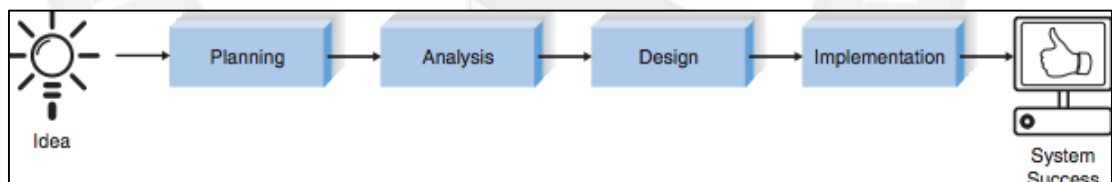


BAB II

LANDASAN TEORI

2.1 *System Development Life Cycle (SDLC)*

System Development Life Cycle atau SDLC merupakan serangkaian proses yang dilakukan oleh sistem analis dan *programmer* untuk mengembangkan sistem informasi dengan cara menentukan kebutuhan, merancang, membangun dan menyampaikan sistem kepada *user*. SDLC adalah proses mengembangkan atau mengubah suatu sistem *software* dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem *software* sebelumnya, berdasarkan *best practice* atau cara-cara yang sudah teruji baik [1]. Pada gambar 2.1, SDLC memiliki 4 fase dasar yaitu perencanaan (*planning*), analisis (*analysis*), desain (*design*), dan implementasi (*implementation*). Secara umum, semua sistem dijalankan melalui 4 fase ini, tetapi setiap sistem memiliki penerapan model SDLC yang berbeda-beda sesuai dengan jenis sistem yang dijalankan [2].



Gambar 2.1 Tahap-Tahap *System Development Life Cycle (SDLC)*

Sumber: A. Dennis, B. H. Wixom, and Roberta M. Roth, *System Analysis & Design*, 5th ed. USA: Wiley, 2015.

2.1.1 Tahap Perencanaan (*Planning*)

Tahap perencanaan adalah proses dasar untuk memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana sistem itu akan dibangun. Pada tahap ini, klien menentukan permintaan sistem kepada *developer* dan *developer* melakukan analisis kelayakan [2].

2.1.2 Tahap Analisis (*Analysis*)

Pada tahap analisis, *developer* akan menganalisis sistem saat ini (*as-is system*) dan mengidentifikasi masalah yang ada, serta merancang sistem yang baru (*to-be system*) dengan menganalisa apa saja kebutuhan dari sistem, mulai dari kebutuhan fungsional sistem maupun kebutuhan non fungsional dari sistem [3]. Ada tiga tahap analisis. Pertama, menganalisa sistem yang sudah ada dan merancang sistem yang baru (*system requirement*). *System requirement* berperan untuk menentukan sistem apa saja yang dapat dilakukan Kemudian, mengumpulkan *requirement gathering* melalui *interview* yang akan dianalisa. *Requirement gathering* bertujuan untuk mengarahkan pengembangan konsep untuk sistem baru dan menggambarkan bagaimana bisnis akan beroperasi dengan sistem yang baru tersebut. Tahap yang terakhir adalah penyusunan proposal sistem. Proposal sistem adalah sebuah dokumen yang menggambarkan persyaratan bisnis apa saja yang harus ada pada sistem yang baru [2].

2.1.3 Tahap Desain (*Design*)

Tahap desain adalah tahapan lanjut dari tahap analisis dimana dalam tahap ini disajikan desain dari aplikasi seperti desain antar muka dan desain *database* yang akan diterapkan ke dalam sistem informasi [3]. Pada fase desain dibagi menjadi empat tahap yang dikerjakan oleh tim *developer*. Pertama, menentukan strategi desain dengan konsultan. Kemudian, mengembangkan desain arsitektur dasar (*hardware, software, dan infrastruktur jaringan*) dan desain *user interface*. Lalu, mengembangkan *database* dan spesifikasi *file*. Setelah itu, tim akan mengembangkan desain programnya. Pada akhir fase desain, analisis kelayakan dan rencana sistem akan diperiksa ulang dan direvisi [2].

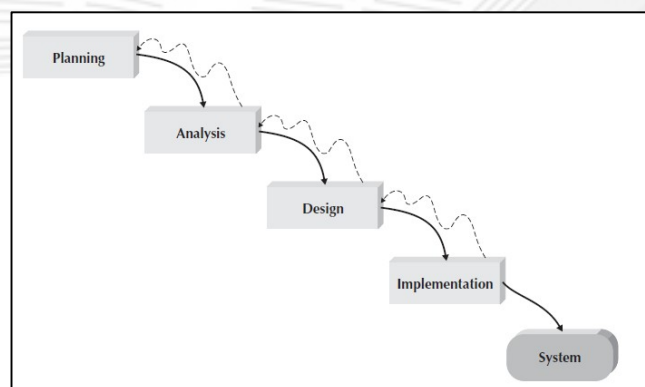
2.1.4 Tahap Implementasi (*Implementation*)

Tahap implementasi adalah fase membangun sistem yang sudah dirancang tersebut. Tahap ini dibagi menjadi 3. Langkah pertama adalah perancangan sistem yang akan diuji untuk memastikan kinerjanya sesuai dengan apa yang dirancang.

Kemudian, melakukan pengimplementasian sistem yang baru. Langkah terakhir adalah tim analis menetapkan *support plan* untuk menganalisis sistem yang telah dibuat dan diuji untuk mengidentifikasi perubahan apa yang diperlukan oleh sistem [2].

2.2 Waterfall

Metode *waterfall* sering juga disebut sebagai model sekuensi linear atau alur hidup klasik. Pengembangan sistem dikerjakan secara terurut mulai dari *planning*, *analysis*, *design*, *implementation*, dan hingga terbentuknya sistem yang baru [3]. Gambar 2.2. merupakan proses pengembangan sistem dengan menggunakan metode *waterfall*. Kelebihan menggunakan metode ini adalah mengidentifikasi *requirement* sebelum pemograman dimulai sehingga meminimalkan perubahan saat perancangan sistem dimulai. Kemudian, metode ini memungkinkan untuk mengontrol proses pengembangan model fase *one by one*, sehingga meminimalkan kesalahan yang mungkin akan terjadi. Di sisi lain, kelemahan dalam menggunakan metode ini adalah desain harus ditentukan sebelum pemograman dimulai dan membutuhkan waktu yang lama dalam penyelesaian proposal sistem bahkan sampai bertahun-tahun [2]. Model *waterfall* ini akan digunakan dalam perancangan *website* terhadap *online shop* penjualan aksesoris Elsoil.



Gambar 2.2 Metode *Waterfall*

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, *System Analysis & Design An Object-Oriented Approach with UML*, 5th ed. USA: Wiley, 2015.

2.3 Unified Modeling Language (UML) 2.5

UML atau *Unified Model Language* dikelompokkan menjadi dua bagian, yaitu *structure modeling* dan *behavior modeling* [2]. UML merupakan suatu bahasa pemodelan yang menggunakan konsep orientasi objek untuk menspesifikasi, memvisualisasi, membangun, dan mendokumentasi hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual [4]. Versi yang akan digunakan dalam penelitian ini yaitu UML versi 2.5 dan pemodelan sistem akan menggunakan *activity diagram*, *use case diagram*, dan *class diagram*.

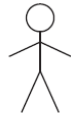
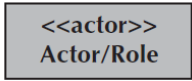
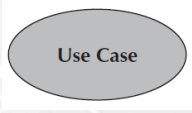
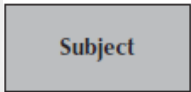

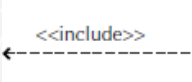


2.3.1 Behavior Diagram

Behavior diagram adalah penggambaran pemodelan perilaku suatu sistem. Diagram ini menganalisis dengan cara menggambarkan hubungan dinamis antara contoh atau objek yang mewakili sistem informasi bisnis. Kemudian, *diagram* meliputi *activity diagram*, *use case diagram*, *state machine diagram*, *interaction diagram*, *sequence diagram*, *interaction overview diagram*, *communication diagram*, dan *timing diagram* [2]. Pemodelan yang akan digunakan pada penelitian ini adalah *activity diagram* dan *use case diagram*.

2.3.1.1 Use Case Diagram

Use Case Diagram adalah penggambaran interaksi antara aktor, *use case*, dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. *Use case* bertujuan untuk mendeskripsikan interaksi antara *user* sebuah sistem dengan sistemnya dengan menjelaskan cara sebuah sistem dipakai. *Use case* memiliki dua istilah, yaitu *system use case* yang berinteraksi dengan sistem dan *business use case* yang berinteraksi dengan konsumen atau kejadian nyata [4]. *Use case* merupakan awal untuk fase pengembangan yang menggambarkan kebutuhan fungsional sistem. Elemen-elemen yang terdapat di dalam *use case diagram* dapat dilihat pada tabel 2.1.

Tabel 2.1 Elemen pada *Use Case Diagram*



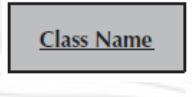

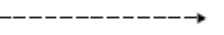

No.	Simbol	Elemen
1.	 <p>Actor/Role</p> 	Actor: <ul style="list-style-type: none"> • Adalah seseorang atau sistem yang memperoleh keuntungan dari dan berada di luar subjek. • Digambarkan dengan <i>stick figure (default)</i> dan apabila bukan orang akan digambarkan dengan kotak yang bertuliskan <code><<actor>></code>. • Diberi label sesuai dengan peran. • Dapat berasosiasi dengan <i>actor</i> lain dengan <i>specialization/superclass</i>. • Diletakan di luar <i>subject boundary</i>.
2.		Use Case: <ul style="list-style-type: none"> • Merepresentasikan fungsi sistem. • Dapat <i>extend</i> ke <i>use case</i> lainnya. • Dapat <i>include</i> ke <i>use case</i> lainnya. • Diletakan di dalam <i>subject boundary</i>. • Diberi label dengan <i>descriptive verb-noun phrase</i>.
3.		Subject Boundary: <ul style="list-style-type: none"> • Nama subjek ditulisa di dalam atau di atas. • Merepresentasikan <i>scope</i> dari subjek tersebut.
4.		Association Relationship: <ul style="list-style-type: none"> • Menghubungkan antara <i>actor</i> dan <i>use case</i> yang berinteraksi
5.		Include Relationship: <ul style="list-style-type: none"> • Merepresentasikan penyertaan fungsionalitas satu <i>use case</i> dengan <i>use case</i> lainnya. • Tanda panah mengarah dari <i>base use case</i> ke <i>used use case</i>.
6.		Extend relationship: <ul style="list-style-type: none"> • Merupakan <i>extension</i> dari <i>use case</i> untuk memasukan perilaku opsional. • Memiliki tanda panah yang mengarah dari <i>extend use case</i> menuju ke <i>base use case</i>.
7.		Generalization Relationship: <ul style="list-style-type: none"> • Mewakili <i>use case</i> khusus ke <i>use case</i> umum • Tanda panah mengarah ke <i>parent/ base use case</i>.

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, *System Analysis & Design An Object-Oriented Approach with UML*, 5th ed. USA: Wiley, 2015.



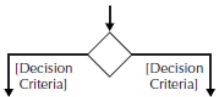
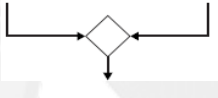


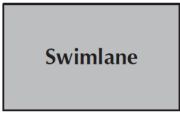
2.3.1.2 Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku dalam proses bisnis yang tidak bergantung pada *object* dan untuk memahami proses bisnis dari model *use case* yang sudah dibuat. Diagram ini berupa aktivitas-aktivitas, objek, *state*, transaksi *state*, dan *event* yang menggambarkan proses aktivitas-aktivitas domain bisnis secara umum. Kemudian, *activity diagram* dapat digunakan untuk memodelkan segala jenis proses mulai dari *high level business workflow* yang terdiri dari banyak *use case*, hingga detail *individual use case* [2]. Elemen-elemen yang digunakan dalam *activity diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Elemen pada *Activity Diagram*

No.	Simbol	Elemen
1.		Action: <ul style="list-style-type: none"> Merupakan perilaku sederhana dan tidak dapat didekomposisi. Diberi label sesuai dengan namanya.
2.		Activity: <ul style="list-style-type: none"> Mewakili satu <i>set action</i>. Diberi label sesuai dengan namanya.
3.		Object Node: <ul style="list-style-type: none"> Mewakili sebuah <i>object</i> yang terhubung dengan <i>set of object flow</i>. Diberi label sesuai dengan nama kelasnya.
4.		Control Flow: <ul style="list-style-type: none"> Menunjukkan urutan eksekusi.
5.		Object Flow: <ul style="list-style-type: none"> Menunjukkan arus sebuah objek dari satu aktivitas (atau <i>action</i>) ke aktivitas lain (atau <i>action</i>).
6.		Initial Node: <ul style="list-style-type: none"> Menggambarkan awal dari serangkaian aktivitas atau <i>action</i>.

Tabel 2.2 Elemen pada *Activity Diagram* (Lanjutan)

No.	Simbol	Elemen
7.		Final-Activity Node: <ul style="list-style-type: none"> Digunakan untuk menghentikan <i>semua control flow</i> dan <i>object flow</i> dalam suatu aktivitas atau <i>action</i>.
8.		Final-Flow Node: <ul style="list-style-type: none"> Digunakan untuk menghentikan <i>control flow</i> dan <i>object flow</i> tertentu.
9.		Decision Node: <ul style="list-style-type: none"> Digunakan untuk mewakili kondisi pengujian untuk memastikan <i>control flow</i> dan <i>object flow</i> turun satu jalur. Diberikan label kriteria keputusan untuk melanjutkan ke jalur tertentu.
10.		Merge Node: <ul style="list-style-type: none"> Digunakan untuk mengembalikan jalur keputusan yang berbeda dari <i>decision node</i>.
11.		Fork Node: <ul style="list-style-type: none"> Digunakan untuk memisahkan perilaku secara paralel atau bersamaan menjadi serangkaian arus aktivitas/<i>action</i>.
12.		Join Node: <ul style="list-style-type: none"> Digunakan untuk menggabungkan kembali aktivitas/ <i>action</i> yang paralel atau bersamaan.
13.		Swimlane: <ul style="list-style-type: none"> Digunakan untuk membagi activity diagram menjadi beberapa baris dan kolom untuk menetapkan aktivitas individu (atau <i>action</i>) kepada individu atau objek yang bertanggung jawab dalam melaksanakan aktivitas (atau <i>action</i>) Diberi label dengan nama individu atau objek yang bertanggung jawab

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, *System Analysis & Design An Object-Oriented Approach with UML*, 5th ed. USA: Wiley, 2015.

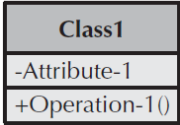
2.3.2 Structure Diagram

Structure diagram adalah penggambaran pemodelan dari struktur data yang mendukung proses bisnis suatu organisasi. Diagram ini menyediakan cara untuk mewakili data dan hubungan statis dalam sistem informasi. Kemudian, *structure diagram* dibagi menjadi 6 yaitu *class diagram*, *composite structure diagram*, *component diagram*, *deployment diagram*, *object diagram*, dan *package diagram* [2]. Pada penelitian ini, peneliti akan menggunakan *class diagram* dalam memodelkan proses sistem *website* penjualan Elsoil.

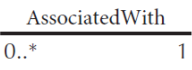

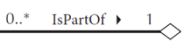

2.3.2.1 Class Diagram

Class Diagram merupakan model statis yang menunjukkan kelas (perilaku dan status) dan hubungan antara kelas dalam sebuah sistem [2]. Tabel 2.3 menjelaskan elemen-elemen yang terdapat pada *class diagram*.

Tabel 2.3 Elemen pada *Class Diagram*

No.	Simbol	Elemen
1.		Class: <ul style="list-style-type: none">• Mewakili seseorang, tempat, atau hal tentang sistem, yang dibutuhkan untuk mengambil dan menyimpan informasi.• Diberi nama dengan dicetak tebal dan diletakkan di bagian tengah atas.• Memiliki daftar atribut di bagian tengah.• Memiliki daftar operasi di bagian bawah.• Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas
2.	<i>Attribute Name / Derived Attribute Name</i>	Attribute: <ul style="list-style-type: none">• Mewakili properti yang mendeskripsikan kondisi dari sebuah objek.• Bisa diturunkan dari atribut lain, dengan menunjukkan garis miring (/) sebelum nama atribut.
3.	<i>Operation Name ()</i>	Operation: <ul style="list-style-type: none">• Mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas.• Dapat diklasifikasikan sebagai <i>constructor</i>, <i>query</i>, atau <i>update operation</i>.

Tabel 2.3 Elemen pada *Class Diagram* (Lanjutan)

No.	Simbol	Elemen
		<ul style="list-style-type: none"> Termasuk tanda kurung () yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.
3.		<p>Association:</p> <ul style="list-style-type: none"> Mewakili sebuah hubungan antara beberapa kelas atau dengan dirinya sendiri. Diberi label menggunakan frase kata kerja atau nama peran. Dapat berada di antara satu atau lebih kelas. Berisi beragam simbol, yang merepresentasikan waktu minimum dan maksimum sebuah <i>class instance</i> dapat diasosiasikan dengan <i>class instance</i> terkait.
5.		<p>Generalization:</p> <ul style="list-style-type: none"> Mewakili hubungan antara beberapa kelas
6.		<p>Aggregation:</p> <ul style="list-style-type: none"> Mewakili bagian logis dari hubungan antara beberapa kelas atau dengan dirinya sendiri. Merupakan bentuk asosiasi khusus.
7.		<p>Composition:</p> <ul style="list-style-type: none"> Mewakili bagian fisik dari hubungan antara beberapa kelas atau dengan dirinya sendiri. Merupakan bentuk asosiasi khusus.

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, *System Analysis & Design An Object-Oriented Approach with UML*, 5th ed. USA: Wiley, 2015.

2.4 Teknik Pemograman Website

Pemograman *website* dalam penelitian ini menggunakan bahasa pemograman HTML, CSS, *JavaScript*, PHP, dan komponen-komponen lainnya.

2.4.1 *Hypertext Markup Language* (HTML)

Hypertext Markup Language (HTML) adalah bahasa pemrograman yang umum digunakan untuk membuat halaman *website*. atau *Hypertext Markup Language*, dibuat oleh Tim Berners-Lee ketika masih bekerja untuk CERN dan dipopulerkan pertama kali oleh browser Mosaic [5]. HTML terdiri dari *Hypertext* dan *Markup Language*. *Hypertext* artinya ada beberapa *text* dapat digunakan sebagai

link untuk menghubungkan satu halaman ke halaman lainnya. Sedangkan *markup language* artinya bahasa pemrograman ini menggunakan tanda (*mark*) untuk menandai bagian dari *text*. Biasanya, dalam membuat suatu *website*, selain HTML, *programmer* juga menggunakan CSS dan *JavaScript* yang akan dibahas di subbab selanjutnya.

Dalam penulisan HTML, HTML memiliki dua bagian, yaitu bagian pembuka dan bagian penutup. Tag HTML ditulis dengan <HTML> sebagai pembuka dan </HTML> sebagai penutup. Sebuah dokumen HTML memerlukan minimal empat set element, yaitu <HTML> ... </HTML> , <head> ... </head>, <title> ... </title>, <body> ... </body>. Saat ini versi terbaru dari HTML adalah HTML5 [4]. Gambar 2.3 adalah contoh penulisan HTML.

```
<!DOCTYPE html>
<html>
<head>
  <title>Example for Nested HTML Elements</title>
</head>
<body>
  <p>This is my paragraph text with <strong>bold</strong> word.</p>
</body>
</html>
```

Gambar 2.3 Contoh *Coding* HTML

Sumber: W3School.

2.4.2 *Cascading Style Sheets* (CSS)

CSS (*Cascading Style Sheet*) adalah salah satu bahasa desain *website* (*style sheet language*) yang mengontrol format tampilan sebuah halaman *website* yang ditulis dengan menggunakan penanda (*markup language*) [6]. CSS ditulis dengan *markup language*. Elemen yang dapat didesain oleh CSS meliputi *font type*, *font size*, *color*, *font style*, *position*, *background*, dan sebagainya. Cara menambahkan *style* ke dalam *website* dengan CSS adalah meletakkannya pada bagian atas halaman diantara <head> dan </head> dengan menuliskan <style> dan diakhiri dengan

</style>. Pada umumnya, *file* css akan disimpan pada satu *file* bernama *style.css* yang mencakup desain semua halaman *website* agar penggunaannya dapat lebih efektif dan efisien. Contoh penggunaan CSS dapat dilihat pada gambar 2.4.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
    <style>
      h1 { color:red; font-size:3em; font-family:Arial; }
    </style>
  </head>
  <body>
    <h1>Hello there</h1>
  </body>
</html>
```

Gambar 2.4 Contoh Coding CSS

Sumber: R. Nixon, Learning PHP, MySQL, *JavaScript*, and CSS: a Step by Step Guide to Creating Dynamic *Website*, 2 Edition, USA: O'Reilly, 2012.

2.4.3 *JavaScript*

JavaScript adalah bahasa *client side scripting* yang berjalan di dalam *web browser*. Fungsinya adalah untuk membuat *website* menjadi lebih dinamis dan interaktif [7]. Kode *JavaScript* bisa disisipkan dalam bagian awal, tengah atau akhir halaman *website* dengan menggunakan tag <script> diakhiri dengan tag </script>. Penggunaan *JavaScript* ini diikuti juga dengan penggunaan HTML dan CSS. Contoh mengkodekan “Hello World” dengan menggunakan *JavaScript* seperti pada gambar 2.5.

```
<html>
  <head><title>Hello World</title></head>
  <body>
    <script type="text/javascript">
      document.write("Hello World")
    </script>
  </body>
</html>
```

Gambar 2.5 Contoh Coding JavaScript

Sumber: R. Nixon, Learning PHP, MySQL, *JavaScript*, and CSS: a Step by Step Guide to Creating Dynamic *Website*, 2 Edition, USA: O'Reilly, 2012.

2.4.4 *Hypertext Preprocessor (PHP)*

PHP atau *Hypertext Preprocessor* merupakan salah satu bahasa pemrograman *open source* yang sering digunakan dalam membangun aplikasi *website*. PHP ini merupakan bahasa *server-side scripting* yang digabungkan dengan HTML untuk membuat dan mengontrol halaman *website* yang dinamis. Ketika program PHP dipanggil oleh *web browser*, program tersebut akan di-*parsing* di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya ditampilkan kembali ke *web browser*. Pemrosesan program PHP yang dilakukan di dalam *web server* sehingga PHP dikatakan sebagai bahasa *server side* [7].

2.4.5 *CodeIgniter*

Setelah menguasai dasar-dasar PHP, banyak developer menggunakan *framework* dalam pekerjaan mereka. *Framework* dapat diartikan sebagai sekumpulan fungsi-fungsi dan class-class untuk tujuan tertentu sehingga mempercepat pekerjaan mereka. Salah satu *framework* yang digunakan adalah *codeigniter*. *Codeigniter* adalah sebuah *framework* PHP yang bersifat *open source* dengan menggunakan metode MVC (*Model, View, Controller*) untuk memudahkan *programmer* dalam membangun sebuah aplikasi *website* tanpa harus membuatnya dari awal. *Framework* ini terdapat tiga komponen utama yaitu *Model, Controller*, dan *View*. *Model* berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete*) dan menangani validasi dari bagian *controller*, serta tidak berhubungan langsung dengan bagian *view*. Kemudian, *controller* bertugas untuk mengontrol aliran hubungan antara *model* dan *view*. Lalu, *view* merupakan file HTML untuk menampilkan data pada *browser* dan diatur pada *controller*. *Framework* yang dikembangkan oleh *Rick Ellis* ini memiliki beberapa kelebihan, yaitu berupa *open source*, berukuran kecil, dan menggunakan konsep MVC.

2.4.6 MySQL

Pada masa awal *website*, banyak situs *website* menggunakan *file text* “flat” dalam menyimpan data, akan tetapi penggunaan ini terdapat beberapa kekurangan, seperti kesulitan mengelola, menggabungkan, dan mencari *file* yang kompleks. Maka dari itu, *database* yang terstruktur menjadi hal yang sangat penting. *Database* merupakan sekumpulan data-data yang dikemas secara terstruktur dan rapi serta dapat dikelola menggunakan suatu *software* untuk menghasilkan suatu informasi. Sedangkan MySQL adalah manajemen sistem basis data untuk mengelola atau memanajemen data dengan menggunakan perintah dasar SQL (*Structured Query Language*).

Beberapa kelebihan MySQL yaitu: [7]

- 1) MySQL dapat berjalan dengan stabil di berbagai *operation system*;
- 2) Bersifat *open source* (gratis);
- 3) Bersifat *multiuser* atau beberapa user dalam waktu yang bersamaan tanpa mengalami masalah;
- 4) MySQL memiliki kecepatan yang baik dalam mengelola *query*.

2.4.7 Testing

Testing merupakan suatu hal yang diperlukan untuk memastikan apakah sistem yang dirancang sesuai dengan prosedur yang seharusnya. *Testing* adalah proses menemukan dan memperbaiki *errors*. *Testing* dapat dibagi menjadi empat tahap, yaitu *unit test*, *integration test*, *system test*, dan *acceptance test*. Pada penelitian ini, metode pengujian yang dilakukan adalah *black box testing* pada *unit testing*. *Black box testing* merupakan suatu metode pengujian yang menguji fungsionalitas *website* tanpa harus mengetahui *programming codes* dari *website* tersebut. Para penguji akan berinteraksi dengan GUI / *Graphical User Interface* untuk mengetahui bagaimana sistem tersebut bekerja sehingga dapat dilakukan verifikasi sistem yang dirancang sudah sesuai [2].

2.5 Penelitian Sebelumnya

Untuk merancang *website* terhadap penjualan aksesoris wanita pada *online shop Elsoil*, peneliti ingin membandingkan kesimpulan dari penelitian sebelumnya yang terkait dengan topik ini. Terdapat 5 *paper* yang membahas mengenai topik yang menyerupai.



Tabel 2.4 Perbandingan penelitian sebelumnya mengenai tema yang menyerupai

No	Judul	Penulis	Masalah	Metode	Hasil / kesimpulan
1.	Sistem Informasi Penjualan Aksesoris Berbasis <i>Web</i> di Toko Aliya <i>Accessories Olshop</i>	Sudarsih (2018)	<ul style="list-style-type: none"> - Bagaimana merancang suatu sistem informasi penjualan berbasis <i>web</i> yang memudahkan konsumen dalam mencari produk dan memperoleh produk di Toko Aliya <i>Accessories Olshop</i>? - Bagaimana menampilkan data order seluruh penjualan pada <i>web</i> Toko Aliya <i>Accessories Olshop</i>? 	Metode pengembangan menggunakan sistem <i>waterfall</i> . dan studi literatur	<ul style="list-style-type: none"> - Dengan adanya suatu sistem informasi yang berbasis <i>e-commerce</i> konsumen dapat dengan sangat mudah mencari produk yang diinginkan sehingga membuat omset penjualan semakin meningkat, karena para konsumen dapat dengan mudah memesan produk yang diinginkan, karena ketersediaan stok juga langsung otomatis dari sistem. - Data order pada <i>website</i> dapat dilihat pada laman <i>admin</i> pada menu order dimana disana ditampilkan semua data order seluruh penjualan pada <i>web</i> Toko Aliya <i>Accessories Olshop</i>.
2.	Perancangan Sistem Informasi Penjualan Berbasis <i>Web</i> (Studi Kasus pada Newbiestore)	Dadan Zaliluddin dan Rohmat (2018)	<ul style="list-style-type: none"> - Bagaimana merancang dan membuat sistem informasi penjualan <i>online</i> berbasis <i>web</i> yang mendukung transaksi penjualan pakaian pada Newbiesore ? - Bagaimana mempermudah proses transaksi penjualan pada Newbiestore ? 	Metode pengembangan menggunakan <i>waterfall</i> yang merupakan bagian dari <i>system development life cycle</i> (SDLC)	<ul style="list-style-type: none"> - Aplikasi ini menyajikan informasi terkait dengan <i>Newbiestore</i> yang ditujukan untuk pembeli. - Dengan adanya aplikasi ini, informasi tentang produk terbaru di <i>Newbiestore</i> dapat terpublikasi dengan baik. - Dengan adanya aplikasi ini wilayah pemasaran <i>Newbiestore</i> menjadi luas
3.	Perancangan <i>Website</i> Sistem Informasi Pemasaran dan Penjualan Berbasis	Ajrul Azwar (2013)	<ul style="list-style-type: none"> - Bagaimana sistem penyampaian informasi pemasaran dan penjualan yang sedang berjalan untuk saat ini di S28? 	Metode pengembangan menggunakan metode <i>prototyping</i> dan pengumpulan data melalui observasi	<ul style="list-style-type: none"> - Dengan adanya <i>website</i> ini maka konsumen sudah dapat memperoleh informasi produk <i>foodmart</i> secara optimal tanpa harus datang langsung ketempat perusahaan beroperasi. - Dengan adanya <i>website</i> ini maka ruang lingkup penjualan dapat merambah ke kota-kota yang ada di Indonesia.

Tabel 2.4 Perbandingan penelitian sebelumnya mengenai tema yang menyerupai (Lanjutan)

No	Judul	Penulis	Masalah	Metode	Hasil / kesimpulan
	<i>E-Commerce</i> pada S28		- Bagaimana perancangan <i>website</i> yang interaktif sehingga menunjang peningkatan terhadap minat para pengunjung untuk tertarik terhadap produk yang di tawarkan?	berorientasi pada objek	<ul style="list-style-type: none"> - <i>Website</i> ini menjadi media alternatif pemesanan produk <i>foodmart</i> bagi konsumen tanpa harus datang langsung ke perusahaan. - Dengan adanya <i>website</i> ini maka konsumen dapat mengetahui informasi-informasi lainnya yang ada di perusahaan tanpa harus datang langsung ke tempat perusahaan beroperasi.
4.	Rancang Bangun Sistem Informasi Sepatu Berbasis <i>Website</i> dengan <i>Waterfall</i>	Ibnu Dewi Lesmono (2018)	- Bagaimana merancang bangun sistem informasi penjualan sepatu berbasis <i>website</i> ?	Metode pengembangan menggunakan <i>waterfall</i> dengan teknik pengumpulan data observasi dan studi kepustakaan	<ul style="list-style-type: none"> - Perancangan <i>website</i> sangat penting untuk mengatasi persaingan bisnis khususnya penjualan sepatu. - Laman dinamis yang dibuat dengan rancangan yang terencana dengan baik akan menghasilkan laman yang menghasilkan informasi yang cepat, tepat, akurat, menarik dan <i>up to date</i> untuk pelanggan. <i>Website</i> juga akan memperjelas dan mempermudah pengolahan data transaksi. - Agar <i>website</i> dapat berjalan dalam jangka panjang, perlu adanya <i>back up</i> data dan diberikan perawatan yang berkala.
5.	Pembuatan Sistem Informasi Penjualan Berbasis <i>Web</i> untuk Memperluas Pangsa Pasar	Sandy Kosasi (2014)	- Bagaimana membuat sistem informasi penjualan berbasis <i>web</i> untuk memperluas pangsa pasar?	Metode pengembangan menggunakan <i>incremental development</i> dengan pendekatan berorientasi objek dan pengumpulan data dengan wawancara dan observasi teknik <i>purposive sampling</i>	<ul style="list-style-type: none"> - Sistem informasi penjualan berbasis <i>web</i> memiliki fitur navigasi yang memberikan kemudahan bagi pengunjung. - Menampilkan pesan-pesan tertentu dalam mengarahkan pengunjung, calon pembeli dapat melakukan proses pemesanan barang secara <i>online</i> kapanpun dan dimanapun serta mendapatkan informasi produk secara <i>up to date</i>. - Untuk realisasi penerapannya membutuhkan kesiapan sumberdaya manusia dan ketersediaan infrastruktur teknologi informasi dalam kelancaran transaksi dan menjalin interaksi yang personal dengan pelanggan perusahaan.