

BAB 2

TINJAUAN PUSTAKA

Dalam pembangunan *e-commerce multichannel management* untuk mengelola produk pada toko online yang ditunjukkan untuk WADEZIG! ini dibutuhkan tinjauan pustaka yang dapat membantu dalam proses pembuatan aplikasi beserta objek-objek yang diperlukan dalam pembuatan aplikasi ini. Berikut beberapa tinjauan pustaka yang digunakan dalam pembuatan aplikasi ini.

2.1 Profil WADEZIG!

Dalam penelitian ini yang menjadi objek penelitian adalah WADEZIG!. Di mana perusahaan ini merupakan perusahaan yang menjual pakaian dan barang-barang berkualitas tinggi. Berikut ini merupakan profil dari WADEZIG!:

Nama	: WADEZIG!
Alamat kantor	: Jl. Permata Taman Sari Raya kav. 7, Arcamanik, Bandung.
Telp. kantor	: +62.22.63722722
Alamat toko offline	: Jl. Trunojoyo No. 4, Bandung.
Telp. Toko offline	: +62.22.20503948
Jam operasional	: 09.00 – 21.00 WIB
Email	: info@wadezig.com

2.1.1 Sejarah Singkat Perusahaan

WADEZIG! merupakan perusahaan yang bergerak pada penjualan produk. Produk yang dijual sendiri sangat beragam mulai dari *t-shirt*, celana, sepatu, topi, hoodie, jaket dan lain-lain. Tidak hanya menjual, namun mulai dari proses merancang, memproduksi dan memasarkan pun dilakukan oleh WADEZIG!.

Perusahaan ini didirikan pada awal tahun 2003 di Bandung, Indonesia. WADEZIG! merupakan merek yang lahir dari kegelisahan tiga pemuda. Mereka

memiliki passion yang sama dan kecintaan yang dalam pada seni, yang membuat mereka selalu haus akan sesuatu yang baru, sesuatu yang berbeda. Namun mereka tidak pernah dapat menemukan apa yang mereka cari. Kebutuhan akan tempat untuk mengekspresikan dan menyuarakan ide juga merupakan faktor fundamental yang mendorong terciptanya WADEZIG!.

Lahir, tumbuh dan berakar pada komunitas seni dan desain, membuat WADEZIG! terkait dan melekat secara pribadi pada bidang seni dan desain. Ini adalah faktor penting dalam pengembangan gaya dan sikap WADEZIG! kritis, terkadang menyenangkan, tetapi selalu cerdas. Itu juga membuat WADEZIG! sejak dulu selalu melihat sesuatu dari sudut pandang subjektif dan personal, sehingga segala pemikiran dan ide yang diberikan selalu terlihat berbeda. Belakangan, itu menjadi salah satu kekuatan grup. Selain itu, ia tumbuh dalam komunitas di mana berbagai macam ide digabungkan, membuat WADEZIG! selalu bereksperimen dalam memadukan berbagai macam ide untuk menciptakan sesuatu yang baru.

Hal tersebut memberi pengaruh yang sangat kuat terhadap produknya. Yang mana setiap produk dari WADEZIG! memiliki ciri khas semangat WADEZIG! Yang merupakan benturan ide-ide yang bertentangan untuk memberikan ide-ide baru. Dari segi desain grafis, WADEZIG! juga merupakan benturan ide, seperti perpaduan desain klasik yang dihadirkan dengan sentuhan teknologi modern, atau perpaduan desain Eropa yang apik dan sangat teoritis dengan gaya desain grunge yang penuh distorsi dan terkesan 'anti skill'.

Produk merek tersebut saat ini tersedia di berbagai outlet distribusi di hampir semua kota besar di Indonesia dan beberapa kota di luar negeri. Selain WADEZIG ini! memiliki toko Offline di Bandung. Kami juga memutuskan untuk masuk ke pasar Online dan kami membuka toko web di situs web kami www.wadezig.com dan membuat halaman penggemar di jejaring sosial seperti Facebook dan Twitter.

Untuk mengamankan dan memperkuat akarnya di komunitas seni dan desain, merek ini secara teratur bertemu dan berkolaborasi dengan para ahli di bidang ini seperti seniman, fotografer, videografer, ilustrator, desainer grafis, musisi, dan banyak lagi. Pertemuan dan kolaborasi ini menuntun WADEZIG! untuk

menciptakan konsep baru dalam merek dan komunitas lokal: “WADEZIG! Artist Series”. Secara teratur mereka memutuskan untuk membuat dan memproduksi pakaian dengan artis dalam dan luar negeri untuk mendukung dan menghargai mereka. WADEZIG! juga aktif dalam komunitas lokal dengan mengadakan dan / atau mendukung berbagai pameran seni dan desain grafis, terutama di kancah lokal.

Setelah 10 tahun, WADEZIG! tidak lagi menjadi '*personal playground*' bagi para pendirinya, tetapi menjadi '*personal playground*' bagi semua orang dengan visi bersama yang sama.

2.2 E-commerce

E-commerce atau perdagangan elektronik adalah penyebaran, jual beli, serta pemasaran barang dan jasa yang berlangsung melalui sistem elektronik. Contoh sistem elektronik yang dipakai adalah internet. Menurut Shely Cahsman (2007:83) “E-commerce merupakan transaksi bisnis yang terjadi dalam jaringan elektronik, seperti internet”. Oleh karena itu dapat disimpulkan bahwa siapa pun yang memiliki sambungan ke internet, maka dapat berpartisipasi dalam kegiatan *e-commerce* ini.

Pada dasarnya *e-commerce* juga berlangsung sebagaimana perdagangan tatap muka (tanpa media elektronik) yang mana meliputi proses promosi, pemasaran, penjualan, pembelian, pembayaran dan penyerahan produk dari penjual ke pembeli (bila di *e-commerce* biasanya dikirim menggunakan jasa pengiriman). Yang berbeda di sini adalah sistem yang digunakan, di mana *e-commerce* menggunakan media elektronik seperti internet. Sehingga bisa disimpulkan bahwa *e-commerce* ini didorong oleh teknologi internet dan teknologi informasi [4].

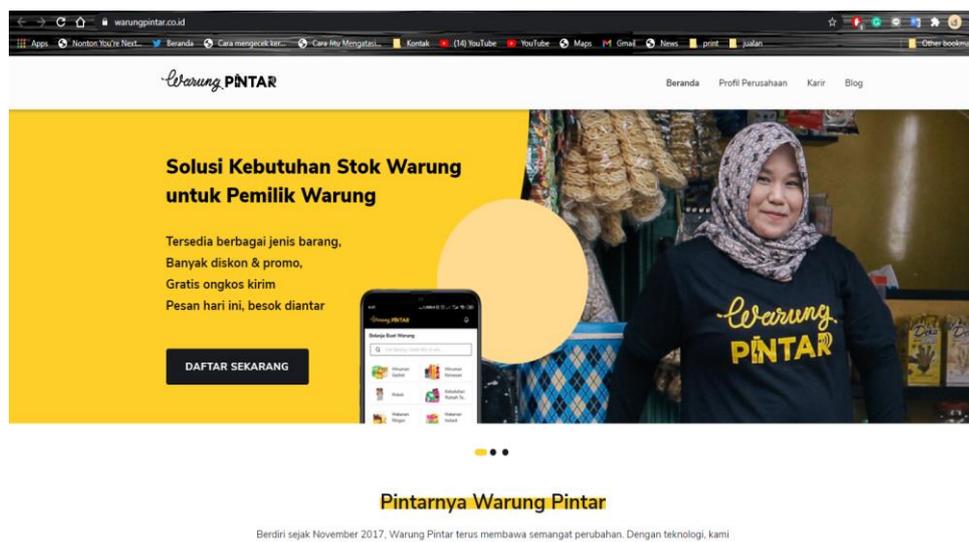
Banyak keuntungan yang didapat ketika menggunakan *e-commerce*. Contohnya adalah dengan menggunakan *e-commerce*, maka penjual dapat meningkatkan keuntungan yang diperoleh dengan biaya yang lebih murah [5]. Selain itu *e-commerce* juga membuat semua proses yang ada menjadi lebih mudah. Sehingga tak aneh bila *e-commerce* digunakan oleh banyak orang saat ini baik penjual maupun pembeli.

2.2.1 Jenis-Jenis E-commerce

Setidaknya ada tujuh jenis e-commerce yang memiliki bentuk bisnis dan karakteristik yang berbeda. Berikut adalah tujuh jenis e-commerce yang ada [6]:

1. Business-to-Business (B2B)

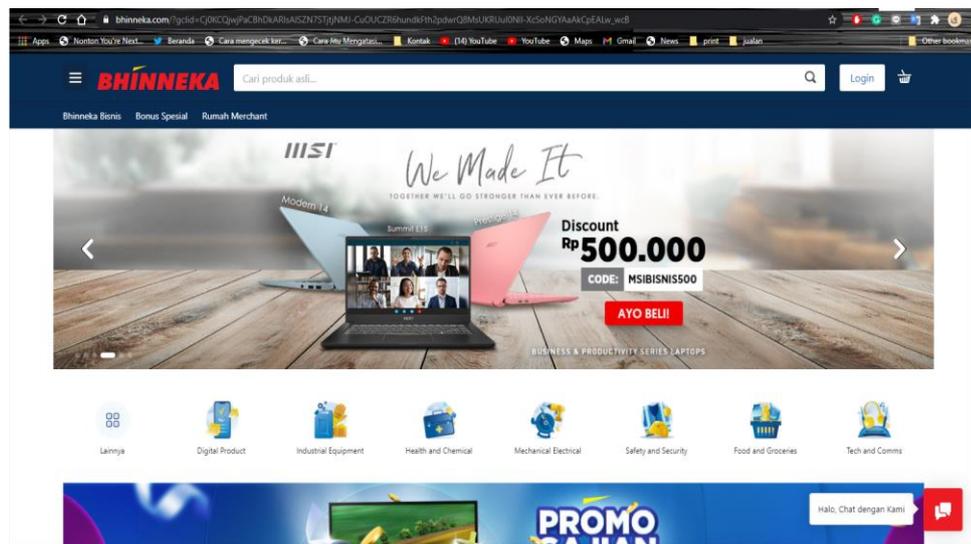
E-commerce jenis ini meliputi semua transaksi yang dilakukan antar perusahaan. Yang mana jenis *e-commerce* ini biasa digunakan oleh produsen dan pedagang tradisional. Umumnya e-commerce jenis ini menerapkan EDI (Electronic Data Interchane) dan email dalam proses pembelian barang sana jasa, informasi dan kolsultasi juga pengiriman dan permintaan proposal bisnis. Contoh dari *e-commerce* ini adalah Bizzy yang sekarang bergabung dengan Warung Pintar.



Gambar 2.1 Contoh Jenis E-Commerce B2B

2. Business-to-Consumer (B2C)

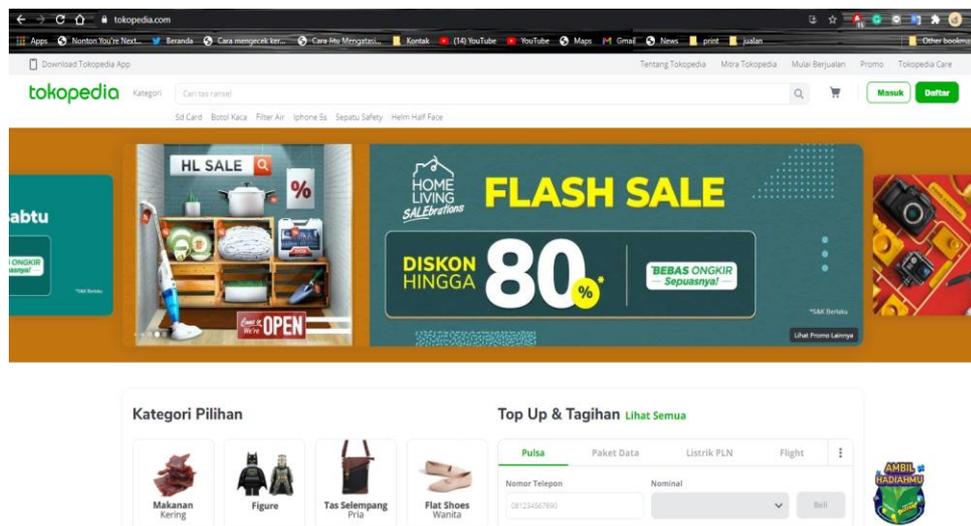
B2C Adalah jenis E-commerce antara perusahaan dan konsumen akhir. Singkatnya seperti bagian ritel dari e-commerce yang biasa dioperasikan oleh perdagangan ritel tradisional. Jenis ini berkembang dengan cepat karena munculnya website serta toko virtual bahkan mal di internet yang menjual beragam kebutuhan masyarakat. Contoh dari e-commerce ini adalah Tiket.com dan Bhinneka.



Gambar 2.2 Contoh Jenis E-Commerce B2C

3. Consumer-to-Consumer (C2C)

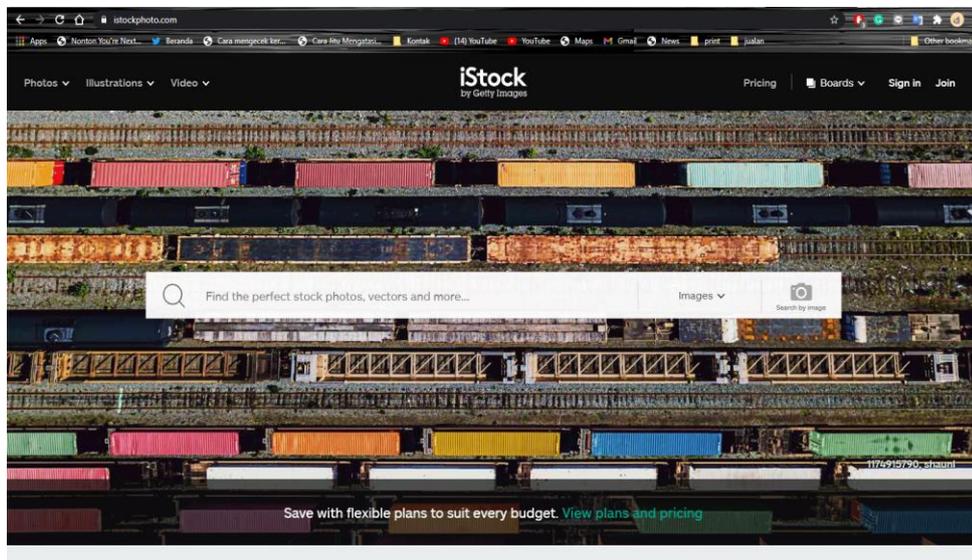
C2C merupakan jenis e-commerce yang meliputi semua transaksi antar konsumen. Umumnya transaksi ini dilakukan melalui pihak ketiga yang menyediakan *platform online* untuk melakukan transaksi tersebut. Contoh dari e-commerce ini adalah Tokopedia, Shopee, Bukalapak dan lain-lain.



Gambar 2.3 Contoh Jenis E-Commerce C2C

4. Consumer-to-Business (C2B)

C2B adalah jenis e-commerce yang menerapkan pembalikan utuh dari transaksi pertukaran barang secara tradisional. Dasar dari e-commerce ini multi sumber daya. Artinya dalam e-commerce ini terdapat sekelompok besar individu menyediakan layanan jasa atau produk mereka bagi perusahaan atau orang yang mencari jasa dan produk tersebut. Yang mana nantinya akan dipilih salah satu yang dianggap paling cocok dan efektif dengan apa yang diinginkan oleh pembeli atau perusahaan tersebut. Contoh dari e-commerce ini adalah Istock.



Gambar 2.4 Contoh Jenis E-Commerce C2B

5. Business-to-Administration (B2A)

B2A adalah jenis e-commerce yang mencakup semua transaksi yang dilakukan secara online antara perusahaan dan administrasi publik. Jenis e-commerce ini melibatkan banyak layanan, seperti di bidang jaminan sosial, dokumen dan register, dan lain-lain. Contoh dari e-commerce ini adalah www.pajak.go.id.



Gambar 2.5 Contoh Jenis E-Commerce B2A

6. Consumer-to-Administration (C2A)

C2A merupakan e-commerce yang didalam meliputi semua transaksi yang dilakukan antara individu dan administrasi publik. Contoh di bidang pendidikan adalah penyebaran informasi, proses pembelajaran jarak jauh dan lain-lain. Contoh di bidang kesehatan adalah janji pertemuan, informasi mengenai penyakit, pembayaran layanan kesehatan dan lainnya. B2A dan C2A pada dasarnya sama saja, pembedanya hanya terletak pada pihak yang bersangkutan, di mana pada B2A adalah perusahaan-administrasi publik dan C2A adalah pihak individu-administrasi publik.

7. Online-to-offline (O2O)

O2O adalah jenis e-commerce yang menarik pelanggan dari saluran online untuk menuju toko fisik. Contoh dari jenis ini adalah memasang iklan di internet untuk mendapatkan pelanggan di toko fisik atau juga melakukan pemesanan via online namun tetap diambil secara langsung ke toko fisik. Contoh penggunaan e-commerce ini adalah Kudo, MatahariMall, Walmart dan lain-lain.

2.3 Marketplace

Marketplace adalah sebuah media online yang berguna sebagai tempat jual beli dan dapat diakses menggunakan perangkat terhubung dengan internet. Oleh karena itu marketplace merupakan model *E-Business* yang berhubungan dengan penjual dan pembeli. Tidak hanya sebagai tempat jual beli, *marketplace* juga memfasilitasi pengguna mereka dengan fasilitas rekening bersama. Dengan begitu, transaksi yang dilakukan di *marketplace* tersebut akan mudah diawasi dan diminimalisir tingkat kecurangan atau kejahatan yang mungkin terjadi.

Di Indonesia sendiri terdapat banyak *marketplace* saat ini. Lima urutan teratas yang mendapat banyak pengunjung perbulan terbanyak pada periode Juli 2020 adalah Shopee, Tokopedia, Bukalapak, Lazada dan Tokopedia. Di mana datang yang diambil pada bulan tersebut adalah Shopee berada di peringkat pertama dengan total pengunjung sebanyak 96.460.000 perbulan, lalu Tokopedia sebanyak 86.410.000 pengunjung perbulan, Bukalapak sebanyak 32.760.000 pengunjung perbulan, Lazada sebanyak 22.220.000 pengunjung perbulan dan BliBli sebanyak 18.340.000 pengunjung perbulan [1].

Penggunaan *marketplace* dapat membantu penjual dalam memasarkan produknya ke lebih banyak pembeli yang tidak hanya berada di daerah sekitar penjual, namun di luar daerah juga [7]. Sedangkan untuk pembeli, penggunaan *marketplace* dapat membantu mereka untuk menemukan dan memilih apa yang mereka inginkan dengan lebih mudah.

2.4 Multichannel Management

Multichannel management adalah bentuk strategi pengelolaan. Berasal dari bahasa inggris, yakni multichannel (banyak saluran) dan management (pengelolaan). Sehingga bisa diartikan bahwa bentuk pengelolaan ini membuat pengguna dapat mengelola produk yang tersebar di banyak *channel* penjualan secara bersamaan. Dengan begitu pengguna bisa lebih mudah dalam mengatur produk yang ia miliki dan tentunya waktu yang diperlukan dalam mengelola produk juga menjadi lebih efisien.

Biasanya bentuk pengelolaan ini berjalan dengan cara mengintegrasikan beberapa *channel* penjualan ke dalam sebuah sistem yang sudah disiapkan. Dengan melakukan integrasi tersebut, maka pengguna di dalam sistem tersebut dapat mengelola produk yang tersebar di banyak *channel* penjualan dengan lebih mudah, karena sistem tersebut sudah menyediakan pengelolaan untuk beberapa *channel* penjualan dan perubahan yang ada di suatu *channel* penjualan, dapat berpengaruh ke *channel* penjualan yang lain (yang saling terintegrasi).

2.5 Toko Online

Toko online adalah toko yang memasarkan produknya secara online melalui internet. Berbasis online membuat toko tersebut tidak perlu memiliki bangunan toko asli dan tidak perlu tatap muka antar penjual dan pembeli [8]. Oleh karena itu siapapun yang memiliki internet, pada dasarnya bisa membuka toko online.

Penggunaan toko online mampu mengatasi beberapa masalah yang dialami penjual, seperti mengatasi keterbatasan pemasaran produk, mengatasi keterbatasan modal, mengatasi pengawasan barang dan pendataan yang masih manual sehingga akan menghambat pembuatan laporan dan lain-lain [9].

Dengan segala kemudahan dan keunggulannya, penggunaan toko online ini diminati oleh banyak orang. Baik itu peorangan ataupun perusahaan. Baik itu perusahaan kecil maupun besar. Bahkan banyak penjual yang menutup toko fisik mereka dan beralih ke toko online sebab pengeluaran yang diperlukan lebih murah.

2.6 Internet

Internet berasal dari kata *interconnected-networking*. Yang mana bisa diartikan sebagai rangkaian jaringan pada perangkat yang membuat perangkat tersebut saling terhubung dan jangkauan jaringannya bersifat global yang yang menghubungkan suatu jaringan dengan jaringan lainnya di seluruh dunia [9].

Media yang digunakan untuk menghubungkan jaringan tersebut bisa berupa kabel, kanal satelit dan lain-lain. Dalam menjalankan tugasnya, jaringan internet ini bekerja sesuai dengan protokol atau aturan. Protokol standar yang digunakan untuk

menghubungkan jaringan-jaringan di dalam internet sehingga bisa saling bertukar data disebut TCP/IP atau *Transmission Control Protocol Internet Protocol*. Agar setiap perangkat yang tersambung ke internet memiliki ciri atau identitasnya sendiri, maka setiap perangkat yang terhubung ke internet akan diberikan suatu nomer unik yang disebut dengan alamat IP [10].

2.7 Web

Web atau *world wide web* adalah suatu layanan informasi yang menggunakan konsep tautan (hyperlink) dan dapat diakses menggunakan perangkat yang terhubung ke internet melalui URL web tersebut [9]. Saat ingin mengakses sebuah web, pengguna hanya memerlukan web browser (seperti Chrome, Firefox, Microsoft Edge dan lain-lain) dan juga perangkat yang sudah terhubung ke internet tanpa perlu instal aplikasi lainnya. Hal tersebut memberikan kemudahan bagi para pemakai internet dalam mencari informasi yang mereka butuhkan.

Website pertama kali dibuat oleh Tim Berners Lee pada akhir tahun 1980 dan secara resmi online 11 tahun setelahnya, yaitu pada tahun 1991 [11]. Tujuan awal pembuatan web ini pada awalnya adalah untuk memudahkan pekerjaan para peneliti di tempatnya saat akan bertukar atau melakukan perubahan informasi. Dan seperti yang kita tahu sekarang bahwa web merupakan sebuah layanan yang berkembang dengan pesat. Bahkan sekarang banyak web yang sudah *responsive* (tampilan yang dimuat akan selalu menyesuaikan sekalipun diakses lewat perangkat yang berbeda).

Terdapat 3 unsur yang sangat penting pada sebuah web. Tanpa unsur ini, web tersebut tidak akan bisa ditemukan dan diakses sekalipun pengguna memiliki web browser dan perangkat yang terhubung ke internet. Unsur-unsur penting tersebut adalah sebagai berikut [11]:

1. Domain

Domain adalah alamat yang dimiliki oleh sebuah web. Berguna sebagai rujukan untuk dapat mengakses web yang diinginkan. Nama domain dari setiap web dibuat secara unik dan khas. Artinya setiap web memiliki domain yang berbeda.

2. Hosting

Hosting adalah tempat penyimpanan database (script, gambar, video dan lain-lain) yang dimiliki oleh web tersebut. Setiap web yang dibuat pada awalnya bersifat offline (tidak bisa diakses oleh orang lain) sebelum akhirnya dihosting dan bersifat online.

3. Konten

Sebuah web harus selalu memiliki konten di dalamnya. Sebab sebuah web harus memiliki tujuan yang jelas. Bila tidak ditemukan konten di dalamnya, maka apa tujuan dari web tersebut? Karena umumnya setiap web selalu memiliki konten, baik itu yang beraneka ragam maupun yang itu-itu saja. Konten pada sebuah website dapat berupa teks, gambar, video dan lain-lain.

2.8 PHP

PHP atau Hypertext Preprocessor adalah bahasa pemrograman yang sudah digunakan secara luas yang mana dapat disisipkan ke dalam HTML dan sering dipakai untuk membuat web dinamis [12]. Bahasa pemrograman ini pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Saat itu PHP masih bernama Form Interpreted (FI) yang berwujud sekumpulan skrip untuk mengolah data formulir data dari web. Dan hingga kini PHP masih terus berkembang. Versi terbaru dari PHP adalah PHP 8.

Tujuan dari bahasa pemrograman ini adalah untuk membuat aplikasi yang dapat memberikan hasil pada web browser, tapi secara keseluruhan prosesnya dijalankan di server [13]. Sehingga bahasa pemrograman ini bersifat server-side programming. Hal ini membuat sintaks dan perintah yang diberikan akan sepenuhnya berjalan di server dan disertakan pada halaman HTML, tapi script atau sintaks tersebut tidak akan tampak di sisi *client* [9]. PHP juga bersifat *multiplatform*, artinya dapat dijalankan di berbagai platform (dalam hal ini sistem operasi) seperti Linux, Windows, Mac Os dan lain-lain. Selain itu karena PHP berjalan di server-side programming, maka diperlukan juga sebuah web server simulator seperti Apache [8]. Dari segi database, PHP juga bisa terhubung dengan berbagai macam database seperti MySQL, ORACLE, PostgreSQL dan lain-lain.

PHP sudah digunakan oleh banyak orang hingga sekarang, sehingga tidak perlu khawatir ketika menemukan kesulitan dalam mempelajari bahasa pemrograman ini. Sebab dukungan *technical-support* dan komunitas PHP yang tersebar di mana-mana akan memudahkan kita ketika mencari solusi atas masalah yang terjadi.

Sebagai bahasa pemrograman, tentunya terdapat beberapa aturan. Beberapa aturan tersebut diantaranya adalah harus ditulis diantara tag `<?>` atau `<?php` dan `?>`. Selain itu peraturan yang terkenal adalah setiap satu statement (perintah) biasanya diakhiri dengan tanda titik koma (;). Sehingga ketika menggunakan bahasa ini harus selalu ingat titik koma. Dan masih banyak lainnya aturan mengenai penulisan pada bahasa pemrograman PHP ini.

2.9 MySQL

MySQL adalah salah satu RDBMS (*relationship database management system*) yang populer dan sering digunakan. Dalam pengertiannya MySQL adalah perangkat lunak yang berfungsi sebagai sistem manajemen basis data SQL yang mendukung hubungan antar tabel dan bersifat *open source* [14]. Karena bersifat open source, maka MySQL memberikan kebebasan bagi penggunanya untuk menginstall, menggunakan dan bahkan melakukan perubahan.

Selain bersifat open source dan gratis, beberapa kelebihan dari MySQL adalah bersifat fleksibel dengan berbagai bahasa pemrograman, mudah dipelajari dan digunakan, performa yang baik, memiliki standar industri, keamanan yang tinggi dan lain-lain. Untuk sistem operasinya sendiri MySQL dapat dijalankan di berbagai macam sistem operasi seperti Windows, Linux, macOS dan lain-lain.

Salah satu bahasa pemrograman yang cocok untuk dipakai dengan MySQL adalah PHP [15]. Selain itu MySQL juga sudah menjadi RDBMS yang digunakan sejak lama, sehingga pada penelitian kali ini pun saya memilih MySQL sebagai RDBMS untuk aplikasi yang akan saya buat.

2.10 Laravel

Laravel adalah *framework* berbahasa PHP yang bersifat open source dan gratis dan ditunjukkan untuk pembuatan aplikasi berbasis web [16]. Dengan menggunakan Laravel, penggunaan bahasa pemrograman PHP menjadi lebih powerfull. Karena dalam penggunaannya menjadi lebih simpel, mudah dan aman. Sehingga saat ini Laravel menjadi framework PHP yang populer dan digunakan oleh banyak orang.

Laravel dibuat oleh Taylor Otwell dengan tujuan sebagai *framework* yang dapat membantu pengembangan aplikasi berbasis web dengan konsep atau pola arsitektur *model-view-controller* (MVC) [16]. Konsep tersebut membuat aplikasi yang dibuat menjadi lebih terstruktur, jelas, sederhana dan tentunya memberikan kemudahan bagi pengembang dalam menciptakan aplikasi yang diinginkan.

Contoh beberapa fitur dari Laravel yang memberikan kemudahan pengembangan aplikasi berbasis web adalah *Template Engine*, *Routing*, *Authentication* dan *Modularity*. Selain itu juga terdapat banyak fungsi yang dapat digunakan untuk berinteraksi dengan *database* seperti mengambil semua data, menyaring suatu data dengan kata kunci yang diinginkan dan lain-lain.

Dengan segala kelebihanannya, tidak heran Laravel memiliki jargonnya sendiri, yaitu “*The PHP Framework For Web Artisans*”. Karena dengan memakai Laravel, pengembangan aplikasi berbasis web menjadi lebih elegan, ekspresif dan menyenangkan.

2.11 API

Application Programming Interface atau API adalah suatu dokumentasi yang dibuat oleh pengembang dan biasanya berupa antarmuka, fungsi, kelas dan lain sebagainya yang bisa digunakan untuk mempermudah pengembang dalam membuat suatu aplikasi [17].

Contoh kemudahan yang diperoleh pengembang bila menggunakan API adalah mendapatkan fitur-fitur dari API yang dipakai, sehingga proses pengembangan aplikasi tidak benar-benar dari awal, karena ada fitur yang sudah

tersedia lewat API yang dipakai. Selain itu API juga dapat menghubungkan satu aplikasi dengan aplikasi lainnya [18].

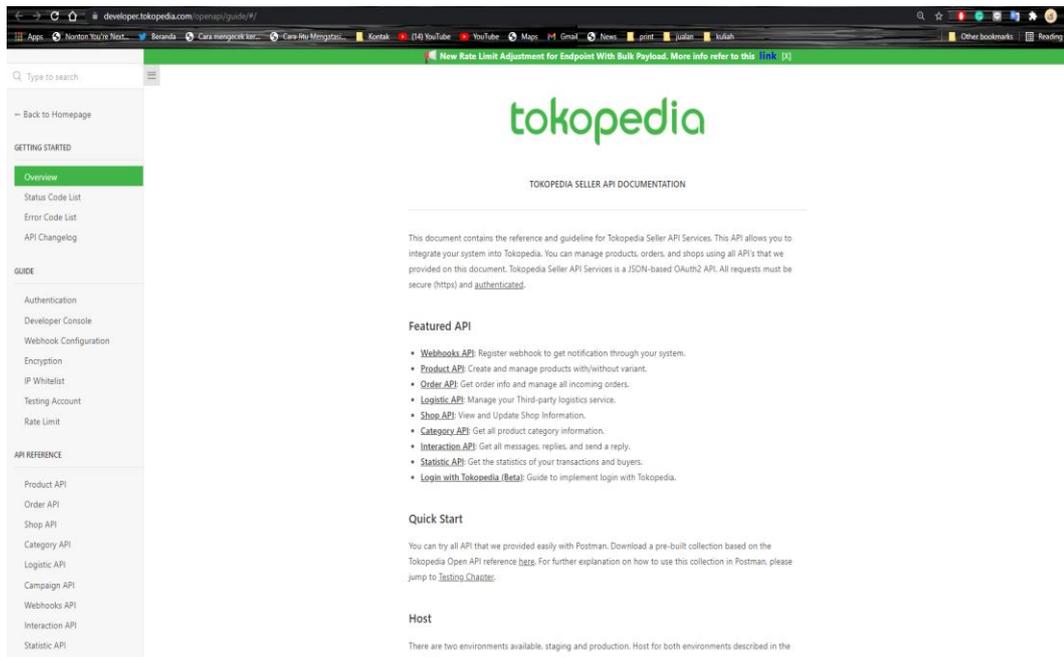
Terdapat banyak API yang tersedia secara publik khususnya di Indonesia. Contoh dari beberapa API tersebut adalah MatahariMall API, Tiket.com API, Jurnal API, Midtrans – Payment Gateway, Ipaymu, Bisatopup, Raja Ongkir, Cek Resi, Kaskus API, Ayo Donor PMI, Halal Product MUI dan lain-lain [19].

2.12 Tokopedia API

Tokopedia API adalah API yang disediakan oleh Tokopedia untuk membantu pedagang dalam mengelola toko mereka khususnya untuk pedagang besar yang menjadi mitra Tokopedia.

Beberapa pedagang (terutama pedagang besar) biasanya memiliki gudang baik satu maupun lebih dari satu dan sudah memiliki sistem inventori sendiri. Hal tersebut menimbulkan masalah di mana terdapat dua sistem yang berbeda untuk mengelola satu toko. Terlebih dari sisi perawatan juga tidak akan efisien jika harus merawat dua sistem untuk toko yang sama. Agar bisa memudahkan para pedagang dalam mengelola usaha mereka, dibuatlah API ini. Di mana fungsi utama dari API ini adalah agar sistem yang sudah dimiliki toko tersebut bisa terintegrasi dengan sistem di Tokopedia.

Untuk dapat memakai API yang disediakan oleh Tokopedia, toko tersebut harus berstatus sebagai *power merchant* atau *official store*. Lalu jika status dari toko tersebut sudah sesuai dengan syarat yang diinginkan Tokopedia, maka langkah selanjutnya adalah menyambungkan toko tersebut sistem atau aplikasi yang akan dibuat oleh pengembang. Untuk dokumentasinya sendiri bisa diakses pada link <https://developer.tokopedia.com/openapi/guide/#/>. Dokumentasi pada Tokopedia API dapat dilihat pada Gambar 2.6.



Gambar 2.6 Dokumentasi Tokopedia API

Dalam dokumentasi tersebut terdapat banyak API yang bisa dimanfaatkan. Berikut adalah fitur-fitur yang bisa dan akan dimanfaatkan dalam sistem yang dibangun memakai Tokopedia API, yaitu:

- 1) Membuat atau menambahkan produk
- 2) Lihat Produk
- 3) Mengubah Produk
- 4) Hapus Produk
- 5) *Webhook Notifcation*

Agar dapat menjalankan fitur-fitur yang sudah di atas, maka diperlukan akses ke endpoint-endpoint yang diperlukan. Berikut adalah penjelasan lebih detail mengenai endpoint yang dipakai:

a. Authentication

Berikut adalah endpoint untuk melakukan authentication agar memperoleh token dari Tokopedia API. Contoh request dan contoh responsnya.

```

bash
curl -X POST \
  'https://accounts.tokopedia.com/token?grant_type=client_credentials' \
  -H 'Authorization: Basic Y2xpZW50X2lkOmNsawVudF9zZWNYZXQK' \
  -H 'Content-Length: 0' \
  -H 'User-Agent: PostmanRuntime/7.17.1'

```

Gambar 2.7 Contoh Request Authentication Tokopedia API

```

json
{
  "access_token": "LSP7x7sRGaewzWZE6IcuA",
  "expires_in": 2592000,
  "token_type": "Bearer"
}

```

Gambar 2.8 Contoh Respon Authentication Tokopedia API

b. Endpoint Get All Categories

Berikut adalah endpoint dari get all categories beserta parameter, contoh request dan contoh responsenya.

```
GET /inventory/v1/fs/:fs_id/product/category
```

Gambar 2.9 Endpoint Get All Categories

Parameter	Type	Required?	Location	Description
fs_id	integer	required	path	Fulfillment service id.
keyword	string	optional	query	Keyword or Product Name to get recommended category. Leave it blank to get full categories.

Gambar 2.10 Parameter Get All Categories

```

curl -X GET \
  https://fs.tokopedia.net/inventory/v1/fs/13004/product/category?keyword=Tas%20Sekolah%20
  -H 'Authorization: Bearer OoHgyxcgQ5a1uCmbbXjJA'

```

Copy to clipboard

Gambar 2.11 Contoh Request Get All Categories

```

{
  "header": {
    "process_time": 0.374549464,
    "messages": "Successfully Retrieve Category",
    "reason": "",
    "error_code": 0
  },
  "data": {
    "categories": [
      {
        "name": "fashion-anak / Tas Anak / Tas Backpack Anak",
        "id": "1979"
      },
      {
        "name": "fashion-anak / Tas Anak / Tas Selempang Anak",
        "id": "1978"
      },
      {
        "name": "fashion-anak / Tas Anak / Tas Koper Anak",
        "id": "1980"
      }
    ]
  }
}

```

Gambar 2.12 Contoh Response Sukses Get All Categories

```

<html>
  <head>
    <title>401 Authorization Required</title>
  </head>
  <body bgcolor="white">
    <center>
      <h1>401 Authorization Required</h1>
    </center>
    <hr>
    <center>nginx</center>
  </body>
</html>

```

Gambar 2.13 Contoh Response Error Get All Categories

c. Endpoint Get All Variants by Category ID

Berikut adalah endpoint dari get all variants by category id beserta parameter, contoh request dan contoh responsenya.

```
GET /inventory/v1/fs/:fs_id/category/get_variant?cat_id=:cat_id
```

Gambar 2.14 Endpont Get All Variants by Category ID

Parameter	Type	Required?	Location	Description
fs_id	integer	required	path	Fulfillment service id.
cat_id	integer	required	path	Category id.

Gambar 2.15 Parameter Get All Variants by Category ID

```
curl -X GET \
  'https://fs.tokopedia.net/inventory/v1/fs/13004/category/get_variant?cat_id=1769' \
  -H 'Authorization: Bearer HB8GNgfdRXS9c_sksTdTBw'
```

Gambar 2.16 Contoh Request Get All Variants by Category ID

```
{
  "header": {
    "process_time": 0.755010807,
    "messages": "Your request has been processed successfully"
  },
  "data": [
    {
      "variant_id": 6,
      "name": "Ukuran Pakaian",
      "identifier": "size",
      "status": 1,
      "has_unit": 1,
      "units": [
        {
          "unit_id": 7,
          "name": "International",
          "short_name": "Intl",
          "values": [
            {
              "value_id": 22,
              "value": "XXS",
              "hex_code": "",
              "icon": ""
            },
            {
              "value_id": 23,
              "value": "XS",
              "hex_code": "",
              "icon": ""
            }
          ]
        }
      ]
    }
  ]
}
```

Gambar 2.17 Contoh Respon Sukses Get All Variants by Category ID

```

{
  "header": {
    "process_time": 0.000048264,
    "messages": "We could not process your request due to malformed request, please check the request format",
    "reason": "invalid fs_id format",
    "error_code": "VRT_DLV_002"
  },
  "data": null
}

```

Gambar 2.18 Contoh Respon Error Get All Variants by Category ID

d. Endpoint Get All Variants by Product ID

Berikut adalah endpoint dari get all variants by product id beserta parameter, contoh request dan contoh responsnya.

```
GET /inventory/v1/fs/:fs_id/product/variant/:product_id
```

Gambar 2.19 Endpoint Get All Variants by Product ID

Parameter	Type	Required?	Location	Description
<code>fs_id</code>	integer	required	path	Fulfillment service id.
<code>product_id</code>	integer	required	path	Product id.

Gambar 2.20 Parameter Get All Variants by Product ID

```

curl -X GET \
  'https://fs.tokopedia.net/inventory/v1/fs/13004/product/variant/15240600' \
  -H 'Authorization: Bearer HB8GNgfdRXS9c_sksTdTBw'

```

Gambar 2.21 Contoh Request Endpoint Get All Variants by Product ID

```
{
  "header": {
    "process_time": 0.92548774,
    "messages": "Your request has been processed successfully"
  },
  "data": {
    "parent_id": 15330203,
    "default_child": 15330204,
    "sizechart": "",
    "variant": [
      {
        "name": "warna",
        "identifier": "colour",
        "unit_name": "",
        "position": 1,
        "option": [
          {
            "id": 41368,
            "value": "Hijau",
            "hex": "#006400"
          },
          {
            "id": 41369,
            "value": "Merah",
            "hex": "#ff0016"
          }
        ]
      },
      {
        "name": "ukuran",
        "identifier": "size",
        "unit_name": "Default",
        "position": 2,
        "option": [
          {
            "id": 41371,
            "value": "XL",
            "hex": ""
          }
        ]
      }
    ]
  }
}
```

Gambar 2.22 Contoh Respon Sukses Get All Variants by Product ID

```

{
  "header": {
    "process_time": 0.903005513,
    "messages": "We could not process your request due to malformed request, please che",
    "reason": "Failed Send GRPC Request",
    "error_code": "PRD_GRPC_001"
  },
  "data": null
}

```

Gambar 2.23 Contoh Respon Gagal Endpoint Get All Variants by Product ID

- e. Endpoint Get All Etalase
- f. Endpoint Create Product V2

Berikut adalah endpoint dari create product v2 beserta parameter, contoh request dan contoh responsenya.

```
POST /v2/products/fs/:fs_id/create
```

Gambar 2.24 Endpoint Create Product V2

Parameter	Type	Required?	Location	Description
fs_id	integer	required	path	Fulfillment service unique identifier
shop_id	integer	required	path	Shop unique identifier
products	object array	required	body	Product object array to create.

Gambar 2.25 Gambar Parameter dari Endpoint Create Product V2

Parameter	Type	Required?	Location	Description
<code>name</code>	string	required	body	Name of the product with length less than or equals 70 characters.
<code>category_id</code>	integer	required	body	Unique identifier for the product's category. To get available categories, please check Get All Categories . Please input the <i>deepest category child ID</i> .
<code>price_currency</code>	string	required	body	Currency code for stated price (IDR or USD)
<code>price</code>	integer	required	body	The possible value between 100 to 100.000.000. If the product variant is added, the price parameter is automatically set to the lowest price among the variant products.
<code>status</code>	string	required	body	Status for the product with the following available values <code>UNLIMITED</code> , <code>LIMITED</code> , and <code>EMPTY</code> .
<code>min_order</code>	integer	required	body	Minimum order required to purchase the product. Can only be a positive integer
<code>weight</code>	float	required	body	Weight of the product.
<code>weight_unit</code>	string	required	body	The unit of the weight with the following available values <code>GR</code> (gram) and <code>KG</code> (kilogram)
<code>condition</code>	string	required	body	The condition of the product with the following available values <code>NEW</code> and <code>USED</code> .
<code>etalase</code>	object	required	body	Etalase of the product. The object contains <code>id</code> and <code>name</code> . To get available etalase, please check Get All Etalase . Required field to input just <code>id</code> with <code>etalase_id</code> responses from Get All Etalase
<code>description</code>	string	optional	body	Description of the product. Maximum characters allowed is 2000.

Gambar 2.26 Parameter Create Product V2

<code>is_must_insurance</code>	boolean	optional	body	Determine if the product must be insured (<code>true</code>) or not (<code>false</code>).
<code>is_free_return</code>	boolean	optional	body	Determine if the product can be returned (<code>true</code>) by buyers or not (<code>false</code>).
<code>sku</code>	string	optional	body	The stock keeping unit for the product. Maximum characters allowed is 50.
<code>stock</code>	integer	optional	body	The stock of the product. <code>0</code> indicates always available. Other than that, the possible values are from <code>1</code> to <code>1000</code> . Stock should be <code>1</code> if want to add variant product.
<code>wholesale</code>	object array	optional	body	Wholesale price and quantity of the product. The object keys includes: <code>min_qty</code> and <code>price</code> .
<code>preorder</code>	object	optional	body	Preorder information. The object keys includes: <code>is_active</code> , <code>duration</code> , and <code>time_unit</code> .
<code>pictures</code>	object array	optional	body	Images information of the product. The object keys includes: <code>file_path</code> .
<code>videos</code>	object array	optional	body	Video link of the product. The object keys includes: <code>url</code> and <code>source</code> . <code>url</code> should only contain the YouTube video id e.g. <code>dQw4w9WgXcQ</code> . Where the type <code>type</code> should be <code>youtube</code> .
<code>variant</code>	object	optional	body	Variant of the product. The object keys includes: <code>variant</code> and <code>product_variant</code> .

Gambar 2.27 Parameter Create Product V2

```

curl -X POST \
  'https://fs.tokopedia.net/v2/products/fs/13004/create?shop_id=479573' \
  -H 'Authorization: Bearer C-UMqWXRqaJugr7w5U0zQ' \
  -H 'Content-Type: application/json' \
  -d '{
"products":[
  {
    "name":"Product Testing V2 1.10",
    "condition":"NEW",
    "description":"Product Testing Descr V2",
    "sku":"TST21",
    "price":10000,
    "status":"LIMITED",
    "stock":900,
    "min_order":1,
    "category_id":1817,
    "price_currency":"IDR",
    "weight":200,
    "weight_unit":"GR",
    "is_free_return":false,
    "is_must_insurance":false,
    "etalase":{
      "id":1402956
    },
    "pictures":[
      {
        "file_path":"https://ecs7.tokopedia.net/img/cache/700/product-1/2017/9/27/55:
      }
    ],
    "wholesale":[
      {
        "min_qty":2,
        "price":9500
      },
      {

```

Gambar 2.28 Contoh Request Create Product V2

```

    {
      "min_qty":3,
      "price":9000
    }
  ],
  "preorder":{
    "is_active":true,
    "duration":5,
    "time_unit":"DAY"
  },
  "videos": [
    {
      "source": "youtube",
      "url": "3T9DAOQIUdo"
    }
  ]
}
]
}'

```

Gambar 2.29 Contoh Request Create Product V2

```

{
  "header": {
    "process_time": 0.404528676,
    "messages": "Your request has been processed successfully"
  },
  "data": {
    "upload_id": 5721
  }
}

```

Gambar 2.30 Contoh Respon Sukses Create Product

```

{
  "header": {
    "process_time": 0.000126612,
    "messages": "We could not process your request due to malformed request, please check",
    "reason": "Invalid field [shop_id] format, value [] should be int",
    "error_code": "PRD_DLV_042"
  },
  "data": null
}

```

Gambar 2.31 Contoh Respon Gagal Create Product

g. Endpoint Check Upload Status

Berikut adalah endpoint dari check upload status beserta parameter, contoh request dan contoh responsenya.

```
GET /v2/products/fs/:fs_id/status/:upload_id
```

Gambar 2.32 Endpoint Check Upload Status

Parameter	Type	Required?	Location	Description
<code>fs_id</code>	integer	required	path	Fulfillment service unique identifier.
<code>shop_id</code>	integer	required	query	Shop unique identifier.
<code>upload_id</code>	integer	required	query	Upload id of the product to check.

Gambar 2.33 Parameter Check Upload Status

```
curl -X GET \
  'https://fs.tokopedia.net/v2/products/fs/13245/status/12345?shop_id=479573' \
  -H 'Authorization: Bearer HB8GNgfdRXS9c_sksTdTBw'
```

Gambar 2.34 Contoh Request Check Upload Status

```

{
  "header": {
    "process_time": 0.045122173,
    "messages": "Your request has been processed successfully"
  },
  "data": {
    "upload_data": [
      {
        "upload_id": 0,
        "status": "DONE",
        "total_data": 1,
        "unprocessed_rows": 0,
        "success_rows": 1,
        "failed_rows": 0,
        "processed": 1
      }
    ]
  }
}

```

Gambar 2.35 Contoh Respon Sukses Check Upload Status

```

{
  "header": {
    "process_time": 9.7e-8,
    "messages": "We could not process your request due to malformed request, please che",
    "reason": "Invalid field [upload_id] format, value [283a] should be int",
    "error_code": "PRD_DLV_042"
  },
  "data": null
}

```

Gambar 2.36 Contoh Respon Gagal Check Upload Status

h. Endpoint Get All Products V2

Berikut adalah endpoint dari get all products v2 beserta parameter, contoh request dan contoh responsnya.

```
GET /v2/products/fs/:fs_id/:page/:per_page?product_id=:product_id
```

Gambar 2.37 Endpoint Get All Products V2

Parameter	Type	Required?	Location	Description
<code>fs_id</code>	integer	required	path	Fulfillment service id.
<code>page</code>	integer	required	path	Page number to shown.
<code>per_page</code>	integer	required	path	The total number of products to be shown on one page.
<code>product_id</code>	integer	optional	query	Product ID.

Gambar 2.38 Parameter Get All Products V2

```
bash
curl -X GET \
  'https://fs.tokopedia.net/v2/products/fs/13004/1/20' \
  -H 'Authorization: Bearer HB8GNgfdRXS9c_sksTdTBw'
```

Gambar 2.39 Contoh Request Get All Products V2

```

{
  "data": [
    {
      "product_id": 34,
      "name": "Kemeja Pria",
      "sku": "BF10",
      "shop_id": 479573,
      "shop_name": "I`nti.Cosmetic",
      "category_id": 1805,
      "desc": "A shirt is a cloth garment for the upper body. Originally an undergarment",
      "stock": 100,
      "price": 10000,
      "status": "Active"
    },
    {
      "product_id": 14286600,
      "name": "STABILO Paket Ballpoint Premium Bionic Rollerball - Multicolor",
      "sku": "",
      "shop_id": 479573,
      "shop_name": "I`nti.Cosmetic",
      "category_id": 1774,
      "desc": "Paket\n pulpen premium membuat kegiatan menulis kamu bisa lebih berwarra",
      "stock": 11,
      "price": 75000,
      "status": "Active"
    }
  ]
}

```

Gambar 2.40 Contoh Respon Sukses Get All Product V2

```

{
  "data": [],
  "status": "200 Ok",
  "error_message": []
}

```

Gambar 2.41 Contoh Respon Sukses Tapi Tidak Ada Produk

```

<html>
  <head>
    <title>401 Authorization Required</title>
  </head>
  <body bgcolor="white">
    <center>
      <h1>401 Authorization Required</h1>
    </center>
    <hr>
    <center>nginx</center>
  </body>
</html>

```

Gambar 2.42 Contoh Respon Gagal Get All Product V2

i. Endpoint Get Product By ID

Berikut adalah endpoint dari get product by id beserta parameter, contoh request dan contoh responsenya.

```
GET /inventory/v1/fs/:fs_id/product/info
```

Gambar 2.43 Endpoint Get Product by ID

Parameter	Type	Required?	Location	Description
fs_id	integer	required	path	Fulfillment service unique identifier
product_id	integer	optional	query	Can input more than one product_id
product_url	string	optional	query	Can input more than one product_url

Gambar 2.44 Parameter Get Product by ID

```

curl -X GET \
  'https://fs.tokopedia.net/inventory/v1/fs/13004/product/info?product_id=15341594' \
  -H 'Authorization: Bearer HB8GNgfdRXS9c_sksTdtBw'

```

Gambar 2.45 Contoh Request Get Product by ID

j. Endpoint Get Product By SKU

Berikut adalah endpoint dari get product by SKU beserta parameter, contoh request dan contoh responsenya.

```
GET /inventory/v1/fs/:fs_id/product/info
```

Gambar 2.46 Endpoint Get Product by SKU

Parameter	Type	Required?	Location	Description
fs_id	integer	required	path	Fulfillment service unique identifier
sku	string	optional	query	Product's SKU

Gambar 2.47 Parameter Get Product by SKU

```
curl -X GET \
  'https://fs.tokopedia.net/inventory/v1/fs/13004/product/info?sku=TST21' \
  -H 'Authorization: Bearer HB8GNgfdRXS9c_sksTdTBw'
```

Gambar 2.48 Contoh Request Get Product by SKU

k. Endpoint Edit Product V2

Berikut adalah endpoint dari edit product v2 beserta parameter, contoh request dan contoh responsenya.

```
PATCH /v2/products/fs/:fs_id/edit
```

Gambar 2.49 Endpoint Edit Product V2

Parameter	Type	Required?	Location	Description
<code>fs_id</code>	integer	required	path	Fulfillment service unique identifier
<code>shop_id</code>	integer	required	path	Shop unique identifier
<code>products</code>	object array	required	body	Product object array to create.

Gambar 2. 50 Parameter Edit Product V2

Parameter	Type	Required?	Location	Description
<code>id</code>	integer	required	body	Product ID
<code>sku</code>	string	optional	body	The stock keeping unit for the product. Maximum characters allowed is 50.
<code>name</code>	string	optional	body	Name of the product with length less than or equals 70 characters.
<code>category_id</code>	integer	optional	body	Unique identifier for the product's category. To get available categories, please check Get All Categories . Please input the <i>deepest category child ID</i> .
<code>price_currency</code>	string	optional	body	Currency code for stated price (IDR or USD)
<code>price</code>	integer	optional	body	The possible value between 100 to 100.000.000. If the product variant is added, the price parameter is automatically set to the lowest price among the variant products.
<code>status</code>	string	optional	body	Status for the product with the following available values <code>UNLIMITED</code> , <code>LIMITED</code> , and <code>EMPTY</code> .
<code>min_order</code>	integer	optional	body	Minimum order required to purchase the product. Can only be a positive integer
<code>weight</code>	float	optional	body	Weight of the product.
<code>weight_unit</code>	string	optional	body	The unit of the weight with the following available values <code>GR</code> (gram) and <code>KG</code> (kilogram)
<code>condition</code>	string	optional	body	The condition of the product with the following available values <code>NEW</code> and <code>USED</code> .

Gambar 2.51 Parameter Product pada Edit Product V2

<code>etalase</code>	object	optional	body	Etalase of the product. The object contains <code>id</code> and <code>name</code> . To get available etalase, please check Get All Etalase . Required field to input just <code>id</code> with <code>etalase_id</code> responses from Get All Etalase
<code>description</code>	string	optional	body	Description of the product. Maximum characters allowed is 2000.
<code>is_must_insurance</code>	boolean	optional	body	Determine if the product must be insured (<code>true</code>) or not (<code>false</code>).
<code>is_free_return</code>	boolean	optional	body	Determine if the product can be returned (<code>true</code>) by buyers or not (<code>false</code>).
<code>stock</code>	integer	optional	body	The stock of the product. <code>0</code> indicates always available. Other than that, the possible values are from <code>1</code> to <code>1000</code> . Stock should be <code>1</code> if want to add variant product.
<code>wholesale</code>	object array	optional	body	Wholesale price and quantity of the product. The object keys includes: <code>min_qty</code> and <code>price</code> .
<code>preorder</code>	object	optional	body	Preorder information. The object keys includes: <code>is_active</code> , <code>duration</code> , and <code>time_unit</code> .
<code>pictures</code>	object array	optional	body	Images information of the product. The object keys includes: <code>file_path</code> .
<code>videos</code>	object array	optional	body	Video link of the product. The object keys includes: <code>url</code> and <code>source</code> . <code>url</code> should only contain the YouTube video id e.g. <code>dQw4w9WgXcQ</code> . Where the type <code>type</code> should be <code>youtube</code> .
<code>variant</code>	object	required*	body	Variant of the product. The object keys includes: <code>variant</code> and <code>products</code> . There are some field in variant entity that mandatory. For more detail please refer to this link

Gambar 2.52 Parameter Product dari Edit Product V2

```
{
  "header": {
    "process_time": 3.945634896,
    "messages": "Your request has been processed successfully"
  },
  "data": {
    "upload_id": 5721
  }
}
```

Gambar 2.53 Contoh Respon Sukses pada Edit Product V2

```
{
  "header": {
    "process_time": 0.381013787,
    "messages": "Our server encounters an error, please try again later",
    "reason": "Data Not Found",
    "error_code": "OPENAPI_DB_003"
  },
  "data": null
}
```

Gambar 2.54 Contoh Respon Gagal pada Edit Product ID

1. Endpoint Delete Product V2

Berikut adalah endpoint dari delete product V2 beserta parameter, contoh request dan contoh responsnya.

```
POST /v3/products/fs/:fs_id/delete
```

Gambar 2.55 Endpoint Delete Product V2 Tokopedia API

Parameter	Type	Required?	Location	Description
fs_id	integer	required	path	Fulfillment service unique identifier
shop_id	integer	required	query	Shop ID
product_id	[]integer	required	body	List of Product ID

Gambar 2.56 Parametert Delete Product V2 Tokopedia API

```

curl -X POST \
  https://fs.tokopedia.net/v3/products/fs/13004/delete?shop_id=479573 \
  -H 'Authorization: Bearer OoHgyxcgQ5a1uCbmbbXjjA' \
  -H 'Content-Type: application/json' \
  -d '{
    "product_id": [15362375]
  }'

```

Gambar 2.57 Contoh Request Delete Product V2 Tokopedia API

```

{
  "header": {
    "process_time": 0.588856684,
    "messages": "Your request has been processed successfully"
  },
  "data": {
    "total_data": 1,
    "succeed_rows": 1,
    "failed_rows": 0,
    "failed_rows_data": null
  }
}

```

Gambar 2.58 Contoh Respon Sukses Delete Product V2 Tokopedia API

```

{
  "header": {
    "process_time": 0.029730173,
    "messages": "Your request has been processed successfully"
  },
  "data": {
    "total_data": 1,
    "succeed_rows": 0,
    "failed_rows": 1,
    "failed_rows_data": [
      "Error Product ID [115306196] : There are some problem when retrieve product in"
    ]
  }
}

```

Gambar 2.59 Contoh Respon Gagal Delete Product V2 Tokopedia API

m. Endpoint Webhook Register

Berikut adalah endpoint dari webhook register beserta parameter, contoh request dan contoh responsenya.

```
POST /v1/fs/:fs_id/register
```

Gambar 2.60 Endpoint Webhook Register

Parameter	Type	Required?	Location	Description
fs_id	integer	required	query	Fulfillment service id.
fs_id	integer	required	body	Fulfillment service id.
order_notification_url	string	required	body	Callback url on which to receive new payment verified order notifications.
order_cancellation_url	string	required	body	Callback url on which to receive order cancellation notifications.
order_status_url	string	required	body	Callback url on which to receive order status update.
chat_notification_url	string	required	body	Callback url on which to receive new chat notification.
product_creation_url	string	required	body	Callback url which triggered when product creation from Create Product Endpoint V2 finished uploading.
product_changes_url	string	required	body	Callback url on which to receive new product changes.
campaign_notification_url	string	required	body	Callback url on which to receive new campaign notification.
webhook_secret	string	required	body	Secret for generating HMAC while making callbacks.

Gambar 2.61 Parameter Webhook Register

```

curl -X POST \
  https://fs.tokopedia.net/v1/fs/13004/register \
  -H 'Authorization: Bearer OoHgyxcgQ5a1uCbmbbXjjA' \
  -H 'Content-Type: application/json' \
  -d '{
    "fs_id":13004,
    "order_notification_url": "http://yourstore.com/v1/order/notification",
    "order_cancellation_url": "http://yourstore.com/v1/order/cancellation",
    "order_status_url": "http://yourstore.com/v1/order/status",
    "order_request_cancellation_url":"http://yourstore.com/v1/order/request_cancel",
    "chat_notification_url":"http://yourstore.com/v1/chat/notification",
    "product_creation_url":"http://yourstore.com/v1/product/creation",
    "product_changes_url": "http://yourstore.com/v1/product/changes",
    "campaign_notification_url": "http://yourstore.com/v1/campaign/notification",
    "webhook_secret": "webhook_secret"
  }'

```

Gambar 2.62 Contoh Request Webhook Register

```

{
  "data": "success",
  "status": "200 Ok",
  "error_message": []
}

```

Gambar 2.63 Contoh Respon Sukses Webhook Register

```

{
  "data": null,
  "status": "400 Bad Request",
  "error_message": [
    "bad input data"
  ]
}

```

Gambar 2.64 Contoh Respon Gagal Webhook Register

n. Endpoint Webhook List Registered

Berikut adalah endpoint dari webhook list registered beserta parameter, contoh request dan contoh responsenya.

```
GET /v1/fs/:fs_id
```

Gambar 2.65 Endpoint Webhook List Registered

Parameter	Type	Required?	Location	Description
fs_id	integer	required	query	Fulfillment service id.

Gambar 2.66 Parameter Webhook List Registered

```
curl -X GET \
  https://fs.tokopedia.net/v1/fs/13004 \
  -H 'Authorization: Bearer JhgtgUAQ50JIrQMjCiHA'
```

Gambar 2.67 Contoh Request Webhook List Registered

```
{
  "data": {
    "fs_id": 13004,
    "order_notification_url": "http://yourstore.com/v1/order/notification",
    "order_cancellation_url": "http://yourstore.com/v1/order/cancellation",
    "order_status_url": "http://yourstore.com/v1/order/status",
    "order_request_cancellation_url": "http://yourstore.com/v1/order/request_cancel",
    "chat_notification_url": "http://yourstore.com/v1/chat/notification",
    "product_changes_url": "http://yourstore.com/v1/product/changes",
    "campaign_notification_url": "http://yourstore.com/v1/campaign/notification",
    "webhook_secret": "webhook_secret"
  },
  "status": "200 Ok",
  "error_message": []
}
```

Gambar 2.68 Contoh Respon Sukses Webhook List Registered

```

<html>
  <head>
    <title>401 Authorization Required</title>
  </head>
  <body bgcolor="white">
    <center>
      <h1>401 Authorization Required</h1>
    </center>
    <hr>
    <center>nginx</center>
  </body>
</html>

```

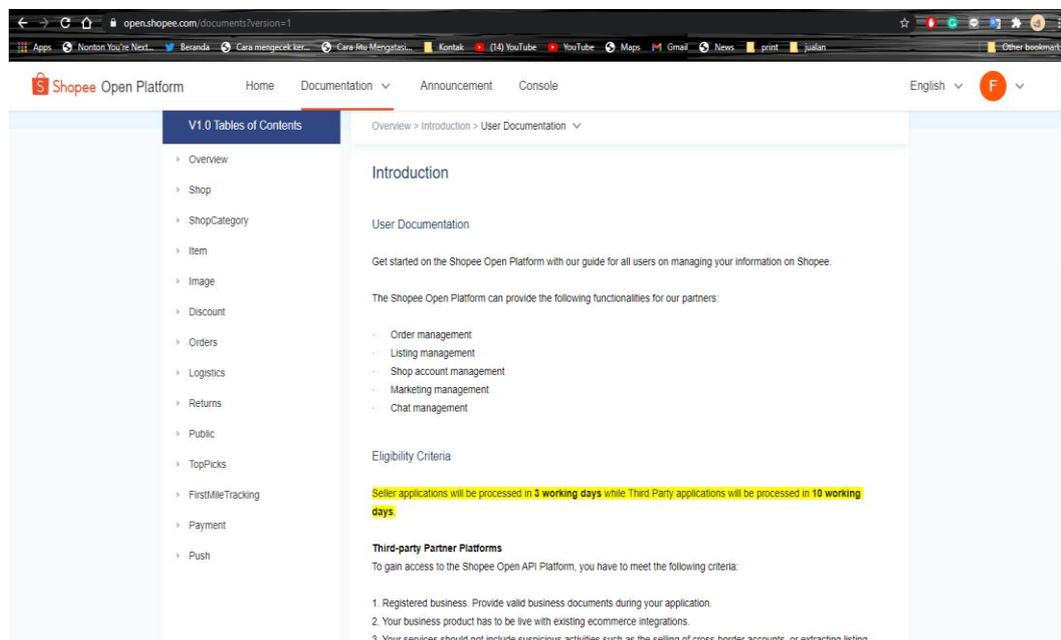
Gambar 2.69 Contoh Respon Gagal Webhook List Registered

2.13 Shopee API

Shopee API adalah API yang disediakan oleh Shopee untuk membantu pedagang dalam mengelola toko mereka khususnya untuk pedagang besar yang menjadi mitra Shopee.

Beberapa pedagang (terutama pedagang besar) biasanya memiliki gudang baik satu maupun lebih dari satu dan sudah memiliki sistem inventori sendiri. Hal tersebut menimbulkan masalah di mana terdapat dua sistem yang berbeda untuk mengelola satu toko. Terlebih dari sisi perawatan juga tidak akan efisien jika harus merawat dua sistem untuk toko yang sama. Agar bisa memudahkan para pedagang dalam mengelola usaha mereka, dibuatlah API ini. Di mana fungsi utama dari API ini adalah agar sistem yang sudah dimiliki toko tersebut bisa terintegrasi dengan sistem di Shopee.

Untuk wilayah Indonesia, syarat agar toko tersebut dapat memakai API yang disediakan oleh Shopee adalah toko tersebut harus berstatus sebagai *mall sellers*, *managed sellers* atau *sellers with at least 30 orders in the last 30 days*. Lalu jika status dari toko tersebut sudah sesuai dengan syarat yang diinginkan Shopee, maka langkah selanjutnya adalah menyambungkan toko tersebut sistem atau aplikasi yang akan dibuat oleh pengembang. Untuk dokumentasinya sendiri bisa diakses pada link <https://open.shopee.com/documents?version=1> dan dokumentasi pada Shopee API dapat dilihat pada Gambar 2.70.



Gambar 2.70 Dokumentasi Shopee API

Dalam dokumentasi tersebut terdapat banyak API yang bisa dimanfaatkan. Berikut adalah fitur-fitur yang bisa dan akan dimanfaatkan dalam sistem yang dibangun memakai Shopee API, yaitu

- 1) Membuat atau menambahkan produk
- 2) Lihat Produk
- 3) Mengubah Produk
- 4) Hapus Produk
- 5) *Push*

Agar dapat menjalankan fitur-fitur yang sudah dijelaskan, maka diperlukan akses ke endpoint-endpoint yang diperlukan. Berikut adalah penjelasan lebih detail mengenai endpoint yang dipakai:

a. Authentication

Berikut adalah endpoint beserta parameter yang digunakan untuk *authentication* agar nantinya sistem bisa mendapatkan token dari Shopee API.

Contoh request

https://partner.shopeemobile.com/api/v1/shop/auth_partner?id=xxxxxxx&token=xxxxxxx&redirect=xxxxxxx.com

Endpoint :

https://partner.shopeemobile.com/api/v1/shop/auth_partner

https://partner.test-stable.shopeemobile.com/api/v1/shop/auth_partner (test environment)

Parameter:

id: Partner ID.

token: enkripsi yang sesuai dengan aturan shopee.

redirect: URL setelah *authentication* selesai dilakukan.

a. Endpoint Category

Endpoint dari `get_category` dan parameternya dapat dilihat pada Gambar 2.71.

Surroundings	HTTP Address		
URL	https://partner.shopeemobile.com/api/v2/product/get_category		
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/product/get_category		

Name	Description		
method	GET		
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2		

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
access_token	string	c09222e3fc40ff b25fc947f738b1 abf1	The token for API access, using to identify your permission to the api. Valid for multiple use and expires in 4 hours.
shop_id	int	600000	Shopee's unique identifier for a shop. Required param for most APIs.
sign	string	e318d3e932719 916a9f9ebb57e 2011961bd47ab fa54a36e040d0 50d8931596e2	Signature generated by partner_id, api path, timestamp, access_token, shop_id and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2.71 Endpoint dan Parameter Get Category

b. Add Item

Endpoint dari add_item dan parameternya dapat dilihat pada Gambar 2.72, Gambar 2.73 dan Gambar 2.74.

Surroundings	HTTP Address
URL	https://partner.shopeemobile.com/api/v2/product/add_item
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/product/add_item

Name	Description
method	POST
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
access_token	string	c09222e3fc40ffb25fc947f738b1abf1	The token for API access, using to identify your permission to the api. Valid for multiple use and expires in 4 hours.
shop_id	int	600000	Shopee's unique identifier for a shop. Required param for most APIs.
sign	string	e318d3e932719916a9f9ebb57e2011961bd47abfa54a36e040d050d8931596e2	Signature generated by partner_id, api path, timestamp, access_token, shop_id and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2.72 Endpoint dan Paramater dari Add Item

Name	Type	Sample	Description
original_price Required	float	123.3	Item price
description Required	string	item description test	Description of item
weight	float	1.1	Weight of item
item_name Required	string	Item Name Example	Item name
item_status	string	UNLIST	Item status, could be UNLIST or NORMAL
<input type="checkbox"/> dimension	object		Item dimension
normal_stock Required	int	33	Item stock
<input type="checkbox"/> logistic_info Required	object[]		Logistic channel setting
<input type="checkbox"/> attribute_list	object[]		This field is optional(expect Indonesia) depending on the specific attribute under different categories. Should call shopee.item.GetAttributes to get attribute first. Must contain all all mandatory attribute.
category_id Required	int	14695	ID of category
<input type="checkbox"/> image Required	object		Item images

Gambar 2.73 Parameter Add Item

<input type="checkbox"/> pre_order	object		Pre order setting
item_sku	string		SKU tag of item
condition	string	NEW	Condition of item, could be USED or NEW
<input type="checkbox"/> wholesale	object[]		Wholesale setting
video_upload_id	string[]	["sg_f4bde9bc-ff3c-485e-a6dd-3161dab4b942_000000"]	Video upload ID returned from video uploading API. Only accept one video_upload_id at most.
<input type="checkbox"/> brand	object		
item_dangerous	int	0	This field is only applicable for local sellers in Indonesia and Malaysia. Use this field to identify whether a product is a dangerous product. 0 for non-dangerous product and 1 for dangerous product. For more information, please visit the market's respective Seller Education Hub.

Gambar 2.74 Parameter Add Item

c. Get Item Base Info

Endpoint dari `get_item_base_info` beserta parameternya dapat dilihat pada Gambar 2.75 dan Gambar 2.76.

Surroundings	HTTP Address
URL	https://partner.shopeemobile.com/api/v2/product/get_item_base_info
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/product/get_item_base_info

Name	Description
method	GET
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
access_token	string	c09222e3fc40ff b25fc947f738b1 abf1	The token for API access, using to identify your permission to the api. Valid for multiple use and expires in 4 hours.
shop_id	int	600000	Shopee's unique identifier for a shop. Required param for most APIs.
sign	string	e318d3e932719 916a9f9ebb57e 2011961bd47ab fa54a36e040d0 50d8931596e2	Signature generated by partner_id, api path, timestamp, access_token, shop_id and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2.75 Endpoint dan Parameter Get Item Base Info

Name	Type	Sample	Description
item_id_list Required	int[]	34001,34002	item_id list; limit [0,50]

Gambar 2.76 Parameter Get Item Base Info

d. Update Item

Endpoint dari update_item beserta parameternya dapat dilihat pada Gambar 2.77, Gambar 2.78, Gambar 2.79.

Surroundings	HTTP Address
URL	https://partner.shopeemobile.com/api/v2/product/update_item
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/product/update_item

Name	Description
method	POST
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
access_token	string	c09222e3fc40ff b25fc947f738b1 abf1	The token for API access, using to identify your permission to the api. Valid for multiple use and expires in 4 hours.
shop_id	int	600000	Shopee's unique identifier for a shop. Required param for most APIs.
sign	string	e318d3e932719 916a9f9ebb57e 2011961bd47ab fa54a36e040d0 50d8931596e2	Signature generated by partner_id, api path, timestamp, access_token, shop_id and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2.77 Endpoint dan Parameter dari Update Item

Name	Type	Sample	Description
description	string	Hello product product WlQPdMV4SIV oG7QD1v0fEec NoCVEBNx6	Description of item.
weight	float	0.9	Weight of item.
<input type="checkbox"/> pre_order	object		Pre Order setting.
item_name	string	Hello Pgkk50jdNgEnl WvX	Item name.
<input type="checkbox"/> attribute_list	object[]		Item attributes.
<input type="checkbox"/> image	object		Images of item.
item_sku	string	abc	SKU tag for item.
item_status	string	UNLIST	Item status, could be UNLIST or NORMAL.
<input type="checkbox"/> logistic_info	object[]		Logistic channel setting.
<input type="checkbox"/> wholesale	object[]		Wholesale setting.
item_id Required	int	2800143058	ID of item.
category_id	int	34106	ID of category.
<input type="checkbox"/> dimension	object		Dimension of item.

Gambar 2.78 Parameter dari Update Item

condition	string	USED	Condition of item, could be NEW or USED.
video_upload_id	object[]	["sg_f4bde9bc-ff3c-485e-a6dd-3161dab4b942_000000"]	Video upload ID returned from video uploading API.
<input type="checkbox"/> brand	object		
item_dangerous	int	0	This field is only applicable for local sellers in Indonesia and Malaysia. Use this field to identify whether a product is a dangerous product. 0 for non-dangerous product and 1 for dangerous product. For more information, please visit the market's respective Seller Education Hub.

Gambar 2.79 Parameter dari Update Item

e. Delete Item

Endpoint dari delete_item beserta parameternya dapat dilihat pada Gambar 2.80 dan Gambar 2.81.

Surroundings	HTTP Address		
URL	https://partner.shopeemobile.com/api/v2/product/delete_item		
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/product/delete_item		

Name	Description		
method	POST		
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2		

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
access_token	string	c09222e3fc40ffb25fc947f738b1abf1	The token for API access, using to identify your permission to the api. Valid for multiple use and expires in 4 hours.
shop_id	int	600000	Shopee's unique identifier for a shop. Required param for most APIs.
sign	string	e318d3e932719916a9f9ebb57e2011961bd47abfa54a36e040d050d8931596e2	Signature generated by partner_id, api path, timestamp, access_token, shop_id and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2.80 Endpoint dan Parameter Delete Item

Name	Type	Sample	Description
item_id Required	int	34001	The identity of product item.

Gambar 2. 81 Endpoint dan Parameter Delete Item

f. Get Push Config

Endpoint dari delete_item beserta parameternya dapat dilihat pada Gambar 2.82.

Surroundings	HTTP Address
URL	https://partner.shopeemobile.com/api/v2/push/get_push_config
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/push/get_push_config

Name	Description
method	GET
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
sign	string	e318d3e932719 916a9f9ebb57e 2011961bd47ab fa54a36e040d0 50d8931596e2	Signature generated by partner_id, api path, timestamp and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2. 82 Endpoint dan Parameter Get Push Config

g. Set Push Config

Endpoint dari delete_item beserta parameternya dapat dilihat pada Gambar 2.83 dan Gambar 2.84.

Surroundings	HTTP Address
URL	https://partner.shopeemobile.com/api/v2/push/set_push_config
Test URL	https://partner.test-stable.shopeemobile.com/api/v2/push/set_push_config

Name	Description
method	POST
data definition	https://open.shopee.com/documents?module=87&type=2&id=62&version=2

Name	Type	Sample	Description
partner_id	int	1	Partner ID is assigned upon registration is successful. Required for all requests.
timestamp	timestamp	1610000000	This is to indicate the timestamp of the request. Required for all requests. Expires in 5 minutes.
sign	string	e318d3e932719 916a9f9ebb57e 2011961bd47ab fa54a36e040d0 50d8931596e2	Signature generated by partner_id, api path, timestamp and partner_key via HMAC-SHA256 hashing algorithm. More details: https://open.shopee.com/documents?module=87&type=2&id=58&version=2

Gambar 2.83 Endpoint dan Paramter Set Push Config

Name	Type	Sample	Description
callback_url	string	www.baidu.com	The callback url of push mechanism.
 push_config	object	{ "shop_update": 0, "reserved_stock _change": 0, "promotion_upd ate": 0, "banned_item": 0, "order_status": 0, "order_tracking _no": 0, "item_promotion ": 0 }	Detail configuration of push mechanism.
blocked_shop_id	int[]	[1,2,3]	Use this filed to set shops that need to be blocked.

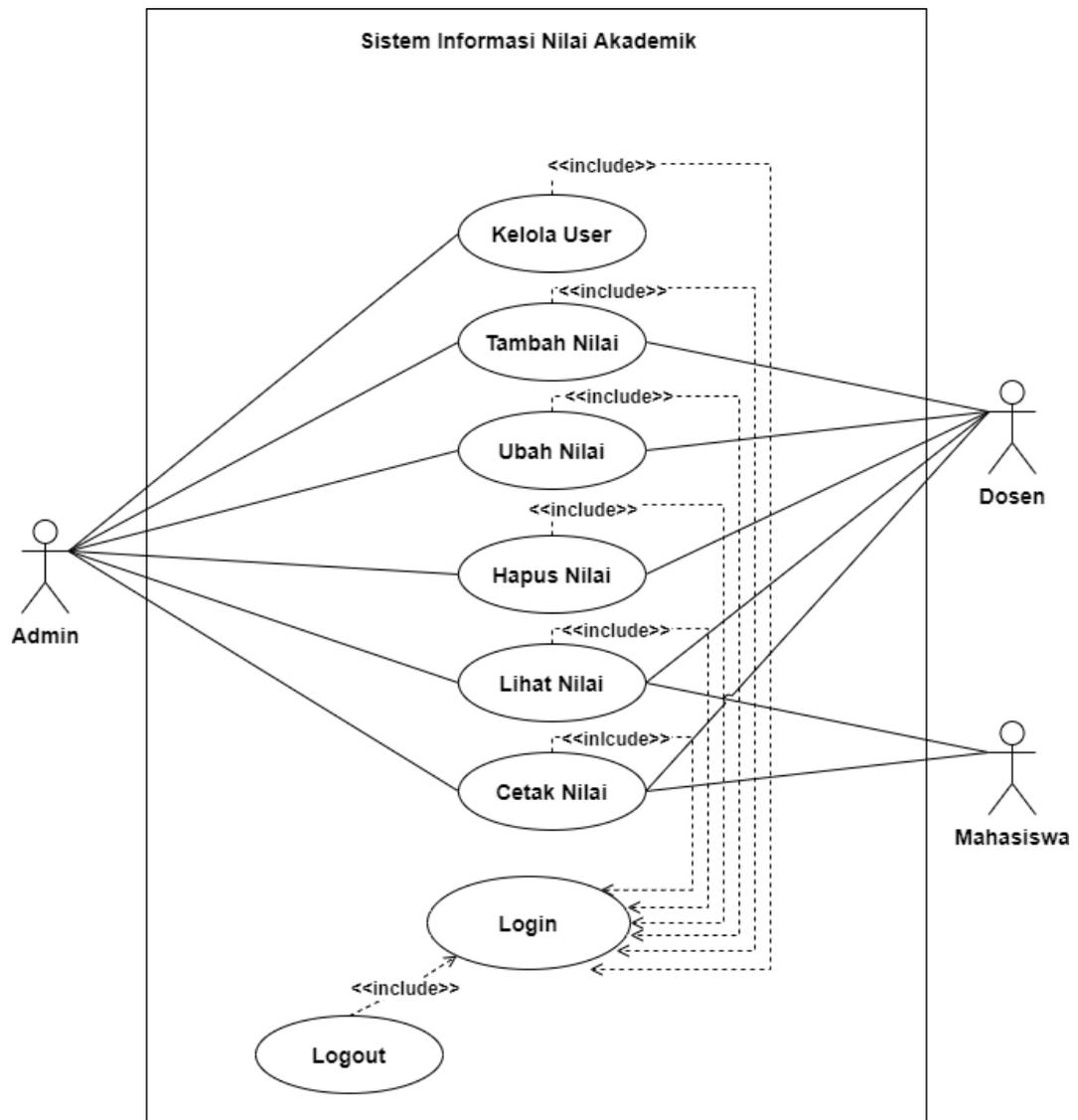
Gambar 2. 84 Parameter Set Push Config

2.14 UML

UML atau Unified Modeling Language merupakan metode yang dapat digunakan untuk membuat analisa perancangan bagi suatu sistem yang menggunakan OOP atau Object Oriented Programming [21]. Terdapat beberapa diagram yang dipakai untuk UML antara lain adalah sebagai berikut:

1. *Use Case Diagram*

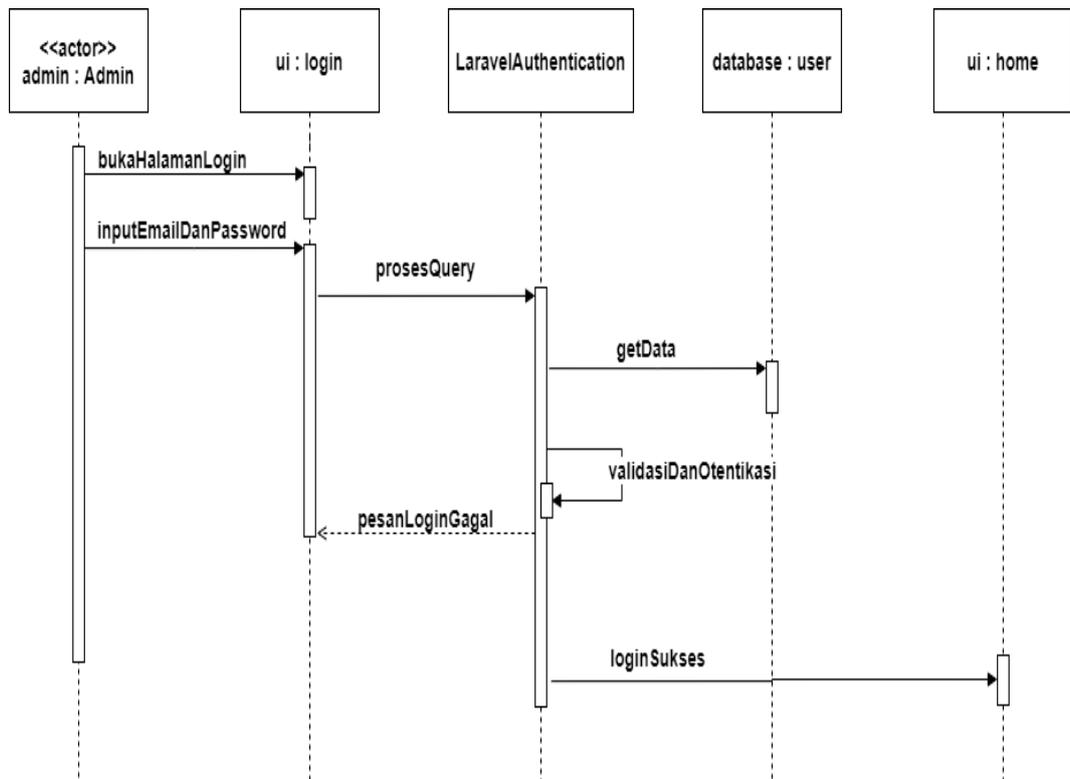
Use Case Diagram merupakan pemodelan yang menggambarkan interaksi antara user atau aktor dalam menggunakan / memanfaatkan sistem atau external view dari sistem yang akan dibuat [22]. Komponen yang membentuk suatu diagram use case adalah aktor (pihak yang berperan pada sistem), use case(aktivitas atau sarana yang disiapkan oleh sistem) dan hubungan atau *link*. Berikut adalah contoh dari use case bisa dilihat pada Gambar 2.85.



Gambar 2.85 Use Case Diagram

2. Sequence Diagram

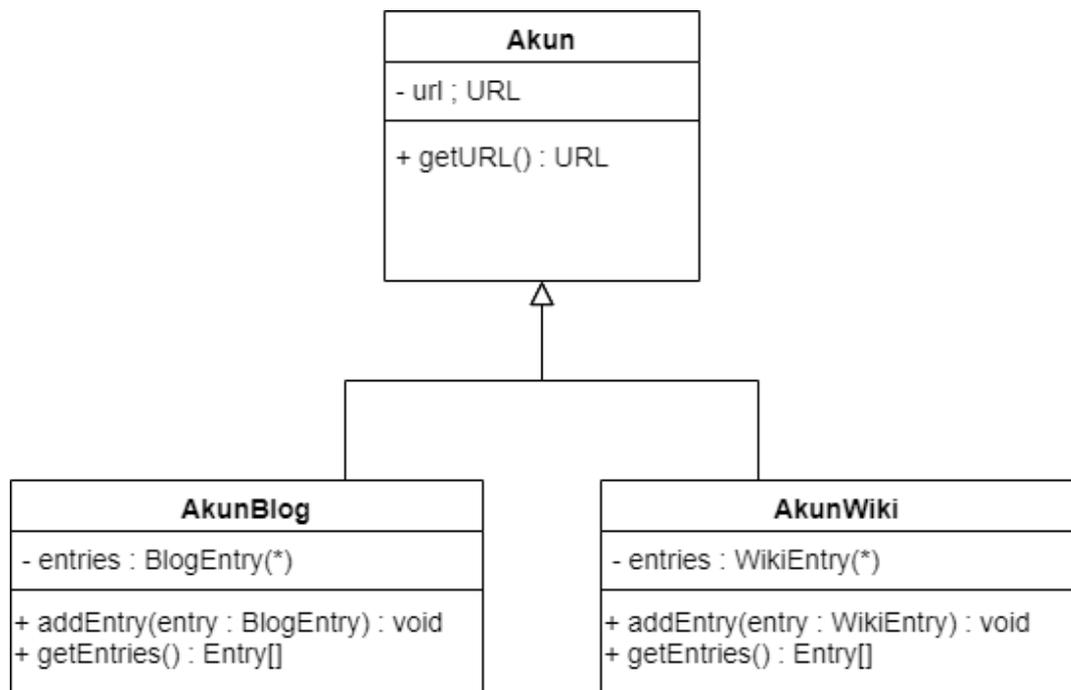
Sequence diagram merupakan pemodelan yang digunakan untuk menggambarkan perilaku aktor pada sebuah sistem secara detail berdasarkan waktu. Berikut adalah contoh dari sequence diagram bisa dilihat pada gambar 2.86 di bawah ini.



Gambar 2.86 Sequence Diagram

3. Class Diagram

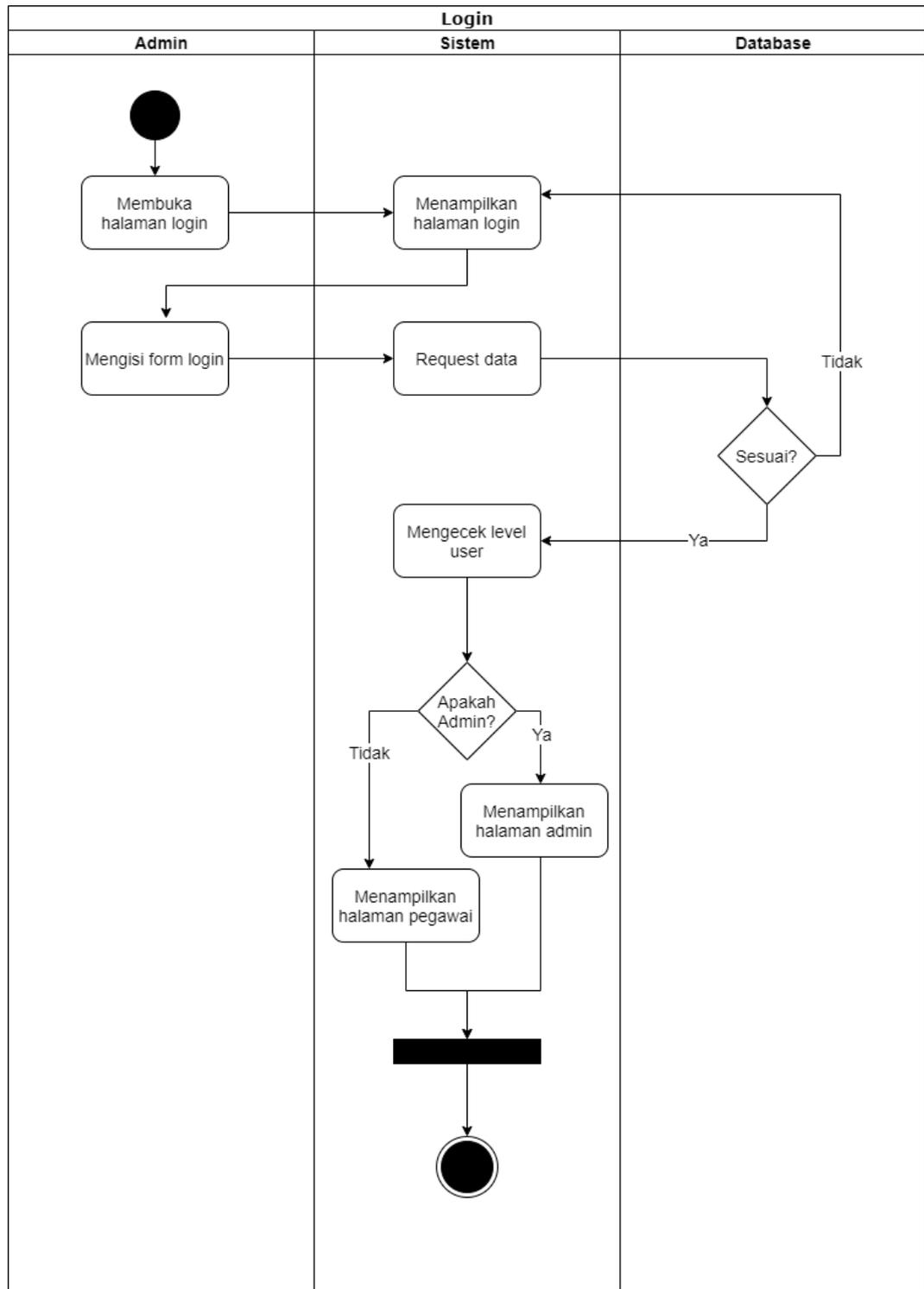
Class diagram merupakan pemodelan yang digunakan untuk menggambarkan hubungan antara class yang ada di mana setiap class memiliki atribut dan fungsi sesuai dengan proses yang terjadi. Berikut adalah contoh dari class diagram dapat dilihat pada gambar 2.87.



Gambar 2.87 Class Diagram

4. Activity Diagram

Activity Diagram merupakan pemodelan yang menggambarkan bagaimana aktivitas yang terjadi dalam sistem yang akan dirancang. Berikut adalah contoh dari activity diagram dapat dilihat pada Gambar 2.88.



Gambar 2.88 Activity Diagram