

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 *Object Oriented Analysis and Design (OOAD)*

OOAD adalah pendekatan populer untuk menganalisis dan merancang aplikasi bisnis dan teknik dengan menerapkan paradigma Berorientasi Objek dan pemodelan visual selama SDLC untuk mendorong komunikasi pemangku kepentingan yang lebih baik dan meningkatkan kualitas produk. Iterasi demi iterasi, model analisis untuk OOA, dan model desain untuk OOD, disempurnakan dan terus berkembang didorong oleh faktor kunci seperti risiko dan nilainya bagi bisnis atau sistem yang akan dibuat.

Singkatnya OOAD menganalisis dan merancang aplikasi perangkat lunak dengan menerapkan paradigma Berorientasi Objek dan pemodelan visual selama Siklus Hidup Pengembangan sehingga menghasilkan komunikasi yang lebih baik antara Pengembang, pelanggan, dan pemangku kepentingan ditambah peningkatan kualitas produk. (Gechman, 2019).

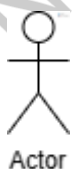


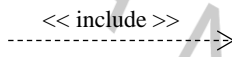
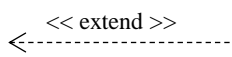
2.1.2 *Unified Modeling Language (UML)*

UML adalah suatu bahasa yang digunakan untuk mendefinisikan, memvisualisasikan, membuat, dan mendokumentasikan informasi yang dihasilkan dalam proses pengembangan perangkat lunak. UML digunakan untuk pemodelan bisnis dan sistem non perangkat lunak lainnya. Pemanfaatan UML melibatkan kemampuan untuk memodelkan sistem menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat dipahami oleh manusia maupun mesin. (Habibi, Masruro, & Khonsa, 2020). Terdapat berbagai macam jenis diagram UML yang digunakan dalam proses pengembangan sistem, di antaranya adalah sebagai berikut:

1. Use Case Diagram

Diagram *use case* mencakup aktor-aktor yang merupakan visualisasi entitas pihak atau sistem yang berinteraksi dengan sistem yang sedang dikembangkan. (Prihandoyo, 2018). Berikut ini adalah beberapa simbol yang digunakan dalam diagram *use case*:

Tabel 2.1 Simbol Use Case



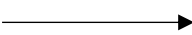
Simbol	Nama	Keterangan
	<i>Actor</i>	Mewakili entitas eksternal, seperti pengguna, sistem lain, atau perangkat keras, yang berinteraksi dengan sistem yang sedang dikembangkan
	<i>Use Case</i>	Simbol yang menggambarkan kasus proses
	<i>Association</i>	Garis penghubung antara aktor dan suatu kasus proses atau sistem
	<i>Include</i>	Simbol yang menunjukkan bahwa suatu kasus proses termasuk dari fungsionalitas kasus proses lain
	<i>Extend</i>	Simbol yang menggambarkan suatu kasus proses merupakan fungsionalitas tambahan dari kasus proses lain


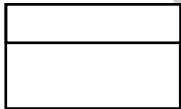
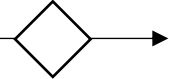
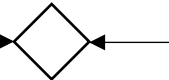
		saat memenuhi kondisi tertentu.
--	--	---------------------------------

2. Activity Diagram

Diagram ini digunakan untuk memvisualisasikan langkah-langkah atau aktivitas yang terjadi dalam suatu proses atau sistem, termasuk urutan, keputusan, dan paralelisme antara aktivitas-aktivitas tersebut. *Activity diagram* memberikan gambaran yang jelas mengenai aliran kerja dan mempermudah pemahaman terhadap interaksi antara berbagai elemen dalam sistem atau proses yang sedang dianalisis. (Syarif & Nugraha, 2020). Berikut ini adalah beberapa simbol yang digunakan dalam diagram *activity*:

Tabel 2.2 Simbol Activity Diagram

Simbol	Nama	Keterangan
	<i>Initial State</i>	Simbol yang menandakan titik awal atau langkah pertama dalam alur kerja atau aktivitas yang direpresentasikan dalam diagram
	<i>Final State</i>	Simbol yang menunjukkan titik akhir atau langkah terakhir dalam alur kerja atau aktivitas yang direpresentasikan dalam diagram
	<i>Association</i>	Simbol yang menghubungkan


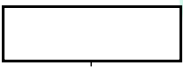

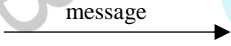
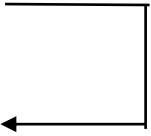
		aktivitas dengan aktivitas selanjutnya
	<i>Activity</i>	Simbol yang menggambarkan suatu aktivitas, biasanya diisi dengan kata kerja
	<i>Swimlane</i>	Menggambarkan pengelompokkan dan memisahkan aktor atau unit organisasi dalam diagram aktivitas berdasarkan peran atau tanggung jawab
	<i>Decision</i>	Menggambarkan adanya lebih dari satu kondisi tertentu yang dapat dipilih untuk pengambilan keputusan
	<i>Merge</i>	Menggambarkan penggabungan alur yang terjadi oleh <i>Decision</i>

3. *Sequence Diagram*

Sequence diagram adalah bentuk diagram UML yang digunakan untuk menggambarkan interaksi antara objek-objek yang ada di dalam sistem dan di sekitarnya, termasuk pengguna, tampilan, dan elemen lainnya. Diagram ini menggunakan pesan-pesan yang digambarkan sepanjang sumbu waktu untuk menunjukkan urutan dan hubungan antara objek-objek tersebut. Melalui *sequence diagram*, kita dapat memvisualisasikan bagaimana objek-objek

berinteraksi dan berkomunikasi satu sama lain selama menjalankan fungsi atau skenario tertentu. Dengan menggunakan simbol-simbol dan notasi yang sesuai, *sequence diagram* memberikan gambaran yang jelas tentang pesan-pesan yang dikirim dan diterima antara objek-objek dalam sistem. (Syarif & Nugraha, 2020). Berikut ini adalah beberapa simbol yang digunakan dalam diagram *sequence*:

Tabel 2.3 Simbol Sequence Diagram

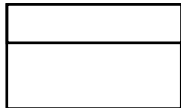

Simbol	Nama	Keterangan
	<i>Actor</i>	Representasi dari aktor yang terlibat dalam interaksi di dalam <i>sequence</i>
	<i>Object</i>	Representasi dari objek yang terlibat dalam interaksi di dalam <i>sequence</i>
	<i>Activation</i>	Menggambarkan masa aktif dan berakhirnya suatu objek
	<i>Message</i>	Komunikasi antar objek mengenai informasi proses yang sedang terjadi
	<i>Self Message</i>	Menunjukkan interaksi atau pesan yang dikirim oleh objek kepada dirinya sendiri

4. Class Diagram

Class diagram adalah jenis diagram UML yang memvisualisasikan interaksi antara sistem dan objek-objek yang

terkait, termasuk pengguna, tampilan, dan lainnya. Diagram ini menggambarkan hubungan antara kelas-kelas dan objek-objek tersebut dalam bentuk pesan atau komunikasi yang dijelaskan dalam konteks waktu. *Class diagram* membantu dalam memahami struktur dan interaksi di dalam sistem dengan cara yang jelas dan terorganisir. Dengan menggunakan simbol-simbol yang tepat, class diagram menyediakan representasi visual yang memberikan gambaran tentang hubungan dan atribut yang dimiliki oleh objek-objek dalam sistem yang sedang dianalisis. (Syarif & Nugraha, 2020). *Class Diagram* merupakan salah satu teknik visual dalam pemodelan yang digunakan dalam proses desain dan pembuatan perangkat lunak berbasis objek. UML (Unified Modeling Language) adalah standar notasi yang digunakan untuk menyusun berbagai jenis diagram yang mencakup proses bisnis serta representasi kelas-kelas yang spesifik. *Class Diagram* membantu dalam menggambarkan struktur statis dari sistem perangkat lunak, termasuk hubungan antara kelas-kelas, atribut, dan metode yang dimiliki oleh setiap kelas. Dengan menggunakan notasi dan simbol yang ditentukan oleh UML, *Class Diagram* menyediakan cara yang sistematis untuk mewakili elemen-elemen penting dalam desain perangkat lunak berbasis objek. (Prihandoyo, 2018). Berikut ini adalah beberapa simbol yang digunakan dalam diagram *class*:

Tabel 2.4 Simbol Class Diagram

Simbol	Nama	Keterangan
	<i>Class</i>	Menggambarkan sekelompok objek dengan atribut dan fungsi yang serupa
	<i>Association</i>	Menggambarkan hubungan antar kelas dengan kelas lainnya

0 .. *	<i>Cardinality</i>	Menggambarkan jenis hubungan jumlah antar kelas dengan kelas lainnya
1 .. *		
1 .. 1		

2.1.3 Hypertext Processor (PHP)

a. Pengertian

PHP, yang merupakan kependekan dari *Hypertext Preprocessor*, adalah sebuah bahasa pemrograman yang sering digunakan dalam pengembangan, pembuatan, dan perluasan situs web. Umumnya, PHP digunakan bersama dengan HTML. Di masa lalu, PHP memiliki peran penting dalam industri pengembangan *web* dan hingga saat ini masih menjadi bahasa pemrograman *server-side* utama yang digunakan untuk membuat situs *web*. PHP dianggap sebagai bahasa pemrograman sisi *server* karena kode PHP diproses oleh mesin *server* sebelum hasilnya dikirim ke peramban web pengguna. Awalnya, PHP adalah singkatan dari *Personal Home Page* yang menggambarkan penggunaan awalnya untuk membuat halaman *web* pribadi. Namun, seiring berjalannya waktu, PHP telah mengalami perkembangan yang signifikan dan menjadi bahasa pemrograman *web* yang andal. PHP telah menunjukkan kehandalannya sebagai bahasa pemrograman *web* yang fleksibel dan dapat diandalkan. Dengan dukungan dan komunitas yang besar, PHP terus berkembang dan menjadi pilihan populer bagi para pengembang untuk menciptakan dan memperluas situs *web* dengan berbagai kompleksitas dan skala.

b. Fungsi

Dalam pembuatan situs *web* HTML lah yang digunakan untuk membuat situs *web*. Situs *web* yang dibuat menggunakan HTML disebut situs *web* statis, di mana konten dan halaman *web* tidak dapat berubah secara dinamis. Tetapi, jika menggunakan

PHP konten *web* dapat menjadi dinamis, di mana konten dapat berubah sesuai dengan kondisi tertentu. *Web* yang menggunakan PHP dapat mengubah konten berdasarkan input pengguna dan menyimpan data pada database dan lain-lain. Karena fungsinya yang demikian, PHP juga sering disebut sebagai *scripting language*. (Oetomo & Mahargiono, 2020).

2.1.4 Database

Database adalah himpunan objek atau data mentah yang tersusun secara sistematis sehingga dapat diambil atau dimanipulasi dengan mudah dan efisien saat diperlukan. Dalam database, informasi diatur dan diindeks dalam baris, kolom, dan tabel untuk mempercepat pencarian. Ambil contoh database Universitas dengan data mahasiswa dan fakultas seperti yang disebutkan di bagian definisi data. Semua tanggal mahasiswa dan fakultas yang dikumpulkan akan disimpan dalam tabel dengan cara yang terorganisir dengan baik, sehingga kapan pun diperlukan, dapat diambil untuk pelaporan dan manipulasi secara efisien. (Negi, 2019). Menurut (Setyawan & Pratiwi, 2020) berikut adalah tujuan dan fungsi database:

a. *Availability*

Fungsi pertama dari database adalah menyediakan informasi penting saat dibutuhkan. Ini adalah fungsi penting dari database, meskipun tidak terletak di tempat yang sama dan disimpan di *disk*, tetapi dengan metode penyimpanan yang sistematis, membuat informasi menjadi mudah diperoleh.

b. *Speed*

Fungsi dari database ini adalah agar pengguna dapat dengan mudah mengaksesnya kapan pun pengguna membutuhkannya. Pengguna tidak perlu menunggu sampai nanti, apalagi menjadwalkan waktu tertentu untuk mendapatkannya.

c. *Completeness*

database harus menyimpan informasi lengkap yang dapat memenuhi kebutuhan umum penggunanya. Meskipun kata yang digunakan di sini bersifat relatif secara keseluruhan, tetapi setidaknya data tersebut dapat membantu memudahkan untuk ditambahkan ke kumpulan data dan memastikan bahwa pengguna dapat mengubah struktur data yang ada dengan mudah.

d. *Accuracy and Security*

Untuk menghindari kesalahan sebanyak mungkin, pengguna dapat mengatur file database dengan benar untuk menghindari kesalahan selama entri data dan juga selama data store atau penyimpanan. Selain itu, peran database adalah keamanan. Pada sistem database yang baik, layanan keamanan informasi memastikan bahwa hanya pihak yang berwenang yang dapat mengakses, menambahkan, mengubah, atau menghapus informasi.

e. *Storage Efficiency*

Pengorganisasian data telah dilakukan dengan baik untuk menghindari duplikasi data, yang dapat mempengaruhi besar penyimpanan database. Coding dan hubungan antar data yang baik berguna untuk menghemat penyimpanan database.

2.1.5 *E-Commerce*

E-commerce dapat diartikan sebagai arena perdagangan atau pertukaran informasi antara penjual dan pembeli di dunia maya. Tak bisa dipungkiri, arena jual beli di dunia maya menghubungkan jutaan komputer dalam satu jaringan yang sangat luas (*Internet*). Munculnya perdagangan elektronik tidak terlepas dari pesatnya perkembangan teknologi informasi, khususnya *internet*. Dengan bantuan *e-commerce*, sebuah perusahaan dapat menjangkau seluruh dunia untuk memasarkan produk atau jasanya tanpa dibatasi oleh batasan geografis. Ketika menerapkan *e-commerce* dalam

memasarkan produk, harus diketahui bahwa bahan baku terpenting adalah informasi. *E-commerce* merupakan salah satu katalisator munculnya prinsip ekonomi baru yang dikenal saat ini sebagai ekonomi *digital*. Toko *online* dirancang untuk memenuhi kebutuhan gaya hidup masyarakat modern yang membutuhkan kenyamanan dan kecepatan di segala bidang. *E-commerce* juga sering disebut sebagai *Market Making* atau pembuat pasar karena secara langsung telah menciptakan pasar di dunia maya yang dapat mempertemukan penjual dan pembeli dari berbagai belahan dunia hanya dengan koneksi *internet*. (Rerung, 2018). Terdapat beberapa jenis *e-commerce* yaitu sebagai berikut:

a. *Business to Business (B2B)*

B2B mengacu pada bisnis atau transaksi antara dua atau lebih perusahaan atau bisnis. Sebagai contoh, ketika produsen menjual produk mereka kepada distributor atau grosir.

b. *Business to Consumer (B2C)*

B2C mengacu pada bisnis atau transaksi antara perusahaan atau bisnis dengan konsumen akhir. Sebagai contoh, ketika seseorang melakukan pembelian produk secara online dari toko *e-commerce* seperti Amazon.

c. *Customer to Customer (C2C)*

Melibatkan transaksi langsung antara konsumen, tanpa melibatkan perusahaan. Sebagai contoh, melalui platform jual-beli online seperti eBay atau Tokopedia, individu dapat membeli dan menjual produk kepada konsumen lainnya.

d. *Consumer-to-Business (C2B)*

Terlibat dalam transaksi di mana konsumen menawarkan produk atau layanan kepada perusahaan. Sebagai contoh, ketika seorang influencer atau freelancer menyediakan layanan mereka kepada perusahaan untuk keperluan pemasaran atau konsultasi.

2.1.6 Black Box Testing

Metode pengujian perangkat lunak yang dikenal sebagai *Black Box Testing* didasarkan pada spesifikasi dan persyaratan yang menentukan fungsionalitas yang diharapkan dari aplikasi, bukan rincian implementasi cara kerjanya. (Khorikov, 2020).

Berdasarkan (Amey, 2022). Berikut adalah kelebihan dan kekurangan dari metode pengujian perangkat lunak *Black Box Testing*.

Tabel 2. 5 Kelebihan dan Kekurangan Black Box Testing

Kelebihan	Kekurangan
Realistis	Kemungkinan tidak mencakup kode secara keseluruhan
Meliputi fungsi utama	Tidak mencakup <i>error cases</i>
Mengambil sudut pandang pengguna	Sulit untuk di <i>de-bug</i>
Tidak perlu mempelajari detail implementasi	Mengulang test yang sudah pernah dilakukan atau repetitif

Keuntungan dari *black box testing* adalah realistis, melakukan tindakan yang akan diambil sendiri oleh pengguna. Pengujian yang lain biasanya mencakupi kasus yang lebih jarang seperti kesalahan dan kegagalan sistem, serangan keamanan, dan pemuatan, dan sebagainya hal-hal tersebut merupakan kasus-kasus penting, tetapi *black box testing* berkemungkinan besar tidak akan mencakupinya. Sementara *black box testing* mencakup kasus penggunaan utama dan bagian produk yang harus berfungsi dengan baik.

Tes ini mengambil sudut pandang pengguna, tanpa kelonggaran untuk kode atau arsitektur sistem yang berarti kemungkinan besar akan menghadapi masalah yang akan memengaruhi pelanggan nyata. Tes ini juga relatif mudah dijalankan. Pihak yang melakukan pengujian tidak perlu mengetahui bahasa pemrograman atau memahami kodenya dan tidak perlu mempelajari teknik untuk pengujian keamanan, juga tidak memiliki

alat untuk menghasilkan beban berlebihan seperti area pengujian lainnya. *Black box testing* ini sistem ini dapat dilakukan oleh pemilik produk atau tim dokumentasi, atau dapat dialihdayakan ke tim eksternal.

2.1.7 Aplikasi

Aplikasi merujuk pada perangkat lunak atau program komputer yang berfungsi di dalam sistem khusus dan dirancang untuk menjalankan perintah atau tugas tertentu. (Pane, Zamzam, & Fadillah, 2020). Kategori aplikasi dapat dibedakan berdasarkan cara pengembangannya, yaitu sebagai berikut;

1. Aplikasi *desktop*, yaitu kumpulan dari program-program yang berjalan di dalam perangkat PC atau laptop.
2. Aplikasi *Web*, yaitu kumpulan dari program-program yang diakses melalui *internet* dan dijalankan di dalam browser web.
3. Aplikasi *mobile*, yaitu kumpulan dari program-program yang dirancang khusus untuk perangkat mobile, seperti ponsel pintar atau tablet, dan memiliki berbagai macam penggunaan.

Berbagai macam jenis aplikasi adalah sebagai berikut:

1. Aplikasi *internet*, merujuk pada aplikasi yang membantu pengguna dalam mengakses *internet*, seperti Yahoo!, Gmail, Google, dan sejenisnya.
2. Aplikasi perkantoran, digunakan untuk melaksanakan tugas-tugas perkantoran dan mempercepat proses, seperti MS Office.
3. Aplikasi grafis, merujuk pada aplikasi yang digunakan untuk mengedit foto dan membuat desain, contohnya adalah Photoshop.

4. Aplikasi *programming*, aplikasi yang dirancang untuk membuat perangkat lunak atau program, seperti Delphi 7, Visual Basic, dan lainnya.
5. Aplikasi multimedia, aplikasi yang mendukung produksi informasi dalam berbagai format, seperti teks, video, dan audio. Contohnya adalah Winamp, pemutar musik, dan lain sebagainya.
6. Aplikasi *games*, aplikasi yang berisi permainan dan dimaksudkan untuk hiburan semata.

2.2 Tinjauan Studi

Dalam rangka mendukung penulisan tugas akhir, dilakukanlah tinjauan studi. Berikut adalah beberapa referensi studi yang telah dikaji.:

1. Jurnal penelitian yang disusun oleh Erik Chandra dan Agus Umar Hamdani dengan judul “PENERAPAN SISTEM INFORMASI PENJUALAN ALAT LISTRIK BERBASIS *E-COMMERCE* STUDI KASUS TOKO LISTRIK MILLENNIUM” pada Jurnal IDEALIS Vol .2 No. 6, November 2019. Dalam penelitian ini, metode *Object Oriented Analysis and Design* (OOAD) digunakan untuk menganalisis dan merancang aplikasi *e-commerce* untuk Toko Listrik Millennium. Tujuan dari metode ini adalah untuk memaksimalkan hasil penjualan toko dengan menggunakan aplikasi *e-commerce* tersebut. Toko Listrik Millennium adalah bisnis milik perorangan yang menjual alat listrik. Pemilik memanfaatkan media sosial untuk mempromosikan produk mereka dan meningkatkan usahanya dengan menjual produk mereka baik secara *offline* maupun secara *online*. Toko listrik Millennium sering menghadapi beberapa masalah selama proses bisnisnya, seperti ketidakakuratan laporan harian dan data barang. Masalah ini menghambat kemampuan toko untuk meningkatkan hasil penjualan. Oleh karena itu, diperlukan adanya sebuah situs *web e-commerce* yang dapat membantu dalam pengolahan data laporan harian dan data barang dengan lebih efisien. (Chandra & Hamdani, 2019).

2. Penelitian ini dilakukan oleh Fitri Purwaningtias dengan judul “*E-COMMERCE* PENJUALAN BERBASIS METODE OOAD (Studi Kasus : PT Musi Utama Bercahaya Palembang)” pada Jurnal Cendikia Vol. XV 2018. Dalam penelitian ini, PT Musi Utama Bercahaya Palembang mengadopsi metode *Object Oriented Analysis and Design* (OOAD) untuk merancang aplikasi *e-commerce*. PT Musi Utama Bercahaya Palembang merupakan perusahaan yang menjual garam. Saat ini, perusahaan masih menggunakan sistem operasional konvensional di mana pelanggan harus datang langsung ke perusahaan untuk melakukan pembelian. Proses pembelian ini memakan waktu sekitar 30 menit untuk membuat berkas transaksi. Sistem tersebut tidak efektif karena memakan banyak waktu dalam melakukan proses transaksi. Selain itu, adanya kemungkinan berkas yang terlalu lama dapat rusak atau hilang. Akumulasi berkas dan arsip data juga menjadi masalah yang mungkin terjadi. Penelitian ini bertujuan untuk mengidentifikasi masalah-masalah tersebut dan menciptakan sistem informasi *e-commerce* untuk penjualan garam dengan menggunakan metode OOAD. Tujuan utama penelitian ini adalah meningkatkan kinerja penjualan yang ada sehingga perusahaan dapat melakukan penjualan dengan cepat dan efisien. (Purwaningtias, 2018).
3. Penelitian ini dilakukan oleh Bayu Dwi Juniansyah, Erliyan Redy Susanto, dan Agung Deni Wahyudi dengan penelitian yang berjudul “PEMBUATAN *E-COMMERCE* PEMESANAN JASA EVENT ORGANIZER UNTUK ZERO SEVEN ENTERTAINMENT” pada Jurnal TEKNOKOMPAK, Vol. 14, No. 1, 2020. Penelitian ini mengkaji pembuatan aplikasi *e-commerce* untuk *Event Organizer* (EO) Zero Seven Entertainment di Bandar Lampung. Pengembangan sistem menggunakan bahasa pemrograman PHP dan phpMyAdmin sebagai DBMS, serta menerapkan metode pengembangan sistem *Extreme Programming*. Tujuan dari penelitian ini adalah mempermudah bisnis EO dalam mempromosikan acara, barang, dan jasa yang mereka tawarkan. Selain itu, aplikasi ini menyediakan kemudahan dalam

melakukan pemesanan langsung melalui situs web, sehingga pelanggan tidak perlu menunggu untuk mendapatkan tempat di lokasi acara. Validasi hasil penelitian dilakukan menggunakan standar ISO 9126. Diharapkan bahwa aplikasi yang dikembangkan dapat membantu EO dalam mengatur dan mengelola acara melalui media yang disediakan oleh perusahaan. (Juniansyah, Susanto, & Wahyudi, 2020).

4. Jurnal penelitian yang disusun oleh Ardi Mardiana, Nunu Nurdiana, dan Risa Risnawati dengan judul “PENGEMBANGAN *E-COMMERCE* PADA TOKO OLEH –OLEH KHAS MAJALENGKASTUDI KASUS IBU POPON” pada jurnal INFOTECH Volume 5 Nomor 2 Desember Tahun 2019. Dalam penelitian ini, metode *Extreme Programming* digunakan untuk mengembangkan sistem untuk aplikasi *e-commerce* Toko Oleh-oleh Khas Majalengka Ibu Popon. Dalam aplikasi *e-commerce* Ibu Popon, bagian *One Time Payment* (OTP) masih dikirim melalui SMS *gateway* dan hanya di bagian registrasi, sehingga perlu ditambahkan fitur lain untuk melindungi data keamanan konsumen, seperti verifikasi melalui email dan penambahan OTP saat melakukan pembayaran untuk melindungi keamanan konsumen. Dengan kode keamanan OTP toko yang menggunakan email ini, penelitian ini akan mengembangkan bagian ini. (Nurdiana & Mardiana, 2019).
5. Jurnal penelitian yang disusun oleh Irmawati Carolina dan Arief Rusman dengan judul “PENERAPAN EXTREME PROGRAMMING PADA SISTEM INFORMASI PENJUALAN PAKAIAN BERBASIS WEB (STUDI KASUS TOKO ST JAYA)” pada jurnal INOVTEK POLBENG - SERI INFORMATIKA, vol. 4, no. 2, 2019. Studi ini mempelajari bagaimana membuat aplikasi *e-commerce* untuk Toko ST Jaya menggunakan metode pengembangan sistem *Extreme Programming*. Penelitian ini dilakukan untuk mengatasi masalah yang sering timbul selama proses penjualan, seperti pencatatan penjualan yang berulang dan ketidaksesuaian jumlah stok pakaian. Tujuan dari penelitian ini adalah mengembangkan sistem informasi berbasis web

yang dapat mengelola data pakaian, termasuk proses penjualan dan pembuatan laporan. (Carolina & Rusman, 2019).

