Pengembangan Sistem Informasi *E-Commerce* dengan Pemanfaatan API Midtrans Menggunakan *Framework* Laravel (Studi Kasus : Byboot.id).

PROPOSAL SKRIPSI

Disusun oleh:

Muhamad Maulana Zuhad Aditya

NIM: 185150701111001



PROGRAM STUDI TEKNOLOGI INFORMASI

JURUSAN SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2022

DAFTAR ISI

[DAFTAR ISI iii](#_Toc100921115)

[DAFTAR GAMBAR v](#_Toc100921116)

[DAFTAR TABEL vi](#_Toc100921117)

[BAB 1 PENDAHULUAN 1](#_Toc100921118)

[1.1 Latar Belakang 1](#_Toc100921119)

[1.2 Rumusan Masalah 2](#_Toc100921120)

[1.3 Tujuan 2](#_Toc100921121)

[1.4 Manfaat 3](#_Toc100921122)

[1.5 Batasan Masalah 3](#_Toc100921123)

[1.6 Sistematika Pembahasan 3](#_Toc100921124)

[BAB 2 LANDASAN KEPUSTAKAAN 4](#_Toc100921125)

[2.1 Kajian Pustaka 4](#_Toc100921126)

[2.2 Byboot.id 6](#_Toc100921127)

[2.3 E-Commerce 6](#_Toc100921128)

[*2.4 Software Development Life Cycle* 8](#_Toc100921129)

[*2.4.1 Waterfall Model* 8](#_Toc100921130)

[2.5 Unified Modelling Language (UML) 9](#_Toc100921131)

[*2.5.1 Activity Diagram* 9](#_Toc100921132)

[*2.5.2 Use Case Diagram* 9](#_Toc100921133)

[*2.5.3 Sequence Diagram* 10](#_Toc100921134)

[*2.5.4 Class Diagram* 10](#_Toc100921135)

[*2.5.5 Physical Data Model* 10](#_Toc100921136)

[2.6 Teknologi Pengembangan Sistem 10](#_Toc100921137)

[*2.6.1 Web Service* 10](#_Toc100921138)

[2.6.2 *Javascript Object Notation* (JSON) 11](#_Toc100921139)

[2.6.3 PHP 11](#_Toc100921140)

[2.6.4 Laravel 11](#_Toc100921141)

[2.6.5 Midtrans 12](#_Toc100921142)

[2.6.6 RajaOngkir 12](#_Toc100921143)

[2.7 Pengujian Perangkat Lunak 12](#_Toc100921144)

[*2.7.1 Blackbox Testing* 12](#_Toc100921145)

[2.7.2 *User Acceptance Testing* (UAT) 12](#_Toc100921146)

[*2.7.3 Compability Testing* 13](#_Toc100921147)

[BAB 3 METODOLOGI 14](#_Toc100921148)

[3.1 Studi Literatur 14](#_Toc100921149)

[3.2 Pengumpulan Data 15](#_Toc100921150)

[3.3 Analisis Kebutuhan 15](#_Toc100921151)

[3.4 Perancangan Sistem 15](#_Toc100921152)

[3.5 Implementasi Sistem 15](#_Toc100921153)

[*3.5.1* Implementasi *E-commerce* 16](#_Toc100921154)

[3.5.2 Implementasi *Payment Gateway* 16](#_Toc100921155)

[3.6 Pengujian 16](#_Toc100921156)

[3.7 Kesimpulan dan Saran 16](#_Toc100921157)

[BAB 4 ANALISIS KEBUTUHAN 17](#_Toc100921158)

[4.1 Deskripsi Umum Sistem 17](#_Toc100921159)

[4.2 Identifikasi Aktor 17](#_Toc100921160)

[4.3 Identifikasi Kebutuhan 18](#_Toc100921161)

[4.3.1 Kebutuhan Fungsional 18](#_Toc100921162)

[4.3.2 Kebutuhan Non Fungsional 20](#_Toc100921163)

[*4.4* Pemodelan *Use Case* 20](#_Toc100921164)

[*4.4.1 Use Case Diagram* 20](#_Toc100921165)

[*4.4.2 Use Case Scenario* 21](#_Toc100921166)

[BAB 5 PERANCANGAN SISTEM 30](#_Toc100921167)

[5.1 Arsitektur Sistem 30](#_Toc100921168)

[*5.2 Sequence Diagram* 31](#_Toc100921169)

[*5.2.1 Sequence Diagram Register* 31](#_Toc100921170)

[DAFTAR REFERENSI 32](#_Toc100921171)

DAFTAR GAMBAR

[Gambar 2.1 Komponen *E-Commerce* 7](#_Toc100521537)

[Gambar 2.2 Diagram *Waterfall Model* 8](#_Toc100521538)

[Gambar 2.3 Penulisan JSON 11](#_Toc100521539)

[Gambar 3.1 Diagram Alir Penelitian 14](#_Toc100521540)

[Gambar 4.1 Kode Kebutuhan 18](#_Toc100521541)

[Gambar 4.2 *Use Case Diagram Website* Byboot.id 21](#_Toc100521542)

DAFTAR TABEL

[Tabel 4.1 Identifikasi Aktor 17](#_Toc100616144)

[Tabel 4.2 Kebutuhan Fungsional Pengunjung dan Pelanggan 18](#_Toc100616145)

[Tabel 4.3 Kebutuhan Fungsional Admin 20](#_Toc100616146)

[Tabel 4.4 Kebutuhan Non Fungsional 20](#_Toc100616147)

[Tabel 4.5 *Use Case Scenario* Melakukan *Register* 21](#_Toc100616148)

[Tabel 4.6 *Use Case Scenario* Melihat Produk 22](#_Toc100616149)

[Tabel 4.7 *Use Case Scenario* Melihat Detail Produk 22](#_Toc100616150)

[Tabel 4.8 *Use Case Scenario* Melakukan Pencarian Produk 23](#_Toc100616151)

[Tabel 4.9 *Use Case Scenario* Melakukan *Login* 23](#_Toc100616152)

[Tabel 4.10 *Use Case Scenario* Melakukan *Logout* 24](#_Toc100616153)

[Tabel 4.11 *Use Case Scenario* Menambahkan Produk ke Keranjang 24](#_Toc100616154)

[Tabel 4.12 *Use Case Scenario* Melakukan *Checkout* 25](#_Toc100616155)

[Tabel 4.13 *Use Case Scenario* Melakukan Pembayaran 25](#_Toc100616156)

[Tabel 4.14 *Use Case Scenario* Mengubah Data Pribadi 26](#_Toc100616157)

[Tabel 4.15 *Use Case Scenario* Melihat Riwayat Transaksi 26](#_Toc100616158)

[Tabel 4.16 *Use Case Scenario* Melihat Status Transaksi 27](#_Toc100616159)

[Tabel 4.17 *Use Case Scenario Login* Admin 27](#_Toc100616160)

[Tabel 4.18 *Use Case Scenario Logout* Admin 28](#_Toc100616161)

[Tabel 4.19 *Use Case Scenario* Mengelola Produk 28](#_Toc100616162)

[Tabel 4.20 *Use Case Scenario* Mengelola Pesanan 29](#_Toc100616163)

# PENDAHULUAN

## Latar Belakang

*Electronic Commerce* (E-Commerce) adalah segala bentuk jual beli yang dilakukan secara daring atau menggunakan perangkat elektronik (Pradana, 2015). Saat ini terdapat berbagai macam platform *E-Commerce* yang memudahkan orang-orang untuk bertransaksi dengan berbagai macam produk yang tersedia didalamnya. Hal ini membuat *E-Commerce* sangat diminati oleh pengguna untuk melakukan transaksi bisnis. Dalam penggunaan sistem ini pembeli hanya perlu memilih barang, membayar, dan menunggu pesanan dikirimkan datang. Sedangkan penjual, hanya perlu memasarkan barang melalui platform *E-Commerce,* menunggu pesanan masuk dan mengirimkan barang ke pelanggan.

*E-Commerce* saat ini sudah mencakup berbagai sektor mulai dari fashion, makanan, rumah tangga, produk kecantikan, dan sebagainya. Menurut data Katadata (2021), produk fashion merupakan produk dengan penjualan tertinggi pada platform *E-Commerce* selama tahun 2020-2021 mengalahkan produk kesehatan dan kecantikan yang berada di urutan kedua*.* Salah satu produk fashion yang paling tinggi penjualannya adalah sepatu. Di Indonesia sendiri terdapat banyak merk sepatu yang sudah memiliki pasar sendiri yang sudah besar, contohnya Nike, Adidas, Geoff-Max, dan Ventela. Byboot.id juga merupakan salah satu produsen sepatu lokal yang berada di Provinsi Jawa Timur.

Saat ini, Byboot.id melakukan transaksi jual beli melalui pesan yang ada di Instagram atau WhatsApp. Untuk pemesanan itu penjual mengalami permasalahan yang cukup besar yaitu sulitnya memverifikasi pembayaran yang dilakukan oleh pembeli karena hanya menggunakan transfer manual ke rekening penjual. Hal ini mengakibatkan Byboot.id kehilangan banyak pembeli karena terdapat pembeli yang mengeluh proses verifikasi transfer ini. Dapat dibuktikan pada data pemesanan yang dicatat secara manual pada kuartal 4 tahun 2021 yang terdapat 10 pesanan dibatalkan akibat terkendala masalah pembayaran.

Byboot.id merupakan sebuah Usaha Mikro Kecil Menengah (UMKM) di bidang *fashion* yang focus pada pembuatan produk sandal, khususnya sandal wanita. Berada di desa Plemahan, Kecamatan Sumobito, Kabupaten Jombang. Byboot.id berdiri sejak tahun 2018 dan telah menjual produk baik secara *offline* maupun *online.* Penjualan secara *online* hanya melalui *platform* Whatsapp dan Instagram. Transaksi *online* ini baru dimulai sejak tahun 2020. Namun, penjualan *online* yang dilakukan lewat platform tersebut dinilai kurang efektif dikarenakan jangkauannya kecil. Pemilik menginginkan sebuah media untuk memperkenalkan produk mereka secara khusus. Sehingga calon pembeli bisa melihat produk mereka secara detail dan bisa melihat katalog mereka secara lebih lengkap.

Selain itu, saat ini Byboot.id mengalami permasalahan yang cukup besar yaitu pada proses verifikasi pembayaran yang masih manual yaitu harus mengirimkan bukti transfer ke admin. Proses manual ini bisa mengakibatkan lamanya waktu tunggu oleh pembeli dan bisa mengakibatkan pembeli membatalkan pembeliannya.

Untuk mengatasi permasalah diatas, Byboot.id memerlukan solusi untuk mengatasi masalah yang terjadi pada proses pembayaran. Diperlukan berbagai macam jenis pembayaran untuk memudahkan pengguna dalam melakukan transaksi. Midtrans sebagai *Payment Gateway* merupakan bentuk solusi dari permasalahan tersebut. *Payment Gateway* merupakan sitem yang menjembatani situs web dan lembaga keuangan untuk melakukan proses transaksi secara online. Midtrans merupakan salah satu layanan *payment gateway* yang terkemuka di Indonesia. Midtrans adalah sistem pembayaran yang memfasilitasi penjual dan pembeli untuk melakukan transakasi jual beli (Yuninda, 2019). Serta untuk mendapatkan *website* yang bisa memenuhi kebutuhan pemilik untuk memudahkan pembeli untuk melihat produk mereka, dan juga membuat *website* yang dinamis untuk memudahkan pemilik dan pembeli untuk melakukan transaksi dibutuhkan sebuah *framework Php* yang memiliki fitur untuk membuat *website* tersebut. Laravel merupakan salah satu *framework Php* yang dapat memenuhi semua kebutuhan tersebut dengan aman, cepat, dan sederhana (Nugroho & Nugroho, 2021).

Berdasarkan uraian diatas, pemilik Byboot.id membutuhkan sebuah website yang dapat mencatat penjualan produk secara otomatis dan menerapkan sistem pembayaran melalui *payment gateway* agar dapat melakukan transaksi secara online selama 24 jam. Pengembangan *E-Commerce* ini menggunakan *framework* Laravel dan memanfaatkan layanan *payment gateway* Midtrans.

## Rumusan Masalah

1. Bagaimana analisis kebutuhan *e-commerce* Byboot.id dengan memanfaatkan layanan API *Payment Gateway* Midtrans?
2. Bagaimana rancangan dan implementasi *website e-commerce* Byboot.id ke dalam *framework* Laravel?
3. Bagaimana implementasi fitur Midtrans pada proses pembayaran pada *website e-commerce* Byboot.id?

## Tujuan

Tujuan dari penelitian ini, yaitu:

Mendapatkan hasil analisis kebutuhan *website e-commerce* Byboot.id yang memanfaatkan layanan *API Payment Gateway Midtrans.*

Mendapatkan hasil perancangan dan implementasi *website e-commerce* Byboot.id ke dalam *framework* Laravel dengan memanfaatkan *API Payment Gateway Midtrans.*

Mendapatkan hasil implementasi pada proses pembayaran pada *website e-commerce* dengan menggunakan Midtrans.

## Manfaat

Manfaat dari penelitian ini, yaitu:

Memberikan informasi kepada calon pembeli mengenai produk yang disediakan oleh Byboot.id.

Meringankan pekerjaan pemilik took sepatu Byboot.id pada saat proses verifikasi pembayaran dan juga pada saat pencatatan pesanan.

Memberikan kemudahan pada pembeli untuk melakukan transaksi pembelian serta memberikan banyak metode pembayaran yang dapat dipilih.

## Batasan Masalah

Batasan masalah berdasarkan permasalahan diatas, yaitu:

Penelitian ini hanya akan sampai pada pengembangan sistem, tanpa melanjutkan hingga sampai pemeliharaan sistem.

Aplikasi ini ditujukan untuk digunakan oleh pemilik dan pembeli dari toko sepatu Byboot.id.

Implementasi dari sistem ini ditujukan untuk mengelola produk, pesanan masuk, dan konfirmasi pembayaran yang masuk secara otomatis.

Penelitian ini menggunakan bahasa pemrograman PHP dan *framework* Laravel.

Integrasi *payment gateway* pada aplikasi ini dilakukan dengan menggunakan *API sandbox* dari midtrans.

## Sistematika Pembahasan

Sistematika pembahasan dari penelitian ini dibagi menjadi tiga bab untuk memudahkan pembaca untuk memahami alur pembahasan pada penelitian ini

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang dari permasalahan yang dihadapi oleh Byboot.id yang kemudian diangkat menjadi, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan permasalahan, dab sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas mengenai teori-teori yang mendukung dan dijadikan referensi penelitian untuk membuat *website e-commerce* toko sepatu *Byboot.id.*

BAB 3 METODOLOGI

Bab ini berisi tentang metode penelitian yang digunakan untuk menyelesaikan penelitian pembuatan *website e-commerce* toko sepatu Byboot.id.

# LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi kajian pustaka, profil perusahaan serta landasan teori. Kajian Pustaka juga berisi ringkasan penelitian sebelumnya yang mendukung penelitian ini dan digunakan sebagai acuan oleh peneliti untuk menyelesaikan permasalahan. Profil perusahaan berisi detail profil toko sepatu Byboot.id. Landasan teori berisi teori-teori yang mendukung dan berhubungan dengan topik yang sedang diteliti.

## Kajian Pustaka

Kajian Pustaka yang menjadi landasan untuk penelitian ini akan digunakan untuk membahas penelitian terdahulu yang terkait dengan penelitian ini mengenai pengembangan sistem informasi dengan menggunakan layanan *web service.*

Penelitian pertama adalah penelitian yang dilakukan oleh Chairunnisa Dwinanda Asti, Widhy Hayuhardika Nugraha Putra dan Welly Purnomo pada tahun 2021 dengan judul “Pengembangan Website E-Commerce dengan Pemanfaatan Sistem *Payment Gateway* Midtrans (Studi Kasus: Butik Rizza Collection)”. *Website* ini dibangun untuk memudahkan para pelanggan dari butik untuk melakukan pemesanan busana *request* yang sebelumnya harus dilakukan dengan cara datang langsung ke toko, menjadi hanya harus memesan melalui *website* saja. Kemudian untuk sistem pembayaran terintegrasi dengan *payment gateway* Midtrans, sehingga proses verifikasi pembayaran dilakukan secara otomatis.

Penelitian kedua adalah penelitian yang dilakukan oleh Erikson Damanik pada tahun 2012 dengan judul “Perancangan Sistem Informasi Pembayaran *Online* Menggunakan *Payment Gateway*”. Penelitian ini terkait tentang sistem pengamanan yang dilakukan oleh sistem *payment gateway* yaitu menggunakan SSL (*Secure Socket Layer*). Peneliti akan membandingkan *hash* dikirim oleh penjual dengan *hash* yang dikirim oleh *merchant*. Dengan mengetahui perbandingan tersebut maka peneliti bisa mengetahui bagaimana komunikasi antara *e-commerce* dengan sistem *payment gateway* yang digunakan.

Penelitian ketiga adalah penelitian yang dilakukan oleh Yuninda Eka Nisrina, Widhy Hayuhardika Nugraha Putra dan Buce Trias Hanggara pada tahun 2019 dengan judul “Pengembangan *E - Commerce* Dengan Pemanfaatan Sistem *Payment Gateway* (Studi Kasus: Wisata Kampung Sapi *Adventure*)”. Penelitian ini terkait tentang pengembangan sistem informasi dari Wisata Kampung Sapi *Adventure.* Pengembangan sistem informasi ini dilakukan untuk memudahkan transaksi pembelian tiket masuk oleh pelanggan, yang sebelumnya harus dilakukan secara manual dengan mengirimkan foto bukti *transfer* kepada admin yang kemudian dilakukan pendataan secara manual. Hal ini dapat mengakibatkan data yang tersimpan tidak valid karena proses verifikasi dilakukan secara manual. Untuk mengatasi permasalahan tersebut, peneliti menggunakan integrasi dengan sistem *payment gateway* dengan menggunakan konsep SOA (*Service Oriented Architecture*). Dengan menggunakan solusi tersebut proses verifikasi akan dilakukan secara otomatis, sehingga memudahkan pihak petugas dari Wisata Kampung Sapi *Adventure.*

Penelitian keempat adalah penelitian yang dilakukan oleh Hardhani Eko Nugroho dan Agung Nugroho pada tahun 2021 dengan judul “Analisis Dan Perancangan E-Commerce pada Toko Sepatu Dope13store Menggunakan Framework Laravel”. Penelitian ini terkait dengan pembuatan sebuah *website e-commerce* toko sepatu untuk meningkatkan penjualan dan efektivitas penyampaian informasi kepada pelanggan atau calon pelanggan toko sepatu Dope13store. Untuk mengatasi hal tersebut peneliti menggunakan *framework* Laravel untuk membuat *website e-commerce* karena dengan menggunakan Laravel dianggap bisa memaksimalkan penggunaan PHP dan mempercepat proses pembuatan *website.* Dengan pembuatan *website e-commerce* ini diharapkan dapat membantu efektivitas dan efisien kerja dalam proses penyampaian informasi serta proses penjualan dari toko sepatu Dope13store.’

Penelitian kelima adalah penelitian yang dilakukan oleh Dony Aditya Putra, Gusti Made Arya Sasmita dan AA.Kt.Agung Cahyawan Wiranatha pada tahun 2020 dengan judul “*E-Commerce Marketplace Petshop* Menggunakan Integrasi Rajaongkir API dan iPaymu *Payment Gateway* API”. Penelitian ini terkait dengan pengembangan sistem informasi berbasis *website* dengan menggunakan *framework* CodeIgniter, Google Maps API, RajaOngkir API dan iPaymu sebagai sistem *payment gateway.* Dengan menggunakan API dari RajaOngkir dapat memudahkan pelanggan mengecek besaran ongkos kirim yang dibebankan berdasarkan alamat yang dipilih. Dengan menggunakan fitur yang disediakan oleh pihak ketiga yang digunakan oleh peneliti diharapkan dapat memudahkan transaksi dari *Petshop.*

Penelitian keenam adalah penelitian yang dilakukan oleh Reza Yusuf Maulana pada tahun 2021 dengan judul “Pemanfaatan API Midtrans sebagai *Payment Gateway* pada Pembuatan *Website E-Commerce* Menggunakan *Framework* Laravel (Studi Kasus : Chesspedia)”. Penelitian ini terkait dengan pengembangan dari *website e-commerce* penjualan permainan catur yang sudah ada namun dinilai kurang efektif dalam hal verifikasi pembayaran karena masih dilakukan secara manual. Untuk mengatasi hal tersebut peneliti menggunakan sistem *payment gateway* Midtrans sebagai aplikasi yang menjembatani pembayaran pelanggan dengan toko Chesspedia, sehingga proses verifikasi pembayaran dilakukan secara otomatis. Kemudian untuk pengujian fungsional dari sistem yang telah dibangun menggunakan metode *blackbox* dengan hasil 100% valid untuk semua fitur dari *website.* Pengujian non-fungsional menggunakan *System Usability Scale* (SUS) dengan responden pengguna *website,* hasilnya mendapatkan *score* 74 dengan kategori *acceptable.*

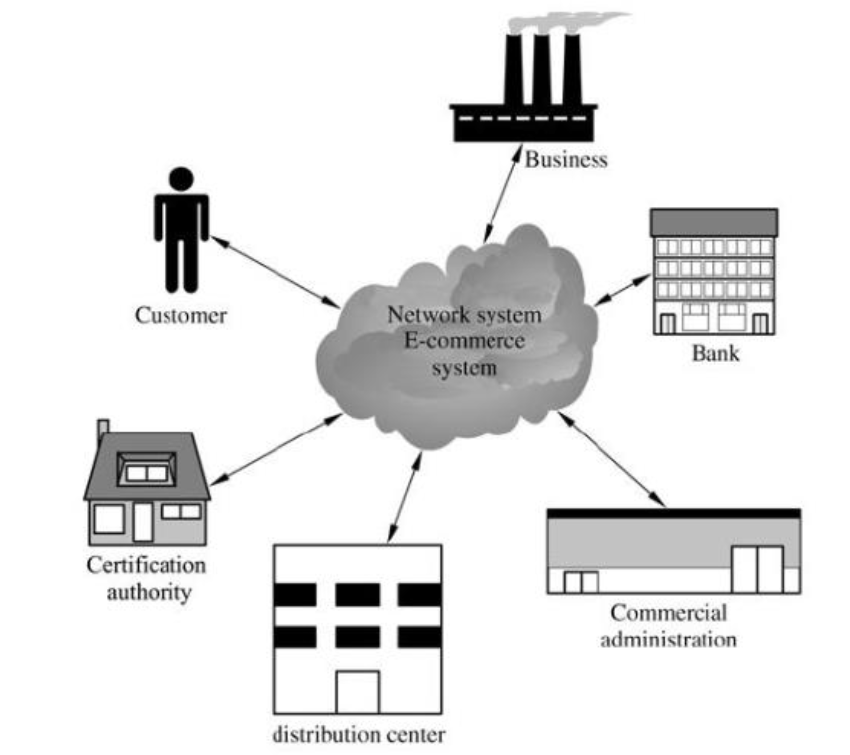
Penelitian ketujuh dalah penelitian yang dilakukan oleh Fariz Reynaldo Pratema, Nurudin Santoso dan Lutfi Fanani pada tahun 2020 dengan judul “Pengembangan Aplikasi E-Commerce Menggunakan Payment Gateway Midtrans”. Penelitian ini terkait tentang pembuatan aplikasi toko online dengan memanfaatkan sistem *payment gateway* Midtrans. Pembuatan aplikasi ini menggunakan metode *Waterfall Model.* Hal ini dilakukan karena kebutuhan sistem yang dari awal sudah jelas. Aplikasi yang dibuat oleh peneliti ini dikembangkan untuk *platform android* akan tetapi memungkinkan untuk pengembangan lebih lanjut untuk *platform* lain seperti IOS atau *web.* Dengan penelitian ini diharapkan dapat mengatasi permasalahan yang dialami toko Anfield *Collection* yang menjual berbagai kerajian serta meningkatkan efisiensi dan efektivitas proses bisnis dan pemasaran.

## Byboot.id

Byboot.id merupakan sebuah toko yang bergerak dibidang fashion dengan focus menjual produk sandal dan sepatu wanita. Toko ini sudah berdiri sejak tahun 2018 yang hingga saat ini masih melayani pembelian via *direct message* Instagram dan pesan Whatsapp saja. Terletak di desa Plemahan, kecamatan Sumobito, kabupaten Jombang. Seiring berjalannya waktu dan banyaknya *e-commerce,* Byboot.id juga berencana membuat sebuah *website e-commerce* untuk meningkatkan penjualan produk mereka dan juga untuk mempermudah proses transaksi dan pencatatan penjualan.

## E-Commerce

Menurut Kozinets et al,. (2010) *e-commerce* merupakan segala jenis transaksi seperti penjualan, pembelian, transfer, ataupun pertukaran produk yang dilakukan secara *online.* Menurut Kotler dan Armstrong (2018) *e-commerce* adalah media yang bisa diakses oleh pebisnis dalam melakukan aktivitas bisnis dan digunakan konsumen untuk mendapat informasi dengan bantuan computer dan jaringan internet yang berfungsi untuk memberikan konsumen bantuan dalam proses penentuan pilihan.Strategi bisnis untuk *e-commerce* dapat berhasil dengan mengambil bentuk dari proses bisnis tradisional dan diaplikasikan melalui sosial media melalui jaringan internet (Pradana, 2015). Terdapat berbagai bentuk *e-commerce* yang saat ini banyak digunakan seperti toko *online,* pemesanan tiket, reservasi hotel atau penginapan sampai dengan pemesanan tiket wisata. Menurut Qin (2009) *e-commerce* memiliki beberapa komponen utama seperti yang terdapat pada gambar 2.2.



Gambar 2.1 Komponen *E-Commerce*

Sumber: (Qin, 2009)

Gambar 2.1 merupakan penjelasan dari komponen utama dari sebuah *e-commerce.*

*Network System*

*Network* atau jaringan yang dimaksud disini merupakan internet atau intranet. Internet adalah jaringan dasar yang digunakan oleh *e-commerce* untuk menyampaikan informasi bisnis komersial. Intranet adalah jaringan yang digunakan bisnis secara local untuk urusan internal.

*Customer*

*Customer* disini yang dimaksud adalah pengguna dari *e-commerce.* Pengguna dibedakan menjadi konsumen pribadi dan konsumen bisnis. Konsumen bisnis bertugas untuk mengelola semua bagian yang ada pada bisnis *e-commerce* seperti staf, penghasilan, barang, produksi, pasokan, dan penjualan melalui internet, dan *Management Information System (*MIS*).*

*Certification Authority*

CA disini merupakan CA yang sudah diakui secara hukum. Bertanggung jawab untuk menerbitkan, mengelola sertifikat digital dan memfasilitasi pihak-pihak yang terlibat dalam penjualan *online* untuk mengidentifikasi satu sama lain.

*Online Bank*

*Online Bank* berfungsi untuk memberikan layanan pada penjualan untuk memfasilitasi transaksi sepanjang waktu.

*Commercial Administration*

Administrasi dari kegiatan komersial utama terdiri dari departemen industry, bea cukai, pajak dan perdagangan.

*Distribution Center*

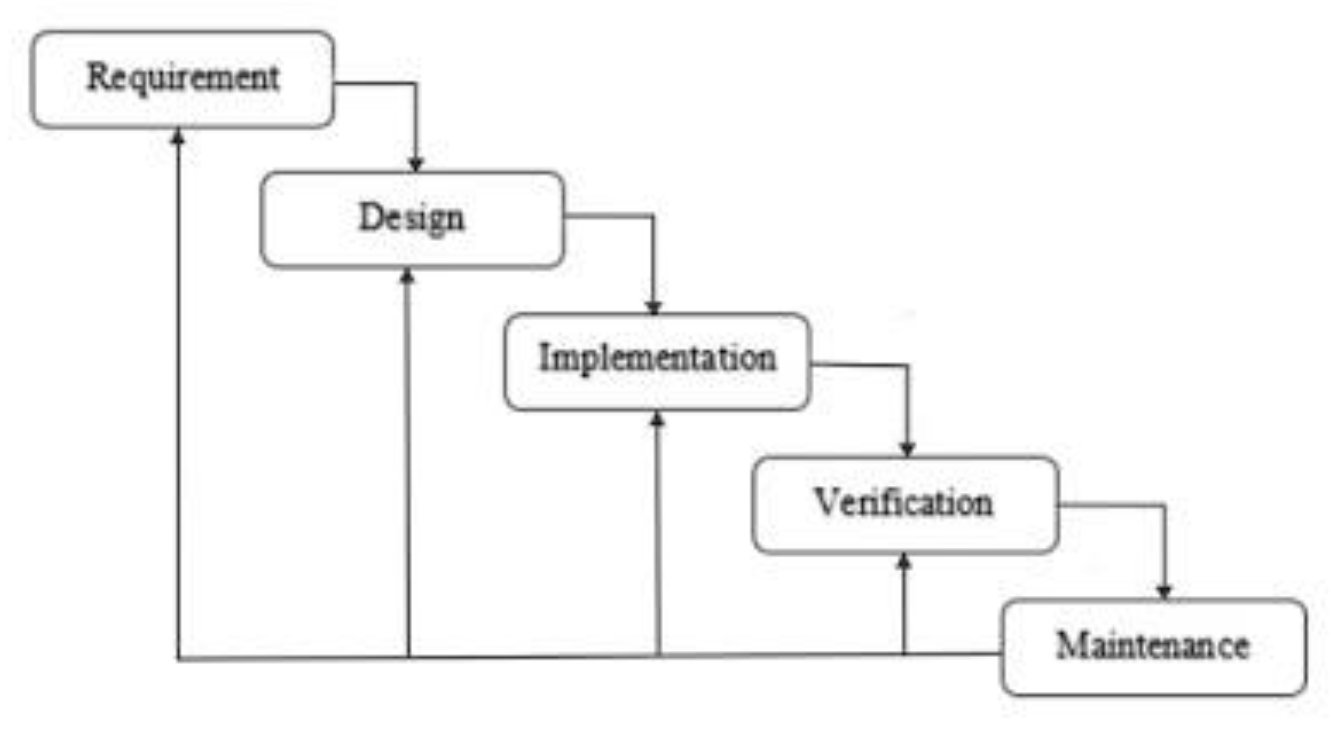
Pusat distribusi bertugas dan bertanggung jawab untuk mengirim barang ke konsumen dan melacak *traffic* barang.

## *Software Development Life Cycle*

SDLC memiliki banyak model yang penggunaannya menyesuaikan kebutuhan dan keadaan proyek yang sedang dihadapi sehingga pengembang bisa memilih model mana yang lebih efektif. Setiap model memiliki keunikan dan ciri khas masing-masing, sehingga penggunaannya bisa meningkatkan keberhasilan pada proses pengembangan suatu aplikasi (Pressman, 2012).

### *Waterfall Model*

*Waterfall model* merupakan salah satu *classic life cycle.* Pendekatan dari model ini memiliki ciri khas yaitu sistematis, berurutan dalam pengembangan suatu aplikasi. Tahapan dimulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga distribusi pada *user.* Tahapan tadi dapat dilihat pada gambar 2.2.



Gambar 2.2 Diagram *Waterfall Model*

Sumber: (Pressman, 2012)

Dalam pengembangan suatu aplikasi, *waterfall* memiliki beberapa tahapan yang berurutan yaitu: *requirement, system design, coding & testing,* Penerapan program, pemeliharaan. Berikut adalah penjelasan dari masing-masing tahapan.

*Requirement Analyst*

Tahapan ini bertujuan untuk memahami aplikasi yang diharapkan oleh *user* dan Batasan perangkat lunak yang menggunakannya. Informasi ini di++bisa diperoleh dari wawancara, diskusi, atau survey secara langsung. Kemudian informasi diolah dan dianalisis untuk mendapatkan data yang berguna untuk pengembang.

*System Design*

Dari informasi yang didapat dari tahap sebelumnya kemudian menjadi spesifikasi kebutuhan dan dipelajari kemudian menyiapkan desain sistem pada tahap ini. Desain sistem akan menentukan spesifikasi perangkat keras yang dibutuhkan dan menjadi dasar dari arsitektur sistem secara menyeluruh.

*Implementation*

Tahap ini merupakan awal dari sistem dikembangkan. Kemudian menjadi unit-unit kecil, unit kecil ini akan dikembangkan masing-masing dan diuji fungsionalitas yang disebut sebagai *unit testing.*

*Integration & Testing*

Pada tahap ini seluruh unit akan diintegrasikan ke dalam sistem setelah melalui uji fungsionalitas. Setelah integrasi selesai, sistem akan diuji untuk mengecek setiap kegagalan atau kesalahan.

*Operation & Maintenance*

Pada tahap ini perangkat lunak sudah selesai dikembangkan, dijalankan serta dilakukan pemeliharaan. Pemeliharaan disini termasuk merperbaiki masalah yang tidak ditemukan pada tahap sebelumnya. Perbaikan sistem dan peningkatan jika ada kebutuhan baru juga termasuk dalam tahap ini.

## Unified Modelling Language (UML)

*Unified Modelling Language* (UML) merupakan suatu metode yang digunakan secara visual untuk merancang sebuah sistem berorientasi pada objek (Sora, 2015). Terdapat beberapa jenis diagram UML yang dapat menggambarkan sistem yaitu *activity diagram, class diagram, use case diagram,* dan *sequence diagram.*

### *Activity Diagram*

*Activity diagram* merupakan diagram yang menggambarkan proses yang dilakukan oleh sebuah sistem secara berurutan (Jeffrey Whitten, 2007). *Activity diagram* berfungsi untuk menggambarkan proses bisnis secara berurutan dalam sebuah proses, memperlihatkan urutan kegiatan proses pada sebuah sistem, dibuat berdasarkan *use case* pada *use case diagram.*

### *Use Case Diagram*

Use case diagram merupakan diagram yang menggambarkan hubungan antara aktor dengan sistem (Michael, 2019). Use case diagram bisa mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.

### *Sequence Diagram*

Menurut Desy (2018) *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display/form*) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan scenario atau rnagkaiann Langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali denga napa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

### *Class Diagram*

*Class diagram* menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain (Desy, 2017). Class Diagram memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka. Class Diagram bersifat statis; menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan.

### *Physical Data Model*

*Phisycal data model* menggambarkan struktur asli dari sebuah database berupa tabel dan kolom, atau pesan antara proses komputer (Matthew, 2011). Struktur model data akan sering disesuaikan dengan kebutuhan dari proses yang bekerja untuk memastikan bisa bekerja dengan optimal.

## Teknologi Pengembangan Sistem

### *Web Service*

*Web service* adalah kumpulan dari *method* yang ada pada sebuah server yang bisa dipanggil oleh klien untuk melakukan sebuah perintah (Kapojos, 2012). *Method* yang ada pada server ini bisa dipakai dalam bahasa permograman apapun sesuai yang diinginkan, namun kita harus menyesuaikannya. Menurut Marthasar dkk (2010) terdapat 3 komponen *web service* yaitu:

1. *Extensible Markup Language* (XML)

XML adalah sebuah teks yang digunakan untuk berkomunikasi antar aplikasi dalam *web services.* Karena XML berbentuk teks maka XML mudah untuk ditransportasikan menggunakan protocol HTTP.

1. *Simple Object Access Protocol* (SOAP)

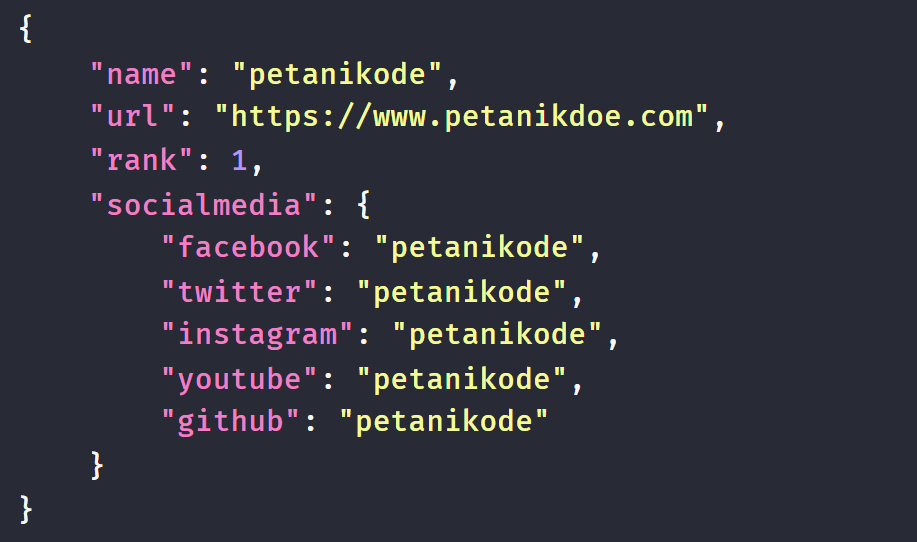
SOAP merupakan sebuah format dokumen yang digunakan oleh *web service* dalam bentuk XML untuk melakukan proses *request* dan *response* antara *web service* dengan aplikasi yang memanggilnya*.*

1. *Web Service Definition Language* (WSDL)

WSDL juga merupakan format dokumen dalam bentuk XML yang isinya merupakan informasi secara detail mengenai sebuah *web service* seperti parameter dari sebuah *method,* hasil, dan tipe data.

### *Javascript Object Notation* (JSON)

JSON merupakan sebuah format dari data yang biasa digunakan untuk pertukaran informasi atau penyimpanan data (Muhardian, 2022). JSON lebih popular daripada XML karena lebih mudah dimengerti dan dibaca oleh orang awam. JSON juga bisa dibaca dan dipakai oleh berbagai macam bahasa pemrograman yang saat ini sering digunakan. Berikut merupakan contoh penulisan JSON pada gambar 2.3.



Gambar 2.3 Penulisan JSON

Sumber: (PetaniKode, 2022)

### PHP

PHP merupakan bahasa pemrograman yang bisa terhubung dengan basis data, *file* dan folder sehingga bisa membuat sebuah *website* bisa menampilkan konten yang dinamis (Yulianto, 2007). PHP termasuk bahasa pemrograman yang bisa berjalan pada berbagai sistem operasi seperti windows, linux, ataupun mac. Program PHP ditulis dalam teks biasa dengan format nama mempunyai akhiran “.php”.

### Laravel

Laravel adalah salah satu *framework* PHP yang saat ini sedang popular digunakan dikalangan *web developer.* Laravel merupakan *framework* PHP yang dikembangkan oleh Taylor Otwell pada tahun 2011, Otwell mengembangkan Laravel karena pada saat itu dia tidak menemukan *framework* PHP yang cocok untuk aplikasi yang sedang dikembangkannya (Yudhanto dan Prasetyo, 2018). Laravel pada dasarnya menekankan prinsip kesederhanaan dan fleksibilitas dalam penggunaannya, juga menggunakan konsep *Model, View, Controller* (MVC) seperti *framework* PHP lainnya.

### Midtrans

Midtrans merupakan salah satu sistem *payment gateway* yang popular di Indonesia. Midtrans adalah sistem pembayaran yang menyediakan fasilitas bagi penjual dan pembeli untuk melakukan transaksi (Midtrans (PT Midtrans), 2019). Midtrans menyediakan berbagai metode pembayaran yang dapat dipilih oleh pembeli sesuai dengan keinginan mereka. Midtrans juga menyediakan *tools* yang bisa terintegrasi langsung pada *e-commerce* untuk memantau penjualan produk sampai mengecek produk yang terjual.

### RajaOngkir

RajaOngkir merupakan sebuah *website* yang menyediakan fitur untuk mengecek besaran ongkos kirim ke seluruh wilayah di Indonesia menggunakan jasa ekspedisi seperti JNE, J&T, POS Indonesia, TIKI, ESL, dan RPX. Selain hanya digunakan untuk mengecek ongkir bagi para pengembang, RajaOngkir juga menyediakan API untuk digunakan mengecek ongkir bagi pelanggan *e-commerce* yang menggunakan jasa API dari RajaOngkir. Penggunaan API dari RajaOngkir ini cukup mudah karena hanya diperlukan API key dan keyword seperti kota asal, kota tujuan, dan berat dalam kilogram. Dengan *input* tersebut RajaOngkir bisa mengirim *response* jenis kurir yang tersedia hingga harga ongkirnya.

## Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan bagian yang sangat penting dari pengembangan suatu aplikasi karena untuk melakukan perbaikan dan pengukuran dari kualitas aplikasi yan sedang dibangun (Rouf, 2012). Pengujian juga dapat memberikan perspektif baru secara objektif dan independent yang bisa dimanfaatkan untuk melihat resiko dan benefit dari pengimplementasian teknologi yang digunakan.

### *Blackbox Testing*

*Black box testing* atau biasa disebut dengan tes fungsional adalah sebuah teknik yang digunakan untuk melakukan tes berdasarkan informasi yang terdapat pada spesifikasi (Nidhra dan Dondeti, 2012). Dengan menggunakan metode ini orang yang melakukan tes tidak akan mempunyai akses ke *source code* dari perangkat lunak yang sedang di tes.

### *User Acceptance Testing* (UAT)

*User acceptance testing* (UAT) merupakan sebuah proses untuk memverifikasi apakah solusi dari suatu permasalahan sudah bekerja sesuai dengan yang *user* inginkan (Otaduy et al., 2017). UAT ini dilakukan untuk menentukan apakah sistem yang telah dibangun sudah memenuhi kebutuhan pengguna dan mendukung semua scenario bisnis. Karena itu UAT dilakukan oleh *client* dan *end-user.* Menurut situs Softwaretesting.com (2022) terdapat 3 cakupan proses untuk melakukan UAT, yaitu:

Perencanaan

Perencanaan harus dilakukan diawal proses, karena jika dilakukan diakhir dikhawatirkan akan terjadi masalah yang menyebabkan terlambatnya pengujian sehingga penyelesaian proyek akan mundur dari jadwal. Perencanaan ini bisa berupa penyusunan *test plan*.

Persiapan

Persiapan pada UAT ini mencakup mengumpulkan data pengujian yang biasanya cukup rumit dan membutuhkan sumber daya yang cukup besar. Selain itu juga harus memastikan bahwa perangkat yang dilakukan untuk pengujian adalah perangkat yang digunakan adalah perangkat yang biasa digunakan sehari-hari.

Pengelolaan dan Eksekusi

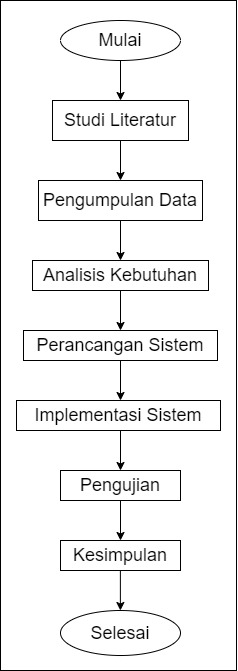
Setelah kedua proses diatas dilakukan maka selanjutnya adalah eksekusi pengujian itu sendiri, *user* yang melakukan pengujian bertanggung jawab untuk mengidentifikasi kasus yang dites, membuat data tes, dan melakukan UAT. Setelah pengujian dilakukan maka *user* bisa menyimpulkan apakah sistem sudah memenuhi semua persyaratan atau tidak. Apabila terjadi kesalahan maka akan dicatat dan tim akan melacak kesalahan tersebut untuk diperbaiki.

### *Compability Testing*

*Compability testing* atau pengujian kompabilitas adalah pengujian yang dilakukan untuk memastikan bahwa sistem yang dibangun dapat berjalan dengan baik di berbagai macam perangkat (Yusuf, 2021). Untuk aplikasi *website* maka pengujian dilakukan pada perangkat dengan berbagai macam ukuran dan berbagai macam *browser.*

# METODOLOGI

Bab ini akan menjelaskan tentang metodologi yang digunakan untuk penelitian ini mulai dari identifikasi masalah, analisis kebutuhan, perancangan sistem, dan pengambilan kesimpulan. Berikut merupakan diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

## Studi Literatur

Proses yang digunakan untuk melakukan tinjauan teori atau studi literatur yang bertujuan untuk membantu penelitian. Teori-teori yang digunakan yaitu:

Penelitian terdahulu yang berkaitan tentang implementasi *Payment Gateway Midtrans.*

Byboot.id

Proses rekayasa perangkat lunak

*Unified Modelling Language* (UML) yang berupa *use case diagram* dan *class diagram.*

Teknologi pengembangan sistem menggunakan teknologi REST pada *payment gateway* Midtrans sebagai *web service,* PHP, dan Laravel sebagai sistem *website.*

## Pengumpulan Data

Proses awal dari analisis kebutuhan berupa pengumpulan data yang akan dijadikan dasar dari penelitian. Terdapat 2 jenis data berdasarkan sumbernya yaitu data primer dan data sekunder.

Data primer merupakan data yang didapatkan langsung dari sumbernya. Data ini diperoleh dari proses wawancara pada pemilik Byboot.id mengenai permasalahan yang dihadapi.

Data sekunder merupakan data hasil olahan dan proses dari pihak luar, yang bisa berupa jurnal atau penelitian yang terkait. Data sekunder disini diperoleh dari jurnal, yakni jurnal yang membahas tentang implementasi sistem pembayaran menggunakan *payment gateway* Midtrans.

## Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan dari sistem yang akan dibangun. Analisis kebutuhan dimulai dari mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat dalam sistem. Data dari analisis kebutuhan diperoleh dari analisis hasil wawancara yang sebelumnya sudah dilakukan sehingga bisa mendapatkan sebuah pemahaman yang sama antara pihak pengembang sistem dan pemilik Byboot.id. Sampai akhirnya bisa diketahui fungsional, data dan dokumen yang dibutuhkan untuk menyelesaikan permasalahan yang sedang dihadapi.

## Perancangan Sistem

Perancangan sistem dilakukan untuk mengetahui tahapan-tahapan yang harus dilakukan berdasarkan hasil analisis kebutuhan yang sudah dilakukan agar menghasilkan rancangan dan implementasi sistem yang sesuai. Permodelan dapat menggunakan *unified modelling language* (UML) sebagai alat bantu untuk membuat rancangan *e-commerce*. UML yang digunakan bisa seperti *class diagram* dan *use case diagram.*

## Implementasi Sistem

Implementasi sistem menggunakan pengkodean baris kode (*source code*). Implementasi akan dilakukan sesuai dengan hasil analisis kebutuhan dan perancangan sistem yang sudah dilakukan. Bahasa pemrograman yang digunakan dalam implementasi sistem ini antara lain PHP, HTML, Javascript dan CSS.

### Implementasi *E-commerce*

Implementasi pembuatan *website e-commerce* toko sepatu Byboot.id ini menggunakan integrasi antara layanan *payment gateway* Midtrans dan *framework* Laravel. Laravel digunakan untuk mempermudah pembuatan *website* dan untuk implementasi antarmuka akan menggunakan Bootstrap.

### Implementasi *Payment Gateway*

Implementasi *payment gateway* pada *website e-commerce* toko sepatu Byboot.id ini akan menggunakan API dari salah satu penyedia layanan sistem pembayaran local yaitu Midtrans. Integrasi antara *website e-commerce* dan Midtrans ini akan mempermudah pemilik dan pembeli dari toko septau Byboot.id untuk melakukan transaksi, dan pencatatan transaksi.

## Pengujian

Pengujian sistem dari *website e-commerce* toko sepatu Byboot.id akan menggunakan metode *blackbox.* Pengujian akan dilakukan saat pembuatan *e-commerce* sudah selesai. Pengujian ini dilakukan untuk melacak *error* atau kesalahan yang ada pada *website.* Selain itu juga untuk menghindari adanya kesalahan yang mungkin bisa ditemukan oleh pengguna sebelum diluncurkan. Pengujian juga akan menggunakan *compatibility test* agar sistem informasi bisa diakses oleh berbagai device dan *browser.* Serta akan dilakukan UAT untuk memastikan bahwa *website* bisa memenuhi kebutuhan dan mengatasi permasalah yang terjadi.

## Kesimpulan dan Saran

Tahap akhir dari pembangunan sistem informasi *e-commerce* toko sepatu Byboot.id yakni pengambilan kesimpulan dan saran. Tahap ini dilakukan untuk menyimpulkan dari permasalahan yang dihadapi. Selain itu kesimpulan juga akan diberikan saran yang dapat digunakan untuk pengembangan sistem informasi dalam penelitian selanjutnya.

# ANALISIS KEBUTUHAN

Analisis kebutuhan bertujuan untuk mengetahui dan memberikan gambaran tentang sistem yang harus terpenuhi oleh pengguna dari sistem yang akan dibangun. Kemudian hasil dari analisis kebutuhan ini akan dijadikan sebagai acuan untuk pembuatan sistem informasi *e-commerce* Byboot.id. Pembahasan bab ini akan dibagi menjadi deskripsi umum sistem, pemodelan proses bisnis, identifikasi actor, identifikasi kebutuhan dan pemodelan kebutuhan.

## Deskripsi Umum Sistem

Sistem yang akan dibangun merupakan sebuah *website e-commerce* yang bisa digunakan oleh pelanggan untuk berbelanja dan melakukan transaksi melalui *website* ini dan juga dapat digunakan oleh admin untuk mengelola produk dan pesanan yang masuk dari pelanggan. Pelanggan dapat langsung mengakses *website* dan melihat katalog produk, deskripsi, harga dan gambar secara detail tanpa harus melakukan pendaftaran ke *website.* Jika pelanggan ingin melakukan transaksi maka pelanggan diharuskan melakukan pendaftaran terlebih dahulu. Saat melakukan *checkout,* pelanggan dapat memilih kurir pengiriman dan durasi yang diinginkan. Kemudian untuk pembayaran, pelanggan juga bebas memilih berbagai jenis metode pembayaran yang disediakan seperti melalui dompet digitak OVO, Gopay, dan Shoopepay, lalu juga melalui transfer bank, kartu kredit, dan kartu debit.

Admin dapat melakukan pengelolaan produk lewat panel admin yang disediakan saat admin melakukan *login.* Pada panel ini, admin dapat melihat semua pesanan yang masuk, dan dapat mengubah status pesanan tersebut jika pembayaran sudah terverifikasi oleh *payment gateway* Midtrans. Saat pesanan sudah disetujui admin akan melanjutkan proses pesanan menuju proses pengiriman dan menambahkan nomor resi dari ekspedisi sesuai dengan pilihan pelanggan. Selain melakukan pengelolaan pesanan admin juga dapat mengelola produk seperti menambah produk, mengubah produk, dan menghapus produk.

## Identifikasi Aktor

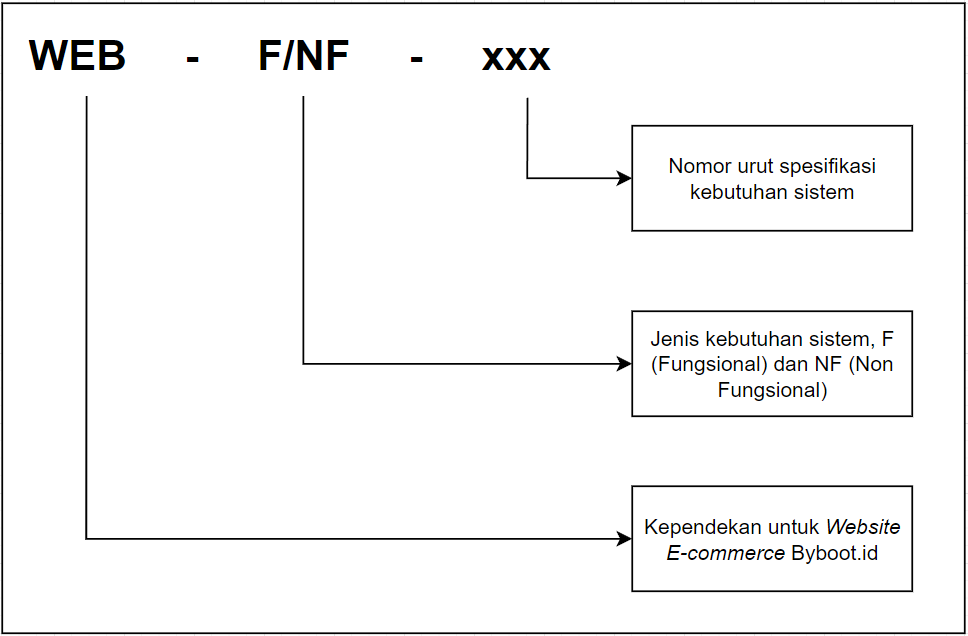
Actor yang melakukan interaksi pada sistem. Pada sistem informasi ini actor yang terlibat adalah pengguna. Dimana pengguna adalah entitas manusia yang melakukan interaksi dengan sistem ini. Penjelasan actor yang akan berinteraksi akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

|  |  |  |
| --- | --- | --- |
| No. | Aktor | Deskripsi |
| 1 | Pengunjung | Merupakan orang melakukan kunjungan pada *website e-commerce* Byboot.id |
| 2 | Pelanggan | Merupakan pengunjung yang sudah melakukan pendaftarandan masuk pada *website e-commerce* Byboot.id |
| 3 | Admin | Memiliki hak untuk mengelola dan mengatur segala sesuatu yang menyangkut *website e-commerce* Byboot.id |

## Identifikasi Kebutuhan

Identifikasi kebutuhan merupakan tahap untuk memperoleh daftar kebutuhan apa saja yang diperlukan oleh sistem yang akan dibangun. Dalam identifikasi kebutuhan terdapat dua jenis kebutuhan yaitu kebutuhan fungsional dan non fingsional. Penamaan kode dapat dilihat pada Gambar 4.1.



Gambar 4.1 Kode Kebutuhan

### Kebutuhan Fungsional

Kebutuhan fungsional akan dibagi berdasarkan jenis actor yang terlibat. Kebutuhan fungsional pada sistem informasi *website e-commerce* Byboot.id akan dibagi menjadi dua sesuai dengan jumlah actor, yaitu untuk kebutuhan fungsional pengunjung dan pelanggan, serta kebutuhan fungsional untuk admin sebagaimana akan ditunjukkan pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.2 Kebutuhan Fungsional Pengunjung dan Pelanggan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Pelanggan** | | |
| **Kode Fungsi** | **Nama Fungsi** | **Deskripsi** |
| 1 | WEB-F-PEL1 | *Register* | Sistem dapat melakukan pendaftaran *user* |
| 2 | WEB-F-PEL2 | Melihat Produk | Sistem dapat digunakan untuk melihat daftar produk |
| 3 | WEB-F-PEL3 | Melihat Detail Produk | Sistem dapat digunakan untuk melihat detail produk |
| 4 | WEB-F-PEL4 | Melakukan Pencarian Produk | Sisten dapat digunakan untuk melakukan pencarian produk |
| 5 | WEB-F-PEL5 | *Login* | Sistem dapat digunakan untuk *login* |
| 6 | WEB-F-PEL6 | *Logout* | Sistem dapat digunakan untuk *logout* |
| 7 | WEB-F-PEL7 | Menambahkan Produk ke Keranjang | Sistem dapat digunakan untuk menambahkan produk kedalam keranjang |
| 8 | WEB-F-PEL8 | Melakukan *Checkout* | Sistem dapat digunakan untuk melakukan *checkout* produk yang sudah ada dalam keranjang |
| 9 | WEB-F-PEL9 | Melakukan Pembayaran | Sistem dapat digunakan untuk melakukan pembayaran produk yang sudah di*-checkout* |
| 10 | WEB-F-PEL10 | Mengubah Data Pribadi | Sistem dapat digunakan untuk mengubah data pribadi pelanggan |
| 11 | WEB-F-PEL11 | Melihat Riwayat Transaksi | Sistem dapat digunakan untuk melihat daftar riwayat transaksi pelanggan |
| 12 | WEB-F-PEL12 | Melihat Status Transaksi | Sistem dapat digunakan untuk melihat status transaksi pelanggan yang sedang berjalan |

Dan pada Tabel 4.3 akan menjelaskan kebutuhan fungsional untuk actor admin yang terdapat empat kebutuhan fungsional.

Tabel 4.3 Kebutuhan Fungsional Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Admin** | | |
| **Kode Fungsi** | **Nama Fungsi** | **Deskripsi** |
| 1 | WEB-F-ADM1 | *Login* | Sistem dapat melakukan masuk untuk mengakses panel admin |
| 2 | WEB-F-ADM2 | *Logout* | Sistem dapat digunakan untuk *logout* |
| 3 | WEB-F-ADM3 | Mengelola Produk | Sistem dapat digunakan untuk mengelola produk |
| 4 | WEB-F-ADM4 | Mengelola Pesanan | Sisten dapat digunakan untuk mengelola pesanan |

### Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan untuk menunjang pengalaman pengguna dalam penggunaan sistem informasi *website e-commerce* ini sehingga dapat memudahkan pengguna untuk menggunakan sistem ini. Kebutuhan non fungsional akan dijelaskan pada Tabel 4.4.

Tabel 4.4 Kebutuhan Non Fungsional

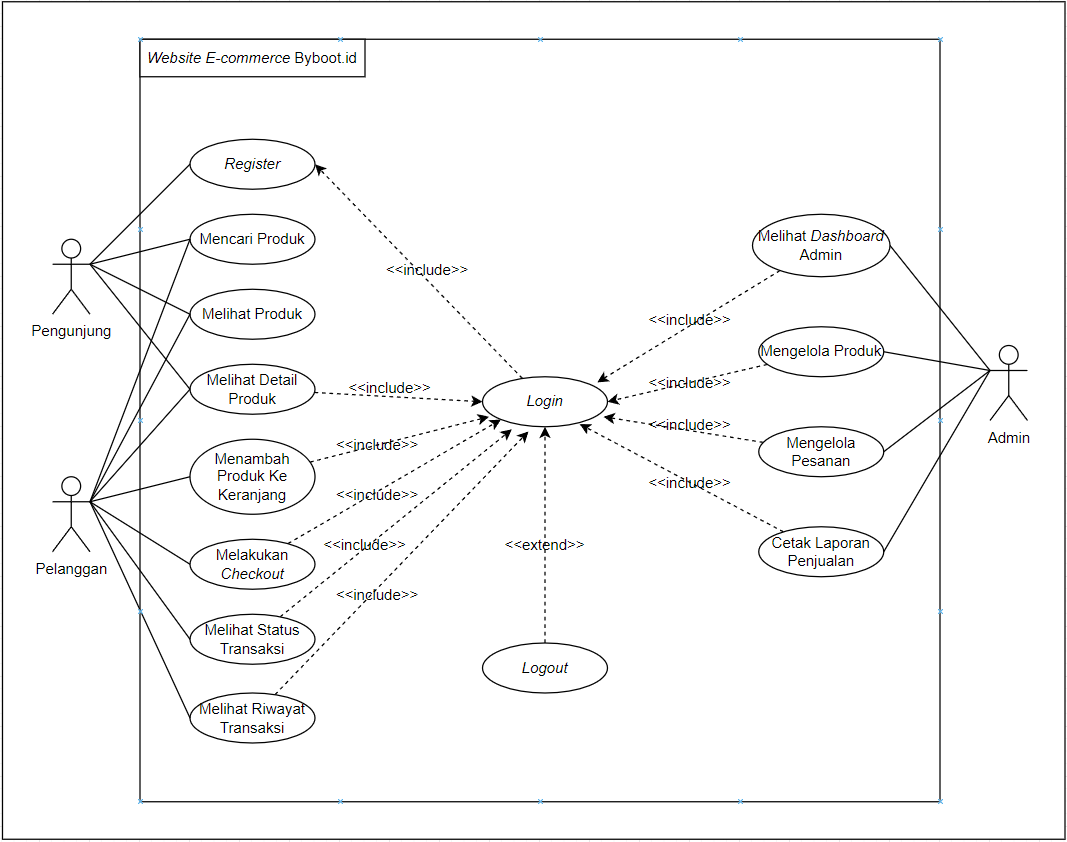
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Fungsi** | **Nama Fungsi** | **Deskripsi** |
| 1 | WEB-NF-ALL1 | *Usability* | Kemudahan penggunaan sistem oleh semua actor/user |

## Pemodelan *Use Case*

Pemodelan *use case* adalah suatu cara bagi pengembang dan pengguna untuk menjelaskan kebutuhan fungsional suatu sistem dalam bentuk diagram yang dimaksudkan untuk membantu pengembang dan pengguna memahami kebutuhan sistem dan dijadikan sebagai acuan untuk validasi.

### *Use Case Diagram*

*Use case diagram* merupakan jenis pemodelan dari *Unified Markup Language* (UML) yang dapat digunakan untuk menggambarkan jenis interaksi yang terjadi dan digunakan untuk menunjukkan tindakan yang dilakukan oleh aktor pada sistem, dengan fokus pada deskripsi persyaratan dan fitur sistem.



Gambar 4.2 *Use Case Diagram Website* Byboot.id

### *Use Case Scenario*

#### Use Case Scenario Melakukan Register

*Use case scenario* untuk melakukan *register* adalah untuk actor dapat mendaftarkan diri pada sistem informasi *website* dengan memasukkan data pribadi seperti email, nama, dan password. Ketika berhasil maka actor akan diarahkan menuju halaman *login* untuk memasukkan Kembali email dan password yang sudah didaftarkan tadi dan jika sesuai maka akan dialihkan ke menu home. Penjelasan mengenai *use case scenario* melakukan *register* dapat dilihat pada Tabel 4.5.

Tabel 4.5 *Use Case Scenario* Melakukan *Register*

|  |  |
| --- | --- |
| Melakukan *Register* | |
| *Code* | WEB-F-PEL1 |
| *Actors* | Pengunjung |
| *Description* | Skenario untuk melakukan *register* |
| *Use Case Goal* | Aktor dapat melakukan *register* |
| *Preconditions* | Aktor sudah mengakses/membuka *website* Byboot.id |
| *Main Flow* | Actor memilih menu *register*  Halaman akan dialihkan menuju halaman *register*  Aktor mengisikan data yang dibutuhkan seperti email, nama, dan password  Sistem akan memverifikasi akun   1. Actor akan dialihkan ke halaman *login* |
| *Alternative Flow* | Jika *main flow* langkah ke-3 gagal :  Ditampilkan informasi mengapa proses pendaftaran gagal serta sebab gagalnya |
| *Postconditions* | Aktor berhasil melakukan pendaftaran |

#### Use Case Scenario Melihat Produk

Pada *use case scenario* ini actor dapat melihat produk yang tersedia pada *website e-commerce* Byboot.id tanpa harus melakukan *register.* Produk yang tersedia akan diurutkan berdasarkan kategori masing-masing sehingga memudahkan pengunjung untuk melihat produk yang diinginkan. Penjelasan *use case scenario* ini aka nada pada table 4.6.

Tabel 4.6 *Use Case Scenario* Melihat Produk

|  |  |
| --- | --- |
| Melihat Produk | |
| *Code* | WEB-F-PEL2 |
| *Actors* | Pengunjung, Pelanggan |
| *Description* | Skenario untuk melihat produk |
| *Use Case Goal* | Aktor dapat melihat produk |
| *Preconditions* | Aktor berada pada halaman selain *home* |
| *Main Flow* | Actor memilih menu home  Halaman akan dialihkan menuju halaman *home,* dan langsung terdapat daftar produk yang tersedia |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil melihat produk |

#### Use Case Scenario Melihat Detail Produk

Pada *use case scenario* ini actor dapat melihat detail produk yang telah dipilih yang tersedia pada *website e-commerce* Byboot.id tanpa harus melakukan *register.* Detail produk akan berisi informasi mengenai produk yang dipilih seperti bahan, ukuran yang tersedia, dan lain-lain. Penjelasan *use case scenario* ini aka nada pada table 4.7.

Tabel 4.7 *Use Case Scenario* Melihat Detail Produk

|  |  |
| --- | --- |
| Melihat Produk | |
| *Code* | WEB-F-PEL3 |
| *Actors* | Pengunjung, Pelanggan |
| *Description* | Skenario untuk melihat detail produk |
| *Use Case Goal* | Aktor dapat melihat detail produk |
| *Preconditions* | Aktor berada pada halaman *home* |
| *Main Flow* | Actor memilih salah satu produk  Halaman akan dialihkan menuju halaman detail produk dan sistem akan menampilkan detail produk |
| *Alternative Flow* | Jika produk sudah habis terjual :  Sistem akan menampilkan keterangan bahwa produk sudah habis  Tombol beli dan tambah ke keranjang tidak bisa di pilih |
| *Postconditions* | Aktor berhasil melihat detail produk |

#### Use Case Scenario Melakukan Pencarian Produk

Pada *use case scenario* ini actor akan bisa melakukan pencarian pada produk yang ada pada *website e-commerce* berdasarkan nama produk. Fitur ini akan berada pada *navbar website* dan bisa diakes tanpa harus melakukan *register* terlebih dahulu. Jika berhasil maka sistem akan menampilkan produk yang sesuai dengan pencarian. Penjelasan mengenai *use case scenario* ini akan ada pada Tabel 4.8.

Tabel 4.8 *Use Case Scenario* Melakukan Pencarian Produk

|  |  |
| --- | --- |
| Melakukan Pencarian Produk | |
| *Code* | WEB-F-PEL4 |
| *Actors* | Pengunjung, Pelanggan |
| *Description* | Skenario untuk melakukan pencarian produk |
| *Use Case Goal* | Aktor dapat melakukan pencarian produk |
| *Preconditions* | Aktor sudah mengakses/membuka *website* Byboot.id |
| *Main Flow* | Actor memilih kolom pencarian yang ada pada *navbar*  Actor memasukkan kata kunci pencarian  Sistem akan mencari produk yang sesuai  Sistem akan menampilkan daftar produk yang sesuai |
| *Alternative Flow* | Jika *main flow* langkah ke-3 gagal :  Ditampilkan informasi bahwa produk yang dicari tidak ada |
| *Postconditions* | Aktor berhasil melakukan pencarian produk |

#### Use Case Scenario Login

Pada *use case scenario* ini actor bisa masuk pada website dengan menggunakan akun yang sudah didaftarkan. Actor akan memasukkan *email* dan *password* dan jika sesuai maka akan dialihkan menuju halaman *home,* sehingga bisa mengakses fitur yang mengharuskan login terlebih dahulu. Penjelasan *use case scenario* ini akan ada pada Tabel 4.9.

Tabel 4.9 *Use Case Scenario* Melakukan *Login*

|  |  |
| --- | --- |
| Melakukan *Login* | |
| *Code* | WEB-F-PEL5 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk melakukan *login* |
| *Use Case Goal* | Aktor dapat melakukan *login* |
| *Preconditions* | Aktor sudah mengakses/membuka *website* Byboot.id |
| *Main Flow* | Actor memilih menu *login*  Halaman akan dialihkan menuju halaman *login*  Aktor mengisikan data yang dibutuhkan seperti *email* dan *password*  Sistem akan memverifikasi akun   1. Actor akan dialihkan ke halaman *home* |
| *Alternative Flow* | Jika *main flow* langkah ke-3 gagal :  Ditampilkan informasi mengapa proses login gagal serta sebab gagalnya |
| *Postconditions* | Aktor berhasil melakukan *login* |

#### Use Case Scenario Logout

Pada *use case scenario* ini actor akan bisa melakukan *logout* dari website dan akan menghapus session yang tersimpan tentang data akunnya. Penjelasan tentang *use case scenario* ini akan ada pada Tabel 4.10.

Tabel 4.10 *Use Case Scenario* Melakukan *Logout*

|  |  |
| --- | --- |
| Melakukan *Logout* | |
| *Code* | WEB-F-PEL6 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk melakukan *logout* |
| *Use Case Goal* | Aktor dapat melakukan *logout* |
| *Preconditions* | Aktor sudah *login* kedalam sistem |
| *Main Flow* | Actor memilih menu *dropdown* yang ada pada *navbar*  Aktor memilih menu *logout*  Aktor dialihkan ke halaman *home* |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil melakukan *logout* |

#### Use Case Scenario Menambahkan Produk ke Keranjang

Pada *use case scenario* ini actor akan bisa menambahkan produk yang diinginkan kedalam keranjang belanja. Untuk menggunakan fitur ini berada dalam halaman detail produk dan disamping tombol masukkan ke *wishlist.* Setelah tombol ini ditekan maka produk akan masuk kedalam keranjang dan bisa melanjutkan ke proses *checkout.* Fitur ini hanya tersedia jika stok dari produk masih ada. Penjelasan *use case scenario* ini akan ada Tabel 4.11.

Tabel 4.11 *Use Case Scenario* Menambahkan Produk ke Keranjang

|  |  |
| --- | --- |
| Menambahkan Produk ke Keranjang | |
| *Code* | WEB-F-PEL7 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk menambahkan produk ke keranjang |
| *Use Case Goal* | Aktor dapat menambahkan produk ke keranjang |
| *Preconditions* | Aktor sudah *login* dan berada pada halaman detail produk |
| *Main Flow* | Actor memilih tombol tambah ke keranjang  Sistem akan menambahkan produk ke keranjang belanja |
| *Alternative Flow* | Jika stok produk telah habis:  Tombol tambah ke keranjang akan dimatikan |
| *Postconditions* | Aktor berhasil menambahkan produk ke keranjang |

#### Use Case Scenario Melakukan Checkout

Pada *use case scenario* ini actor akan bisa melakukan *checkout* pada produk yang sudah berada pada keranjang. *Checkout* bisa dilakukan dengan memasukkan data diri seperti nama lengkap, alamat lengkap, nomor hp, dan kurir. Setelah data telah terisi tekan tombol bayar untuk melakukan pembayaran. Penjelasan mengenari *use case scenario* ini akan ada pada Tabel 4.12.

Tabel 4.12 *Use Case Scenario* Melakukan *Checkout*

|  |  |
| --- | --- |
| Melakukan *Checkout* | |
| *Code* | WEB-F-PEL8 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk melakukan *checkout* |
| *Use Case Goal* | Aktor dapat melakukan *checkout* |
| *Preconditions* | Aktor berada pada halaman keranjang |
| *Main Flow* | Actor memilih menu *checkout*  Sistem akan menampilkan halaman *checkout*  Aktor mengisikan data yang dibutuhkan seperti nama lengkap, nomor hp, alamat lengkap dan kurir ekspedisi  Aktor memilih menu bayar |
| *Alternative Flow* | Jika data tidak lengkap:  Ditampilkan informasi mengapa proses *checkout* gagal |
| *Postconditions* | Aktor berhasil melakukan *checkout* |

#### Use Case Scenario Melakukan Pembayaran

Pada *use case scenario* ini actor akan memilih menu bayar setelah mengisi data pada menu *checkout.* Kemudian akan ditampilkan halaman pembayaran oleh *Payment Gateway* Midtrans, selanjutnya actor akan memilih metode pembayaran yang tersedia. Jika telah melakukan pembayaran akan dialihkan menuju halaman *home.* Penjelasan mengenai *use case scenario* ini akan ada pada Tabel 4.13.

Tabel 4.13 *Use Case Scenario* Melakukan Pembayaran

|  |  |
| --- | --- |
| Melakukan Pembayaran | |
| *Code* | WEB-F-PEL9 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk melakukan pembayaran |
| *Use Case Goal* | Aktor dapat melakukan pembayaran |
| *Preconditions* | Aktor berada pada halaman *checkout* dan sudah mengisi data |
| *Main Flow* | Actor memilih menu bayar  Sistem akan menampilkan halaman pembayaran pesanan  Aktor akan memilih metode pembayaran yang tersedia  Aktor akan melakukan pembayaran sesuai dengan metode pembayaran yang dipilih |
| *Alternative Flow* | Jika aktor gagal melakukan pembayaran:  Status pesanan pada halaman status transaksi akan bernilai “belum dibayar” |
| *Postconditions* | Aktor berhasil melakukan pembayaran |

#### Use Case Scenario Mengubah Data Pribadi

Pada *use case scenario* ini aktor akan dapat melakukan perubahan pada data pribadi seperti nama, nomor hp, dan alamat. Fitur ini tersedia pada halaman profil pengguna pada *dropdown menu* yang ada pada *navbar.* Setelah melakukan perubahan data pribadi halaman akan dialihkan menuju profil pengguna dengan data sesuai dengan yang sudah diubah. Penjelasan mengenai *use case scenario* ini akan ada pada Tabel 4.14.

Tabel 4.14 *Use Case Scenario* Mengubah Data Pribadi

|  |  |
| --- | --- |
| Mengubah Data Pribadi | |
| *Code* | WEB-F-PEL10 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk mengubah data pribadi |
| *Use Case Goal* | Aktor dapat melakukan perubahan data pribadi |
| *Preconditions* | Aktor berada pada halaman profil pengguna |
| *Main Flow* | Actor memilih menu edit profil  Halaman akan dialihkan menuju halaman edit data pengguna  Aktor mengisikan data yang dibutuhkan seperti nama, nomor hp, email dan alamat.  Sistem akan mengalihkan halaman menuju profil pengguna |
| *Alternative Flow* | Jika *main flow* langkah ke-3 gagal :  Ditampilkan informasi mengapa prosesedit data gagal serta sebab gagalnya |
| *Postconditions* | Aktor berhasil melakukan perubahan data pribadi |

#### Use Case Scenario Melihat Riwayat Transaksi

Pada *use scenario* ini aktor akan bisa melihat semua riwayat transaksi yang pernah dilakukan. Fitur ini bisa dilihat dengan memilih menu riwayat transaksi yang ada pada *dropdown menu* yang ada pada *navbar.* Penjelasan *use case scenario* ini akan ada pada Tabel 4.15.

Tabel 4.15 *Use Case Scenario* Melihat Riwayat Transaksi

|  |  |
| --- | --- |
| Melihat Riwayat Transaksi | |
| *Code* | WEB-F-PEL11 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk melihat riwayat transaksi |
| *Use Case Goal* | Aktor dapat melihat semua riwayat transaksi yang pernah dilakukan |
| *Preconditions* | Aktor berada pada halaman *home* |
| *Main Flow* | Actor memilih *dropdown menu* yang ada pada *navbar*  Aktor memilih menu riwayat transaksi  Sistem akan menampilkan data riwayat transaksi pengguna |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil melihat semua riwayat transaksi |

#### Use Case Scenario Melihat Status Transaksi

Pada *use case scenario* ini aktor akan bisa melihat status transaksi yang sedang dilakukan. Sehingga aktor bisa mengecek status *order* yang sedang dilakukan serta bisa melihat nomor resi jika pesanan sudah dikirim oleh penjual. Penjelasan *use case scenario* ini aka nada pada Tabel 4.16.

Tabel 4.16 *Use Case Scenario* Melihat Status Transaksi

|  |  |
| --- | --- |
| Melihat Status Transaksi | |
| *Code* | WEB-F-PEL12 |
| *Actors* | Pelanggan |
| *Description* | Skenario untuk melihat status transaksi |
| *Use Case Goal* | Aktor dapat melihat status transaksi yang sedang dilakukan |
| *Preconditions* | Aktor berada pada halaman *home* |
| *Main Flow* | Actor memilih *dropdown menu* yang ada pada *navbar*  Aktor memilih menu pesanan  Sistem akan menampilkan daftar transaksi yang sedang berlangsung  Aktor memilih transaksi yang ingin dilihat detail transaksinya  Sistem akan menampilkan detail transaksi yang dipilih |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil melihat status transaksi yang sedang berlangsung |

#### Use Case Scenario Login Admin

Pada *use case scenario* ini actor bisa masuk pada website dengan menggunakan akun yang sudah didaftarkan. Actor akan memasukkan *email* dan *password* dan jika sesuai maka akan dialihkan menuju halaman *home,* sehingga bisa mengakses fitur yang mengharuskan login terlebih dahulu. Penjelasan *use case scenario* ini akan ada pada Tabel 4.17.

Tabel 4.17 *Use Case Scenario Login* Admin

|  |  |
| --- | --- |
| Melakukan *Login* Admin | |
| *Code* | WEB-F-ADM1 |
| *Actors* | Admin |
| *Description* | Skenario untuk melakukan *login* admin |
| *Use Case Goal* | Aktor dapat melakukan *login* sebagai admin |
| *Preconditions* | Aktor sudah mengakses/membuka *website* Byboot.id |
| *Main Flow* | Actor memilih menu *login*  Halaman akan dialihkan menuju halaman *login*  Aktor mengisikan data yang dibutuhkan seperti *email* dan *password*  Sistem akan memverifikasi akun   1. Actor akan dialihkan ke halaman *dashboard* admin |
| *Alternative Flow* | Jika *main flow* langkah ke-3 gagal :  Ditampilkan informasi mengapa proses login gagal serta sebab gagalnya |
| *Postconditions* | Aktor berhasil melakukan *login* |

#### Use Case Scenario Logout Admin

Pada *use case scenario* ini actor akan bisa melakukan *logout* dari website dan akan menghapus session yang tersimpan tentang data akunnya. Penjelasan tentang *use case scenario* ini akan ada pada Tabel 4.18.

Tabel 4.18 *Use Case Scenario Logout* Admin

|  |  |
| --- | --- |
| Melakukan *Logout* Admin | |
| *Code* | WEB-F-ADM2 |
| *Actors* | Admin |
| *Description* | Skenario untuk melakukan *logout* |
| *Use Case Goal* | Aktor dapat melakukan *logout* |
| *Preconditions* | Aktor sudah *login* sebagai admin |
| *Main Flow* | Actor memilih menu *dropdown* yang ada pada *navbar*  Aktor memilih menu *logout*   1. Aktor dialihkan ke halaman *home* |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil melakukan *logout* |

#### Use Case Scenario Mengelola Produk

Pada *use case scenario* ini aktor sebagai admin akan bisa mengelola produk. Pengelolaan produk bisa berupa menambah, menghapus, dan mengupdate produk. Penjelasan *use case scenario* ini akan ada pada Tabel 4.19.

Tabel 4.19 *Use Case Scenario* Mengelola Produk

|  |  |
| --- | --- |
| Mengelola Produk | |
| *Code* | WEB-F-ADM3 |
| *Actors* | Admin |
| *Description* | Skenario untuk mengelola produk |
| *Use Case Goal* | Aktor dapat mengelola produk |
| *Preconditions* | Aktor sudah *login* sebagai admin |
| *Main Flow* | Actor memilih menu produk  Sistem akan menampilkan halaman produk  Aktor bisa menambah, mengubah dan menghapus produk |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil mengelola produk |

#### Use Case Scenario Mengelola Pesanan

Pada *use case scenario* ini aktor akan bisa mengelola pesanan yang masuk. Pengelolaan bisa berupa perubahan status transaksi, dan pengisian nomor resi pada transaksi. Penjelasan *use case scenario* ini akan ada pada Tabel 4.20.

Tabel 4.20 *Use Case Scenario* Mengelola Pesanan

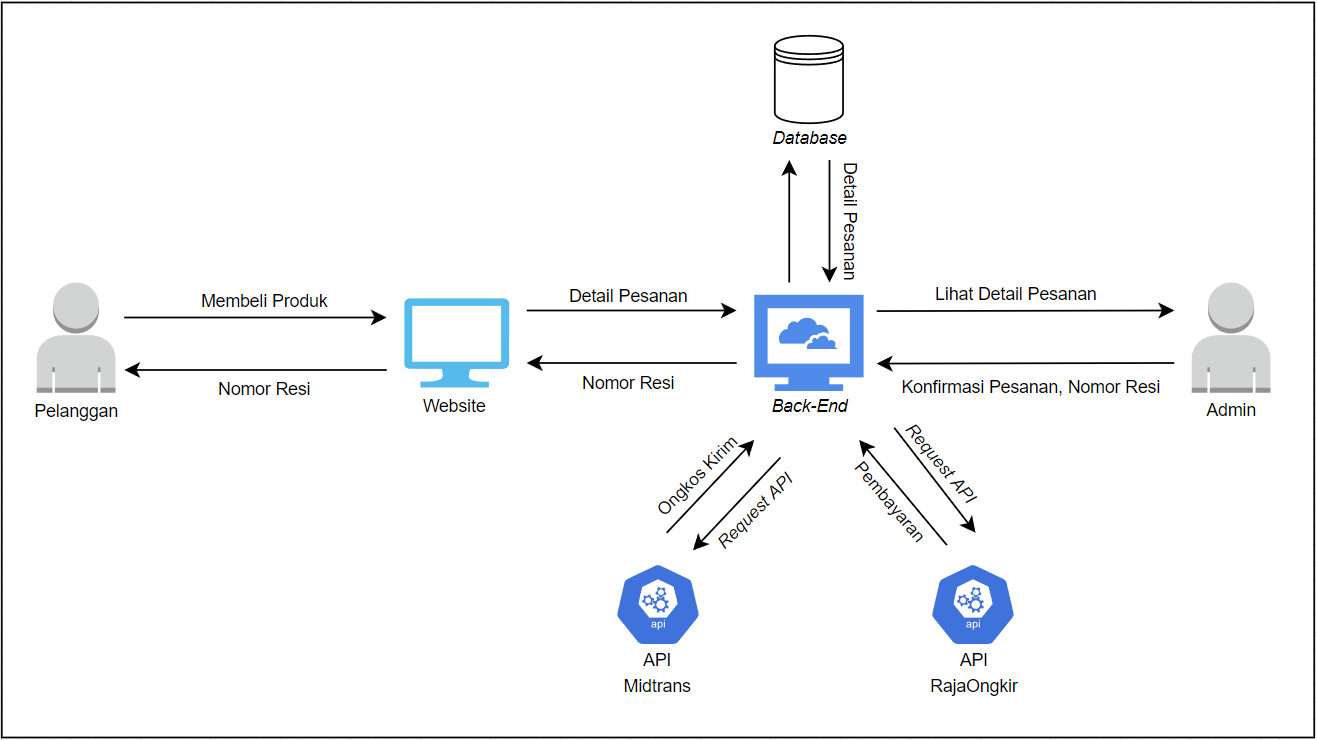
|  |  |
| --- | --- |
| Mengelola Pesanan | |
| *Code* | WEB-F-ADM4 |
| *Actors* | Admin |
| *Description* | Skenario untuk mengelola pesanan |
| *Use Case Goal* | Aktor dapat mengelola pesanan |
| *Preconditions* | Aktor sudah *login* sebagai admin |
| *Main Flow* | Actor memilih menu pesanan  Sistem akan menampilkan halaman pesanan  Aktor bisa mengubah status pesanan, dan menambahkan nomor resi |
| *Alternative Flow* | - |
| *Postconditions* | Aktor berhasil mengelola pesanan |

# PERANCANGAN SISTEM

Bab 5 ini akan berisi pembuatan dan perancangan diagram yang dibutuhkan untuk acuan perancangan tampilan sistem informasi yang akan dibangun. Diagam yang akan dibuat adalah berupa *sequence diagram* yang merupakan salah satu diagram dari UML. Selain diagram akan juga dibuat arsitektur sistem, struktur *database* yang akan digunakan untuk penyimpanan serta rancangan antarmuka sistem.

## Arsitektur Sistem

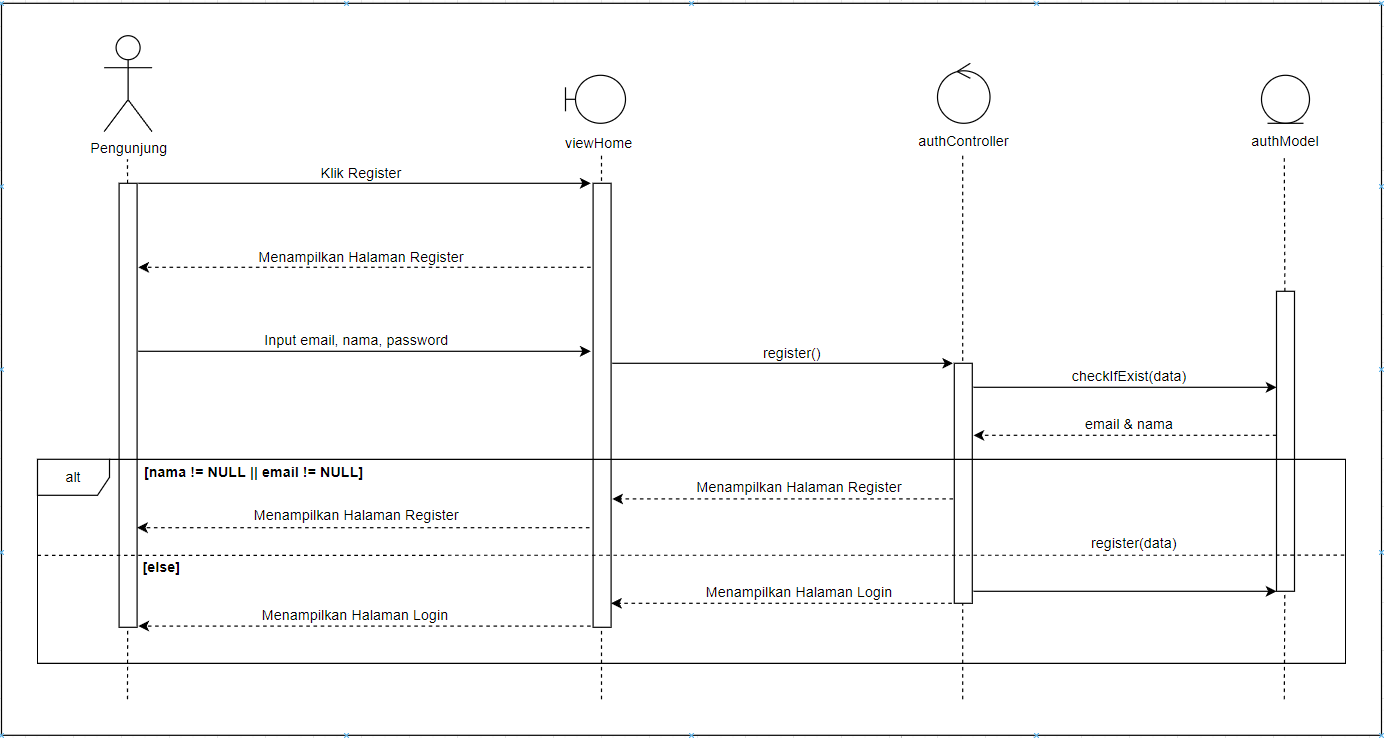
Secara umum arsitektur dari sistem informasi *website e-commerce* Byboot.id adalah pelanggan akan melakukan pembelian produk pada *website.* Detail pesanan akan disimpan pada *database* lengkap dengan ongkos kirim yang didapatkan dari API RajaOngkir. Kemudian admin akan bisa mengonfirmasi pesanan dengan mengubah status pesanan dan juga melihat detail pesanan yang sudah dibuat oleh pelanggan. Jika pembayaran sudah diterima maka admin akan mengubah status pesanan tadi dan melakukan proses pengiriman dan menambagkan nomor resi pada detail pesanan. Rancangan arsitektur dapat dilihat pada Gambar 5.1.



Gambar 5.1 Arsitektur Sistem

## *Sequence Diagram*

### *Sequence Diagram Register*



Gambar 5.2 *Sequence Diagram Register*

DAFTAR REFERENSI

Asti, C., Putra, W., & Purnomo, W. (2021). Pengembangan Website E-Commerce dengan Pemanfaatan Sistem Payment Gateway Midtrans (Studi Kasus: Butik Rizza Collection). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 5(12), 5213-5220. Diambil dari https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10206.

Damanik, Erikson (2012). Perancangan Sistem Informasi Pembayaran *Online* Menggunakan *Payment Gateway*. *Jurnal SIFO Mikroskil ISSN : 1412-0100, Vol. 13 No. 1.* .

Kapojos, F. (2012). Implementasi Service-Oriented Architecture dengan web service untuk aplikasi informasi akademik. Jurnal Teknik Elektro dan Komputer, 1(1).

Katadata, (2021). Produk Fesyen Jadi Primadona di E-Commerce. [online] Available at: https://databoks.katadata.co.id/datapublish/2021/06/09/produk-fesyen-jadi-primadona-di-e-commerce [Diakses 01 February 2022].

Kotler, Philip And Gary Armstrong., 2018., Principle Of Marketing, 17e Global Edition, Pearson Education Limited, New York.

Kozinets, R. V., De Valck, K., Wojnicki, A. C., & Wilner, S. J. (2010). Networked narratives: Understanding word-of-mouth marketing in online communities. Journal of marketing, 74(2), 71-89.

Marthasar, G. I., Aminudin, & Munarko, Y. (2010). Implementasi Web Service Untuk Mendukung Interoperabilitas Pada Aplikasi E-Commerce. *The 12th Industrial Electronics Seminar, Indonesia*, *2010*(Ies), 978–979.

Muhardian, A. (2022). JSON itu Apa Sih? dan Apa Gunanya dalam Pemrograman?. Online at <https://www.petanikode.com/json-pemula/>, accessed 15 February 2022.

Nidhra, S., & Dondeti, J. (2012). Black box and white box testing techniques-a literature review. International Journal of Embedded Systems and Applications (IJESA), 2(2), 29-50.

Nisrina, Yuninda Eka (2019). Pengembangan *E-Commerce* Dengan Pemanfaatan Sistem *Payment Gateway* (Studi Kasus: Wisata Kampung Sapi *Adventure*). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer ISSN : 2548-964X, Vol. 3 No. 10.* .

Nugroho, H. E., & Nugroho, A. (2021). ANALISIS DAN PERANCANGAN E-COMMERCE PADA TOKO SEPATU DOPE13STORE MENGGUNAKAN FRAMEWORK LARAVEL. Information System Journal, 4(1), 38-44.

Otaduy, I., & Díaz, O. (2017). User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. Journal of Systems and Software, 133, 212-229.

Permatasari, Desy I. (2017). *Workshop Pengembangan Perangkat Lunak* [Unggahan blog]*.* Diakses dari <http://desy.lecturer.pens.ac.id/Workshop%20Pengembangan%20Perangkat%20Lunak/>.

Pradana, M. (2015). Klasifikasi jenis-jenis bisnis e-commerce di Indonesia. Neo-Bis, 9(2), 32-40.

Pressman, Roger S. 2012. Rekayasa Perangkat Lunak – Buku Satu, Pendekatan Praktisi (Edisi 7). Yogyakarta: Andi.

Putra, D. A., Sasmita, A., Made, G., & Wiranatha, A. K. A. C. E-Commerce Marketplace Petshop Menggunakan Integrasi Rajaongkir API dan iPaymu Payment Gateway API. Jurnal Ilmiah Teknologi dan Komputer, 1(1), 13-22.

Qin, Z., & Qin, Z. (2009). Introduction to E-commerce (Vol. 2009). Heidelberg: Springer.

Rouf, A. (2012). Pengujian perangkat lunak dengan menggunakan metode white box dan black box. HIMSYATECH, 8(1).

Sora, 2015.*PengertianUML DanJenis-Jenisnya*. [online] Available at:<http://www.pengertianku.net/2015/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html>.

Softwaretesting. (2022). What Is User Acceptance Testing (UAT): A Complete Guide. Online at <https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/#User_Acceptance_Testing_Process>, accessed 16 February 2022.

West, M. (2011). *Developing High Quality Data Models* [Halaman Web]*.* Diakses dari <https://www.sciencedirect.com/topics/computer-science/physical-data-model>*.*

Yudhanto, Y., & Prasetyo, H. A. (2018). Panduan Mudah Belajar Framework Laravel. Elex Media Komputindo.

Yuliano, T. (2007). Pengenalan Php. IlmuKomputer. com.

Yusuf Maulana, R. (2021). PEMANFAATAN API MIDTRANS SEBAGAI PAYMENT GATEWAY PADA PEMBUATAN WEBSITE E-COMMERCE MENGGUNAKAN FRAMEWORK LARAVEL (STUDI KASUS: CHESSPEDIA) (Doctoral dissertation, Universitas Brawijaya).