

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berdasarkan penelitian terdahulu yang telah dilakukan oleh Rina Gustina dan Henny Leidiyana menerapkan konsep pengembangan sistem informasi melalui teknologi web yang dapat diakses melalui internet. Dalam penelitian tersebut, digunakan Framework Laravel sebagai alat bantu dalam pengembangan aplikasi web, dengan fitur-fitur yang lengkap seperti sistem routing, ORM (Object Relational Mapping), dan templating engine. Metodologi pengembangan sistem yang diterapkan mirip dengan model waterfall, meliputi tahapan analisis kebutuhan, desain, pengembangan, pengujian, dan pemeliharaan [3]. Terdapat beberapa kesamaan antara penelitian ini dengan penelitian Rina Gustina dan Henny Leidiyana, yakni penggunaan *framework* Laravel sebagai alat bantu dalam mengembangkan sistem informasi website, serta penerapan model pengembangan sistem waterfall. Namun, terdapat pula perbedaan signifikan. Penelitian yang dilakukan penulis memiliki fokus untuk merancang dan mengembangkan website yang mampu memenuhi kebutuhan dari *band* Blues Clues, dengan tujuan utama meningkatkan interaksi pengguna dan meningkatkan brand awareness mereka, sedangkan penelitian yang dilakukan Rina Gustina dan Henny Leidiyana memiliki tujuan untuk mengatasi ketidak efisiensinya sistem penggajian manual yang dilakukan oleh PT. Eveshine Convertindo. Perbedaan lainnya adalah dalam penggunaan diagram, di mana penelitian Rina Gustina dan Henny Leidiyana menggunakan berbagai jenis diagram, termasuk *use case* diagram, *activity* diagram,

sequence diagram, dan ERD (*Entity Relationship Diagram*). Sementara penelitian ini hanya fokus pada tiga diagram, yaitu *use case* diagram, *sequence* diagram, dan *activity* diagram, karena penelitian ini difokuskan pada perancangan sisi *front-end* dari website.

Penelitian selanjutnya oleh Jebi Hayi Tamami, Yusi Triyoni Mursityo, dan Fajar Pradana yang melakukan penelitian mengenai Pengembangan Sistem Informasi Penjualan Awesam Merchandise dengan metode Rational Unified Process. Pengumpulan data dilakukan dengan cara mengamati dan berbicara langsung atau wawancara dengan tim Awesam Malang, guna memahami cara kerja bisnis dan mengidentifikasi masalah yang mungkin terjadi. Dalam upaya pengembangan, mereka menerapkan metode Rational Unified Process (RUP) yang didasarkan pada konsep Sales Force Automation (SFA). Penelitian ini bertujuan membantu Awesam dalam mengelola transaksi dengan lebih baik, meningkatkan kualitas pelayanan kepada pelanggan, serta menghadapi persaingan di industri [4]. Penelitian ini memiliki kesamaan dengan penelitian yang dilakukan penulis, keduanya membahas tentang sistem informasi penjualan merchandise pada objek yang diteliti. Namun, perbedaannya terletak pada metode pengembangan sistem yang digunakan. Dalam penelitian ini, Jebi dkk menggunakan metode pengembangan yang disebut Rational Unified Process (RUP), sementara penulis menggunakan metode Waterfall.

2.2 Perancangan dan Pengembangan *Website*

Dalam perancangan dan pengembangan *website* ini akan membahas lebih mendalam mengenai konsep, dan tujuan dan manfaat. Penjelasan lebih rinci tentang masing-masing poin adalah sebagai berikut:

2.2.1 Konsep *Website*

Secara umum, *Website* merupakan kumpulan halaman yang terdiri dari beberapa laman yang mengandung informasi dalam bentuk digital seperti teks, gambar, dan animasi. Situs *website* ini dapat diakses melalui internet oleh orang-orang di seluruh dunia yang memiliki koneksi internet [5].

Menurut Lukmanul Hakim [6], *Website* adalah sarana dalam internet yang menghubungkan berbagai dokumen baik dalam skala lokal maupun jarak jauh. Dokumen-dokumen ini, yang disebut sebagai halaman *web*, dapat diakses melalui *browser* seperti Google Chrome, Mozilla Firefox, dan lainnya. Di dalam *website*, terdapat link yang memungkinkan pengguna untuk berpindah dari satu halaman ke halaman lainnya, baik itu dalam *server* yang sama maupun di *server* di seluruh dunia. Dengan menggunakan *browser*, pengguna dapat membaca atau mengakses halaman-halaman ini.

2.2.2 Tujuan dan Manfaat *Website*

Tujuan dari pembuatan *website* untuk mempermudah *sharing* informasi antar komputer yang terhubung melalui jaringan internet. Namun, seiring perkembangan zaman, fungsi sebuah *website* telah meluas dan tidak hanya terbatas pada *sharing* informasi saja. Saat ini, sebuah *website* memiliki peran yang tak kalah penting dengan berbagai fungsi tambahan, seperti menyediakan informasi,

meningkatkan kehadiran online, meningkatkan kredibilitas, interaksi dengan pengguna, meningkatkan penjualan, pemasaran dan promosi, membangun komunitas, mengumpulkan data, dan menjangkau pasar global. Dengan demikian, *website* telah menjadi sebuah *platform* yang serba guna dengan manfaat yang luas dan beragam [7].

Manfaat dari sebuah *website* sendiri itu berbeda-beda tergantung pada kebutuhan pemiliknya, baik itu memiliki beberapa tujuan dan sasaran yang berbeda atau hanya satu tujuan. Beberapa manfaat penting dari pembuatan *website* meliputi potensi dalam menghasilkan pendapatan, baik melalui penjualan produk maupun jasa. Selain itu, *website* dapat menjadi alternatif yang lebih hemat dalam promosi dan interaksi dengan khalayak melalui *platform online*. *Website* juga memiliki peran penting dalam membangun kehadiran digital dan meningkatkan branding sebuah entitas atau bisnis. Di samping itu, *website* dapat memberikan hiburan dan interaksi langsung dengan pengguna, menciptakan hubungan yang lebih dekat serta meningkatkan kesadaran tentang suatu hal atau brand [8].

2.3 Software Perancangan Sistem

Dalam pengembangan sebuah *website*, terdapat beberapa perangkat lunak (*Software*) yang berperan penting dalam perancangan sistem. Berikut merupakan penjelasan mengenai beberapa *Software* yang digunakan:

2.3.1 HTML (*Hypertext Markup Language*)

HTML adalah bahasa standar yang digunakan untuk menampilkan konten di halaman *web*. Dalam HTML, terdapat beberapa fungsi yang dapat dilakukan, antara lain:

1. Mengatur dan mendesain tampilan isi halaman *web*.
2. Membuat tabel di halaman *web*.
3. Mempublikasikan halaman *web* secara online.
4. Membuat formulir yang memungkinkan pengguna untuk memberikan input dan mengelola registrasi serta transaksi melalui halaman *web*.
5. Menampilkan gambar di area *browser*.

Dengan menggunakan HTML, pengembang *web* dapat membangun struktur dasar halaman *web* dan mengatur tampilan konten dengan mudah [9].

2.3.2 CSS (*Cascading Style Sheets*)

CSS (*Cascading Style Sheets*), adalah seperangkat aturan yang digunakan untuk mengendalikan penampilan dan tata letak elemen-elemen dalam sebuah halaman *website*. Dengan menggunakan CSS, kita dapat menciptakan tampilan yang terstruktur.

CSS memungkinkan kita untuk mengatur berbagai macam elemen dalam halaman *website*, seperti ukuran gambar, warna teks, warna latar belakang pada

teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna saat mouse mengarah pada elemen tertentu (*mouse hover*), spasi antar paragraf, spasi antar teks, margin pada sisi kiri, kanan, atas, bawah, dan lainnya. Dengan adanya CSS, memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda [9].

2.3.3 Javascript

Javascript adalah sebuah bahasa pemrograman yang terdiri dari kumpulan *script* yang dijalankan di dalam dokumen HTML. Dalam pengembangan aplikasi berbasis *web*, Javascript dapat digunakan untuk meningkatkan tampilan dan sistem yang ada pada halaman *web*. Karakteristik utama dari Javascript meliputi [10]:

1. Merupakan bahasa pemrograman tingkat tinggi (*high-level programming*)
2. Berjalan di sisi klien (*client-side*)
3. Berorientasi objek
4. Memiliki tipe data yang longgar (*loosely typed*).

Dalam mengimplementasikan Javascript, terdapat beberapa aplikasi atau alat standar yang digunakan. Pertama, *Software* teks editor seperti Notepad++, VScode, Sublime text, Adobe Dreamweaver, dan NetBeans digunakan untuk menulis dan mengedit *script* Javascript. Kedua, *web browser* seperti Mozilla Firefox, Google Chrome, Internet Explorer, Opera Mini, Safari, dan lainnya digunakan untuk menjalankan skrip Javascript di halaman *web*. Ketiga, HTML

digunakan sebagai bahasa *markup* utama untuk membangun struktur dan konten halaman *web* yang akan diperlengkapi dengan fungsionalitas Javascript [10].

Namun, perlu diketahui bahwa Javascript memiliki beberapa kelemahan. Salah satunya adalah tidak didukung oleh beberapa versi *browser* lama. Hal ini dapat menyebabkan adanya perbedaan dalam kemampuan menjalankan atau mendukung fitur-fitur Javascript pada berbagai *browser* yang berbeda. Oleh karena itu, dalam pengembangan *web*, perlu mempertimbangkan kesesuaian *browser* dan versi Javascript yang digunakan agar aplikasi yang telah dibuat dapat berjalan dengan baik di berbagai *platform* dan peramban *web* [10].

2.3.4 Bootstrap

Bootstrap adalah sebuah perpustakaan (*library*) yang sering digunakan untuk membangun aplikasi *web* dan situs *web* yang responsif secara cepat, mudah, dan gratis. Bootstrap, yang juga dikenal sebagai Twitter Bootstrap, terdiri dari kumpulan file CSS dan HTML yang digunakan untuk menciptakan tata letak, tampilan, dan komponen-komponen seperti *grid*, *layout*, tipografi, tabel, formulir, navigasi, dan lain-lain. Selain itu, Bootstrap juga menyediakan plugin jQuery yang berguna untuk menciptakan komponen antarmuka pengguna yang menarik, seperti transisi, jendela modal, *dropdown*, *scrollspy*, *tooltip*, *tab*, *alert*, dan sebagainya [11].

Dengan menggunakan Bootstrap, pengembang dapat mendesain *web* yang responsif dengan mudah. Hal ini karena tampilan *web* yang dibangun dengan Bootstrap akan secara otomatis menyesuaikan dengan lebar peramban yang digunakan oleh pengguna. Dengan demikian, tampilan *web* akan tetap teratur dan

rapi, terlepas dari perangkat yang digunakan oleh pengguna, baik itu *handphone*, tablet, laptop, atau komputer desktop. Dengan menggunakan Bootstrap, pengguna tidak perlu khawatir tentang gangguan tampilan akibat perbedaan resolusi layar [11].

2.3.5 *Web Browser*

Web browser merupakan perangkat lunak yang berfungsi untuk menerima dan menampilkan informasi dari internet. Dengan menggunakan *web browser*, pengguna dapat mengakses berbagai sumber informasi seperti halaman *web*, gambar, video, dan konten lainnya [12].

Browser juga dikenal dengan istilah "penjelajah *web*" atau "peramban" yang merupakan aplikasi atau program yang dijalankan pada perangkat komputer untuk mengakses dan menjelajahi konten yang ada di *World Wide Web* (WWW) melalui jaringan internet. Saat ini, teknologi *browser* telah berkembang pesat dan tidak hanya mampu menampilkan teks atau tulisan, tetapi juga dapat menampilkan konten seperti gambar, musik, suara, video, file PDF, dan data lainnya [12].

Beberapa *jenis browser* yang populer saat ini antara lain [12]:

1. Microsoft Internet Explorer: Internet Explorer, atau IE, merupakan salah satu *browser* tertua yang diproduksi oleh Microsoft Corp. *Browser* ini telah lama ada dan dikenal oleh pengguna komputer sejak dahulu.
2. Google Chrome: Google Chrome dikembangkan oleh perusahaan teknologi besar, Google. *Browser* ini pertama kali dirilis pada tahun 2008 dan awalnya hanya tersedia untuk Microsoft Windows dalam versi

beta. Google Chrome menjadi salah satu *browser* paling populer karena menyediakan banyak fitur dan memiliki respons yang cepat.

3. Mozilla Firefox: Mozilla Firefox dirilis pertama kali pada tahun 2004 oleh Yayasan Mozilla dan langsung mendapat sambutan positif dari pengguna. Pada tahun 2014, penggunaan *browser* ini telah mencapai 12% hingga 22%, bahkan di Indonesia sendiri penggunaan Mozilla Firefox mencapai 55% dari total pengguna internet.

2.3.6 PHP (*Hypertext Preprocessor*)

Menurut Andi, PHP (*Hypertext Preprocessor*) adalah salah satu bahasa pemrograman yang berjalan di dalam *web server* dan digunakan untuk memproses data di *server*. PHP digunakan untuk mengolah data yang dikirim oleh pengguna melalui klien dan menyimpannya dalam *database web server*. Data tersebut dapat ditampilkan kembali saat diakses. Untuk menjalankan kode program PHP, file harus diunggah ke *server*. Unggah merupakan proses transfer data atau file dari komputer klien ke *web server* [13].

Untuk membuat *website* yang dinamis dan mudah diperbarui secara real-time melalui *browser*, diperlukan program yang mampu mengolah data dari komputer klien atau *server* itu sendiri agar dapat ditampilkan dengan mudah dan nyaman di *browser*. Salah satu program yang dapat dijalankan di *server* dan terpercaya adalah PHP [13].

PHP bekerja dalam dokumen HTML (*Hypertext Markup Language*) untuk menghasilkan konten halaman *web* sesuai permintaan. Dengan PHP, situs *web*

dapat diubah menjadi aplikasi berbasis *web*, bukan hanya sekumpulan halaman statis yang jarang diperbarui [13].

Awalnya, PHP dirancang untuk integrasi dengan *web server Apache*. Namun, saat ini PHP juga dapat bekerja dengan *web server* lain seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami. Yang membedakan PHP dari bahasa pemrograman lain adalah penggunaan tag penanda yang dimulai dengan "<?" atau "<?php" dan diakhiri dengan "?>". Dengan demikian, kita bebas menempatkan skrip PHP di mana pun [13].

2.3.7 VScode (*Visual Studio Code*)

Aplikasi (Software) yang digunakan untuk menulis atau mengembangkan kode program. *Visual Studio Code* adalah sebuah perangkat lunak yang ringan namun memiliki kekuatan sebagai editor kode sumber yang berjalan di desktop. Aplikasi ini telah dilengkapi dengan dukungan bawaan untuk Javascript, TypeScript, dan Node.js, serta memiliki berbagai macam ekstensi yang tersedia untuk bahasa pemrograman lain seperti C++, C#, Python, dan PHP [14].

2.4 Framework Laravel

Laravel adalah sebuah *framework web* berbasis PHP yang bersifat open-source dan gratis, yang dikembangkan oleh Taylor Otwell. *Framework* ini ditujukan untuk pengembangan aplikasi *web* yang mengadopsi pola arsitektur MVC (*Model-View-Controller*). Meskipun struktur pola MVC di Laravel sedikit berbeda dari struktur umum pada pola MVC, namun *framework* ini tetap menyediakan fitur-fitur penting yang diperlukan [15].

2.4.1 Keunggulan *Framework* Laravel

Laravel memiliki keunggulan yang membuatnya menjadi *framework* yang lebih baik dibandingkan dengan *framework* lainnya. Keunggulan-keunggulan tersebut antara lain: performa yang lebih cepat, kestabilan dalam me-*reload* data, keamanan data yang baik, penggunaan fitur canggih seperti *blade* dengan konsep HMVC (*Hierarchical Model View Controller*), tersedianya *library* yang siap digunakan, dan adanya fitur pengelolaan *migrations* untuk membuat skema tabel dalam *database* [16].

2.4.2 Arsitektur MVC (*Model View Controller*)

Pada Laravel, terdapat konsep-konsep arsitektur yang memiliki fungsi masing-masing, antara lain [16]:

1. *Routes*: Berfungsi sebagai penentu akses pada setiap permintaan sesuai dengan alur yang telah ditentukan.
2. *Controller*: Merupakan penghubung antara *model* dan *view*. *Controller* memiliki perintah-perintah untuk memproses bagaimana data ditampilkan dari *Model* ke *View* atau sebaliknya.
3. *Model*: Merupakan sekumpulan data yang mengelola tabel dalam *database*. Struktur pemodelan data pada Laravel terdiri dari tabel, *primary key*, dan *fillable*. Ketiga fungsi ini harus dinyatakan sebagai "*protected*". Pada bagian tabel, diisi dengan nama tabel yang sesuai dalam *database*, pada bagian *primary key* diisi dengan *primary key* dari tabel tersebut, dan pada bagian *fillable* diisi dengan kolom-kolom yang termasuk dalam tabel tersebut.

4. *View*: Merupakan file yang berisi kode HTML yang berfungsi untuk menampilkan data ke dalam *browser*. Format *view* pada Laravel menggunakan istilah *blade*, contohnya adalah "*view.blade.php*".
5. *Migrations*: Merupakan proses perancangan tabel dalam *database*. *Migrations* berfungsi sebagai *blueprint* atau pengaturan sistem kontrol untuk skema *database*.

2.5 Brand awareness

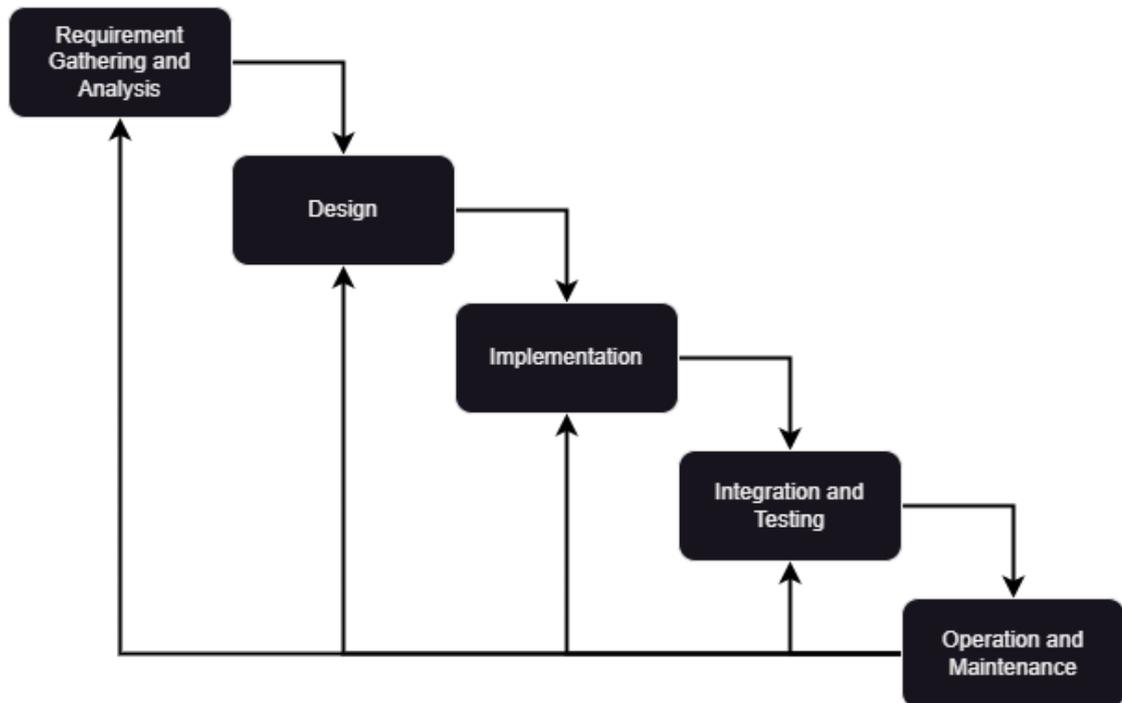
Kesadaran merek (*brand awareness*) merujuk pada sejauh mana konsumen mengenal dan mengingat suatu merek. Ini mencerminkan sejauh mana konsumen dapat mengenali merek berdasarkan nama, logo, produk, atau hal lain yang terkait dengan merek tersebut. Tingkat kesadaran merek yang tinggi memberikan keuntungan kompetitif, mempengaruhi preferensi konsumen, dan membantu membangun hubungan jangka panjang antara merek dan konsumen.

Menurut Wahyudi *Brand awareness* merupakan kondisi di mana pelanggan potensial mengenal suatu merek produk dan dapat mengaitkannya dengan kategori produk yang tepat [17].

2.5.1 Pengertian Brand awareness

Kesadaran merek (*brand awareness*) adalah kemampuan konsumen untuk mengenali suatu merek dalam situasi yang berbeda. Ini terjadi ketika konsumen mengenal dan dapat mengingat merek tersebut melalui pengenalan dan pengingatan terhadap merek tersebut [18].

2.6 Model Pengembangan *Waterfall*



Gambar 2.1 Model Waterfall SDLC (Sequential Development Life Cycle)

Model Waterfall adalah salah satu metode pengembangan perangkat lunak yang termasuk dalam model SDLC (*Sequential Development Life Cycle*). Model *Waterfall* adalah suatu pendekatan dalam pengembangan perangkat lunak yang mengikuti langkah-langkah secara berurutan. Dalam model ini, setiap tahapan seperti analisis, desain, pengkodean, pengujian, dan tahap lainnya harus diselesaikan satu per satu sebelum melanjutkan ke tahap berikutnya. Mirip dengan air yang mengalir secara berurutan, model ini memberikan kerangka kerja yang jelas dan terstruktur dalam pengembangan perangkat lunak. Metode *Waterfall* dilakukan melalui tahap-tahap berikut:

2.6.1 Requirement Gathering and Analysis

Tahap ini melibatkan pengumpulan kebutuhan secara komprehensif untuk dianalisis dan mendefinisikan apa yang perlu dicapai oleh program. Informasi dapat diperoleh melalui wawancara, diskusi, atau survei [19].

2.6.2 Design

Tahap ini melibatkan merancang desain perangkat lunak sebagai panduan sebelum melakukan pemrograman. Desain sistem dapat dibuat menggunakan alat seperti *Flowchart*, *Mind Map*, atau *Entity Relationship Diagram (ERD)* [19].

2.6.3 Implementation

Tahap ini melibatkan mengubah desain yang telah dibuat menjadi kode program yang sebenarnya. Kode yang dihasilkan masih berupa modul-modul yang akan digabungkan pada tahap selanjutnya [19].

2.6.4 Integration and Testing

Tahap ini melibatkan menggabungkan modul-modul yang telah dibuat sebelumnya dan melakukan pengujian untuk memastikan bahwa perangkat lunak yang dibuat sesuai dengan desain dan berfungsi dengan baik [19].

2.6.5 Operation and Maintenance

Tahap ini merupakan tahap akhir dari model *waterfall*. Sistem yang telah selesai dikembangkan dijalankan dan dipelihara. Pemeliharaan meliputi perbaikan kesalahan yang mungkin terlewat pada tahap sebelumnya [19].

2.7 UML (Unified Modelling Language)

UML merupakan salah satu bahasa standar yang secara luas digunakan di sektor industri, memiliki peran penting dalam mendefinisikan kebutuhan,

melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [20]. UML terbagi beberapa macam, antara lain :

2.7.1 Use Case Diagram

Use Case Diagram adalah representasi visual yang digunakan untuk memodelkan perilaku sistem informasi yang akan dibangun. *Use Case* diagram menggambarkan interaksi antara pengguna sistem dan sistem itu sendiri melalui skenario penggunaan yang menggambarkan bagaimana sistem digunakan [20].

2.7.2 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan alur kerja atau aktivitas dalam sebuah sistem pada perangkat lunak. *Activity* diagram membantu dalam menggambarkan proses-proses yang terjadi dalam sistem dengan menggambarkan urutan aktivitas dan keputusan yang dilakukan dalam sistem [20].

2.7.3 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan bagaimana objek-objek berinteraksi satu sama lain dalam suatu sistem. Berikut merupakan komponen dalam *sequence* diagram [21, p. 21]:

- 1) *Activations*, adalah bagian dalam *sequence* diagram yang menggambarkan eksekusi dari fungsi yang dimiliki oleh suatu objek. Hal ini memungkinkan kita untuk melihat urutan langkah-langkah yang dijalankan oleh objek dalam menjalankan suatu fungsi.
- 2) *Actor*, adalah komponen dalam *sequence* diagram yang menggambarkan peran yang melakukan serangkaian aksi dalam suatu proses. Aktor ini dapat

berupa pengguna, sistem eksternal, atau objek lain yang berinteraksi dengan objek-objek dalam sistem.

- 3) *Collaboration boundary*, merupakan komponen dalam sequence diagram yang menunjukkan tempat di mana objek-objek berada dan digunakan untuk memonitor interaksi mereka. Boundary ini memberikan batasan visual untuk menggambarkan ruang lingkup kolaborasi antara objek-objek dalam sistem.
- 4) *Parallel vertical lines*, adalah garis-garis vertikal yang menunjukkan suatu garis proses atau jalur eksekusi dalam sequence diagram. Garis-garis ini dapat mengarah pada suatu state atau titik dalam aliran eksekusi objek-objek yang terlibat.
- 5) *Processes*, mengacu pada tindakan atau aksi yang dilakukan oleh aktor dalam suatu waktu. Dalam sequence diagram, processes ini dapat digambarkan sebagai pesan atau pemanggilan fungsi yang dikirim antara objek-objek.
- 6) *Window*, menggambarkan halaman atau tampilan yang sedang ditampilkan dalam suatu proses. Dalam sequence diagram, window ini dapat mengindikasikan perubahan tampilan atau transisi antara halaman-halaman yang terlibat dalam interaksi antar objek.
- 7) *Loop*, adalah model logika yang berpotensi untuk diulang beberapa kali. Dalam sequence diagram, loop ini dapat menggambarkan proses iteratif di mana serangkaian aksi atau pemanggilan fungsi terjadi berulang kali hingga kondisi tertentu terpenuhi.