARCHAR(255) NULL DEFAULT NULL,  "reverse\_actions" VARCHAR(255) NULL DEFAULT NULL,  "reverse\_reason" VARCHAR(255) NULL DEFAULT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 22 Struktur Tabel sme\_transaction

### **Tabel sme\_product**

Identifikasi/Nama : sme\_product

Deskripsi Isi : id, item\_id, name, short\_description, description, brand, status,

sub\_status, created\_at, updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_product\_transaction, sme\_product\_image,

sme\_product\_skus\_image, sme\_product\_skus

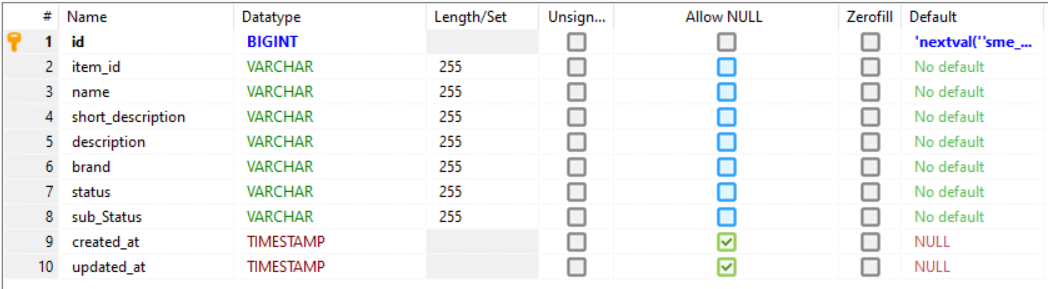
Tabel 22 Tabel sme\_product

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_product | BIGINT | N | - | *primary key* |
| item\_id | berisi informasi id item | VARCHAR | N | - |  |
| name | nama dari produk | VARCHAR | N | - |  |
| short\_description | penjelasan singkat mengenai produk | VARCHAR | N | - |  |
| description | penjelasan mengenai produk | VARCHAR | N | - |  |
| brand | berisi informasi brand dari produk | VARCHAR | N | - |  |
| status | berisi informasi status produk | VARCHAR | N | - |  |
| sub\_status | berisi informasi sub\_status | VARCHAR | N | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_product" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_product\_id\_seq''::regclass)',  "item\_id" VARCHAR(255) NOT NULL,  "name" VARCHAR(255) NOT NULL,  "short\_description" VARCHAR(255) NOT NULL,  "description" VARCHAR(255) NOT NULL,  "brand" VARCHAR(255) NOT NULL,  "status" VARCHAR(255) NOT NULL,  "sub\_Status" VARCHAR(255) NOT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 23 Struktur Tabel sme\_product

### **Tabel sme\_incurred\_fees**

Identifikasi/Nama : sme\_incurred\_fees

Deskripsi Isi : id, bill\_transaction\_id, fee\_type, fee\_amount, created\_at, updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_bill

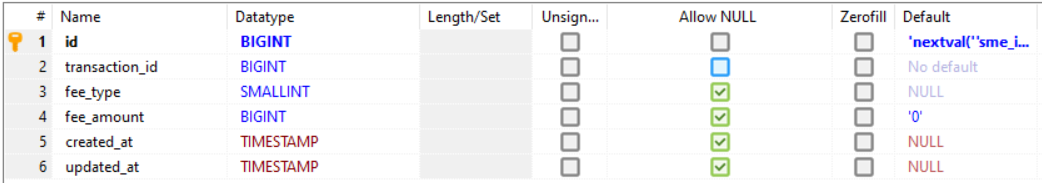
Tabel 23 sme\_incurred\_fees

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_incurred\_fees | BIGINT | N | - | *primary key* |
| transaction\_id | berisi informasi id transaksi | BIGINT | N | - |  |
| fee\_type | berisi informasi tipe pembayaran | SMALLINT | Y | - |  |
| fee\_amount | berisi jumlah pembayaran | BIGINT | Y | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_incurred\_fees" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_incurred\_fees\_id\_seq''::regclass)',  "transaction\_id" BIGINT NOT NULL,  "fee\_type" SMALLINT NULL DEFAULT NULL,  "fee\_amount" BIGINT NULL DEFAULT '0',  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 24 Struktur Tabel sme\_incurred\_fees

### **Tabel sme\_product\_skus**

Identifikasi/Nama : sme\_product\_skus

Deskripsi Isi : id, sme\_product\_id, seller\_sku, shop\_sku, variation\_1, variation\_2,

sellable\_stock, dropshipping\_stock, price, package\_height,

package\_length, package\_widht, package\_weight, status, quantity,

created\_at, updated\_at,

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_product

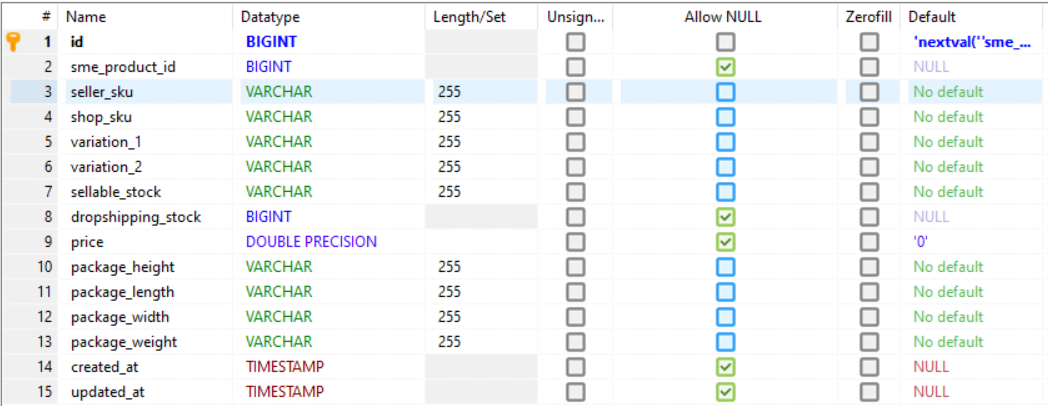
Tabel 24 sme\_product\_skus

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_product\_skus | BIGINT | N | - | *primary key* |
| sme\_product\_id | berisi informasi id sme\_product | BIGINT | Y |  |  |
| seller\_sku | berisi informasi sku seller | VARCHAR | N |  |  |
| shop\_sku | berisi informasi shop sku | VARCHAR | N |  |  |
| variation\_1 | berisi informasi variasi pertama | VARCHAR | N |  |  |
| variation\_2 | berisi informasi variasi kedua | VARCHAR | N |  |  |
| sellable\_stock | berisi informasi stok yang bisa dijual | VARCHAR | N |  |  |
| dropshipping\_stock | berisi informasi dropshipping stock | BIGINT | Y |  |  |
| price | berisi informasi harga | DOUBLE PRECISION | Y |  |  |
| package\_heigth | berisi informasi tinggi kemasan | VARCHAR | N |  |  |
| package\_length | berisi informasi panjang kemasan | VARCHAR | N |  |  |
| package\_width | berisi informasi lebar kemasan | VARCHAR | N |  |  |
| package\_weight | berisi informasi berat kemasan | VARCHAR | N |  |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_product\_sku" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_product\_sku\_id\_seq''::regclass)',  "sme\_product\_id" BIGINT NULL DEFAULT NULL,  "seller\_sku" VARCHAR(255) NOT NULL,  "shop\_sku" VARCHAR(255) NOT NULL,  "variation\_1" VARCHAR(255) NOT NULL,  "variation\_2" VARCHAR(255) NOT NULL,  "sellable\_stock" VARCHAR(255) NOT NULL,  "dropshipping\_stock" BIGINT NULL DEFAULT NULL,  "price" DOUBLE PRECISION NULL DEFAULT '0',  "package\_height" VARCHAR(255) NOT NULL,  "package\_length" VARCHAR(255) NOT NULL,  "package\_width" VARCHAR(255) NOT NULL,  "package\_weight" VARCHAR(255) NOT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 25 Struktur Tabel sme\_product\_skus

### **Tabel sme\_product\_image**

Identifikasi/Nama : sme\_product\_image

Deskripsi Isi : id, sme\_product\_id, image\_url, created\_at, updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_product

Tabel 25 sme\_product\_image

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_product\_image | BIGINT | N | - | *primary key* |
| sme\_product | berisi informasi sme\_product | BIGINT | Y |  |  |
| image\_url | berisi informasi url image | VARCHAR | N |  |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_product\_image" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_product\_image\_id\_seq''::regclass)',  "sme\_product\_id" BIGINT NULL DEFAULT NULL,  "image\_url" VARCHAR(255) NOT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 26 Struktur Tabel sme\_product\_image

### **Tabel sme\_product\_skus\_image**

Identifikasi/Nama : sme\_product\_skus\_image

Deskripsi Isi : id, sme\_product\_skus\_id, image\_url, created\_at, updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_product

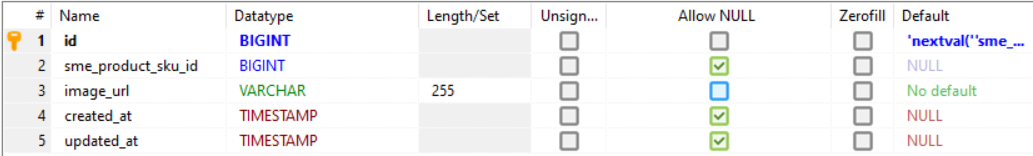
Tabel 26 Tabel sme\_product\_skus\_image

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_product\_skus\_image | BIGINT | N | - | *primary key* |
| sme\_product\_sku\_id | berisi informasi id sku product | BIGINT | Y | - |  |
| image\_url | berisi informasi url image | VARCHAR | N | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_product\_sku\_image" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_product\_sku\_image\_id\_seq''::regclass)',  "sme\_product\_sku\_id" BIGINT NULL DEFAULT NULL,  "image\_url" VARCHAR(255) NOT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

Maka struktur tabel yang dihasilkan:



Gambar 27 Struktur Tabel sme\_product\_skus\_image

### **Tabel sme\_packed\_item**

Identifikasi/Nama : sme\_packed\_item

Deskripsi Isi : id, shipment\_info\_id, shipped\_item\_id, created\_at, updated\_at

Primary Key : id

Detail Description : memiliki relasi terhadap sme\_transaction dan sme\_packing\_info

Tabel 27 Tabel sme\_packed\_item

| **Id Field** | **Deskripsi** | **Tipe & length** | **Boleh**  **NULL** | **Default** | **Keterangan** |
| --- | --- | --- | --- | --- | --- |
| id | id merupakan primary key pada sme\_packed\_item | BIGINT | N | - | *primary key* |
| shipment\_info\_id | berisi informasi id shipment info | BIGINT | N | - |  |
| shipped\_item\_id | berisi informasi id item shipped | BIGINT | N | - |  |

Jika menggunakan SQL Scripts

|  |
| --- |
| CREATE TABLE "sme\_packed\_item" (  "id" BIGINT NOT NULL DEFAULT 'nextval(''sme\_packed\_item\_id\_seq''::regclass)',  "shipment\_info\_id" BIGINT NOT NULL,  "shipped\_item\_id" BIGINT NOT NULL,  "created\_at" TIMESTAMP NULL DEFAULT NULL,  "updated\_at" TIMESTAMP NULL DEFAULT NULL,  PRIMARY KEY ("id")  )  ; |

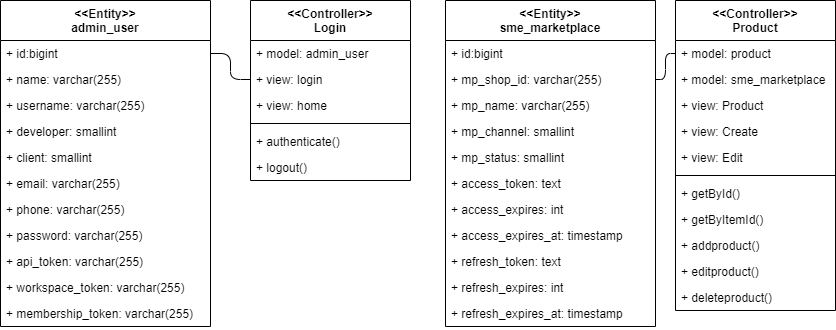
Maka struktur tabel yang dihasilkan:



Gambar 28 Struktur Tabel sme\_packed\_item

## **Class Diagram**

Pada bagian ini dijelaskan *class diagram* dari pengembangan Aplikasi Pengelolaan Marketplace Multi-channel MAGER. Dalam sub bab ini digambarkan *class diagram* yang menunjukkan hubungan antara setiap objek dalam sistem termasuk didalamnya.



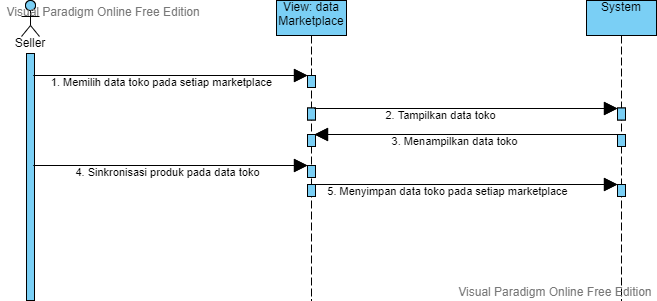
Gambar 29 Class Diagram

## **Sequence Diagram**

Pada sub bab ini akan digambarkan *sequence diagram* yang dimiliki oleh pengembangan Aplikasi Pengelolaan Marketplace Multi-channel MAGER.

### **Sequence Diagram Melihat Produk**

Pada sub bab ini digambarkan *sequence diagram* dari Fitur Melihat Produk Aplikasi Pengelolaan Marketplace Multi-channel MAGER. Berikut adalah sequence diagramnya.



Gambar 30 Sequence Diagram Melihat Produk

#### **Query Specification (jika ada)**

Tidak ada.

#### **Error Message**

Tidak ada.

#### **Algorithm**

Id Process : FS - 01

Objek terkait : Melihat Produk

Event : Click

**Initial State (IS)**:

Tabel sme\_marketplace sudah dibuat strukturnya

**Final State (FS)**:

Menampilkan produk yang ada pada sistem

**Spesifikasi Proses/algoritma**:

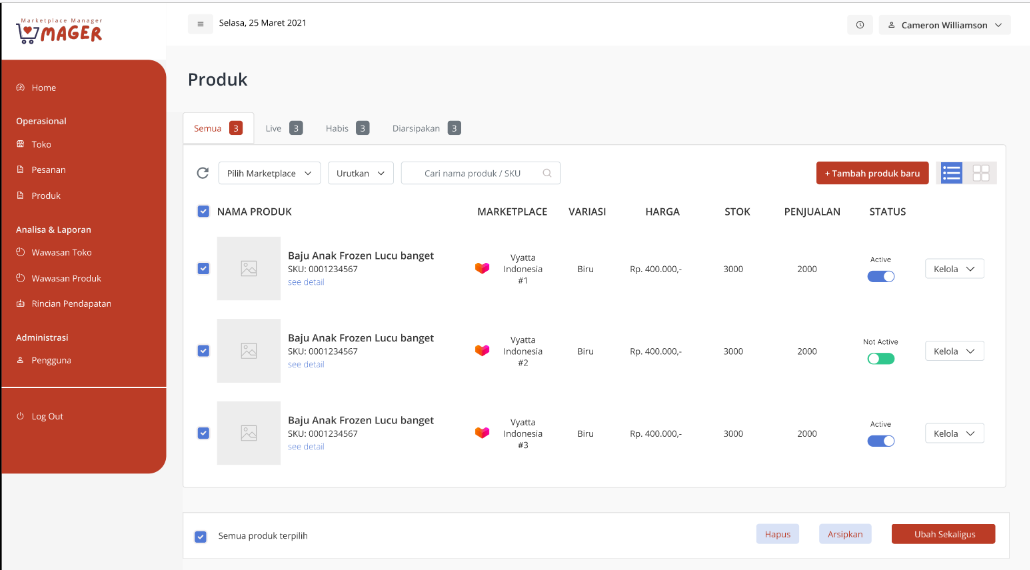
1. Seller memilih menu produk
2. Sistem menampilkan semua daftar produk

#### **Spesifikasi Report**

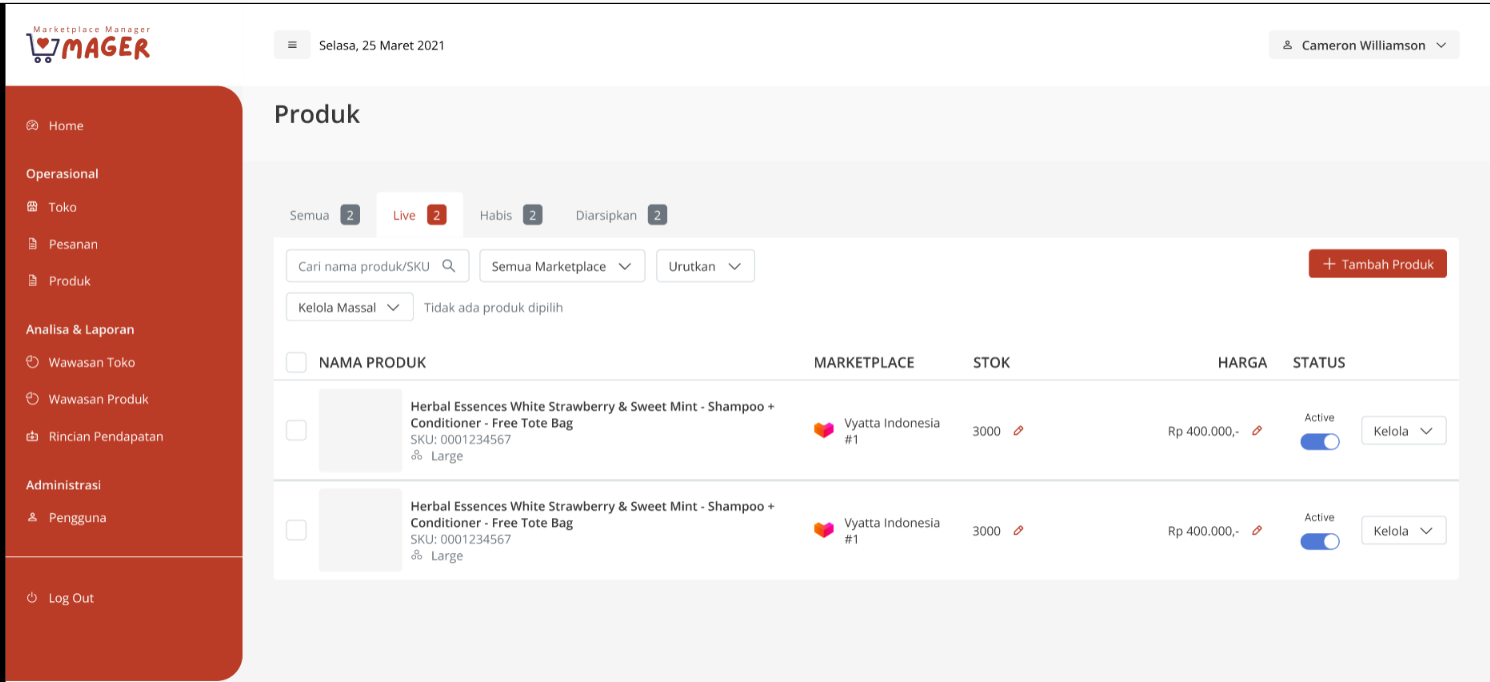
Tidak ada.

#### **Design Screen Layout Melihat Produk**

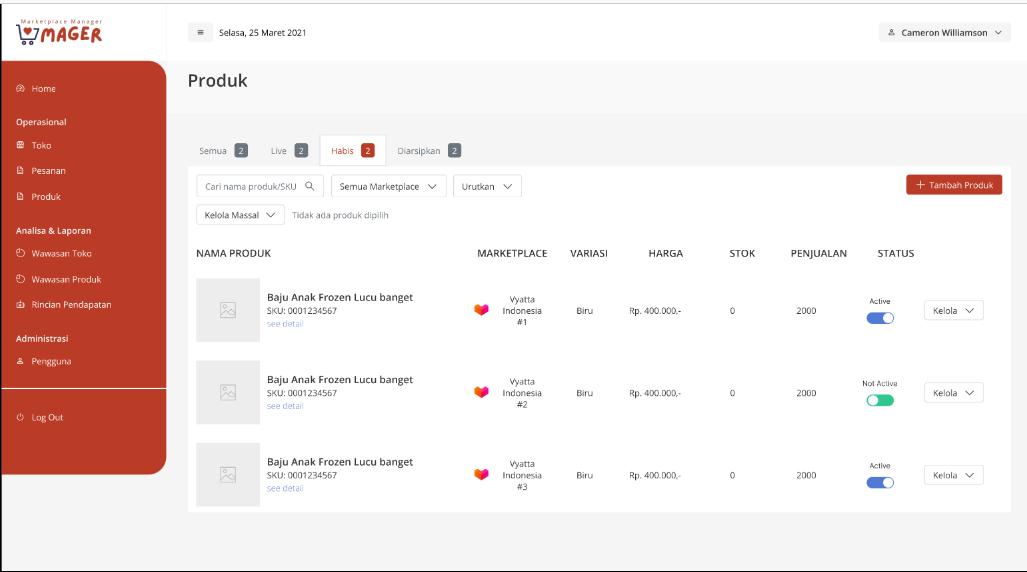
Berikut merupakan gambaran desain dari screen layout melihat produk.



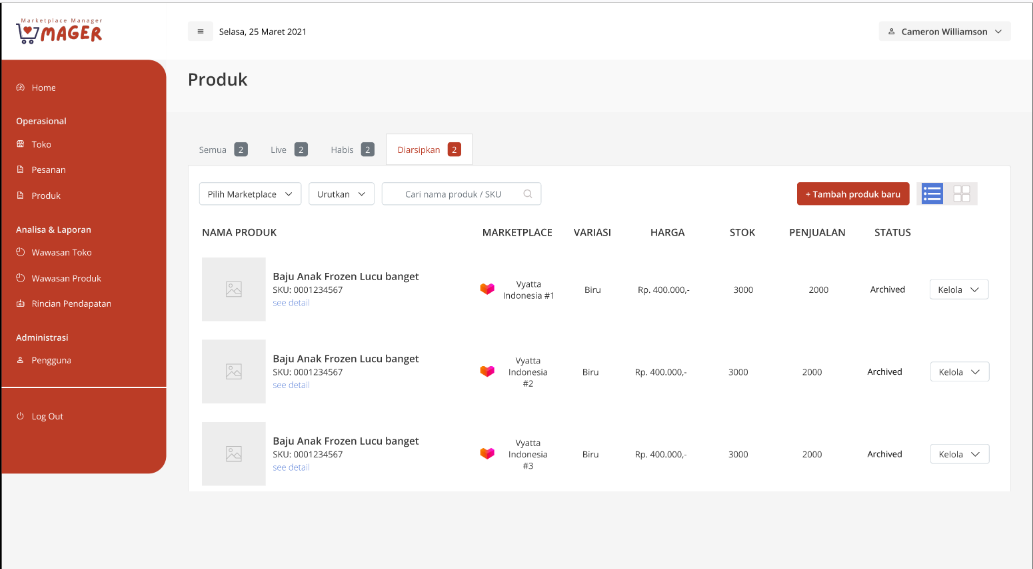
Gambar 31 Design Screen Layout Melihat Produk Semua



Gambar 32 Design Screen Layout Melihat Produk Live



Gambar 33 Design Screen Layout Melihat Produk Habis



Gambar 34 Design Screen Layout Melihat Produk Diarsipkan

#### **Detail Screen Layout**

Berikut merupakan gambaran dari screen layout halaman Melihat produk.



Gambar 35 Detail Screen Layout Melihat Produk

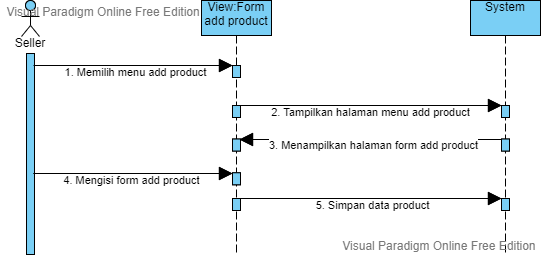
#### **Objects Specification**

Tabel 28 Object Specification Melihat Produk

|  |  |  |
| --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Keterangan** |
| Tambah Produk | *Button* | Jika diklik maka akan mengarah ke halaman *form add product* |

### **Sequence Diagram Menambahkan Produk**

Pada sub bab ini digambarkan *sequence diagram* dari Fitur Menambahkan Produk Aplikasi Pengelolaan Marketplace Multi-channel MAGER. Berikut adalah sequence diagramnya.



Gambar 36 Sequence Diagram Menambahkan Produk

#### **Query Specification (jika ada)**

Tidak ada.

#### **Error Message**

Pesan *error* pada fungsi menambah produk dapat dilihat pada tabel dibawah ini.

Tabel 29 *Error Message*

|  |  |  |
| --- | --- | --- |
| **No.** | **Keterangan** | **Pesan *error* yang disampaikan** |
| 1 | Salah satu field tidak diisi sementara pengguna memilih button **“Simpan Perubahan”** | Client error: `POST https://fs.tokopedia.net/v3/products/fs/15214/create?shop\_id=11637362` resulted in a `400 Bad Request` |

#### **Algorithm**

Id Process : FS-02

Objek terkait : Menambahkan Produk

Event : click

**Initial State (IS)**:

Tabel sme\_marketplace sudah dibuat strukturnya

**Final State (FS)**:

Menghasilkan penambahan produk pada marketplace

**Spesifikasi Proses/algoritma**:

1. Memilih menu produk
2. Menekan tombol penambahan produk
3. Mengisi form pada laman
4. IF sudah sesuai, klik Simpan Perubahan

ELSE

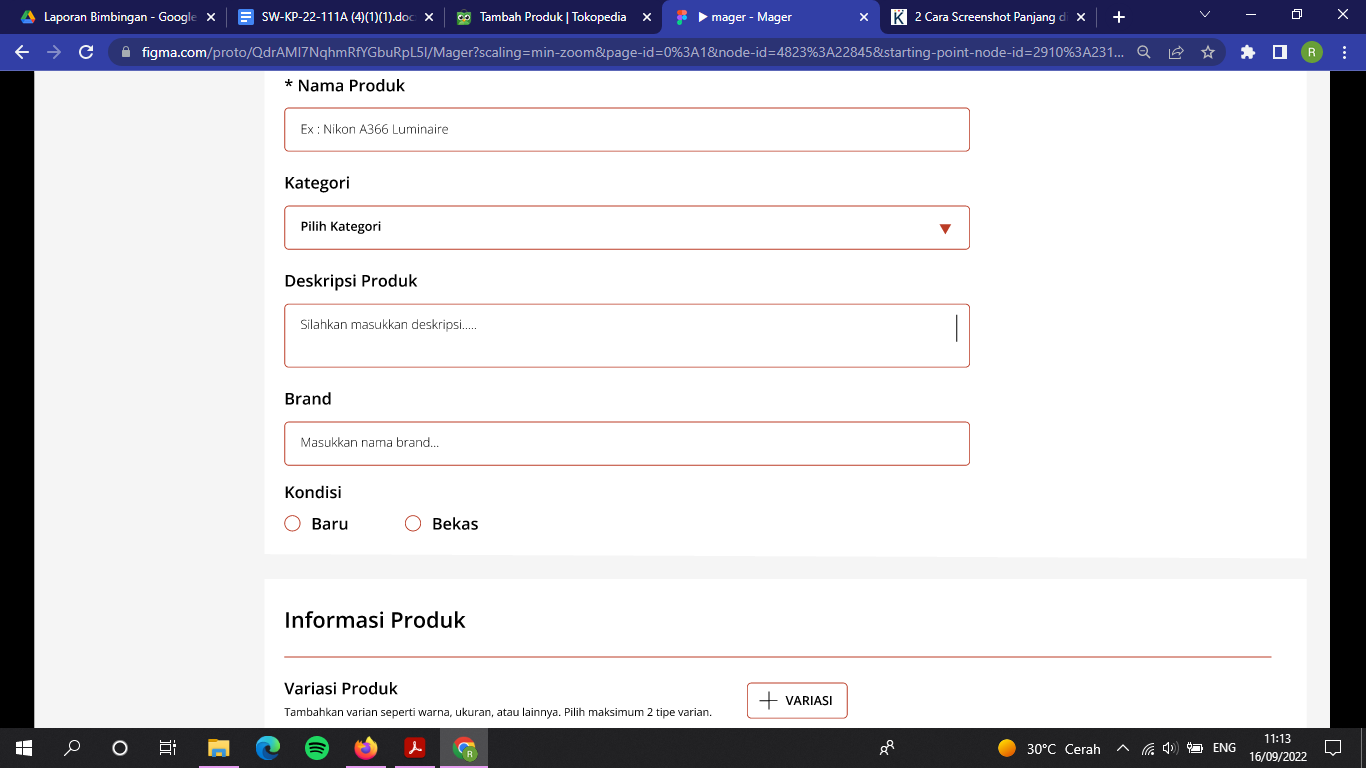
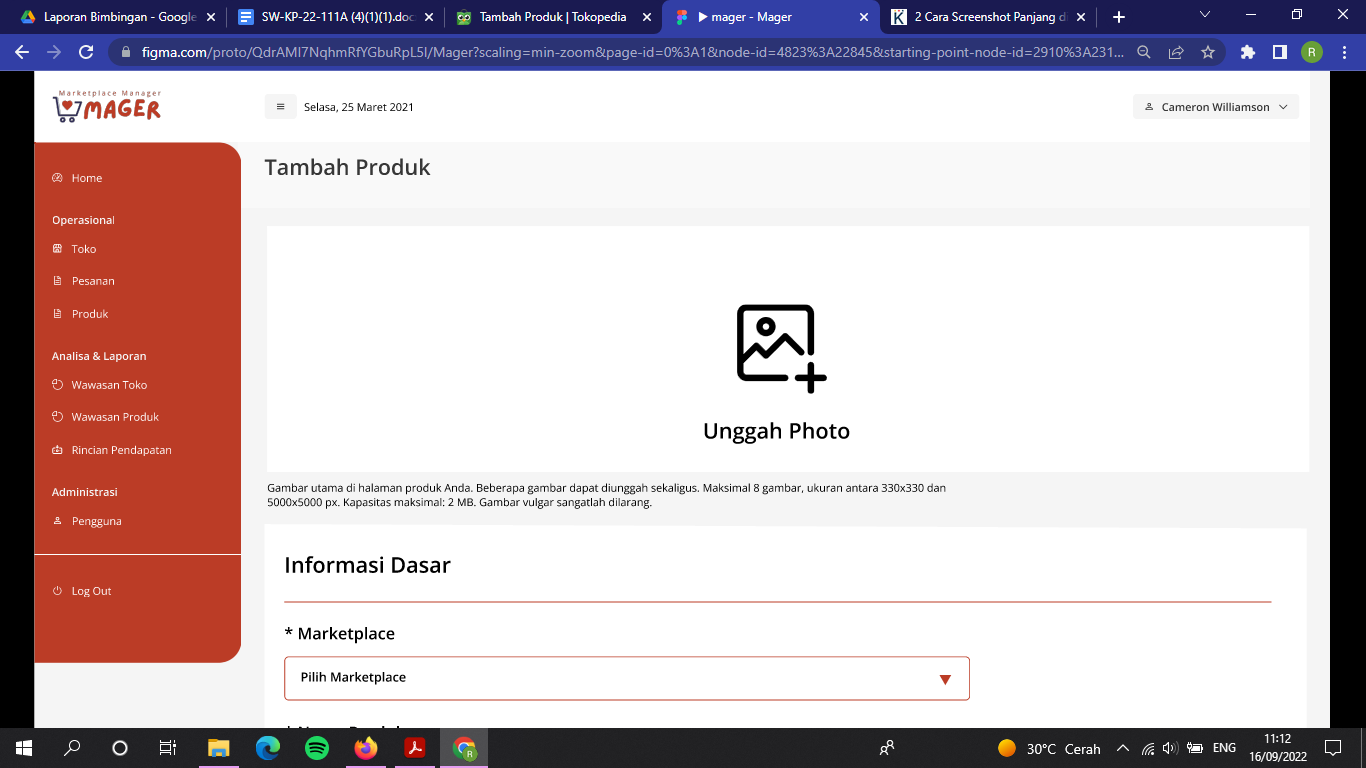
Jika belum, lakukan pengisian form dengan sesuai

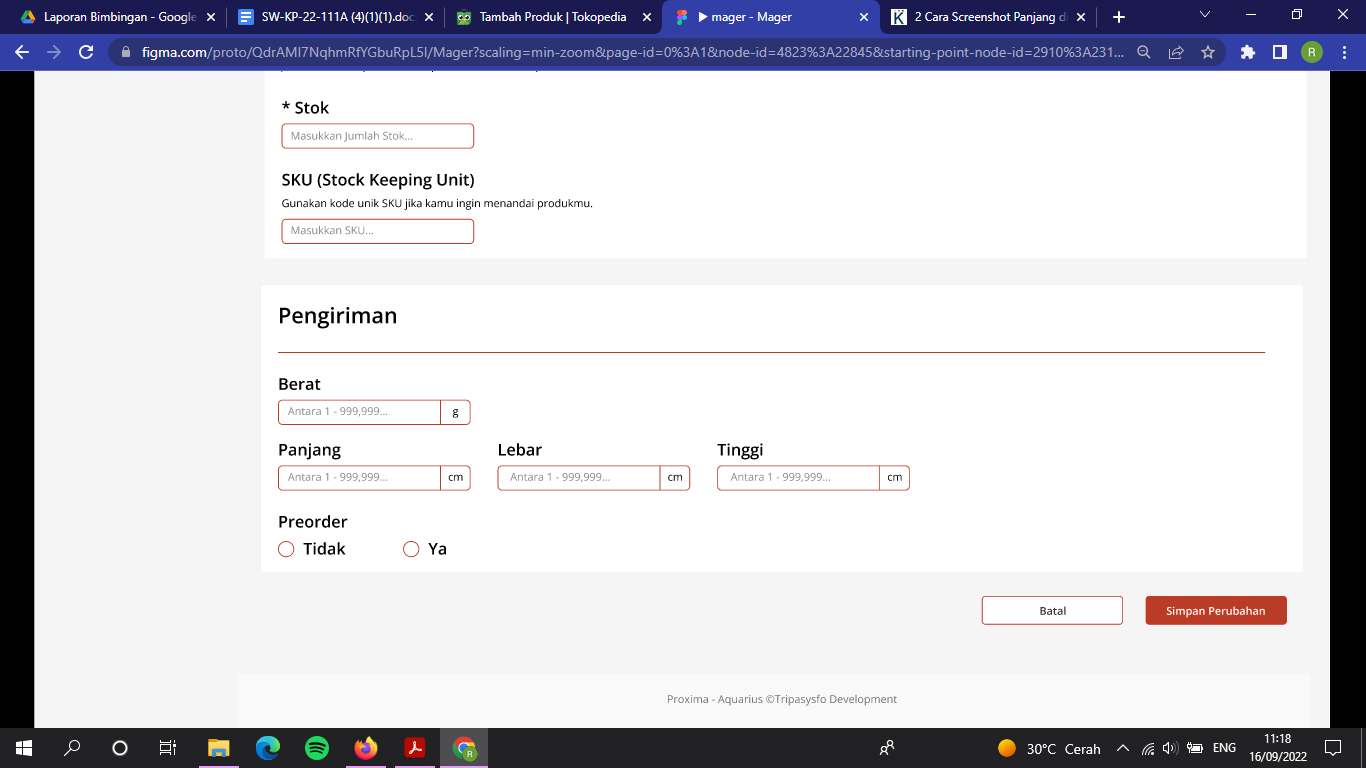
#### **Spesifikasi Report**

Tidak ada.

#### **Design Screen Layout**

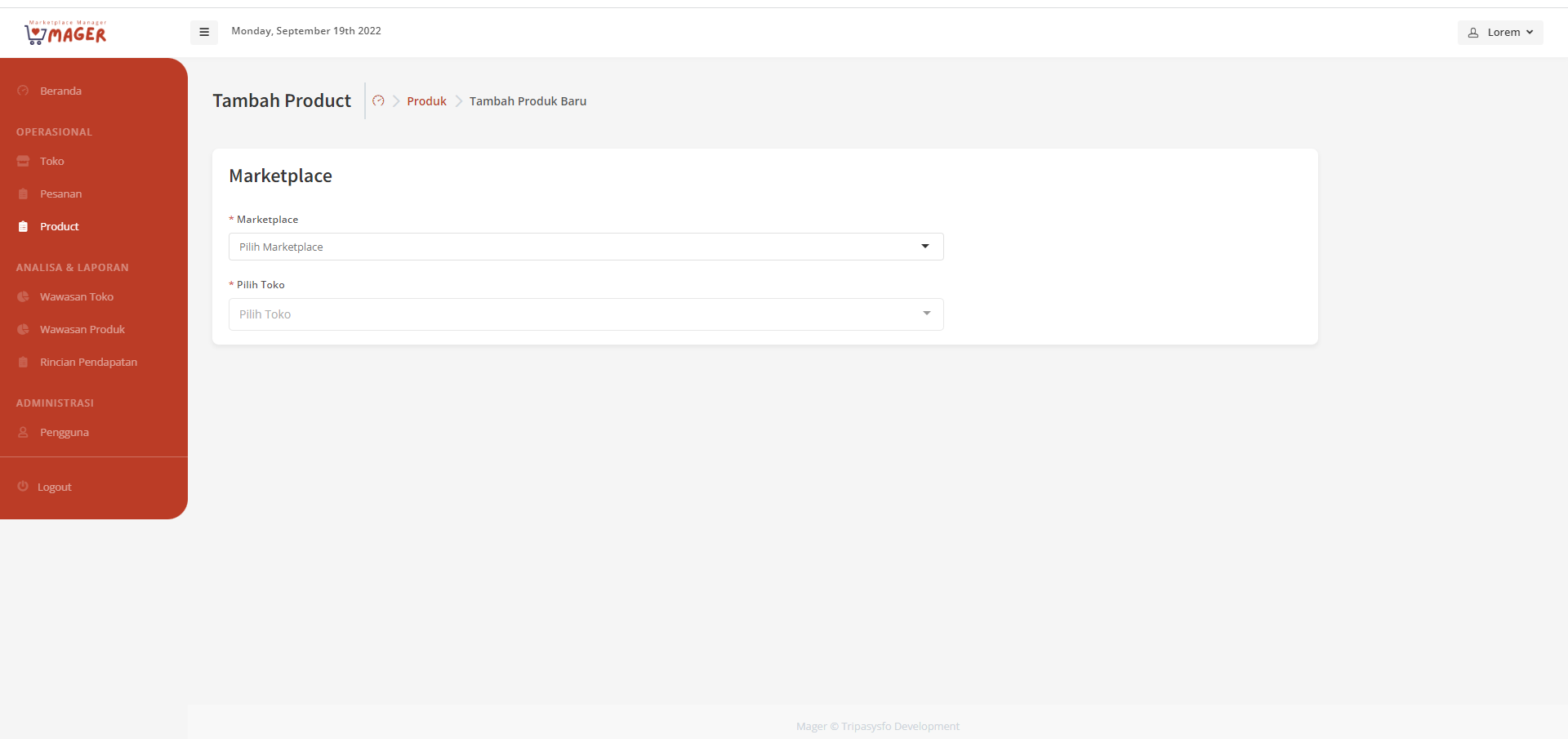
Berikut merupakan gambaran desain dari screen layout menambah produk.



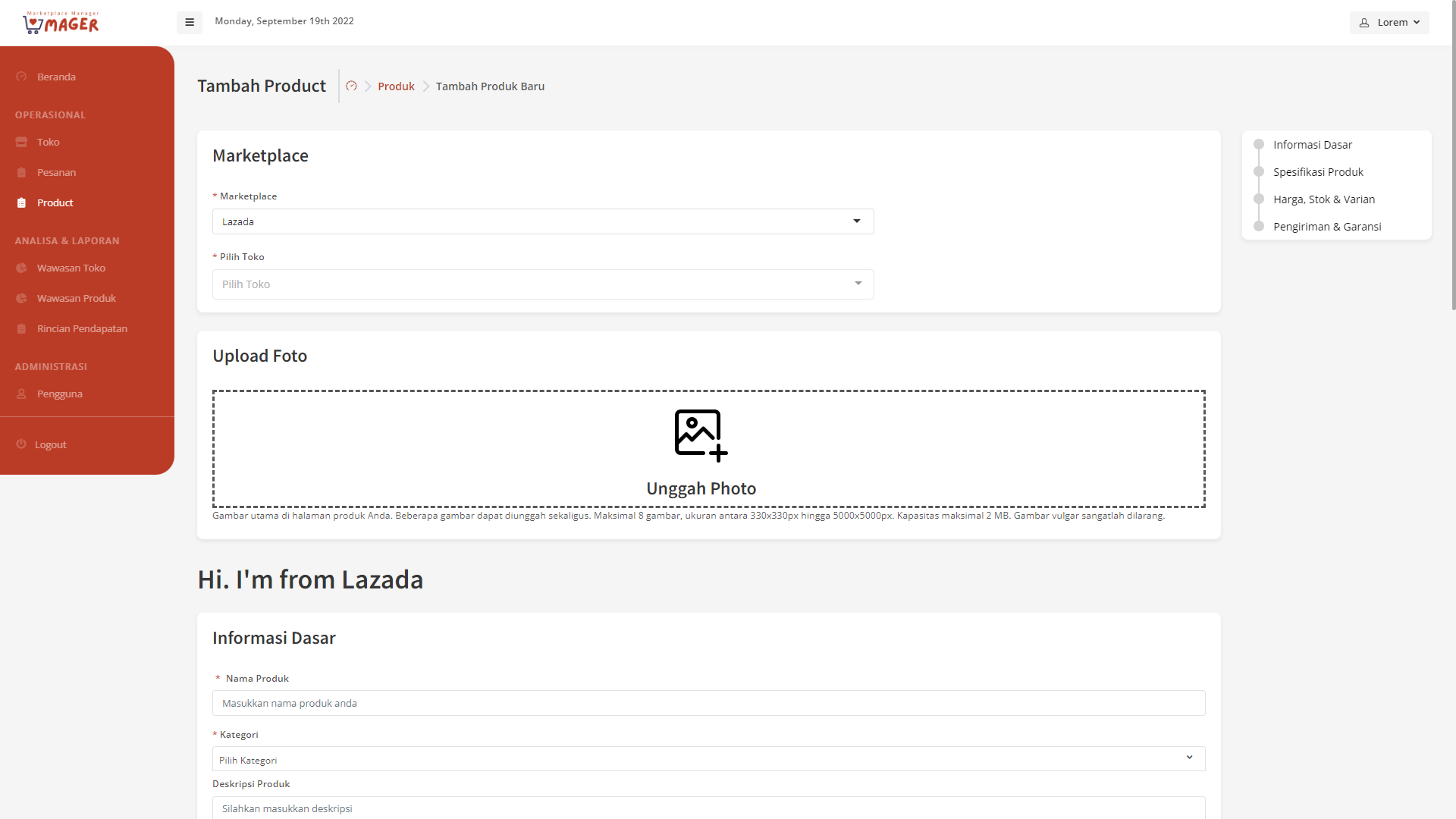


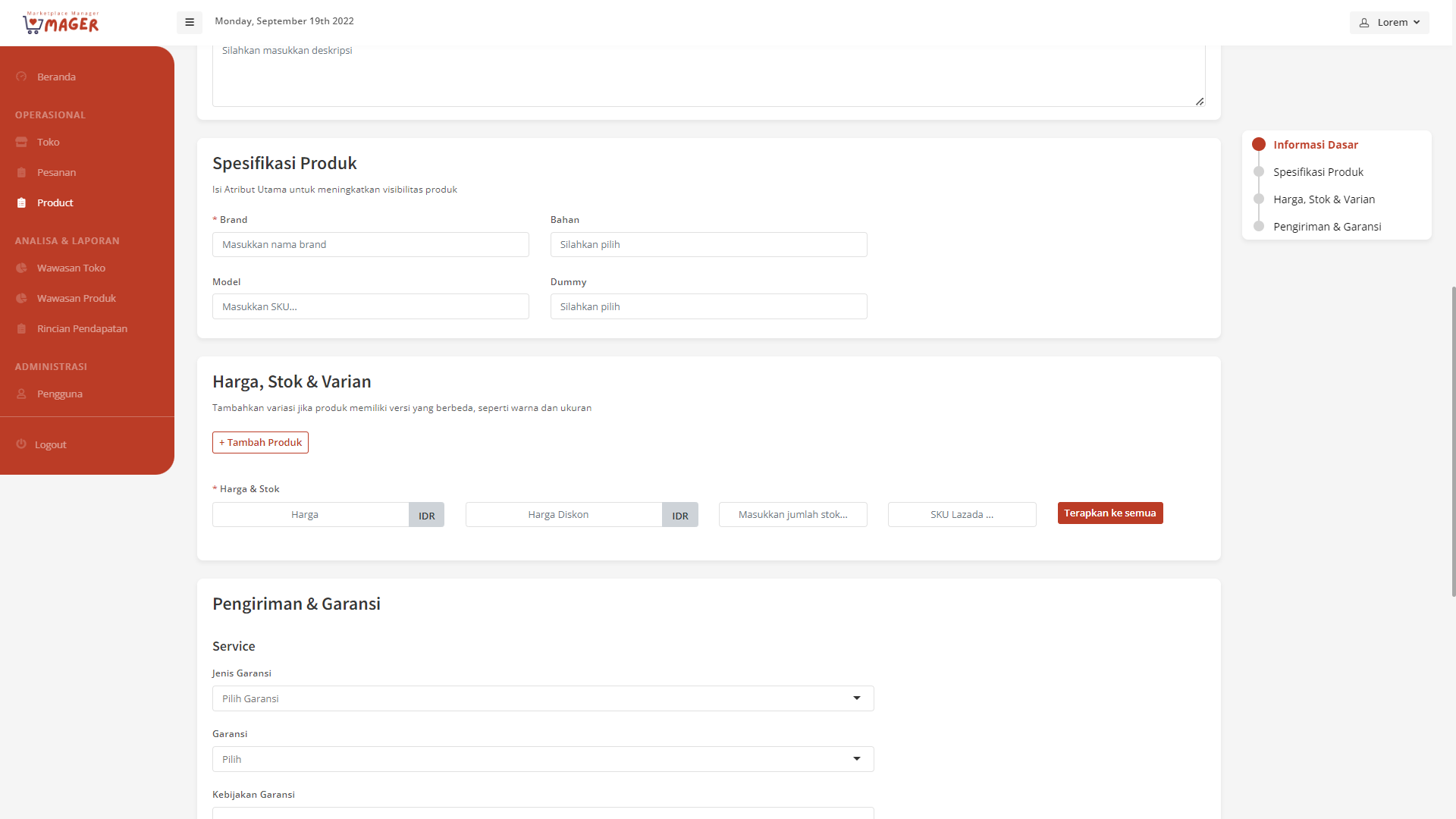
Gambar 37 Design Screen Layout Menambahkan Produk

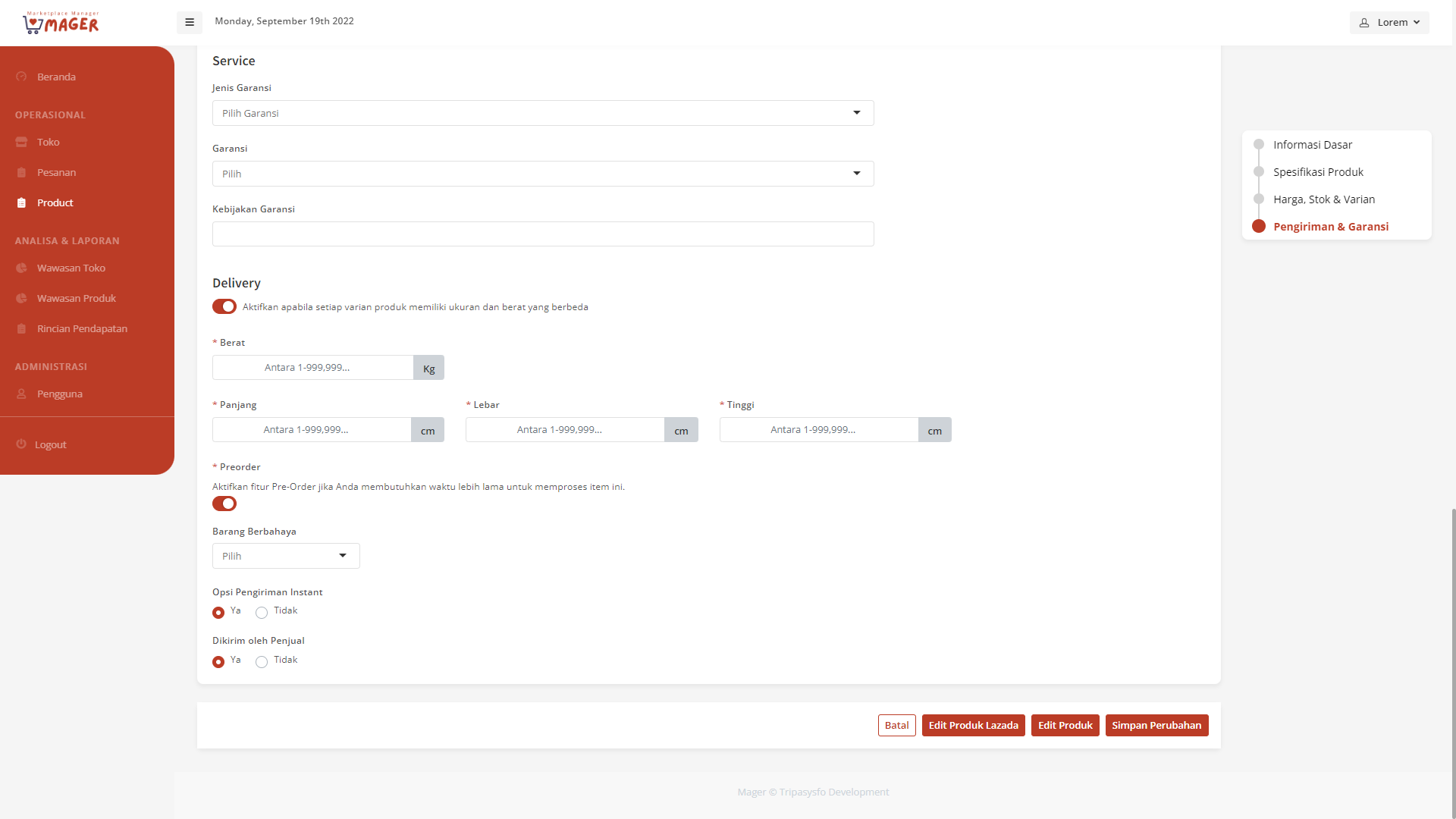
#### **Detail Screen Layout**



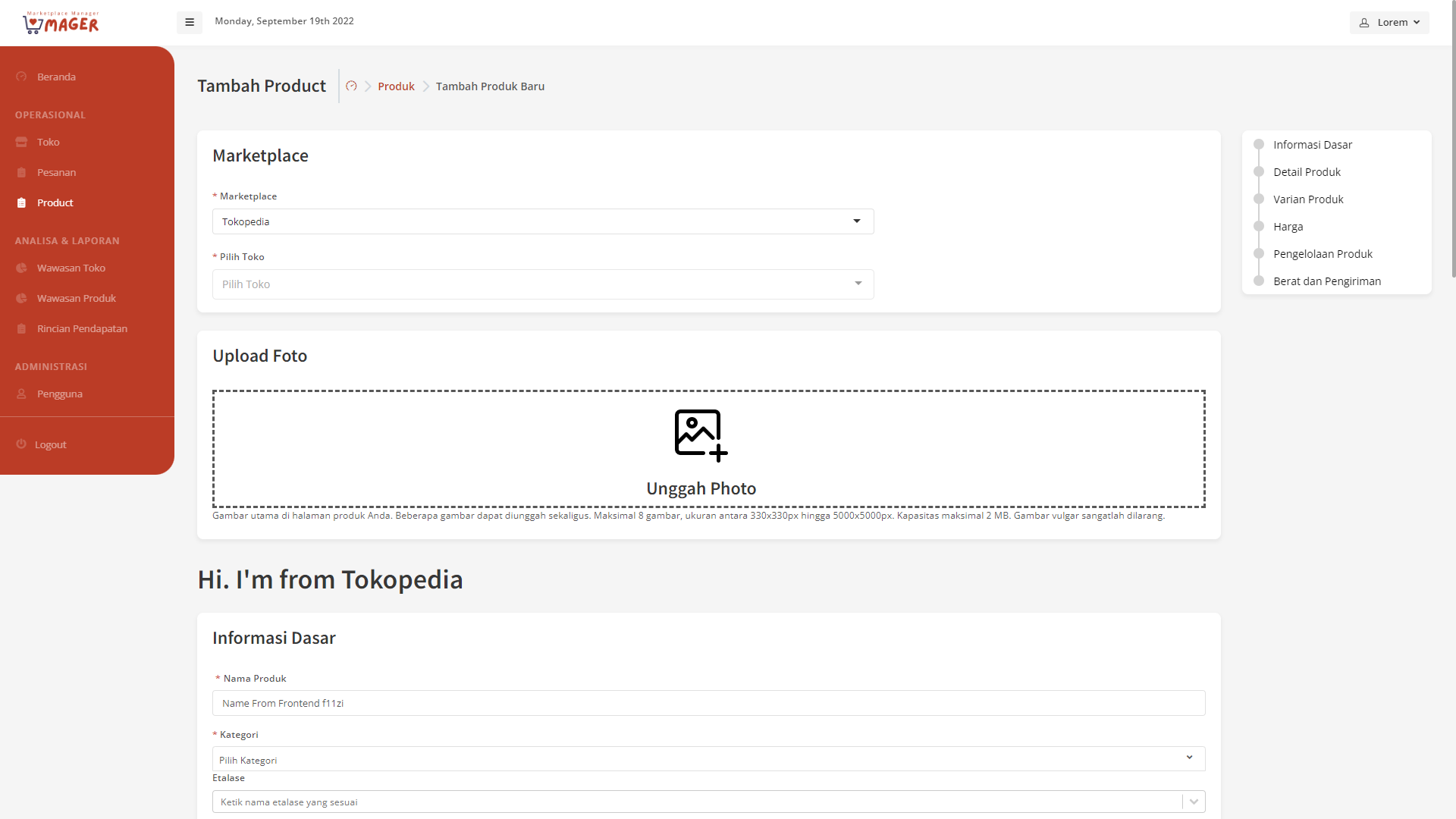
Gambar 38 Detail Screen Layout Memilih Marketplace

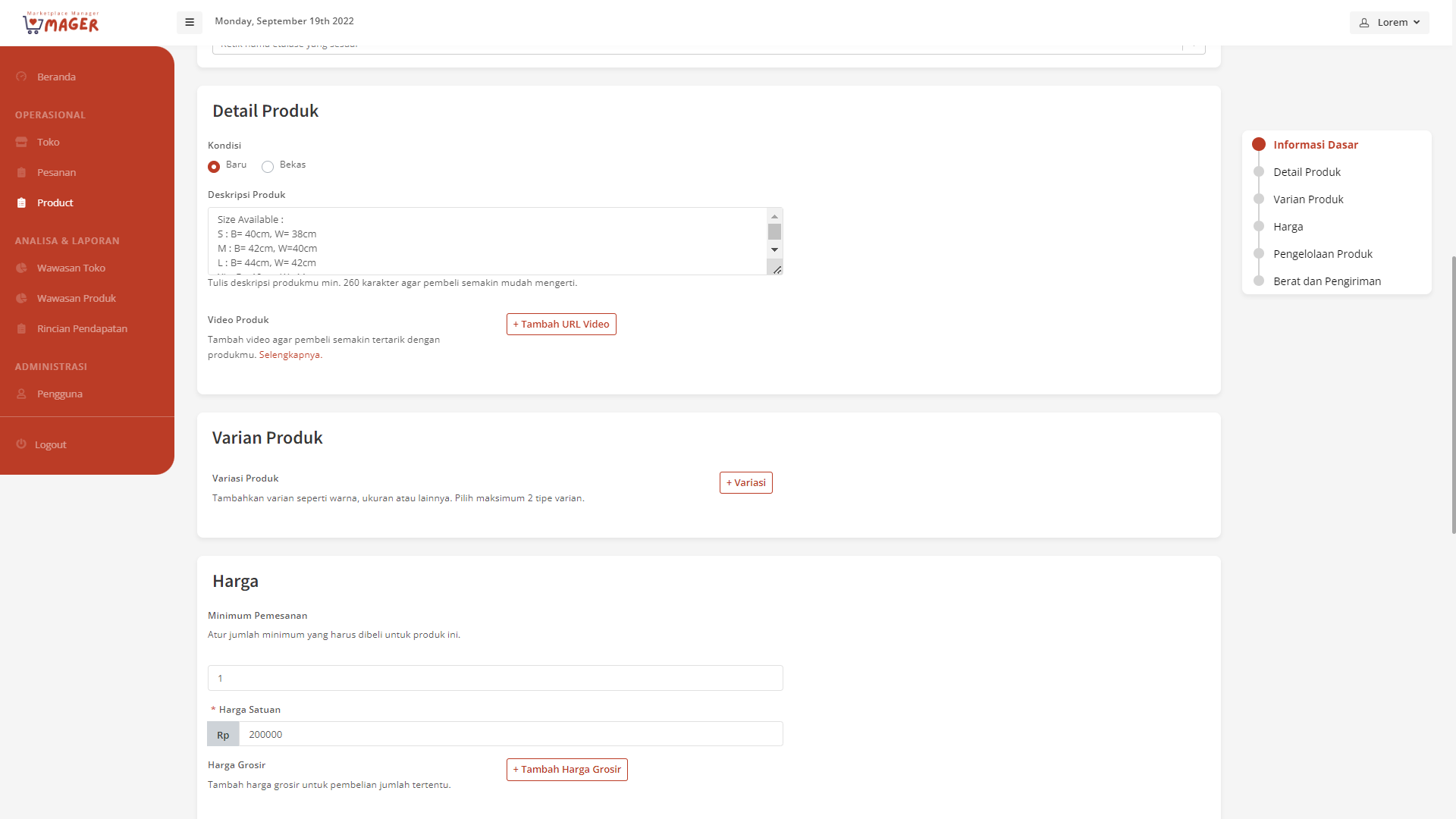


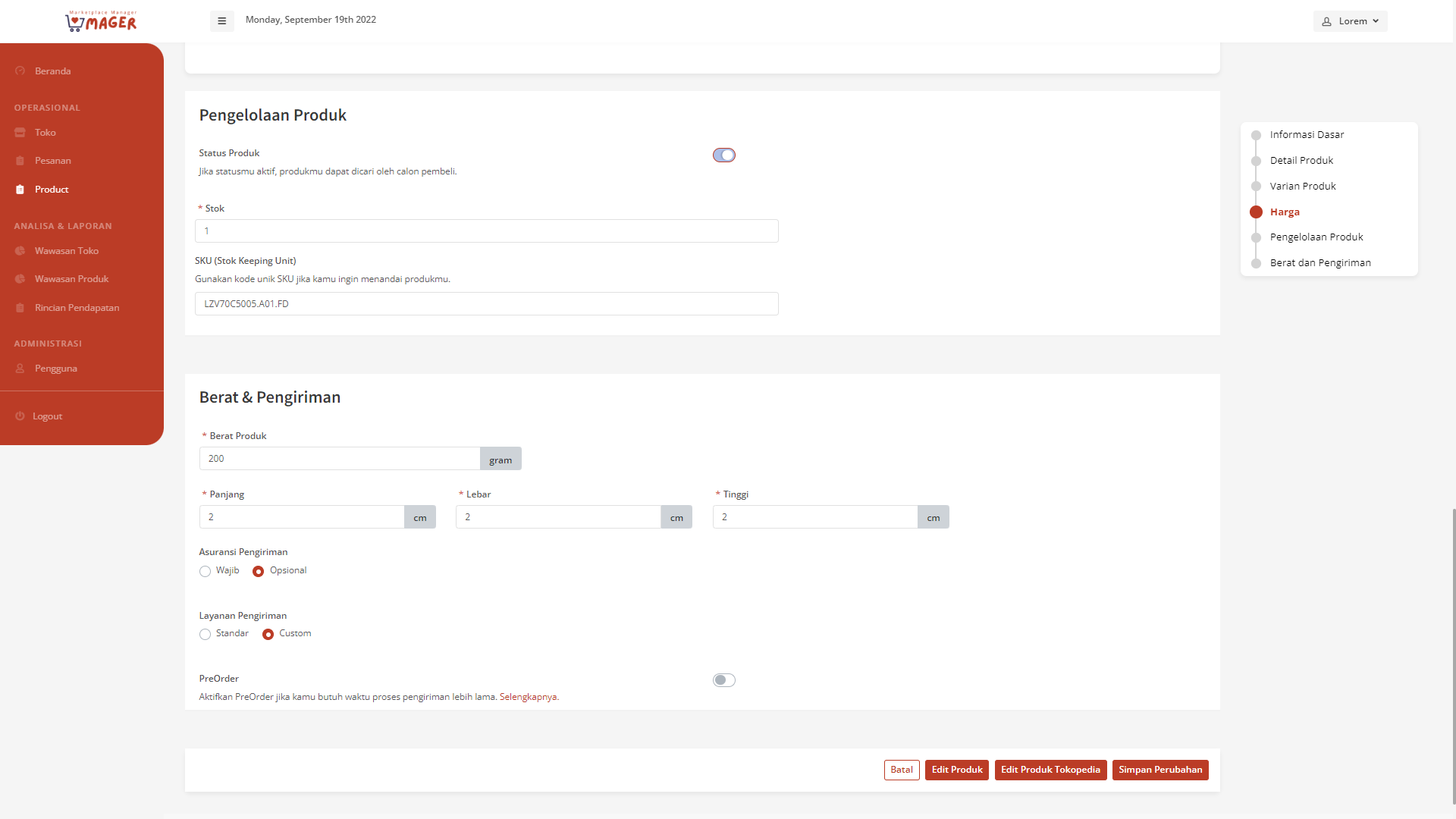




Gambar 39 Detail Screen Layout Menambahkan Produk (Lazada)







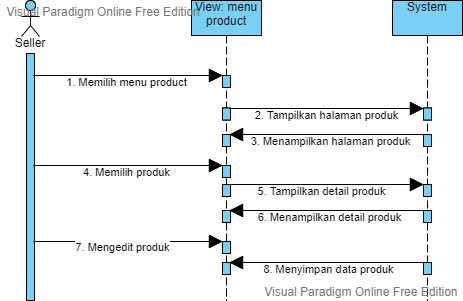
Gambar 40 Detail Screen Layout Menambahkan Produk (Tokopedia)

#### **Objects Specification**

Tabel 30 Object Specification Menambahkan Produk

|  |  |  |
| --- | --- | --- |
| **Id Objek** | **Jenis** | **Keterangan** |
| BatalButton | Button | Jika di klik, akan membatalkan penambahan produk dan melakukan redirect ke laman produk |
| SimpanPerubahanButton | Button | Mengaktifkan algoritma untuk melakukan penambahan produk yang terintegrasi dengan API Marketplace sehingga penambahan produk yang dilakukan tersimpan pada masing - masing database |

### **Sequence Diagram Mengedit Produk**



Gambar 41 Sequence Diagram Mengedit Produk

#### **Query Specification (jika ada)**

Tidak ada.

#### **Error Message**

Pesan *error* pada fungsi mengedit produk dapat dilihat pada tabel dibawah ini.

Tabel 31 Error Message

|  |  |  |
| --- | --- | --- |
| **No.** | **Keterangan** | **Pesan *error* yang disampaikan** |
| 1 | Salah satu field tidak diisi sementara pengguna memilih button **“Simpan Perubahan”** | Client error: `POST https://fs.tokopedia.net/v3/products/fs/15214/create?shop\_id=11637362` resulted in a `400 Bad Request` |

#### **Algorithm**

Id Process : FS-03

Objek terkait : Mengedit Produk

Event : click

Tabel sme\_marketplace sudah dibuat strukturnya

**Final State (FS)**:

Menghasilkan penambahan produk pada marketplace

**Spesifikasi Proses/algoritma**:

1. Memilih menu produk
2. Menekan tombol kelola
3. Menekan tombol edit
4. Mengisi form pada laman
5. IF sudah sesuai, klik Simpan Perubahan

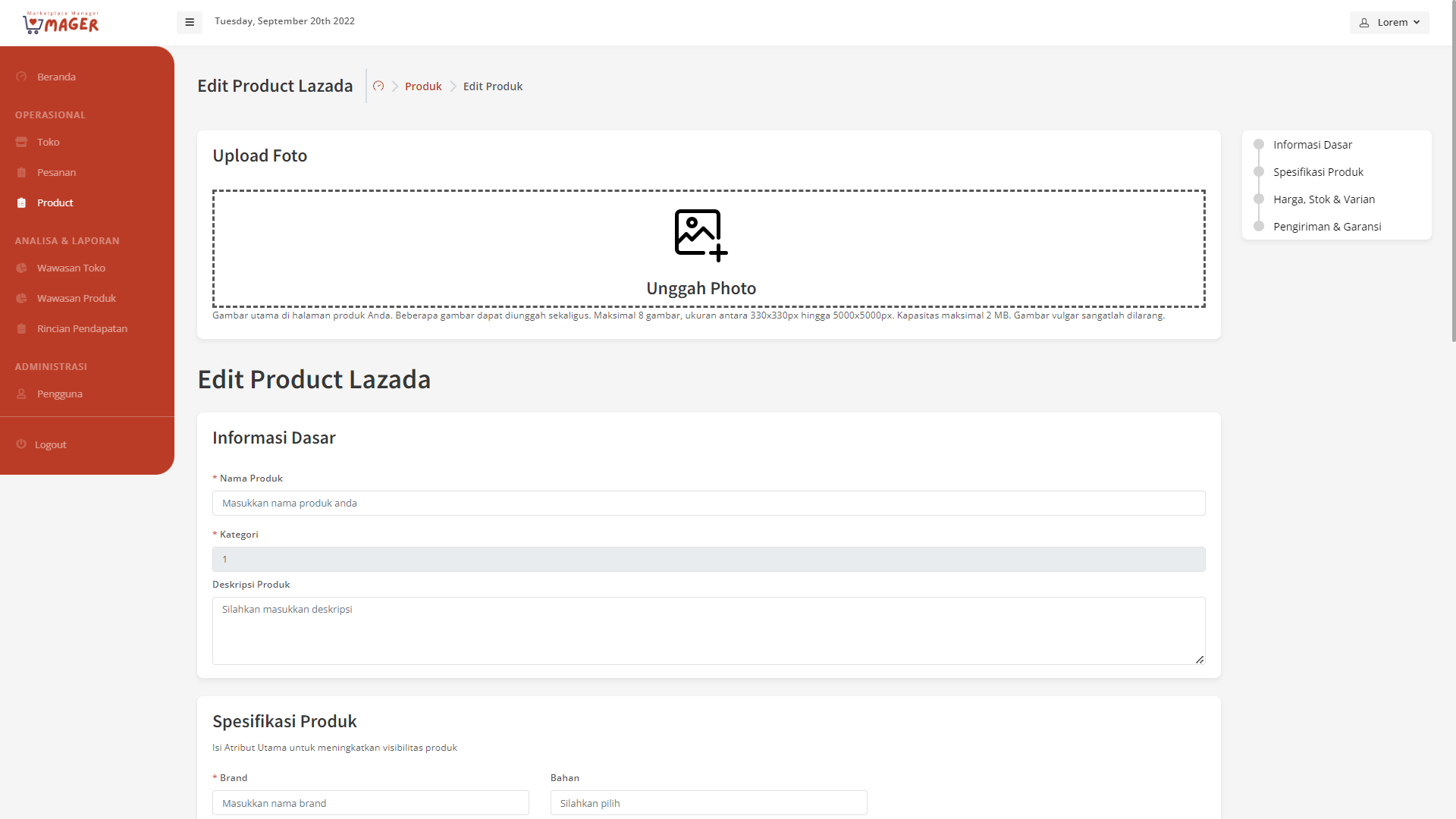
ELSE

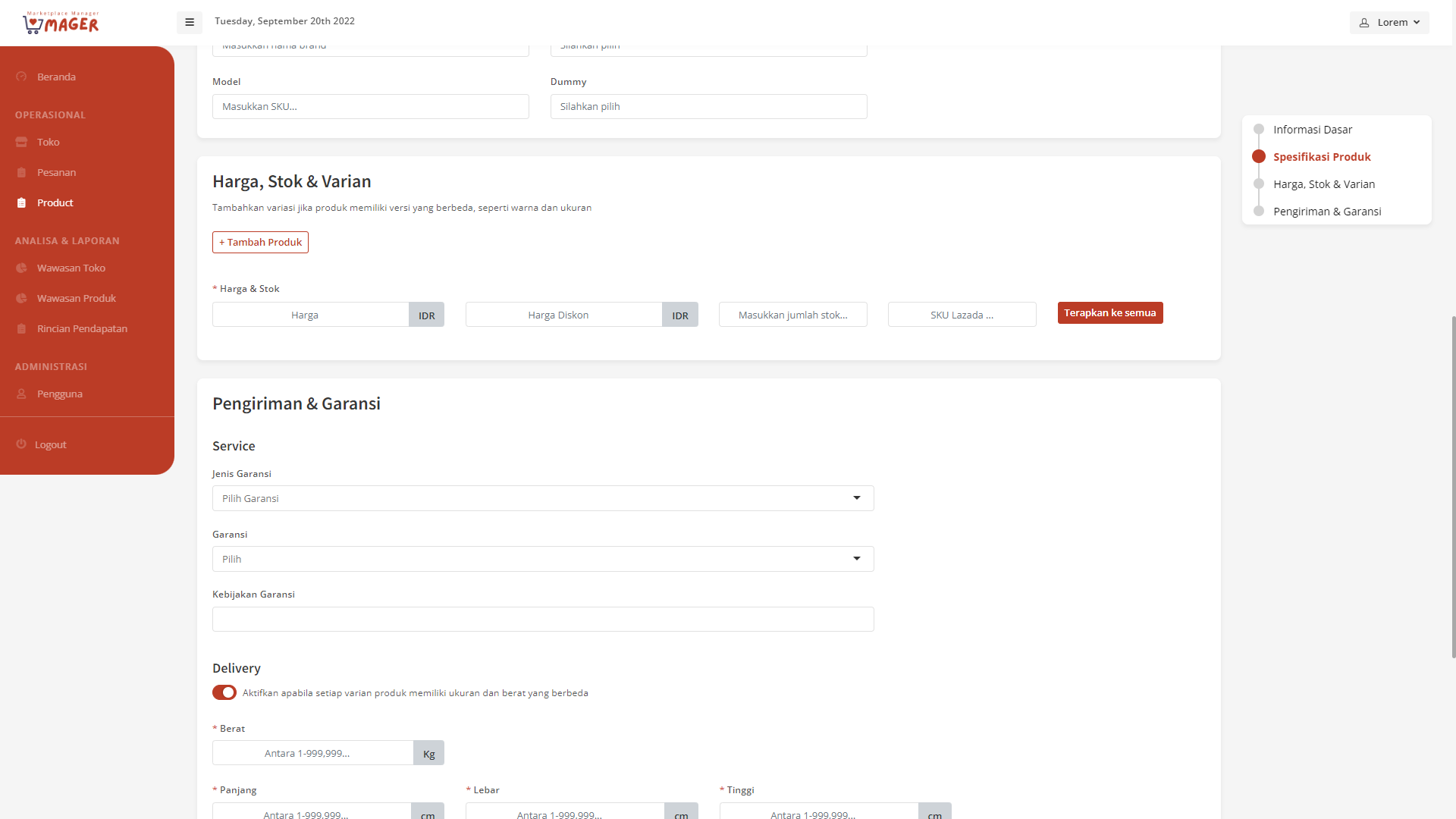
Jika belum, lakukan pengisian form dengan sesuai

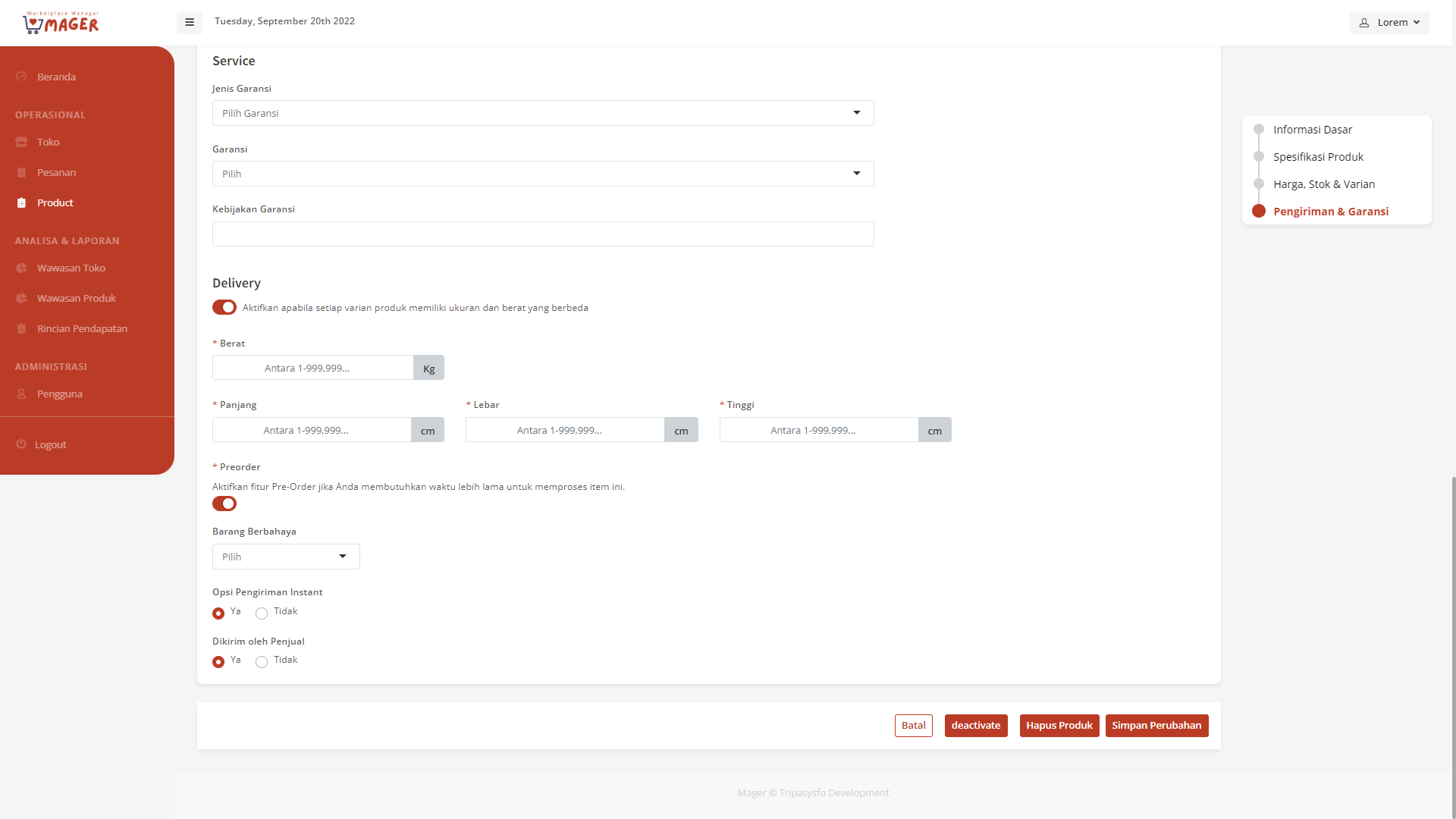
#### **Spesifikasi Report**

Tidak ada.

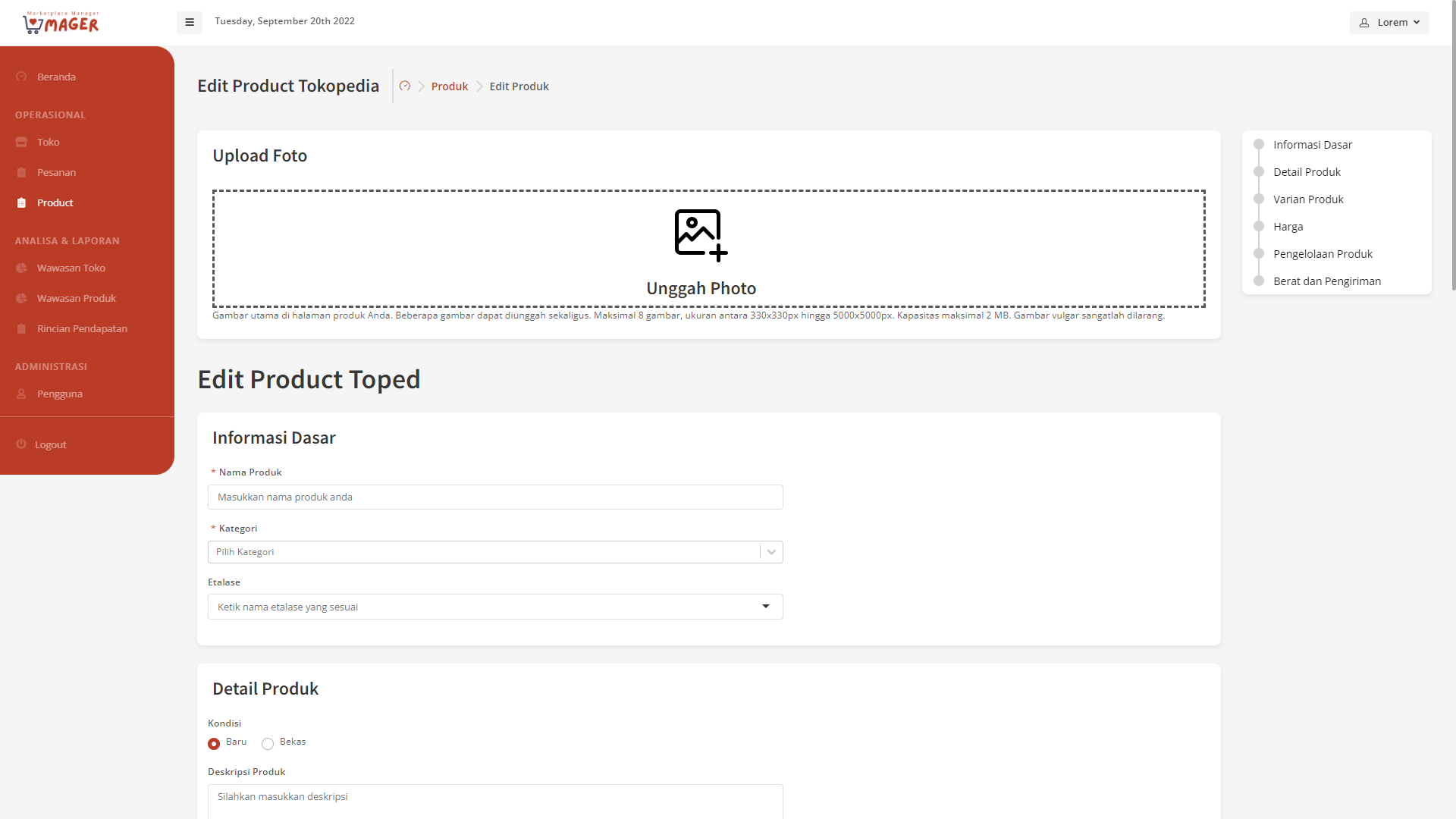
#### **Detail Screen Layout**

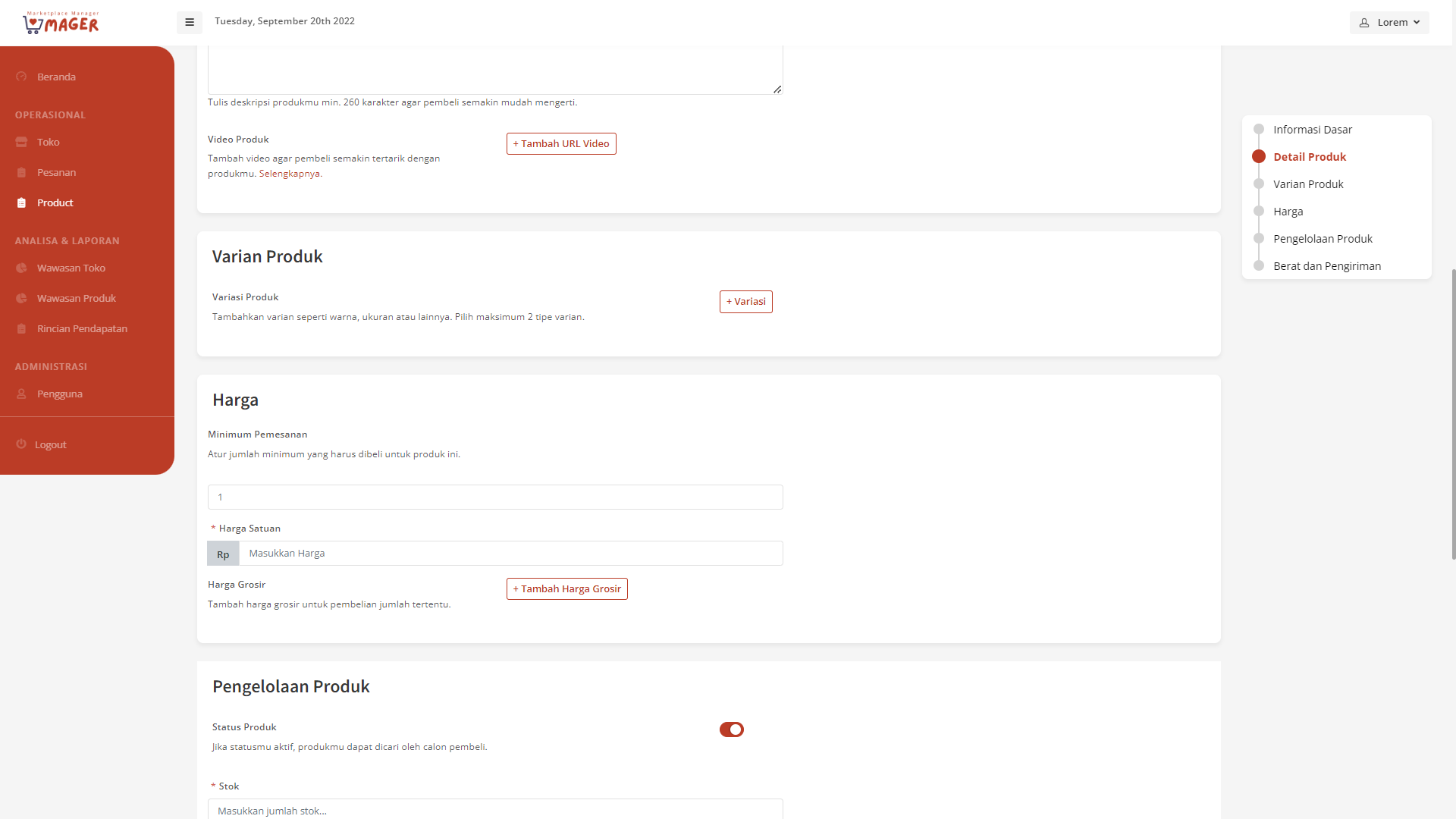


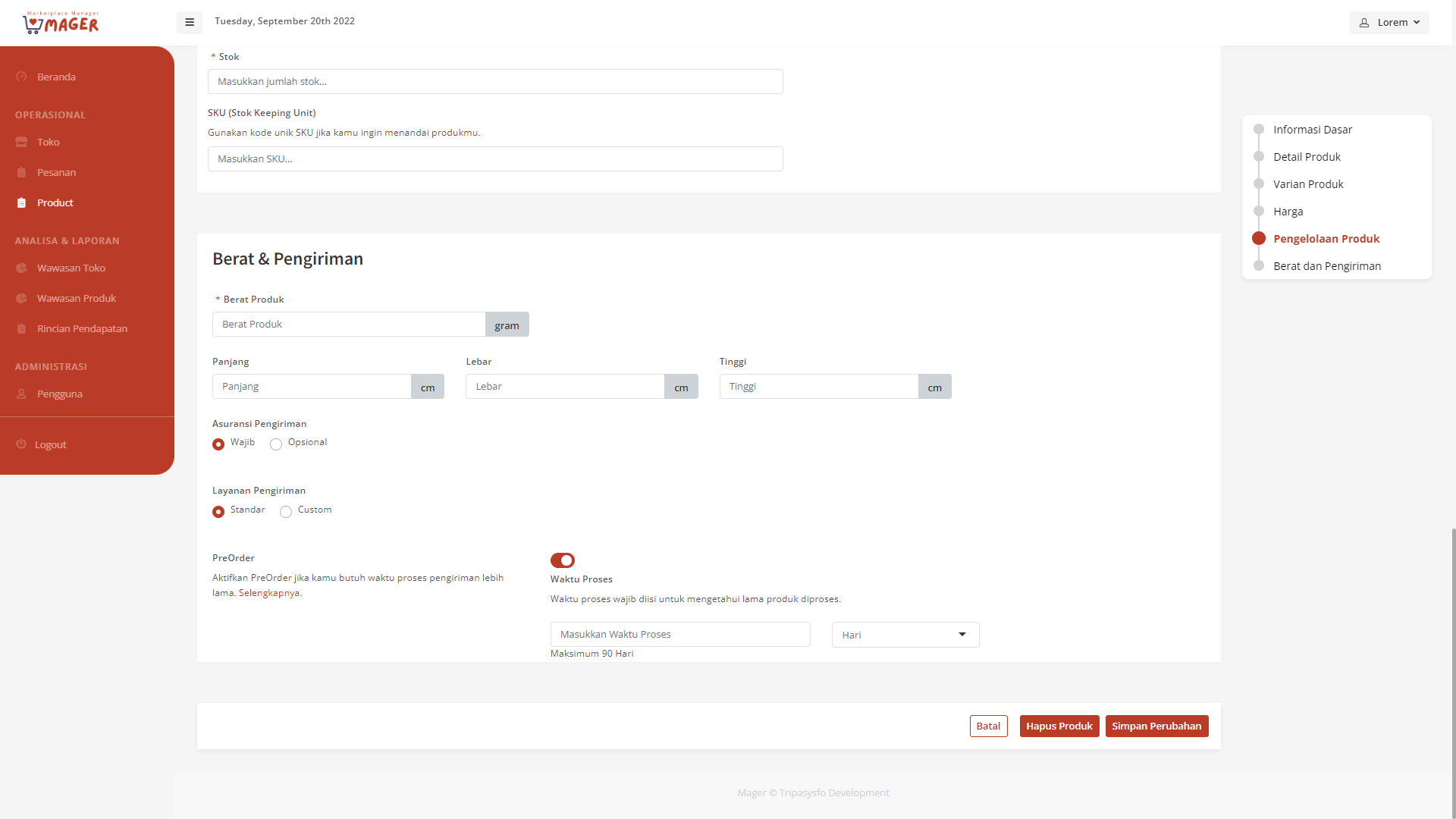




Gambar 42 Detail Screen Layout Mengedit Produk (Lazada)







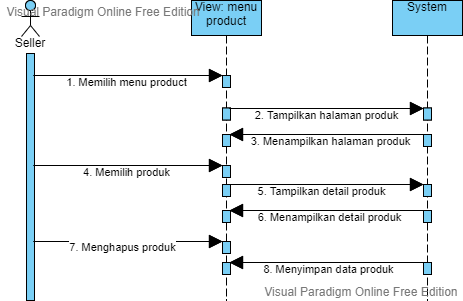
Gambar 43 Detail Screen Layout Mengedit Produk (Tokopedia)

#### **Objects Specification**

Tabel 32 Object Specification Mengedit Produk

|  |  |  |
| --- | --- | --- |
| **Id Objek** | **Jenis** | **Keterangan** |
| Batal Button | Button | Jika di klik, akan membatalkan penambahan produk dan melakukan *redirect* ke laman produk |
| Simpan Perubahan Button | Button | Mengaktifkan algoritma untuk melakukan penambahan produk yang terintegrasi dengan API Marketplace sehingga penambahan produk yang dilakukan tersimpan pada masing-masing *database* |

### **Sequence Diagram Menghapus Produk**



Gambar 44 Sequence Diagram Menghapus Produk

#### **Query Specification (jika ada)**

Tidak ada.

#### **Error Message**

Tidak ada.

#### **Algorithm**

Id Process : FS-04

Objek terkait : Menghapus Produk

Event : click

**Initial State (IS)**:

Tabel sme\_marketplace sudah dibuat strukturnya

**Final State (FS)**:

Menghasilkan penambahan produk pada marketplace

**Spesifikasi Proses/algoritma**:

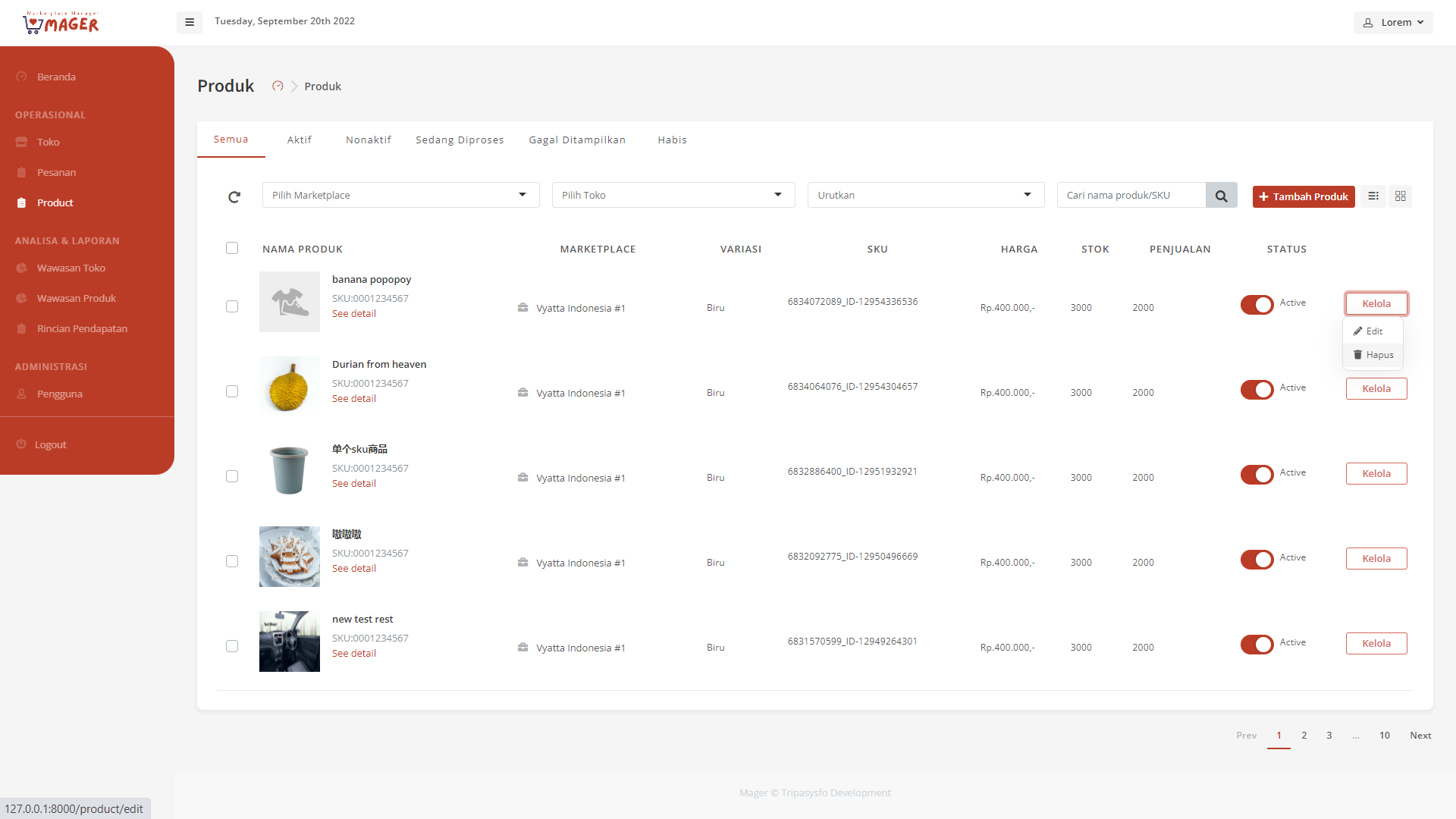
1. Memilih menu produk
2. Menekan tombol kelola
3. Menekan tombol hapus
4. Jika belum, lakukan pengisian form dengan sesuai

#### **Spesifikasi Report**

Tidak ada.

#### **Detail Screen Layout**

Berikut merupakan gambaran dari detail screen layout halaman menghapus produk.



Gambar 45 Detail Screen Layout Halaman Menghapus Produk

#### **Objects Specification**

Tabel 33 Object Specification Menghapus Produk

|  |  |  |
| --- | --- | --- |
| **Id Objek** | **Jenis** | **Keterangan** |
| TambahProdukButton | *Button* | Jika diklik maka akan mengarah ke halaman *form add product* |
| KelolaButton | Dropdown Button | Jika diklik akan menampilkan sub-menu Edit maupun Hapus |
| EditProdukButton | *Button* | Jika diklik maka akan mengarah ke halaman *form edit product* |
| HapusProdukButton | *Button* | Jika diklik maka akan menghapus salah satu produk pada list |

## **Physical File**

Pada sub bab ini akan dijelaskan dekomposisi fisik dari modul yang berisi struktur direktori dan pengumpulan fungsi menjadi *file* yang terdapat pada MAGER.

Pada sub bab *physical file* dijelaskan dekomposisi fisik dari modul yang berisi struktur direktori dan pengumpulan fungsi menjadi *file*. Berikut akan diuraikan nama direktori serta nama *file* dari fungsi yang ada di kebutuhan fungsional.

Tabel 34 Physical File

| **Nama Direktori** | **Nama File** | **Nama Modul** | **Nama Fungsi** | **Keterangan** |
| --- | --- | --- | --- | --- |
| …LasRoha\app\Http\ Controllers\ API\ Auth | RegisterController.php | - | register | Fungsi ini digunakan untuk melakukan registrasi ke aplikasi MAGER. |
| …LasRoha\app\Http\ Controllers\ API\ Auth | LoginController.php | - | login | Fungsi ini digunakan untuk login ke aplikasi MAGER. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ProductController.php | - | index | Fungsi ini digunakan untuk melihat halaman produk. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ProductController.php | - | postProductLazada | Fungsi ini digunakan untuk halaman menambahkan produk melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ProductController.php | - | updateProductLazada | Fungsi ini digunakan untuk halaman update produk melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ProductController.php | - | deleteProductLazada | Fungsi ini digunakan untuk halaman hapus produk melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ItemProductController.php | - | dataCategoryLazada | Fungsi ini digunakan untuk melakukan fetching data Category yang tersedia pada Marketplace Lazada melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ItemProductController.php | - | dataWarrantyLazada | Fungsi ini digunakan untuk melakukan fetching data Warranty/Garansi yang tersedia pada Marketplace Lazada melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ItemProductController.php | - | dataBrandLazada | Fungsi ini digunakan untuk melakukan fetching data Brand/Model yang tersedia pada Marketplace Lazada melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ItemProductController.php | - | dataProductItemLazada | Fungsi ini digunakan untuk melakukan fetching data Item yang tersedia pada Marketplace Lazada melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ProductController.php | - | deactivateProductLazada | Fungsi ini digunakan untuk menonaktifkan produk pada Marketplace Lazada melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ SME | ProductController.php | - | activateProductLazada | Fungsi ini digunakan untuk mengaktifkan produk pada Marketplace Lazada melalui API Lazada. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ProductController.php | - | postProductTokopedia | Fungsi ini digunakan untuk halaman menambahkan produk melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ProductController.php | - | updateProductTokopedia | Fungsi ini digunakan untuk halaman mengedit produk melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ProductController.php | - | deleteProductTokopedia | Fungsi ini digunakan untuk halaman hapus produk melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ItemProductController.php | - | getCategory | Fungsi ini digunakan untuk melakukan fetching data Category yang tersedia pada Marketplace Tokopedia melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ItemProductController.php | - | getActiveCourir | Fungsi ini digunakan untuk melakukan fetching data Kurir yang tersedia pada Marketplace Tokopedia melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ItemProductController.php | - | getAllShowcase | Fungsi ini digunakan untuk melakukan fetching data Showcase yang tersedia pada Marketplace Tokopedia melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ItemProductController.php | - | setInactivateProduct | Fungsi ini digunakan untuk menonaktifkan produk pada Marketplace Tokopedia melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ItemProductController.php | - | setActivateProduct | Fungsi ini digunakan untuk mengaktifkan produk pada Marketplace Tokopedia melalui API Tokopedia. |
| …LasRoha\app\Http\ Controllers\ API\ Toped | ItemProductController.php | - | getProductAnnotation | Fungsi ini digunakan untuk fetching daftar anotasi produk yang tersedia pada Marketplace Tokopedia melalui API Tokopedia. |

## **Traceability**

Pada sub bab ini akan dijelaskan penyimpanan data yang digunakan pada *system* dan keterkaitannya dengan ER.

### **Data**

Pada bagian ini dijelaskan traceability dari tabel aplikasi yang dirancang terhadap *Entity Class* dan ER. *Traceability* ditampilkan pada tabel berikut ini.

Tabel 35 Data

| **Nama Tabel** | **Primary Key** | **Entity Class** | **E/R** | **Deskripsi Isi** |
| --- | --- | --- | --- | --- |
| admin\_user | id | user | user | Merupakan tabel yang berelasi dengan sme\_bill |
| sme\_marketplace | id | user | user | Merupakan tabel yang berelasi dengan sme\_bill |
| sme\_bill | id | market | market | Merupakan tabel yang berelasi dengan sme\_contact, sme\_marketplace, admin\_user, sme\_product\_transaction, sme\_packing\_info, sme\_incurred\_fees, sme\_payout |
| sme\_contact | id | market | market | Merupakan tabel yang berelasi dengan sme\_bill |
| sme\_product\_transaction | id | bill | bill | Merupakan tabel yang berelasi dengan sme\_bill, sme\_product, sme\_transaction |
| sme\_packing\_info | id | bill | bill | Merupakan tabel yang berelasi dengan sme\_bill, sme\_packed\_item |
| sme\_incurred\_fees | id | contact | contact | Merupakan tabel yang berelasi dengan sme\_bill |
| sme\_payout | id | contact | contact | Merupakan tabel yang berelasi dengan sme\_bill |
| sme\_product | id | product\_transaction | product\_transaction | Merupakan tabel yang berelasi dengan sme\_product\_transaction, sme\_product\_skus, sme\_product\_image |
| sme\_transaction | id | product\_transaction | product\_transaction | Merupakan tabel yang berelasi dengan sme\_product\_transaction, sme\_packed\_item |
| sme\_packing\_info | id | packing\_info | packing\_info | Merupakan tabel yang berelasi dengan sme\_bill, sme\_packed\_item, sme\_transaction |
| sme\_packed\_item | id | packing\_info | packing\_info | Merupakan tabel yang berelasi dengan sme\_packing\_info, sme\_transaction |
| sme\_product\_image | id | incurred\_fees | incurred\_fees | Merupakan tabel yang berelasi dengan sme\_product\_skus, sme\_product |
| sme\_product\_skus | id | incurred\_fees | incurred\_fees | Merupakan tabel yang berelasi dengan sme\_product, sme\_product\_skus\_image, sme\_product\_image |
| sme\_product\_skus\_image | id | payout | payout | Merupakan tabel yang berelasi dengan sme\_product\_skus |

### **Requirements**

Daftar *traceability functional* ditampilkan pada tabel berikut ini. Pada sistem terdapat

beberapa fungsi utama, yakni sebagai berikut:

Tabel 36 Requirements

| **Features-ID** | **Nama Method** | **Description** |
| --- | --- | --- |
| FR-01 | Melihat Produk | Fungsi ini digunakan *seller* untuk melakukan Melihat produk dengan akun toko pada masing - masing *marketplace*. Sehingga *output* yang diharapkan adalah menampilkan daftar produk, *marketplace*, variasi, harga, stok, penjualan, status dan kelola produk yang terintegrasi dengan akun toko masing-masing *marketplace* seperti Lazada maupun Tokopedia. |
| FR-02 | Menambahkan Produk | Fungsi ini digunakan *seller* untuk menambahkan produk di toko pada *marketplace*. Dalam penambahan produk terdapat beberapa komponen penting sehingga produk dapat dipublikasikan seperti Informasi Dasar, Spesifikasi atau Detail Produk, Variasi Produk, Harga serta Stok Produk, Pengiriman & Garansi Produk. |
| FR-03 | Mengedit Produk | Fungsi ini digunakan *seller* untuk mengedit informasi produk yang telah dipublikasikan pada akun toko di *marketplace*. Dalam fungsi ini, *seller* dapat melakukan aksi berupa penyesuaian informasi produk seperti jumlah stok maupun harga produk, status produk berupa aktif maupun tidak aktif, serta beragam perubahan lainnya sesuai dengan kondisi sesungguhnya berdasarkan penyesuaian yang dilakukan oleh *seller.* |
| FR-04 | Menghapus Produk | Fungsi ini digunakan *seller* untuk menghapus produk yang sebelumnya dibuat pada toko melalui *marketplace*. Fungsi menghapus produk meliputi semua status produk yang terdapat pada akun toko baik produk yang aktif maupun produk yang tidak aktif. |

# **Testing**

Pada bab ini dijelaskan mengenai persiapan pengujian, perencanaan pengujian dan identifikasi serta hasil pengujian terhadap pengembangan Aplikasi Pengelolaan Marketplace Multi-channel MAGER.

## **Test Preparation**

Pada bagian ini akan dibahas mengenai persiapan yang dilakukan sebelum melakukan pengujian meliputi persiapan prosedur, persiapan *hardware* dan *software.*

### **Procedural Preparation**

Persiapan prosedur yang dilakukan sebelum melakukan pengujian terhadap Aplikasi Pengelolaan Marketplace Multi-channel MAGER adalah sebagai berikut.

1. Mempersiapkan *tools* untuk menjalankan Aplikasi Pengelolaan Marketplace Multi-channel MAGER
2. Mempersiapkan *database* untuk pembangunan Aplikasi Pengelolaan Marketplace Multi-channel MAGER

### **HW & Network Preparation**

Persiapan perangkat keras dan jaringan yang dilakukan sebelum melakukan pengujian terhadap Aplikasi Pengelolaan Marketplace Multi-channel MAGER adalah sebagai berikut.

1. Mempersiapkan laptop/komputer dengan spesifikasi yang sesuai dengan penjelasan pada bab sebelumnya
2. Laptop/Komputer yang digunakan harus terhubung dengan jaringan internet yang stabil.

### **SW Preparation**

Spesifikasi *software* yang harus dipersiapkan untuk melakukan pengujian terhadap Aplikasi Pengelolaan Marketplace Multi-channel MAGER adalah sama dengan kebutuhan *operational software* yang telah dijelaskan pada bab sebelumnya.

## **Test Plan and Identification**

Bagian ini menjelaskan lingkup keseluruhan dari perencanaan pengujian dari fungsi yang dibangun pada Aplikasi Pengelolaan Marketplace Multi-channel MAGER

## **Test Script & Result**

Pada bagian ini, akan menjelaskan secara lebih rinci butir uji yang sudah diuji.

### **Test Script Butir-Uji-1**

Tabel 37 Test Script Butir-Uji-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifikasi** | | BU-01 | | |
| **No. Fungsi** | | - | | |
| **Nama Butir Uji** | | Pengujian tombol di navbar mengarah ke tiap bagian halaman | | |
| **Tujuan** | | Fungsi ini bertujuan untuk mengarahkan pengguna ke bagian halaman yang benar | | |
| **Deskripsi** | | Uji fungsi tombol pada navbar dilakukan ketika pengguna ingin ke halaman yang diinginkan | | |
| **Kondisi Awal** | | Pengguna sudah terhubung pada Aplikasi Marketplace Multi Channel MAGER | | |
| **Tanggal Pengujian** | | 03 Agustus 2022 10:00 WIB | | |
| **Penguji** | | Semua anggota tim | | |
| **Skenario Pengujian** | | | | |
| 1.Pengguna mengakses salah satu halaman pada Aplikasi Marketplace Multi Channel MAGER.  2.Menekan salah satu menu pada navbar.  3.Akan mengarahkan ke bagian halaman yang tepat. | | | | |
| **Kriteria Evaluasi Hasil** | | | | |
| Pengguna diarahkan ke bagian halaman yang tepat | | | | |
| **Kasus dan Hasil Pengujian** | | | | |
| **Data Masukan** | **Yang diharapkan** | | **Pengamatan** | **Kesimpulan** |
| Pengguna menekan salah satu menu pada navbar. | Pengguna diarahkan ke bagian halaman yang tepat | | Pengguna diarahkan ke bagian halaman yang tepat | [x] diterima  [ ] ditolak |
| **Catatan** | | | | |
| Pengujian kasus berjalan dengan baik sesuai dengan ekspektasi. | | | | |

### **Test Script Butir-Uji-2**

Tabel 38 Test Script Butir-Uji-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifikasi** | | BU-02 | | |
| **No. Fungsi** | | - | | |
| **Nama Butir Uji** | | Pengujian *affixed sidebar* untuk mengarah pengguna ke bagian section tertentu pada halaman yang sama | | |
| **Tujuan** | | Fungsi ini bertujuan untuk menunjukkan di mana kita berada, bersama dengan beberapa konteks ke mana kita bisa berpindah pada halaman yang sama | | |
| **Deskripsi** | | Uji fungsi *affixed sidebar* dilakukan ketika pengguna ingin mengetahui posisi/section letak keberadaannya sekarang pada halaman yang sedang diakses | | |
| **Kondisi Awal** | | Pengguna sudah terhubung pada Aplikasi Marketplace Multi Channel MAGER | | |
| **Tanggal Pengujian** | | 03 Agustus 2022 10:00 WIB | | |
| **Penguji** | | Semua anggota tim | | |
| **Skenario Pengujian** | | | | |
| 1.Pengguna mengakses halaman tambah produk atau edit produk pada Aplikasi Marketplace Multi Channel MAGER.  2.Memilih salah satu marketplace Tokopedia ataupun Lazada.  3.Menekan salah satu section pada *affixed sidebar*  4.Halaman akan bergulir dan menampilkan bagian halaman sesuai dengan pilihan pengguna. | | | | |
| **Kriteria Evaluasi Hasil** | | | | |
| Pengguna diarahkan ke section halaman yang sesuai dengan pilihan pengguna. | | | | |
| **Kasus dan Hasil Pengujian** | | | | |
| **Data Masukan** | **Yang diharapkan** | | **Pengamatan** | **Kesimpulan** |
| Pengguna menekan salah satu section pada *affixed sidebar* | Pengguna diarahkan ke section halaman yang sesuai dengan pilihan pengguna. | | Pengguna diarahkan ke section halaman yang sesuai dengan pilihan pengguna. | [x] diterima  [ ] ditolak |
| **Catatan** | | | | |
| Pengujian kasus berjalan dengan baik sesuai dengan ekspektasi. | | | | |

### **Test Script Butir-Uji-3**

Tabel 39 Test Script Butir-Uji-3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifikasi** | | BU-03 | | |
| **No. Fungsi** | | - | | |
| **Nama Butir Uji** | | *Pengujian Upload Foto sesuai Kapasitas jumlah dan ukuran maksimal* | | |
| **Tujuan** | | Fungsi ini bertujuan untuk menguji coba apakah field tetap menerima inputan apabila ukuran serta jumlah upload foto melebihi label yang disediakan | | |
| **Deskripsi** | | Uji fungsi *Upload Foto* dilakukan apabila *seller* ingin melakukan upload foto | | |
| **Kondisi Awal** | | Pengguna sudah terhubung pada Aplikasi Marketplace Multi Channel MAGER | | |
| **Tanggal Pengujian** | | 18 Agustus 2022 10:00 WIB | | |
| **Penguji** | | Semua anggota tim | | |
| **Skenario Pengujian** | | | | |
| 1.Pengguna mengakses halaman tambah produk atau edit produk pada Aplikasi Marketplace Multi Channel MAGER.  2.Memilih salah satu *marketplace* Tokopedia ataupun Lazada.  3.Menginput sejumlah foto pada *;* *Upload Foto*. | | | | |
| **Kriteria Evaluasi Hasil** | | | | |
| Foto yang akan dikirimkan oleh pengguna akan dibatasi sesuai dengan jumlah kapasitas dan ukuran maksimal yang dilanjutkan oleh system. | | | | |
| **Kasus dan Hasil Pengujian** | | | | |
| **Data Masukan** | **Yang diharapkan** | | **Pengamatan** | **Kesimpulan** |
| Pengguna menginput file bukan gambar | Inputan foto tidak diterima serta terdapat notifikasi peringatan ‘Image Only’ | | Inputan foto tidak diterima serta terdapat notifikasi peringatan ‘Image Only’ | [x] diterima  [ ] ditolak |
| Pengguna menginput lebih dari 8 gambar | System membatasi jumlah inputan pengguna hingga maksimal 8 gambar | | System membatasi jumlah inputan pengguna hingga maksimal 8 gambar | [x] diterima  [ ] ditolak |
| Pengguna menginput gambar dengan ukuran lebih dari 2MB | System melakukan *crop* terhadap foto yang akan diupload sehingga ukuran file akan otomatis dikurangi | | System melakukan *crop* terhadap foto yang akan diupload sehingga ukuran file akan otomatis dikurangi | [x] diterima  [ ] ditolak |
| **Catatan** | | | | |
| Pengujian kasus berjalan dengan baik sesuai dengan ekspektasi. | | | | |

## **Test Summary Result & History**

Pada bagian ini akan dijelaskan mengenai hasil pengujian yang dilakukan terhadap fungsi

yang terdapat pada Aplikasi Marketplace Multi Channel MAGER

**Scenario**

Tanggal Pengujian : 18 Agustus 2022

Personil : KP-22-111A

Rekapitulasi Hasil : Terlampir pada tabel dibawah

Tabel 40 Test Summary Result & History

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Script** | **Deskripsi Umum Data** | **Kesimpulan** | **Keterangan** |
| BU-01 | Text | Diterima | Hasil uji diterima |
| BU-02 | Text | Diterima | Hasil uji diterima |

# **SW Item Description & Installation**

Bab ini menjelaskan mengenai tahapan yang akan dilakukan dari mulai tahap persiapan sampai hal yang dibutuhkan selama proses instalasi dan demo dari antarmuka *website* Marketplace Multi-channel MAGER.

## **SW Item & Location**

Pada bagian ini dituliskan nama file dan lokasinya yang merupakan komponen pengembangan Aplikasi Pengelolaan Marketplace Multi-channel MAGER. Software ítem mencakup *software executable file, source code, documentation,* dan apa saja yang harus disimpan.

### **Client Side**

Tabel 41 Client Side

| **Directory** | **File** | **Description** |
| --- | --- | --- |
| /resources/views/ | Home.vue | Tampilan awal untuk Aplikasi Pengelolaan Marketplace Multi-channel MAGER dalam bentuk *Dashboard* |
| auth/Register.vue | Tampilan untuk mengelola registrasi akun oleh pengguna |
| auth/RegisterVerification.vue |
| auth/Login.vue | Tampilan untuk mengelola laman *login* dan lupa *password*. |
| auth/LoginForbidden.vue |
| auth/ForgotPassword.vue |
| auth/ResetPassword.vue |
| admin/Edit.vue | Tampilan untuk mengelola data *profile* termasuk didalamnya mengganti *password* & mengganti avatar *profile* |
| sme/marketplace/Marketplace.vue | Tampilan mengelola toko melalui sistem oleh pengguna |
| sme/marketplace/Create.vue |
| sme/marketplace/Edit.vue |
| sme/marketplace/Detail.vue |
| sme/order/Order.vue | Tampilan mengelola data pesanan ke dalam sistem |
| sme/order/Detail.vue |
| sme/product/Product.vue | Tampilan untuk mengelola produk pada toko melalui sistem oleh pengguna |
| sme/product/Create.vue |
| sme/product/CreateProductLazada.vue |
| sme/product/CreateProductTokopedia.vue |
| sme/product/Edit.vue |
| sme/product/EditProductLazada.vue |
| sme/product/EditProductTokopedia.vue |
| sme/product/ModalProduct.vue |
| sme/product/VariasiProduct.vue |
| sme/insightproduct/Insightproduct.vue | Tampilan untuk mengambil data *trend* produk yang dipesan seperti produk mana yang sering dibeli beserta data penjualannya |
|  | sme/incomedetail/Incomedetail.vue | Tampilan untuk menampilkan data pendapatan dari penjualan keseluruhan toko |

### **Server-side**

Tabel 42 Server-side

| **Directory** | **File** | **Description** |
| --- | --- | --- |
| /routes/ | api.php | File yang berisi url beserta method yang dapat  diakses pada *browser* apabila proyek  dijalankan |
| /resources/js/ | router.js |
| /app/Http/Controllers/API | DashboardController.php | Controller yang digunakan untuk  menjalankan setiap fungsi pada sistem. |
| IncomeDetailController.php |
| InsightMarketplaceController.php |
| InsightProductController.php |
| ItemProductController.php |
| LazadaController.php |
| MarketplaceController.php |
| OrderController.php |
| ProductController.php |
| TokopediaController.php |
| /app/Models/SME | SME/Bill.php | Model yang mewakili setiap tabel yang  terdapat pada basis data. Setiap model dapat  digunakan dalam sistem untuk mengakses  tabel dalam basis data dan menjalankan fungsi  pada sistem. |
| SME/PackingInfo.php |
| SME/ProductTransaction.php |
| SME/IncurredFee.php |
| SME/Payout.php |
| SME/Transaction.php |
| SME/Product.php |
| SME/ProductImage.php |
| SME/ProductSku.php |
| SME/ProductSkuImage.php |

## **SW Installation**

Tidak ada tahapan instalasi dari website Marketplace Multi-channel MAGER.

## **Precondition**

Tidak ada syarat khusus terkait aplikasi ini selain terkoneksi ke internet, namun disarankan mengakses aplikasi ini menggunakan Laptop atau Komputer karena aplikasi ini berbasis *website*.

## **Procedures (Working Instruction)**

Untuk dapat menggunakan aplikasi ini, pengguna cukup mendaftarkan akun dan mengakses *website* Marketplace Multi-channel MAGER melalui *link* berikut ini <https://cobamager.tripasysfo.com/>

## **Reporting**

Setelah semua prosedur selesai dilakukan maka antarmuka website Marketplace Multi-channel MAGER dapat diakses dengan *web browser*. Jika terjadi kesalahan (*error*) maka personal yang dapat dihubungi adalah PT Tripasysfo Development melalui [contact@tripasysfo.com](mailto:contact@tripasysfo.com) atau (+62) 87899717801

# **LAMPIRAN**

Jika ada lampiran lain yang perlu disertakan, sertakan.

# **Sejarah Versi**

*Pada bagian ini, dijelaskan semua versi yang pernah di-deliver, dan ciri serta perubahannya. Untuk Kerja Praktik ini, minimal ada dua versi : versi pada saat presentasi I, dan versi final.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versi** | **Ditulis Oleh** | **Tanggal** | **Disetujui Oleh** | **Tanggal** |
| Draft |  |  | Supervisor |  |
| Final |  |  | Pembimbing ITB |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **Sejarah Perubahan**

*Bagian ini memuat sejarah perubahan dokumen (no. versi terbaru dibandingkan versi sebelumnya).*

**No. dokumen :**

**No. versi :**

| **Halaman** | **Semula** | **Menjadi** | **Alasan perubahan** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |