

*Seminar I*

**RANCANG BANGUN SISTEM INFORMASI B2C *E-COMMERCE*  
BERBASIS WEB (STUDI KASUS: PT. ASTRINDO SENAYASA)**



**Oleh:**

**BAGAS PRASETYO**

**H13116308**

**Pembimbing Utama : Dr. Hendra S.Si., M.Kom.**  
**Pembimbing Pertama : Muhammad Sadno, S.Si., M.Si.**  
**Penguji : 1. Dr. Muhammad Hasbi, M.Sc**  
**2. Jeriko Gormantara, S.Si., M.Si.**

**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN**

**2023**

## DAFTAR ISI

<b>DAFTAR ISI</b> .....	ii
<b>DAFTAR GAMBAR</b> .....	iv
<b>DAFTAR TABEL</b> .....	v
<b>BAB I PENDAHULUAN</b> .....	6
1.1    Latar Belakang .....	6
1.2    Rumusan Masalah .....	8
1.3    Batasan Masalah.....	8
1.4    Tujuan Penelitian.....	8
<b>BAB II TINJAUAN PUSTAKA</b> .....	9
2.1    Konsep Dasar Sistem.....	9
2.1.1    Pengertian Sistem.....	9
2.1.2    Karakteristik Sistem .....	10
2.2    Sistem Informasi.....	11
2.2.1    Pengertian Sistem Informasi .....	11
2.2.2    Komponen Sistem Informasi .....	12
2.2.3    Jenis-jenis Sistem Informasi .....	14
2.3 <i>E-commerce</i> .....	15
2.3.1    Pengertian <i>e-commerce</i> .....	15
2.3.2    Jenis-jenis <i>e-commerce</i> .....	16
2.4 <i>Javascript</i> .....	17
2.5 <i>React JS</i> .....	18
2.6 <i>Redux</i> .....	19
2.7 <i>Next.js</i> .....	20
2.8    Basis Data.....	22
2.9 <i>MongoDB</i> .....	24

2.10	Metode <i>Waterfall</i> .....	26
2.11	UML ( <i>Unified Modeling Language</i> ) Diagram.....	27
2.11.1	<i>Use Case</i> Diagram.....	28
2.11.2	<i>Activity</i> Diagram.....	29
2.12	<i>MongoDB Schema</i> Diagram.....	29
2.13	<i>Black Box Testing</i> .....	30
<b>BAB III METODE PENELITIAN</b> .....		32
3.1	Waktu dan Lokasi Penelitian.....	32
3.2	Metode Pengumpulan Data .....	32
3.3	Tahapan Penelitian .....	33
3.4	Metode Pengembangan Sistem .....	34
3.4.1	Analisis Kebutuhan .....	35
3.4.2	Perancangan Sistem .....	35
3.4.3	Implementasi .....	35
3.4.4	Pengujian Sistem.....	35
3.4.5	Pemeliharaan Sistem .....	36
3.5	Komponen Sistem .....	36
3.6	Perancangan Sistem.....	36
3.7	Rancangan <i>User Interface</i> .....	37
<b>Daftar Pustaka</b> .....		44

## DAFTAR GAMBAR

Gambar 2.1 Alur Data <i>React JS</i> .....	19
Gambar 2.2 Alur Data <i>Redux</i> .....	20
Gambar 2.3 <i>Server Side Rendering</i> .....	21
Gambar 2.4 <i>Static Site Generator</i> .....	22
Gambar 2.5 Struktur BSON .....	24
Gambar 2.6 Simbol-Simbol <i>Use Case Diagram</i> .....	28
Gambar 2.7 Simbol-Simbol <i>Activity Diagram</i> .....	29
Gambar 3.1 Tahapan Penelitian .....	34
Gambar 3.2 <i>Use Case Diagram</i> .....	37
Gambar 3.3 Halaman <i>Home</i> .....	38
Gambar 3.4 Halaman <i>Login/Sign Up</i> .....	39
Gambar 3.5 Halaman Detail Produk .....	40
Gambar 3.6 Halaman Keranjang.....	41
Gambar 3.7 Halaman <i>Checkout</i> .....	41
Gambar 3.8 Halaman Pembayaran.....	42
Gambar 3.9 Halaman Admin .....	43

## DAFTAR TABEL

Tabel 2.1 Jenis Sistem Informasi .....	15
Tabel 3.1 Jadwal Kegiatan Penelitian .....	32

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Di era ini, kemajuan teknologi menjadi fokus utama di seluruh dunia. (Lathifah & Suaidah, 2020). Karena perkembangan teknologi informasi dan komputer semakin maju, manusia semakin membutuhkan informasi yang lebih luas. Salah satu cara untuk memperoleh informasi tersebut adalah dengan memanfaatkan akses internet. Dengan adanya internet, orang-orang dapat menghemat tempat dan waktu yang digunakan untuk mencari informasi yang dibutuhkan. Dengan demikian, aktivitas dan rutinitas sehari-hari untuk mencari informasi yang diperlukan tidak perlu ditinggalkan karena semuanya dapat diakses secara *online*. Peningkatan penggunaan internet oleh masyarakat terlihat dari peningkatan jumlah pelanggan dan pengusaha yang memanfaatkannya, mendorong munculnya suatu tuntutan pelayanan internet melebihi dari apa yang bisa diperoleh di dunia nyata. Ini meliputi kesempatan untuk menjual barang-barang ataupun produk-produk secara daring. Salah satu hal terpenting dalam bisnis melalui internet adalah bagaimana keuntungan dapat diperoleh secara aman dan mudah (Mukaromah & Rosadi, 2015). Perkembangan internet yang pesat dalam 10 tahun terakhir telah secara signifikan mengubah perilaku pengusaha dalam memasarkan produk dan layanan mereka. Teknologi informasi memudahkan produsen untuk memasarkan dan menawarkan produk mereka. Teknologi pada suatu organisasi, perusahaan, dan pemerintahan era sekarang sangat memiliki peran yang begitu besar (Lathifah dkk., 2021).

*E-commerce* merupakan aktivitas jual beli yang dilakukan secara daring (*online*) dengan memanfaatkan dukungan dari teknologi informasi yang diakses melalui *website* maupun perangkat bergerak dengan media telekomunikasi berupa jaringan internet (Zulkarnain, 2019). Pemanfaatan teknologi *e-commerce* dapat dirasakan oleh konsumen, baik konsumen *bisnis-to-consumer* (B2C) maupun konsumen *business-to-business* (B2B). Salah satu faktor yang menyebabkan bisnis saat ini menggunakan *e-commerce* adalah meningkatkan efisiensi dan efektivitas bisnis, dikatakan meningkatkan efisiensi karena dapat meminimalisir biaya

pemasaran, tenaga kerja, maupun biaya operasional. Selain itu dukungan internet pada *e-commerce* meningkatkan efektivitas dengan memungkinkan penjangkauan konsumen yang lebih luas dan lebih cepat. Hal tersebut dimungkinkan karena bisnis dapat membuka toko *virtual* selama 24 jam tanpa jeda dengan menampilkan informasi tentang produk dan prosedur pembelian secara daring pada website milik unit bisnis tersebut. Konsumen juga bisa mendapatkan manfaat lain dari segi finansial karena dapat mengurangi biaya akomodasi. Selain itu, informasi detail tentang produk dan presentasi visual yang bagus, bahkan bisa disertai dengan animasi dan video, akan memudahkan konsumen dalam memilih dan membeli produk tersebut (Atun Mumtahana dkk., 2017).

PT. Astrindo Senayasa awalnya merupakan perusahaan distributor ASUS Indonesia untuk komponen produk. Saat ini PT. Astrindo Senayasa memiliki menjadi distributor berbagai merek ternama seperti ASUS, Intel, Asrock, Sennheiser, Qnap, Sapphire, Sandisk, WD dan Asustor. PT. Astrindo Senayasa memiliki cabang di berbagai kota di Indonesia dengan pusatnya berada di Ibukota Jakarta. Untuk pembelian produk, pengusaha ritel harus mendaftar menjadi *member* untuk dapat mengakses layanan PT. Astrindo Senayasa. Sampai saat ini PT. Astrindo Senayasa hanya melakukan transaksi hanya kepada pengusaha ritel dan tidak menawarkan produknya kepada pengguna langsung (*End User*).

Seiring berkembangnya perusahaan dalam rangka memperluas pangsa pasarnya, PT. Astrindo Senayasa berkeinginan untuk memasuki pasar pengguna langsung (*End User*). Meskipun demikian untuk membuka toko atau gerai fisik di berbagai kota tentunya membutuhkan waktu dan biaya yang besar serta menimbulkan risiko baru bagi perusahaan yang biasanya menjual pada pasar pengusaha retail dan belum mengetahui pasar dari pengguna langsung (*End User*). Sehingga diperlukan suatu sarana atau layanan penjualan berbasis *e-commerce* yang dapat membantu meminimalisir waktu, biaya, dan risiko yang dihadapi. Ini menjadi faktor pendorong bagi penulis untuk merancang dan membangun sistem informasi *e-commerce* berbasis web untuk skripsi tugas akhir, dengan judul **“RANCANG BANGUN SISTEM INFORMASI B2C E-COMMERCE BERBASIS WEB (STUDI KASUS: PT. ASTRINDO SENAYASA)”**. Dengan dibuatnya aplikasi ini diharapkan dapat memenuhi kebutuhan dan bisa

memudahkan perusahaan dalam menjalankan kegiatan bisnis memasarkan produknya kepada pengguna langsung (*End User*).

## **1.2 Rumusan Masalah**

Adapun rumusan masalah dalam penelitian ini adalah :

1. Bagaimana cara merancang sistem informasi *e-commerce* berbasis web pada PT. Astrindo Senayasa?
2. Bagaimana sistem informasi *e-commerce* berbasis web dapat membantu PT. Astrindo Senayasa memasarkan produknya pada pengguna langsung (*End User*)?

## **1.3 Batasan Masalah**

Batasan masalah dilakukan agar penelitian ini dapat memberikan pemahaman yang terarah dan sesuai dengan yang diharapkan. Agar pembahasan tidak menyimpang dari pokok perumusan masalah yang ada. Adapun batasan masalah pada penelitian ini yaitu:

1. Website dirancang dan dibangun menggunakan bahasa pemrograman *javascript* dengan *framework Next.js*, serta pengolahan database menggunakan *MongoDB*.
2. Jenis *e-commerce* yang dibuat adalah sistem komunikasi bisnis antar pelaku bisnis dengan konsumen (*business to consumers*).

## **1.4 Tujuan Penelitian**

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Untuk merancang dan membangun sistem informasi *e-commerce* berbasis web agar PT. Astrindo Senayasa dapat memasarkan produknya kepada pengguna langsung (*End User*).
2. Dapat membantu PT. Astrindo Senayasa memasarkan produknya kepada pengguna langsung (*End User*).



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Konsep Dasar Sistem**

##### **2.1.1 Pengertian Sistem**

Sistem menurut Kamus Besar Bahasa Indonesia (KBBI) merupakan perangkat unsur yang berkaitan untuk membangun sebuah totalitas. Di lihat dari definisi tersebut, sistem merupakan gabungan dari beberapa unsur yang berhubungan untuk mencapai suatu tujuan. Menurut Hutahaeen (2015) sistem merupakan jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan dengan sasaran tertentu. Menurut Winarno (2006) sistem merupakan sekumpulan komponen yang saling bekerja sama untuk mencapai suatu tujuan. Meskipun setiap komponen memiliki fungsi yang berbeda-beda, namun komponen-komponen tersebut saling bekerja sama dan saling bergantung satu sama lain. Menurut Sutabri (2012) secara sederhana, sistem dapat diartikan sebagai kumpulan unsur, komponen, atau variabel yang terorganisasi saling bekerja sama, saling bergantung satu sama lain dan terpadu. Definisi sistem dapat dirinci lebih lanjut dari pengertian umum sistem itu sendiri, yaitu:

- a. Setiap sistem terdiri dari unsur-unsur yang terdiri dari subsistem yang lebih kecil, di mana unsur-unsur tersebut membentuk subsistem tersebut.
- b. Unsur-unsur suatu sistem berhubungan saling terkait sangat erat dan bekerja sama untuk membentuk kesatuan sistem.
- c. Unsur-unsur sistem yang bekerja sama untuk mencapai tujuan yang ditetapkan. Setiap sistem memiliki tujuan khusus yang ingin dicapai.
- d. Sebuah sistem adalah bagian dari sistem yang lebih besar lagi.

Dari pengertian-pengertian tersebut maka sistem secara garis besar merupakan suatu kesatuan dari unsur-unsur atau komponen-komponen yang memiliki fungsi berbeda-beda namun saling terkait dan bekerja sama untuk mencapai suatu tujuan.

### 2.1.2 Karakteristik Sistem

Setiap sistem memiliki sifat atau karakteristik tertentu. Menurut Sutabri (2012) karakteristik sistem meliputi:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari Komponen-komponen yang bekerja sama untuk mencapai tujuan yang sama dan membentuk kesatuan sistem. Komponen-komponen tersebut disebut juga sebagai subsistem. Setiap subsistem memiliki fungsi yang berbeda-beda.

b. Batasan Sistem (*Boundary*)

Batasan yang telah ditetapkan memisahkan satu sistem dari sistem lainnya dan juga memisahkan sistem tersebut dari lingkungan sekitarnya.

c. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar juga berpengaruh terhadap kinerja suatu sistem. Sistem akan mendapatkan manfaat dari lingkungan yang kondusif, sementara lingkungan yang tidak mendukung harus dikendalikan.

d. Penghubung Sistem (*Interface*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

e. Masukan Sistem (*Input*)

Penghubung adalah media yang menghubungkan antara sistem dengan subsistem. Fungsi dari penghubung adalah menyalurkan sumber daya dari satu subsistem ke subsistem lainnya untuk mengintegrasikan sistem secara keseluruhan.

f. Keluaran Sistem (*Output*)

Energi diolah menjadi hasil atau *output* yang berguna. *Output* tersebut dapat menjadi masukan atau *input* untuk subsistem lainnya.

g. Pengolah Sistem (*Process*)

Suatu sistem memiliki proses yang mengubah *input* menjadi *output*.

h. Sasaran Sistem (*Objective*)

Setiap sistem harus memiliki sasaran atau tujuan yang ingin dicapai. Jika sistem tidak memiliki sasaran atau tujuan, maka sistem tersebut tidak

memiliki nilai. Keberhasilan sistem dapat diukur dari kemampuannya untuk mencapai tujuannya.

## **2.2 Sistem Informasi**

### **2.2.1 Pengertian Sistem Informasi**

Husein dan Wibowo (2006) menyatakan sistem informasi adalah seperangkat komponen yang saling berhubungan yang berfungsi untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi. Sistem informasi mencakup informasi tentang orang, tempat, dan sesuatu di dalam organisasi atau lingkungan sekitarnya. Laudon dan Laudon (2017) mendefinisikan sistem informasi sebagai serangkaian komponen teknis yang saling terhubung, yang berfungsi untuk mengumpulkan, menyimpan, memproses, dan mendistribusikan informasi guna mendukung pengambilan keputusan dan pengawasan di sebuah organisasi. Sistem informasi juga membantu manajer dan karyawan dalam menganalisis masalah, memvisualisasikan hal-hal yang rumit, dan menciptakan produk atau inovasi baru. Sistem informasi mengandung informasi penting tentang orang, tempat/lokasi, dan elemen penting lainnya yang terkait dengan organisasi dan lingkungan di sekitarnya. Dari pengertian-pengertian tersebut maka secara garis besar sistem informasi merupakan serangkaian unsur-unsur atau komponen-komponen yang saling berhubungan dan memiliki tugas yaitu mengumpulkan, menyimpan, memproses, dan mendistribusikan suatu informasi yang nantinya dapat digunakan sebagai bahan landasan bagi pengambilan keputusan.

Laudon dan Laudon (2017) mendefinisikan informasi sebagai data yang telah diolah menjadi bentuk yang dapat dimengerti dan bermanfaat bagi manusia, sementara data adalah fakta mentah yang belum diolah menjadi informasi yang dapat dipahami oleh manusia. Sutabri (2016) menyatakan bahwa informasi dikatakan berkualitas jika informasi tersebut memenuhi kualifikasi sebagai berikut:

#### **a. Akurat**

Informasi harus akurat, jelas, dan bebas dari bias karena kesalahan dan bias dapat mengganggu pemahaman informasi. Terdapat risiko kesalahan dalam

proses pengiriman informasi dari sumber hingga penerima, sehingga pengawasan dan perbaikan perlu dilakukan.

b. Tepat waktu

Informasi harus disampaikan tepat waktu karena jika tidak, nilai dari informasi tersebut hilang. Informasi sangat penting dalam pengambilan keputusan dan keterlambatan dapat menyebabkan masalah dan mempengaruhi kinerja organisasi secara negatif.

c. Relevan

Informasi memiliki relevansi yang berbeda-beda bagi setiap individu dan divisi dalam suatu organisasi sehingga informasi harus disesuaikan dengan penerima yang tepat. Sebagai contoh, informasi tentang kerusakan mesin akan lebih relevan bagi staf teknik daripada staf administrasi.

Laudon dan Laudon (2017) berpendapat bahwa terdapat tiga kegiatan sistem informasi yang diperlukan oleh suatu perusahaan dalam mengambil keputusan, mengawasi kegiatan operasional, menganalisis permasalahan, serta menciptakan produk atau inovasi baru. Kegiatan tersebut yakni:

- a. *Input* yaitu pengumpulan data yang diperlukan.
- b. *Process* yaitu memproses data yang telah dikumpulkan menjadi informasi yang dapat dimengerti manusia.
- c. *Output* yaitu penyaluran atau penggunaan informasi yang telah diperoleh dalam pengambilan keputusan. Umpan balik bersifat penting dalam suatu sistem agar dapat digunakan untuk perbaikan *input*.

### 2.2.2 Komponen Sistem Informasi

O'Brien dan Marakas (2013) menjelaskan bahwa sistem informasi terdiri dari komponen-komponen seperti masukan (*input*) yang berasal dari berbagai sumber data, proses pengolahan data menjadi informasi, hasil (*output*) yang berupa produk-produk informasi, serta sistem pengontrolan (*control system*) dan penyimpanan data. Sistem informasi tersebut didukung oleh 5 pilar, yaitu:

- a. *Hardware Resources* (Perangkat Keras)
- b. *Software Resources* (Perangkat Lunak)

- c. *People Resource* (Pengguna)
- d. *Data Resource* (Sumber Data)
- e. *Network Resource* (Sumber Jaringan)

Berdasarkan Hutahaean (2015), sistem informasi terdiri dari blok bangunan atau komponen-komponen, yang meliputi:

- a. Blok Masukkan (*Input Block*)  
*Input* adalah metode dan media dalam pengumpulan data yang dapat berupa dokumen dasar.
- b. Blok Model (*Model Block*)  
Blok Model adalah kombinasi antara metode matematis, logika, dan prosedur yang digunakan untuk mengolah data menjadi *output* yang diharapkan.
- c. Blok Keluaran (*Output Block*)  
Blok Keluaran merupakan informasi dari produk dalam sistem informasi, yang digunakan untuk semua tingkatan manajemen dan semua pengguna sistem.
- d. Teknologi (*Technology*)  
Teknologi digunakan untuk menerima masukan, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, serta membantu pengendalian secara keseluruhan. Teknologi terdiri dari perangkat lunak, perangkat keras, serta teknisi.
- e. Blok Basis Data (*Database Block*)  
Sebuah rangkaian data yang saling terhubung satu sama lain dan disimpan dalam perangkat keras.
- f. Blok Kendali (*Control Block*)  
Pengendalian bertujuan untuk menjaga agar sistem tetap berjalan dengan baik dan mencapai tujuan yang diinginkan. Beberapa hal yang dikendalikan meliputi potensi kerusakan, sabotase, kecurangan, ketidaknormalan, dan lain sebagainya.

### 2.2.3 Jenis-jenis Sistem Informasi

Indrayani dan Humdiana (2009) mengelompokkan sistem informasi ke dalam beberapa jenis yang terkait dengan tingkat manajerial dalam suatu organisasi, antara lain:

a. *Transaction Processing System (TPS)*

*Transaction Processing System (TPS)* adalah sistem informasi yang digunakan untuk merekam data mengenai transaksi bisnis, seperti penjualan produk dengan mencatat informasi seperti harga, jumlah produk yang dijual, dan waktu terjadinya transaksi tersebut. Dengan menggunakan informasi ini, perusahaan dapat menganalisis produk yang paling laris dan produk yang kurang diminati oleh pelanggan, sehingga dapat mengontrol pasokan produk-produk tersebut dalam jumlah yang tepat.

b. *Management Information System (MIS)*

*Management Information System (MIS)* adalah sistem informasi yang mengakomodasi kebutuhan manajemen di suatu organisasi. *Management Information System (MIS)* memberikan laporan kepada manajemen dan memberikan akses *online* untuk informasi kinerja organisasi dan rekaman historis. *Management Information System (MIS)* bertanggung jawab dalam merencanakan, mengendalikan, dan membuat keputusan pada tingkat manajemen. Sistem ini memberikan laporan secara berkala seperti harian, mingguan, bulanan, dan tahunan.

c. *Decision Support System (DSS)*

*Decision Support System (DSS)* adalah sebuah sistem komputer yang digunakan pada level manajemen dalam sebuah organisasi untuk melakukan analisis data secara mendalam dengan menggunakan model berbentuk grafik. Sistem ini sangat fleksibel untuk penggunaannya. Tujuan utama dari *Decision Support System (DSS)* adalah untuk membantu pengambilan keputusan yang bersifat semi-struktur dan tidak terstruktur. DSS terdiri dari tiga komponen dasar, yaitu subsistem dialog, basis data, dan model yang memungkinkan pengguna untuk menjelajahi setiap konsekuensi dengan berinteraksi secara bebas.

d. *Executive Information System (EIS)*

*Executive Information System (EIS)* adalah sebuah sistem yang menyediakan akses mudah dan langsung ke laporan manajemen mengenai kinerja seluruh perusahaan. *Executive Information System (EIS)* juga dapat terhubung dengan layanan informasi *online* dan *email*. Sistem ini dirancang untuk menghemat waktu pengguna dalam memperoleh informasi, yang merupakan bagian penting dalam pengambilan keputusan. Dalam penggunaannya, *Executive Information System (EIS)* memberikan kemudahan akses ke informasi strategis dan penting bagi level eksekutif dalam suatu organisasi.

Tabel 2.1 Jenis Sistem Informasi

Jenis Informasi	Input	Process	Output
TPS	Transaksi	Penyortiran, pengombinasian, pembaruan	Laporan rinci
MIS	Rangkuman data	Laporan rutin, model-model sederhana, Analisis	Laporan akhir berupa informasi
DSS	Data dalam jumlah kecil	Analisis, simulasi, interaktif	Laporan khusus
EIS	Data eksternal dan internal	Grafik, simulasi, interaktif	Proyeksi jawaban pertanyaan

## 2.3 *E-commerce*

### 2.3.1 Pengertian *e-commerce*

Kotler & Amstrong (2012) mendefinisikan *e-commerce* sebagai saluran *online* yang dapat diakses melalui komputer dan digunakan oleh pelaku bisnis untuk melakukan aktivitas bisnisnya, serta oleh konsumen untuk mendapatkan informasi. Pada prosesnya, saluran ini memberikan layanan informasi kepada konsumen untuk

membantu mereka dalam memilih. Menurut Wong (2010) *e-commerce* adalah suatu bentuk aktivitas bisnis yang melibatkan proses jual beli dan pemasaran barang dan jasa melalui sistem elektronik, yang mencakup media seperti radio, televisi, serta jaringan komputer atau internet.

Maka dapat disimpulkan bahwa *e-commerce* merupakan sebuah sistem yang dinamis yang menggabungkan teknologi, aplikasi, dan proses bisnis untuk menghubungkan perusahaan, konsumen, dan komunitas tertentu. Sistem ini memungkinkan terjadinya pertukaran barang antara pengecer dan konsumen yang melibatkan berbagai jenis komoditas dalam skala yang luas. Selain itu, transaksi dilakukan secara elektronik, dan pengiriman barang dari pengecer ke tangan konsumen dapat melalui transportasi dari satu wilayah ke wilayah lain. Dalam sistem ini terdapat hubungan yang saling menguntungkan bagi kedua belah pihak.

### **2.3.2 Jenis-jenis *e-commerce***

Berikut ini terdapat empat jenis *e-commerce* berdasarkan karakteristiknya menurut Kotler (2012) :

1. *Business to business* (B2B)
  - a. Mitra bisnis yang sudah saling mengenal dan sudah menjalin hubungan bisnis yang lama.
  - b. Pertukaran data yang sudah berlangsung berulang dan telah disepakati bersama.
  - c. Model yang umum digunakan adalah *peer to peer*, dimana *processing intelligence* dapat didistribusi oleh kedua pelaku bisnis.
2. *Business to consumer* (B2C)
  - a. Terbuka untuk publik dan informasi dapat disebarluaskan secara bebas kepada masyarakat.
  - b. Layanan yang digunakan juga untuk umum sehingga dapat digunakan oleh banyak orang.
  - c. Permintaan konsumen menjadi faktor penentu layanan yang ditawarkan, sehingga produsen harus dapat merespon permintaan tersebut dengan baik.
  - d. Sistem pendekatan adalah *client-server*.



### 3. *Consumer to Consumer (C2C)*

*Consumer to Consumer (C2C)* adalah model bisnis yang melibatkan transaksi jual-beli secara *online*, dimana website tersebut tidak hanya berperan sebagai platform promosi tetapi juga memberikan fasilitas transaksi uang secara *online*. Dalam konteks ini, terdapat 2 (dua) indikator utama yang menentukan keberhasilan sebuah website *marketplace*:

- a. Semua transaksi secara *online* harus difasilitasi oleh situs web yang terkait.
- b. Bisa digunakan oleh penjual individual.

Dimana kegiatan yang berlangsung harus menggunakan fasilitas transaksi *online* harus digunakan, misalnya dengan menggunakan rekening pihak ketiga untuk memastikan keamanan transaksi. Uang pembayaran hanya akan diterima oleh penjual setelah barang sampai ke tangan pembeli. Penjual tidak dapat menarik uang hasil penjualan sebelum barang diterima oleh pembeli. Jika produk gagal sampai ke tangan pembeli, uang yang telah dibayarkan akan dikembalikan kepada pembeli.

### 4. *Consumer to Business (C2B)*

Dalam *consumer to business (C2B)*, peran antara konsumen dan perusahaan dibalik jika dibandingkan dengan *business to consumer (B2C)*. Pada model bisnis C2B, konsumen bertindak sebagai pencipta nilai yang menawarkan produk atau jasa kepada perusahaan secara elektronik.

## 2.4 *Javascript*

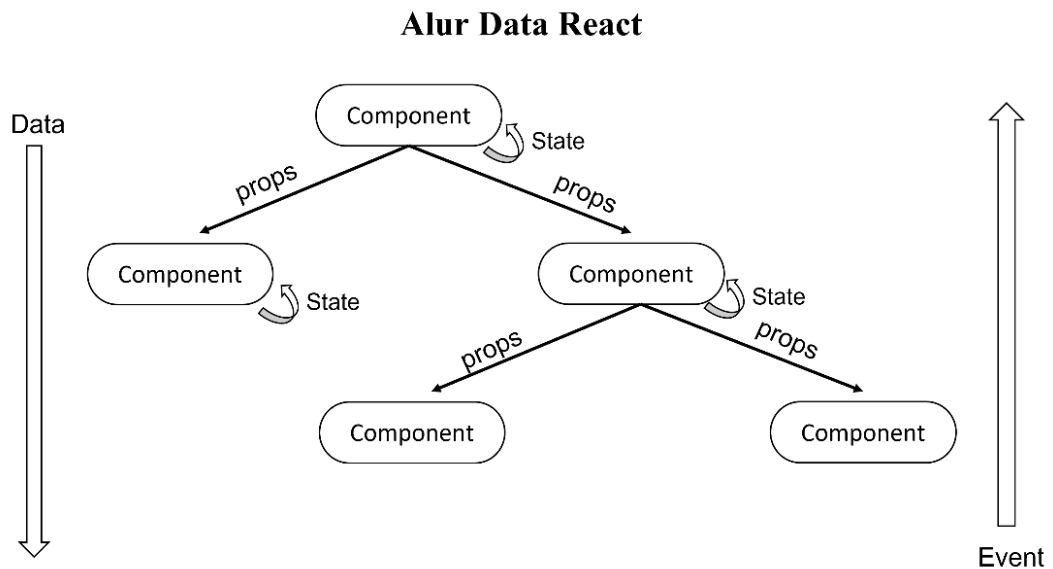
*Javascript* sering digunakan di dalam aplikasi berbasis *web* untuk memberikan pesan menarik kepada pengguna dan tampilan pendukung yang ada pada website. Menurut Wahana Komputer (2010) *JavaScript* merupakan bahasa yang terdiri dari beberapa skrip yang berperan dalam meningkatkan tampilan interaktif dari sebuah dokumen web. *JavaScript* dikembangkan oleh *Netscape* dengan nama awal *LiveScript* yang berfokus pada proses pengolahan data di sisi *client* dan menyajikan komponen web yang lebih interaktif serta berfungsi untuk menambah fungsionalitas dan kenyamanan halaman web.

*Javascript* memungkinkan pembuatan halaman web yang interaktif dan pintar. Contohnya adalah kemampuan *javascript* untuk memeriksa validitas masukan pengguna sebelum masukan dikirim ke *server*, melakukan operasi aritmatika seperti penjumlahan, pengurangan, dan perkalian, serta menampilkan animasi sederhana. Berbeda dengan bahasa berorientasi *object*, *Javascript* adalah bahasa berbasis *object* yang hanya mendukung satu dari tiga konsep dasar yaitu pengkapsulan, dan bahkan pengkapsulan tersebut tidak sepenuhnya didukung.

## 2.5 *React JS*

*React JS* adalah *library JavaScript* yang digunakan untuk mengembangkan komponen antarmuka pengguna (UI) yang dapat digunakan kembali. *React JS* digunakan untuk membangun sebuah website yang dinamis dan tanpa dilakukan *refresh* untuk memuat data selanjutnya *React* memanfaatkan konsep *Document Object Model* (DOM) untuk membangun website yang memiliki performa tinggi, cepat, dan mudah dalam pengembangan. Dalam penggunaannya, *React* mampu menciptakan tampilan website yang lebih sederhana dan interaktif sehingga memberikan pengalaman pengguna yang lebih baik.

Di sisi server, *React* di-render menggunakan *NodeJS*. Komponen *React JS* memiliki seperangkat metode dan properti bawaan yang akan bisa digunakan. Selain itu terdapat *props* dan *state* sebagai konsep penting. *Props* merujuk pada data di dalam komponen yang diberikan dari luar. Sementara itu, *state* serupa dengan *props*, namun hanya tersedia di dalam komponen. Keduanya adalah konsep penting dalam membangun aplikasi *React* yang fleksibel dan efektif. Alur data atau *Data Flow* dari *React* ditujukan pada gambar 2.1.



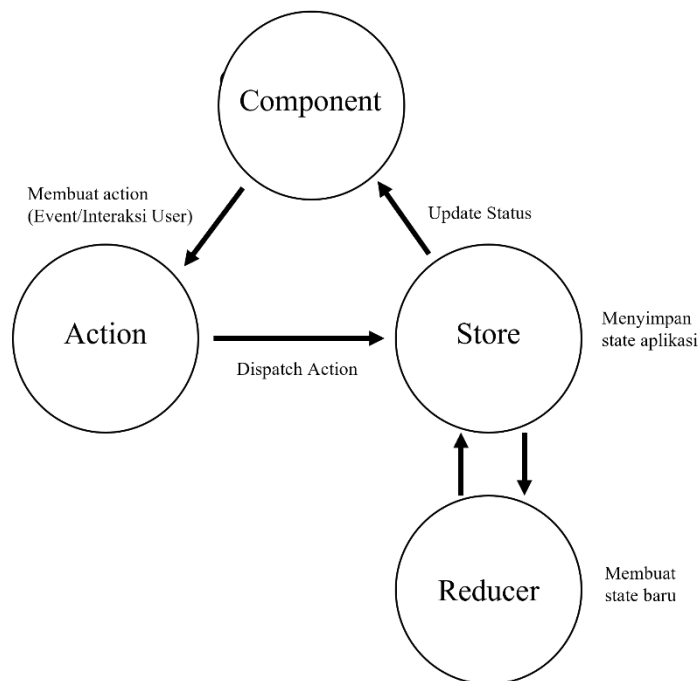
Gambar 2.1 Alur Data *React JS*

## 2.6 *Redux*

*Redux* merupakan *state management* dalam aplikasi yang dibuat pada tahun 2015 oleh Dan Abramov dan Andrew Clark. *State management* ini berfungsi untuk memudahkan komunikasi dan pertukaran data antar komponen dalam sebuah aplikasi. *State management* menciptakan struktur data yang nyata untuk mewakili keadaan aplikasi yang dapat digunakan untuk *read* dan *write*.

Sebagian besar *library*, seperti *React JS* dibangun agar komponen dapat mengelola *state* nya secara internal tanpa memerlukan pustaka atau alat eksternal. Hal ini pun berfungsi dengan baik untuk aplikasi dengan sedikit komponen tapi ketika aplikasi semakin berkembang menjadi lebih besar, mengelola *state* yang digunakan di seluruh komponen menjadi semakin sulit. Itulah menjadi alasan dibutuhkan *redux* yang berfungsi untuk melakukan perubahan *state* yang dibutuhkan oleh setiap fungsional yang ada di suatu aplikasi. Alur data atau *Data Flow* dari *Redux* ditujukan pada gambar 2.2.

## Alur Data Redux



Gambar 2.2 Alur Data *Redux*

### 2.7 *Next.js*

*Next.js* adalah suatu kerangka kerja *React* yang dapat dipakai dalam membuat aplikasi berbasis *JavaScript*. Kerangka kerja ini memungkinkan pengembangan situs jejaring yang mudah digunakan dan dikembangkan dengan cepat. Penggunaan *Next.js* memerlukan *library JavaScript React* untuk membangun antarmuka pengguna yang berinteraksi. Dengan menggunakan kerangka kerja ini, konfigurasi yang dibutuhkan untuk *React* dapat diatur, serta menyediakan struktur, fitur, dan optimisasi pada aplikasi web. Beberapa perusahaan besar seperti Nike, Netflix, dan Playstation telah memanfaatkan *Next.js* dalam pengembangan aplikasi mereka. *Next.js* dibuat karena masalah *pre-render* secara statis di beberapa halaman yang membuat *Search Engine Optimization* (SEO) kurang optimal dikarenakan *file JavaScript* harus di *load* dan menentukan komponen apa yang harus ditampilkan.

*Next.js* memiliki cara kerja dengan menggunakan serangkaian tahapan proses, dimulai dengan *me-render* situs jejaring pada server untuk mempermudah mesin pencari dalam menemukan halaman yang diminta. Setelah itu, server akan

memberikan respon dan menampilkan halaman situs jejaring dalam format HTML (*Hypertext Markup Language*). Secara khusus, *Next.js* memiliki kemampuan yang mencakup *JavaScript*, *HTML*, dan juga *React*. Tiga fitur dalam *Next.js* ini memungkinkan mesin peramban untuk tidak perlu memanipulasi DOM (*Document Object Model*) pada *React* dan *JavaScript*. Hal ini berbeda dengan penggunaan platform lain, di mana mesin peramban harus melakukan manipulasi DOM terhadap *React* dan *JavaScript* agar dapat menampilkan HTML.

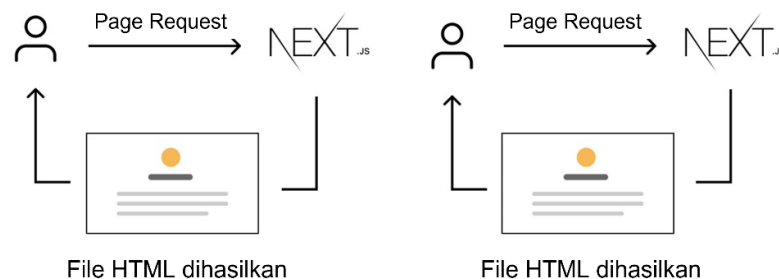
Proses yang digunakan oleh *Next.js* ini disebut *pre-rendering*. *Next.js* memiliki dua jenis proses *rendering*, yaitu:

1. *Server Side Rendering* (SSR)

*Server-side Rendering* (SSR) merupakan teknik *pre-rendering* di mana server mengonversi *package React* dan *JavaScript* menjadi HTML setiap kali browser meminta halaman tersebut seperti yang terlihat pada gambar 2.3.

### Server-side Rendering

File HTML dihasilkan setiap **request**



Gambar 2.3 *Server Side Rendering*

(Sumber: <https://www.niagahoster.co.id>)

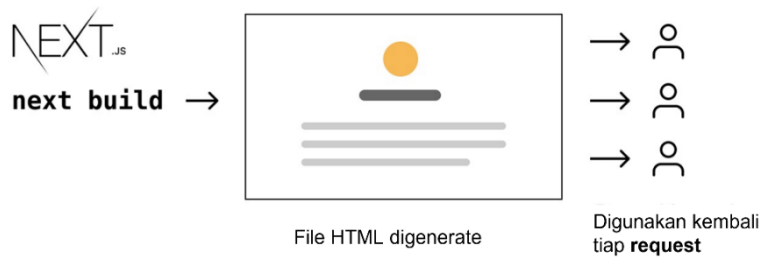
2. *Static Site Generator* (SSG)

*Static Site Generator* (SSG) merupakan proses kompilasi dan *rendering* situs web atau aplikasi pada saat pembuatannya. Dalam *Next.js*, *output* dari proses SSG ini adalah kumpulan dokumen statis, termasuk dokumen HTML itu sendiri. SSG mencakup dua aset, yaitu *JavaScript* dan *CSS*. SSG biasanya lebih cepat dalam memuat halaman pada mesin peramban. Metode

ini sering digunakan dalam pembuatan *landing page*. Metode *Next.js* yang satu ini umumnya menghasilkan HTML pada waktu pembuatan dan kemudian HTML *pre-rendering* ini digunakan kembali pada tiap permintaan seperti yang terlihat pada gambar 2.4.

### Static Generation

File HTML digenerate saat **build-time** dan digunakan kembali tiap **request**



Gambar 2.4 *Static Site Generator*

(Sumber: <https://www.niagahoster.co.id>)

*Next.js* mempunyai beberapa keunggulan seperti *automatic code splitting* yang digunakan sebagai pemecah *code* agar lebih cepat saat me-load halaman, memiliki konsep SSR (*Server Side Rendering*) yang akan membantu untuk memudahkan dalam menggunakan SEO. *Next.js* dilengkapi dengan direktori "*pages*" yang memudahkan dalam proses *routing* dan bisa disesuaikan dengan kebutuhan.

## 2.8 Basis Data

Sistem yang terkomputerisasi memerlukan basis data untuk menjaga data yang telah diproses dan menyediakan informasi saat dibutuhkan. Sukanto dan Shalahuddin (2015) mengemukakan bahwa sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang telah diproses atau informasi, dan menyediakan informasi tersebut saat dibutuhkan. Sedangkan menurut Lubis (2016), basis data adalah kumpulan *file* data yang dibentuk dengan hubungan atau relasi yang logis dan dapat dijelaskan dengan catatan, serta bersifat mandiri. Oleh karena itu, basis data adalah kumpulan data atau informasi yang telah diproses, disimpan, dan dapat digunakan kembali saat dibutuhkan.

Basis data dimaksudkan untuk mengatasi masalah pada sistem yang memakai pendekatan berbasis berkas atau *file* (Kadir, 2003). Basis data mengumpulkan *file* yang saling berhubungan. *Record* di dalam tiap *file* harus dapat dihubungkan dengan *record* di dalam *file* lain. Dalam manajemen *database relational* terdapat komponen utama dalam konsep *database* (Whitten dkk., 2007) yaitu:

1. *Field*, yaitu unit terkecil data yang disimpan dalam database, antara lain:
  - a. *Primary key*, yaitu *field* yang unik dan mengidentifikasi satu *record*.
  - b. *Secondary key*, yaitu *field* yang mengidentifikasi sebuah *record* atau bagian dari beberapa *record* yang terkait.
  - c. *Foreign key*, yaitu *field* yang menunjuk beberapa *record* pada *file* lain.
  - d. *Descriptive field*, yaitu *non-key field*.
2. *Record*, yaitu kumpulan *field* yang diatur dalam format yang *predetermined* (telah ditentukan).

- a. *Fixed-length record structures*

Sebagian besar teknologi *database* memaksakan struktur *record fixed length*, dalam artian setiap instance record mempunyai field yang sama, jumlah *field* yang sama, dan ukuran logika yang sama. Akan tetapi, beberapa sistem *database* akan mengompresi *field-field* dan nilai-nilai yang tidak digunakan untuk menghemat ruang penyimpanan *disk*.

- b. *Variable-length record structures*

Memperbolehkan *record-record* pada *file* yang sama memiliki *length* yang berbeda.

3. *File* dan Tabel

*File* adalah kumpulan semua kejadian dari struktur *record* yang ditentukan.

Tipe-tipe dari *file*, yaitu:

- a. *File induk/master* adalah *file* penting dalam sistem dan akan tetap ada selama siklus hidup sistem informasi berputar.
- b. *File transaksi* adalah *file* yang digunakan untuk merekam data dari suatu transaksi yang terjadi.
- c. *File laporan* adalah *file* yang berisi sistem informasi yang akan ditampilkan.

- d. *File* sejarah adalah *file* yang berisi data masa lalu yang sudah tidak aktif lagi.
- e. *File* pelindung adalah salinan dari *file-file* yang masih aktif di *database* pada saat tertentu yang digunakan bila *file database* rusak.
- f. *File* kerja adalah suatu proses program secara sementara karena memori komputer tidak mencukupi.

Sedangkan tabel adalah suatu kesatuan unit dari *row / record* dengan atribut-atributnya (*column*).

## 2.9 MongoDB

*MongoDB* adalah salah satu sistem basis data (*database*) NoSQL *open source* yang populer, di mana sistem ini menggunakan struktur data JSON untuk menyimpan data. *MongoDB* sering digunakan oleh perancang sistem untuk membangun aplikasi yang terhubung ke *cloud computing*, *grid computing*, atau *big data*. *MongoDB* yang merupakan suatu sistem *database* dengan istilah *open source* berbasis dokumen yang dapat digunakan oleh semua pengguna tanpa perlu lisensi. *MongoDB* awalnya dibuat menggunakan bahasa C++ dan telah dikembangkan oleh *10gen* sejak oktober 2007, dan *MongoDB* baru dipublikasikan pada februari 2009.

*MongoDB* tidak menggunakan sistem yang terdiri dari tabel, kolom, dan baris seperti halnya pada basis data *relasional*. Sebaliknya, *MongoDB* menggunakan koleksi dan dokumen untuk menyimpan dan mengorganisir data. Meskipun dokumen yang disimpan dalam satu koleksi, *MongoDB* memungkinkan dokumen tersebut memiliki atribut yang berbeda-beda. Model data *MongoDB* disebut BSON, yang memiliki struktur yang mirip dengan JSON. Memiliki struktur yang mudah dipahami dan dibaca seperti gambar 2.5 di bawah ini:

```
{
  "nama" : "bagas",
  "kontak" : {
    "alamat" : "Jalan Cendrawasih",
    "kota" : "Makassar",
    "kodepos" : "90126",
    "telp" : "08975500784"
  }
}
```

Gambar 2.5 Struktur BSON



Dengan adanya konsep *key-value* yang terdapat pada sistem *MongoDB*, membuat data yang tersimpan seperti dokumen otomatis memiliki nama *index id* yang unik dan tidak dimiliki sistem lainnya. Sehingga hal tersebut sangat membantu dalam mempercepat suatu proses pencarian data-data yang tersimpan dalam skala *global*. *MongoDB* memiliki beberapa kelebihan, di antaranya adalah sebagai berikut:

1. Performa sistem *MongoDB* lebih cepat daripada MySQL karena format dokumen yang berbentuk JSON.
2. *Replikasi*, yaitu fitur yang memungkinkan *backup* data secara *real-time*, sehingga *MongoDB* dapat digunakan pada blog dan aplikasi lainnya dengan tingkat kecocokan dan efisiensi yang tinggi.
3. *Auto-sharding*, yaitu fitur yang membagi data di dalam *database* dan data yang berukuran besar menjadi beberapa bagian untuk mengoptimalkan kinerja *database*.
4. *Cross-platform*, sistem ini dapat digunakan pada beberapa *operating sistem* seperti *Linux*, *Windows*, *OS X* dan *Solaris*.
5. Proses *Create*, *Read*, *Update*, *Delete* pada *database* sangat memudahkan pengguna.
6. *GridFS*, spesifikasi dalam *MongoDB* yang digunakan untuk menyimpan data yang berukuran sangat besar.

*MongoDB* yang merupakan produk NoSQL memiliki beberapa jenis format penyimpanan data, yaitu sebagai berikut:

1. *Document Database* dimana setiap satu objek data yang ada dapat disimpan kedalam dokumen-dokumen yang ada dimana terdiri dari kunci-kunci atau *keyvalue*, dan *value*/nilai dapat berupa jenis data *array*.
2. *Graph* merupakan penyimpanan data dalam bentuk *graph* yang sering digunakan dalam suatu interaksi seperti jejaring sosial.
3. *Key-value* atau kunci terenskrip pada database adalah kunci unik yang dikeluarkan untuk setiap pengguna. Salah satu contohnya adalah *apache cassandra*.

4. Objek *database* adalah bentuk dari *database* yang telah disimpan dalam suatu objek, seperti yang terjadi pada pemrograman berbasis objek.

## **2.10 Metode *Waterfall***

Pressman (2012) mendefinisikan metode *Waterfall* sebagai suatu metode pengembangan secara sekuensial yang sistematis dan berurutan dalam membangun sebuah perangkat lunak. Proses pembuatannya mengikuti alur dari mulai analisis, desain, kode, pengujian dan pemeliharaan. Metode ini memiliki beberapa kelebihan, antara lain: mudah dipahami dan dapat diterapkan dalam proses pengembangan perangkat lunak. Menurut Sommerville (2011), metode *Waterfall* memiliki tahapan-tahapan sebagai berikut:

### *1. Requirement Analysis*

Proses ini mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh sistem yang akan dikembangkan. Tahapan ini yaitu menentukan klasifikasi data yang akan membantu dan mendukung dalam perancangan desain aplikasi dan basis data untuk mempermudah dalam pengaksesan program yang akan dibuat.

### *2. System and Software Design*

Tahapan ini merupakan tahap penyusunan proses, data, aliran proses, dan hubungan antar data yang paling optimal untuk memenuhi kebutuhan dari hasil yang didapatkan pada tahap sebelumnya. Tahapan ini membuat desain sistem dan aliran proses dari sistem yang akan dirancang agar dapat diimplementasikan menjadi program pada tahap berikutnya.

### *3. Implementation*

Ini adalah tahapan untuk mengkonversi rancangan yang telah dibuat menjadi sebuah sistem yang dapat beroperasi sesuai dengan keperluan. Desain yang telah dibuat kemudian dikodekan dengan menggunakan bahasa pemrograman yang telah ditentukan untuk menciptakan desain sistem dan aliran proses yang telah dirancang sebelumnya.

### *4. System Testing*

Proses pengujian dilakukan untuk memastikan sistem yang dikembangkan berjalan dengan semestinya. Proses pengujian berfokus pada perangkat

lunak dari segi fungsional agar dapat meminimalisir kesalahan dan menghasilkan output yang sesuai dengan kebutuhan.

#### 5. *Operation and Maintenance*

Tahap akhir ini, yaitu proses perawatan mulai dari software dan hardware agar performa dari sistem yang telah dibuat tetap stabil.

### 2.11 UML (*Unified Modeling Language*) Diagram

UML (*Unified Modelling Language*) adalah sebuah cara untuk memvisualisasikan pemodelan yang digunakan sebagai alat untuk merancang sistem berbasis objek. UML juga dapat didefinisikan sebagai bahasa standar visual, desain, dan dokumentasi sistem, atau lebih dikenal sebagai bahasa standar untuk menuliskan *blueprint software*. UML digunakan untuk memudahkan pengembangan perangkat lunak (RPL) dan memenuhi semua kebutuhan pengguna secara efektif, komprehensif, dan tepat. Hal ini termasuk faktor-faktor skalabilitas, *robustness*, keamanan, dan lainnya. Adapun tujuan dan fungsi UML yaitu sebagai berikut:

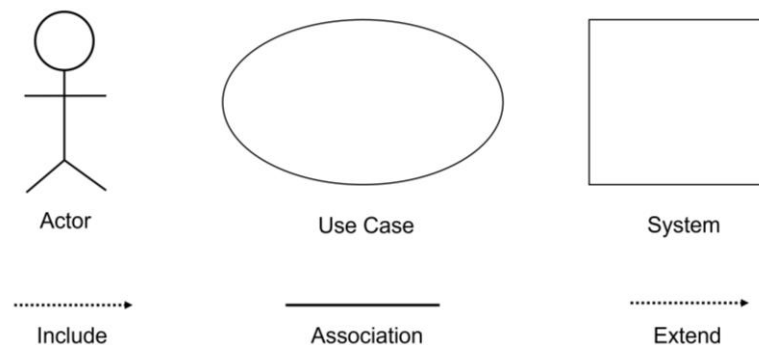
1. Menyediakan representasi visual atau gambar untuk pengguna dari berbagai bahasa pemrograman dan proses umum rekayasa.
2. Mengumpulkan data terbaik yang tersedia untuk membangun model.
3. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
4. Menggambarkan model sistem *software* dan juga dapat memodelkan sistem berorientasi pada objek.
5. Mempermudah pengguna untuk membaca suatu sistem.
6. UML berfungsi sebagai *blueprint* yang jelas untuk menjelaskan informasi lebih rinci tentang bagaimana perancangan *coding* sebuah program.

UML juga bisa digunakan sebagai alat untuk mentransfer pengetahuan tentang sistem aplikasi yang akan dikembangkan dari satu developer ke developer lainnya. UML sangat penting bagi beberapa orang karena ia berfungsi sebagai jembatan yang menghubungkan pengembang sistem dengan pengguna. Di sinilah pengguna dapat memahami sistem yang nantinya akan dikembangkan. UML

menawarkan berbagai jenis diagram untuk memodelkan aplikasi perangkat lunak yang berorientasi pada objek, yaitu:

### 2.11.1 Use Case Diagram

*Use Case* Diagram adalah satu jenis dari diagram UML (*Unified Modelling Language*) yang digunakan untuk menggambarkan kebutuhan fungsional yang diharapkan dari sebuah sistem. Diagram *Use Case* ini berguna untuk menggambarkan secara singkat siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada *Use Case*. Adapun simbol-simbol yang digunakan pada use case diagram dapat dilihat pada gambar 2.6.



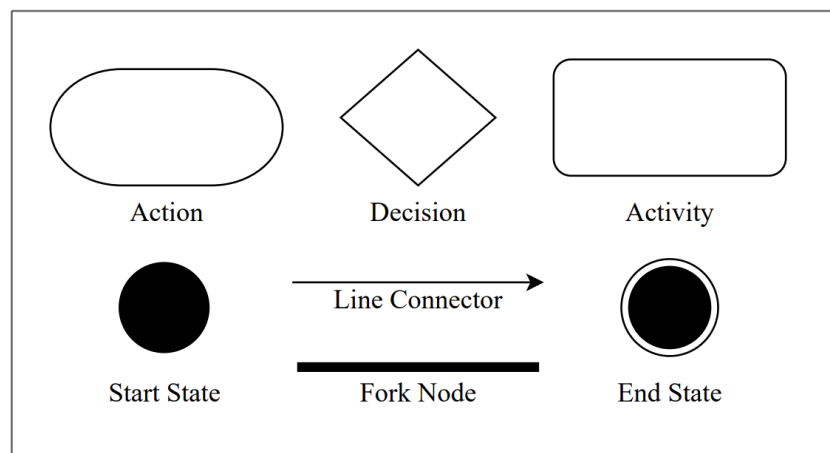
Gambar 2.6 Simbol-Simbol *Use Case* Diagram

Keterangan :

- Aktor/actor**, menjelaskan orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
- Use Case**, menjelaskan tentang tindakan/aksi yang dilakukan oleh aktor.
- System**, menjelaskan paket yang menampilkan sistem secara terbatas.
- Include**, menunjukkan bahwa *use case* satu merupakan bagian dari *use case* lainnya.
- Extend**, menunjukkan bahwa *use case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu.
- Association**, komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.

### 2.11.2 Activity Diagram

*Activity* diagram atau diagram aktivitas, adalah sebuah diagram yang menggambarkan aliran aktivitas atau aliran kerja dalam sebuah sistem. Diagram ini juga digunakan untuk mendefinisikan atau mengelompokkan urutan tampilan dari sistem tersebut. *Activity* Diagram memiliki komponen dengan bentuk yang telah ditentukan yang dihubungkan dengan tanda panah. Panah tersebut menunjukkan urutan aktivitas yang berlangsung dari awal hingga akhir. Simbol-simbol *activity* diagram dapat dilihat pada gambar 2.7.



Gambar 2.7 Simbol-Simbol *Activity* Diagram

Keterangan :

- Action**, *state* dari sistem yang mencerminkan eksekusi dari suatu aksi.
- Decision**, menyatakan pilihan untuk mengambil keputusan/tindakan pada kondisi tertentu.
- Activity**, menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
- Star state**, menyatakan bagaimana objek dibentuk atau diawali.
- End state**, menyatakan bagaimana objek dibentuk dan diakhiri.
- Fork Node**, satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

### 2.12 MongoDB Schema Diagram

*MongoDB Schema Diagram* adalah suatu alat visualisasi yang digunakan untuk merepresentasikan skema dokumen dalam *database MongoDB*. Diagram ini

menunjukkan hubungan antara dokumen dan *field* dalam *database MongoDB*. *MongoDB Schema Diagram* mirip dengan *Entity-Relationship Diagram (ERD)* dalam *database* relasional. Namun, karena *MongoDB* adalah *database NoSQL* yang tidak memiliki struktur tabel dan relasi yang kaku seperti *database* relasional.

*MongoDB Schema Diagram* memiliki beberapa perbedaan. Dalam *MongoDB Schema Diagram*, setiap dokumen dalam *database* direpresentasikan oleh sebuah kotak (*box*) yang berisi nama dokumen. Setiap *field* dalam dokumen direpresentasikan oleh sebuah baris berisi nama *field* dan tipe datanya. Hubungan antar dokumen dalam *database* direpresentasikan oleh garis yang menghubungkan kotak dokumen. Hubungan antar *field* dalam sebuah dokumen direpresentasikan oleh garis yang menghubungkan *field*. *MongoDB Schema Diagram* membantu pengembang dan administrator *database* untuk memahami struktur dokumen dalam *database MongoDB* dan memvisualisasikan hubungan antar dokumen dan *field* dengan lebih mudah.

### **2.13 Black Box Testing**

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Pengujian *black box* sendiri adalah teknik pengujian yang mengabaikan struktur logika internal dari perangkat lunak. Metode ini digunakan untuk menentukan apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* adalah metode desain data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian hasilnya dicek untuk memastikan bahwa sesuai dengan harapan.

Roger S. Pressman (2010) menjelaskan bahwa *black box testing* menekankan pada persyaratan fungsional perangkat lunak, yang memungkinkan *engineer* untuk memberikan *input* yang akan memenuhi persyaratan fungsional dari sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.

5. Kesalahan inisialisasi dan pemutusan kesalahan.

## BAB III

### METODE PENELITIAN

#### 3.1 Waktu dan Lokasi Penelitian

Penelitian ini dilaksanakan dari bulan Februari sampai dengan bulan April 2023, sebagaimana dapat dilihat pada Tabel 3.1. Lokasi penelitian untuk pengembangan aplikasi dilakukan di Fakultas MIPA Universitas Hasanuddin. Dan untuk pengambilan data terkait variabel penelitian dilakukan di kantor cabang PT. Astrindo Senayasa kota Makassar.

Tabel 3.1 Jadwal Kegiatan Penelitian

NO	Tahapan Penelitian	2023											
		Februari				Maret				April			
		1	2	3	4	1	2	3	4	1	2	3	4
1	Pengumpulan Data												
2	Analisis Kebutuhan												
3	Perencanaan												
4	Desain Aplikasi												
5	Pembuatan Aplikasi												
6	Pengujian Aplikasi												
7	<i>Deployment</i>												

#### 3.2 Metode Pengumpulan Data

##### 1. Observasi

Observasi yaitu pengumpulan data dengan melakukan langsung pengamatan terhadap sistem yang sedang berlaku agar bisa mendapatkan data yang aktual dari hasil penelitian yang dilakukan.

##### 2. Wawancara

Wawancara yaitu proses pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara peneliti dan narasumber dalam hal



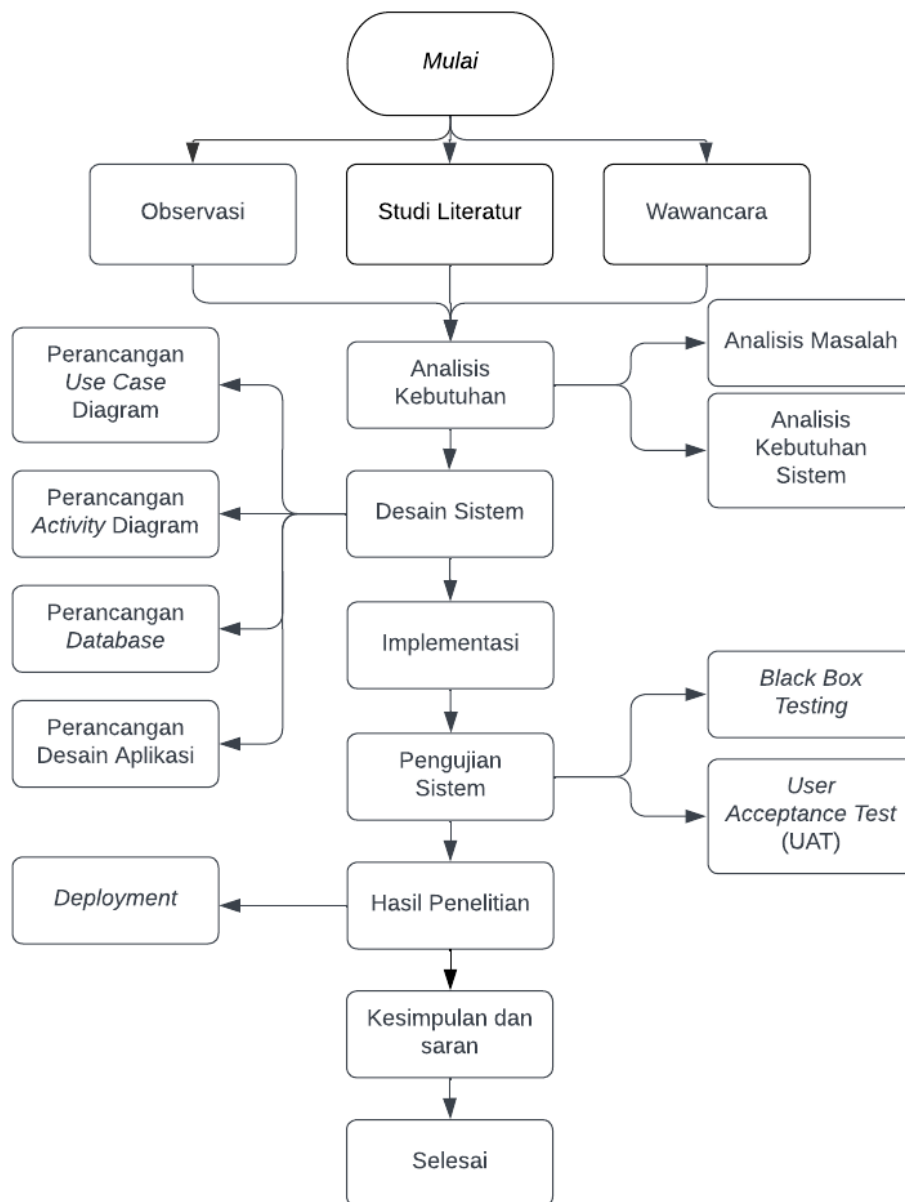
ini pihak PT. Astrindo Senayasa terkait dengan permasalahan yang sedang di teliti.

### 3. Studi Kepustakaan

Studi Kepustakaan yaitu mencari teori-teori yang telah berkembang dalam bidang ilmu yang berpengaruh dalam penelitian ini dan mencari metode teknik penelitian, baik dalam pengumpulan data pengolahan dan menganalisa data.

### 3.3 Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan dimulai dengan pengumpulan data yang menggunakan metode observasi, wawancara, dan studi kepustakaan. Setelah itu, pengembangan sistem akan dilakukan dengan memakai metode *Waterfall*. Sesuai dengan urutan tahapan metode *Waterfall*, penelitian akan dimulai dengan menganalisis kebutuhan, mendesain sistem, mengimplementasikan, melakukan pengujian unit atau pengujian sistem, dan melakukan pemeliharaan. Pengujian sistem akan dilakukan dan jika sistem yang telah dibangun tidak berjalan dengan baik, dan jika diperlukan, seluruh tahapan pengembangan sistem akan diulangi mulai dari analisis kebutuhan hingga pengujian sistem. Jika sistem berhasil, maka seluruh proses pengembangan sistem akan dianggap selesai seperti pada gambar 3.1.



Gambar 3.1 Tahapan Penelitian

### 3.4 Metode Pengembangan Sistem

Metode pengembangan sistem adalah serangkaian prosedur atau langkah-langkah dalam membangun sebuah sistem informasi. Setelah melakukan studi literatur, metode yang paling cocok untuk diterapkan dalam pembuatan sistem informasi ini adalah metode *waterfall*. Adapun tahapan yang terdapat dalam metode *waterfall* meliputi:

#### 3.4.1 Analisis Kebutuhan

Analisis kebutuhan perangkat lunak menentukan apa yang harus dilakukan sistem dan mendefinisikan batasan-batasan operasi dan implementasinya agar dapat mengomunikasikan secara tepat semua fungsi yang diberikan. Pada penelitian ini, peneliti mengumpulkan segala informasi dengan cara wawancara tidak terstruktur kepada pihak PT. Astrindo Senayasa. Proses wawancara yang dilakukan bertujuan untuk mengetahui kebutuhan sistem yang akan dikembangkan. Informasi ini nantinya akan digunakan sebagai acuan untuk mengetahui fitur apa saja yang akan tersedia.

#### 3.4.2 Perancangan Sistem

Pada tahap perancangan sistem, peneliti menerjemahkan kebutuhan sistem yang telah didapatkan pada tahap sebelumnya menjadi sebuah desain perangkat lunak agar dapat diimplementasikan menjadi program. Tahapan ini meliputi perancangan:

- a. *Unified Modelling Language (UML)*
- b. *MongoDb Schema Diagram*
- c. Tampilan Pengguna (*User Interface*)

#### 3.4.3 Implementasi

Tahapan implementasi mengacu pada desain sistem yang telah dirancang dari tahap sebelumnya. Desain tersebut diubah bentuknya menjadi bentuk yang dapat dimengerti oleh komputer, yaitu dengan melakukan proses *coding*. Dalam pengimplementasiannya ke dalam bahasa pemrograman berbasis *web*, peneliti menggunakan *framework* dari *javascript* yaitu *Next.js* sebagai bahasa pemrograman baik untuk *back-end* maupun *front-end* aplikasi.

#### 3.4.4 Pengujian Sistem

Proses pengujian dilakukan untuk memastikan bahwa sistem yang dikembangkan telah berjalan dan sudah sesuai dengan kebutuhan. Tahapan ini, peneliti menggunakan metode pengujian *Black Box* untuk mengetahui apakah *output* pada sistem sudah sesuai dengan kebutuhan. Dan pengujian *User Acceptance Test (UAT)* untuk mengetahui efektivitas penggunaan sistem yang telah dibangun berdasarkan penilaian pengguna.

### 3.4.5 Pemeliharaan Sistem

Pada tahap terakhir, sistem yang telah dikembangkan akan dilakukan proses pemeliharaan. Pemeliharaan memungkinkan peneliti untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perawatan *software*, perbaikan implementasi sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

### 3.5 Komponen Sistem

#### 1. Perangkat Lunak

Perangkat lunak yang akan digunakan adalah sebagai berikut:

- a. *Google Chrome*, untuk menampilkan tampilan *website*.
- b. *Visual Studio Code*, sebagai alat penulisan kode.
- c. *Mongodb Compass*, untuk manajemen *database*.
- d. *Postman*, untuk pengujian *API*.
- e. *Figma*, untuk merancang *user interface* website.

#### 2. Perangkat Keras

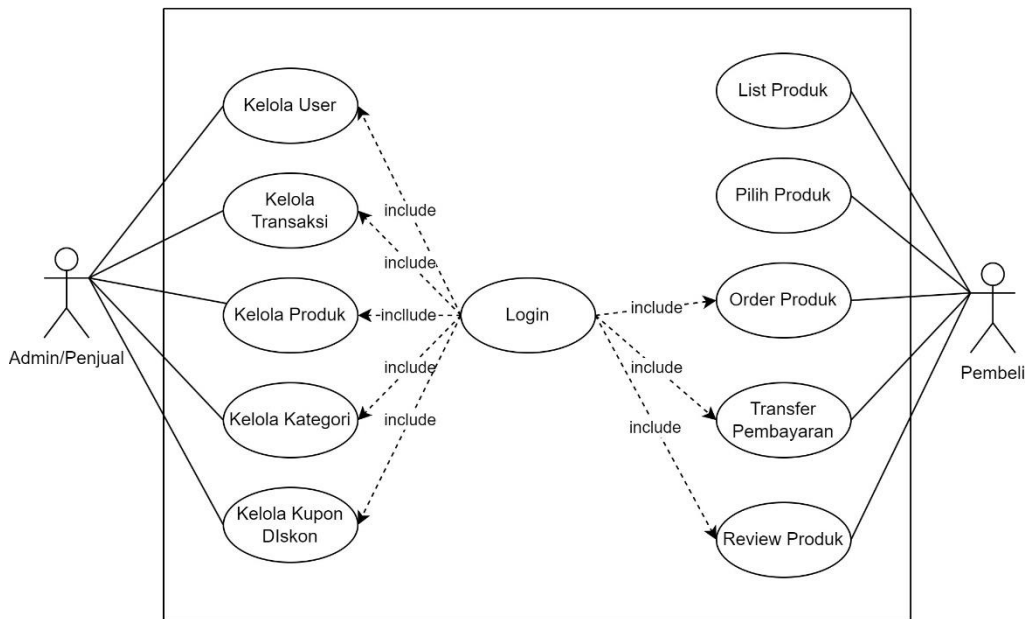
Perangkat keras yang akan digunakan pada penelitian adalah Laptop Asus Vivobook Pro 14X dengan spesifikasi sebagai berikut:

- a. AMD Ryzen 7 5800H
- b. RAM 16 GB
- c. SSD 1 TB

### 3.6 Perancangan Sistem

Perancangan sistem menggunakan *use case* diagram menggambarkan aktivitas yang dapat dilakukan oleh *stakeholder*, sehingga bisa dikatakan sebagai suatu kesatuan dari fitur-fitur utama pada sistem. *Use case* diagram pada penelitian seperti pada Gambar 3.2, terdapat dua aktor yaitu admin/penjual dan pembeli. Admin perlu melakukan *login* untuk bisa mengakses seluruh aktivitas seperti mengelola *user*, transaksi, produk, kategori, dan kupon. Sedangkan, pelanggan dapat mengakses list produk, memilih produk dan perlu melakukan *login* untuk melakukan *order* produk, pembayaran, dan memberikan *review* pada produk yang telah dibeli.

## USE CASE DIAGRAM



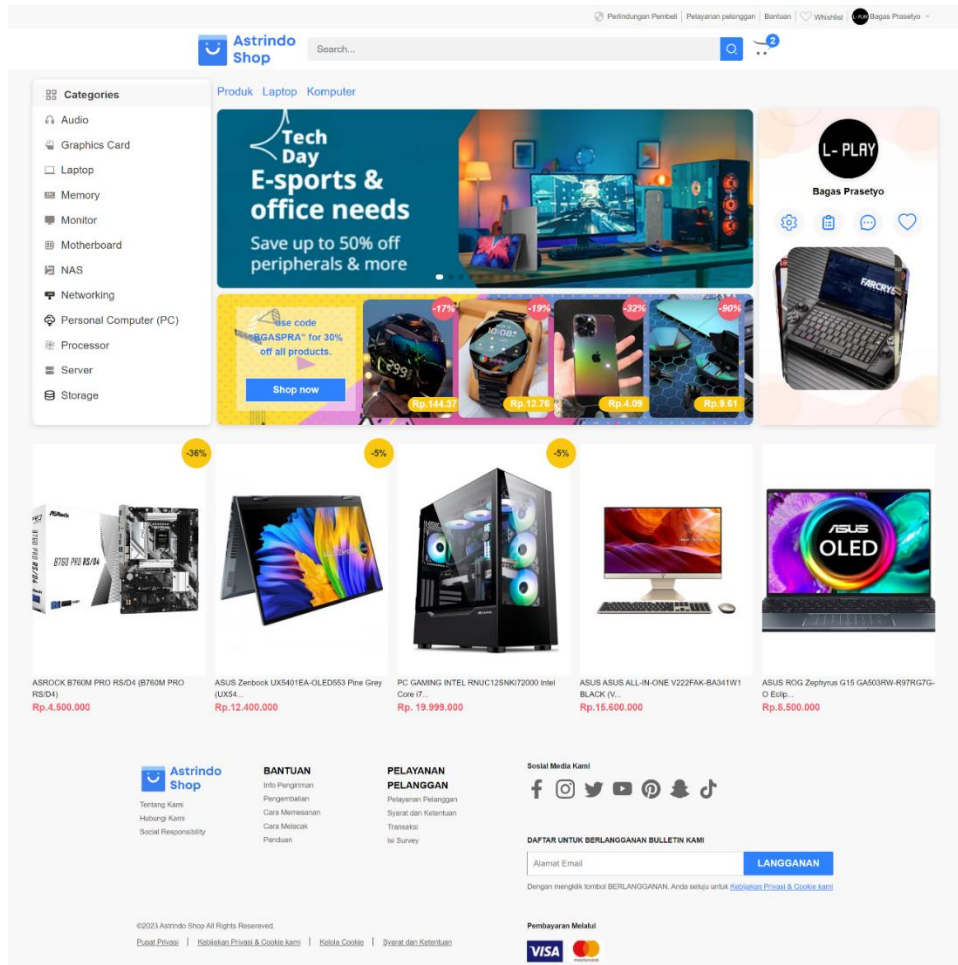
Gambar 3.2 *Use Case Diagram*

### 3.7 Rancangan *User Interface*

Rancangan *User Interface* (UI) merupakan bentuk tampilan visual yang berhubungan langsung dengan para pengguna agar mendapatkan gambaran mengenai fitur-fitur yang akan dihadirkan pada sistem. Merancang *user interface* dari awal berfungsi sebagai acuan untuk membuat tampilan website yang akan digunakan selama pengembangan website berlangsung. Adapun rancangan *user interface* pada penelitian ini mengenai sistem informasi *e-commerce* berbasis *website* yaitu:

#### 1. Halaman *Home*

Halaman *home* adalah halaman utama yang muncul ketika *website* dikunjungi seperti pada gambar 3.3. Pada halaman *home* terdapat fitur pencarian produk, daftar produk, keranjang belanja, serta kategori-kategori produk.



Gambar 3.3 Halaman *Home*

## 2. Halaman *Login/Sign Up*

Halaman login adalah halaman yang digunakan untuk pengguna mendaftarkan akun baru jika belum memiliki akun dan *login* jika telah memiliki akun. Terdapat beberapa pilihan untuk *login* ataupun mendaftar baru yaitu menggunakan alamat *email*, *google*, ataupun *facebook* seperti pada gambar 3.4. Jika pengguna mendaftar menggunakan email maka diperlukan proses verifikasi akun yang akan dikirim ke email pengguna.

Terima Kasih Memilih Kami! [Ke Toko](#)

## Masuk

Silakan masukkan informasi akun Anda untuk melanjutkan.

Email Address

Password

Masuk

Lupa password ?

Atau Lanjutkan dengan

Login dengan Facebook

Login dengan Google

## Daftar Baru

Sudah siap untuk belanja? Masukkan informasi akun Anda untuk memulai.

Nama Lengkap

Alamat Email

Password

Masukkan Ulang Password

Daftar

**BANTUAN**  
Info Pengiriman  
Pengembalian  
Cara Memesan  
Cara Melacak  
Panduan

**PELAYANAN PELANGGAN**  
Pelayanan Pelanggan  
Syarat dan Ketentuan  
Transaksi  
Istis Survey

Sosial Media Kami

DAFTAR UNTUK BERLANGGANAN BULETIN KAMI

Alamat Email

LANGGANAN

Dengan mengklik tombol DAFTAR BERLANGGANAN, Anda setuju untuk [Ketentuan Privasi & Cookie kami](#)

©2023 Astrindo Shop All Rights Reserved.  
Pusat Inovasi | Kebijakan Privasi & Cookie kami | Kelola Cookie | Syarat dan Ketentuan

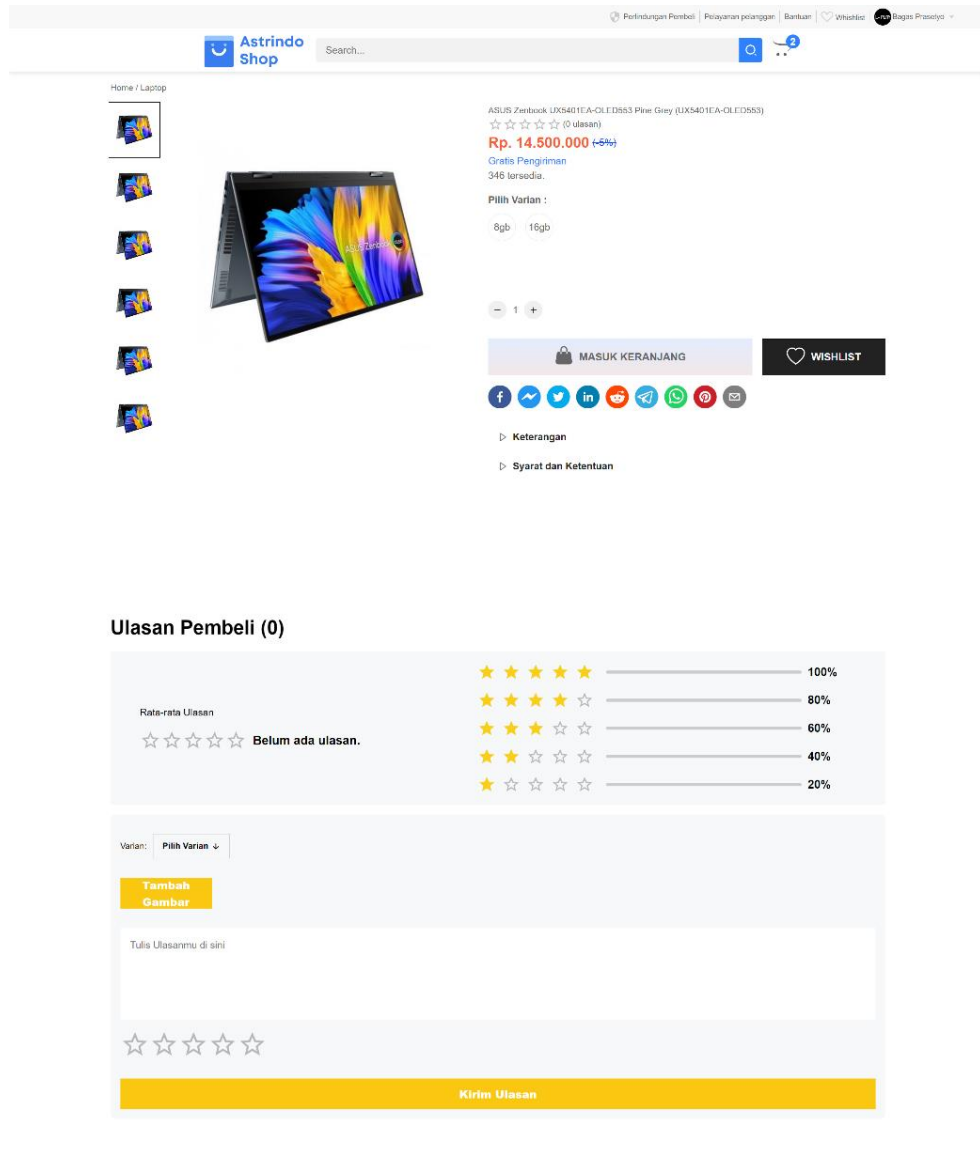
Pembayaran Melalui

VISA

Gambar 3.4 Halaman *Login/Sign Up*

### 3. Halaman Detail Produk

Halaman detail produk adalah halaman yang menampilkan informasi detail tentang sebuah produk, seperti deskripsi produk, harga, foto, spesifikasi, ulasan pengguna, dan lain-lain seperti pada gambar 3.5. Halaman ini bertujuan untuk memberikan informasi yang lengkap dan detail kepada pengguna agar mereka dapat memutuskan untuk membeli produk tersebut atau tidak. Di halaman ini juga terdapat fitur ulasan dari produk tersebut untuk pengguna yang telah membeli produk tersebut memberikan ulasan.

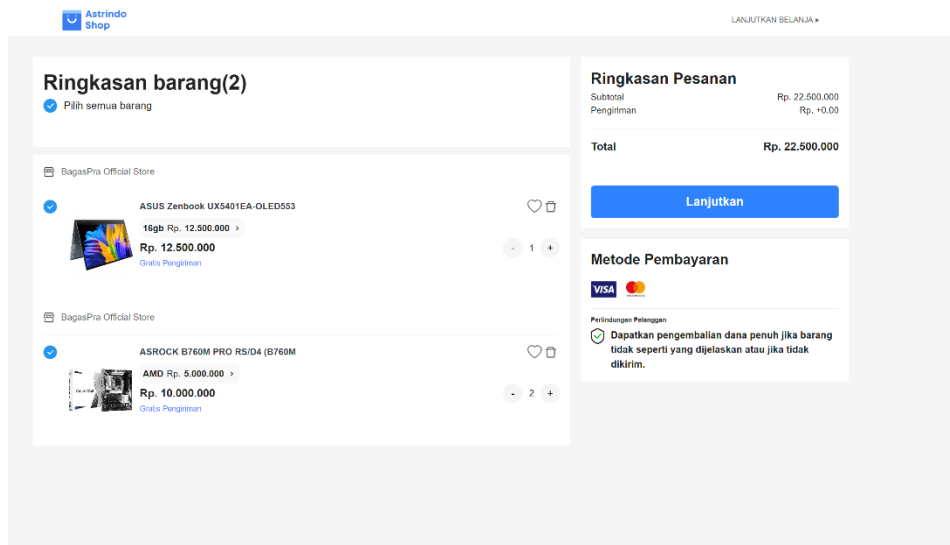


Gambar 3.5 Halaman Detail Produk

#### 4. Halaman Keranjang

Halaman keranjang adalah halaman yang menampilkan semua produk yang telah dipilih oleh pengguna untuk dibeli seperti pada gambar 3.6. Pada halaman ini, pengguna dapat melihat daftar produk yang telah dipilih beserta kuantitas dan harga totalnya. Pengguna juga dapat menghapus produk yang tidak ingin dibeli, mengubah kuantitas produk, atau menambah produk baru ke dalam keranjang belanja. Setelah pengguna selesai memilih produk, mereka dapat melanjutkan proses pembelian dengan melakukan *checkout*.

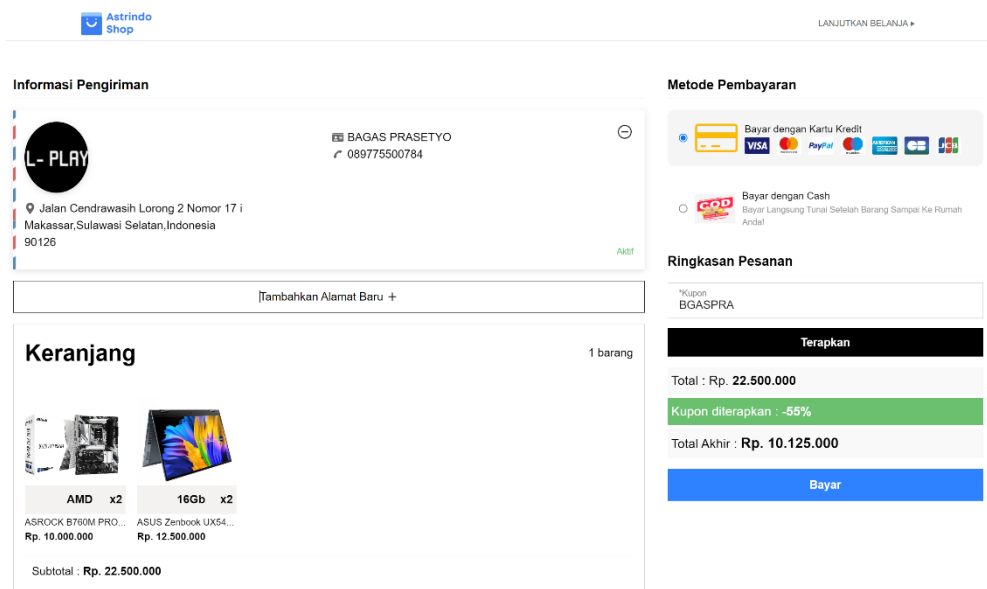




Gambar 3.6 Halaman Keranjang

## 5. Halaman *Checkout*

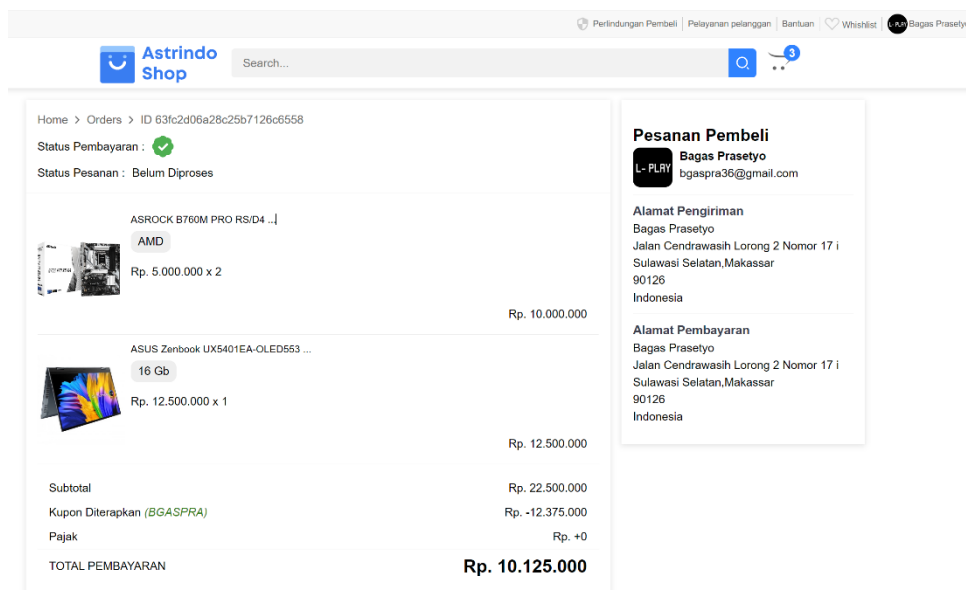
Halaman *checkout* adalah halaman menampilkan informasi terkait pembelian produk yang telah dipilih oleh pengguna . Pada halaman ini, pengguna diminta untuk mengisi informasi pengiriman, dan memilih metode pembayaran seperti pada gambar 3.7. Setelah pengguna memastikan informasi yang diisi sudah benar, pengguna dapat melakukan konfirmasi pembelian dan melakukan pembayaran.



Gambar 3.7 Halaman *Checkout*

## 6. Halaman Pembayaran

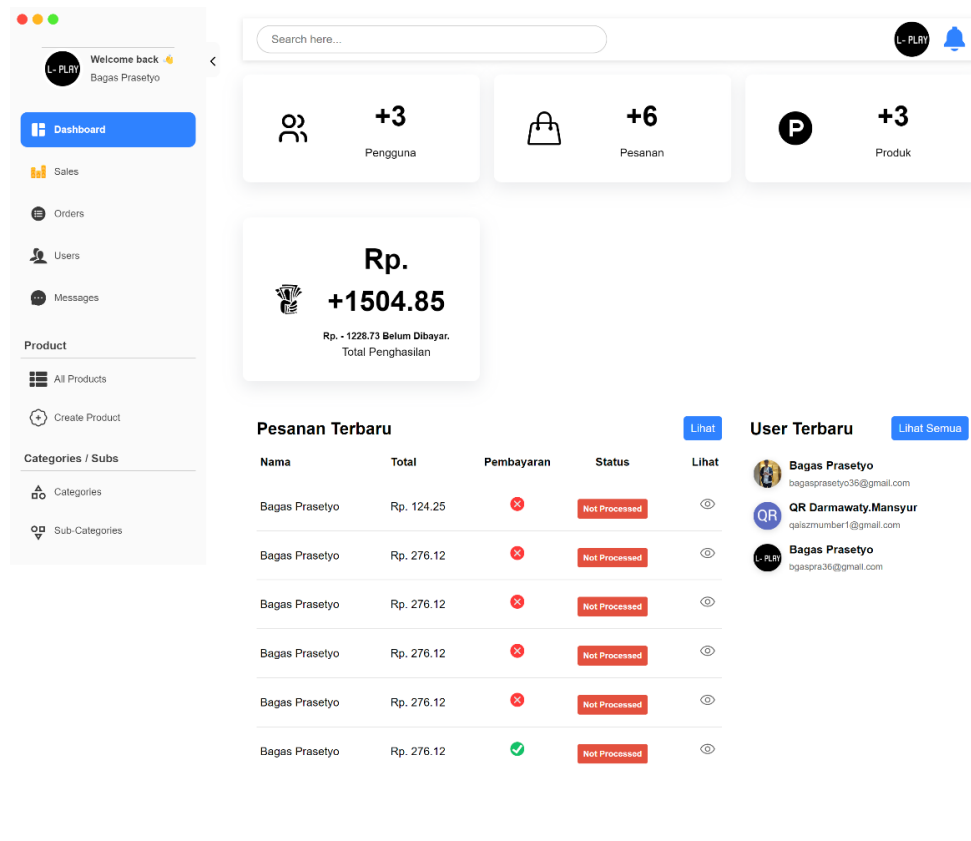
Halaman pembayaran adalah halaman yang menampilkan informasi terkait pembayaran untuk produk atau layanan yang telah dipilih oleh pengguna seperti pada gambar 3.8. Pada halaman ini, pengguna akan diminta melakukan pembayaran sesuai dengan metode pembayaran yang telah dipilih sebelumnya, memasukkan informasi pembayaran seperti nomor kartu kredit atau rekening bank, dan melakukan pembayaran.



Gambar 3.8 Halaman Pembayaran

## 7. Halaman Admin

Pada halaman ini, admin dapat melakukan berbagai tindakan seperti menambah, mengedit, atau menghapus produk, mengatur harga, mengatur stok, dan melakukan analisis penjualan seperti pada gambar 3.9. Pada halaman ini juga admin dapat mengatur kategori maupun subkategori yang ada pada web. Selain itu, halaman admin *e-commerce* juga menyediakan laporan penjualan, informasi pengiriman, informasi pembayaran, serta alat untuk memantau aktivitas pengguna.



Gambar 3.9 Halaman Admin

## Daftar Pustaka

- Hutahaean, J. (2015). *Konsep sistem informasi*. Deepublish.
- Indrayani, E. (2009). Humdiana “Sistem Informasi Manajemen” Mitra Wacana Media.
- Kadir, A. (2003). Pemrograman Web Mencakup: HTML, CSS, JavaScript & PHP. *Yogyakarta: Andi*.
- Kadir, A. (2009). Dasar perancangan dan implementasi database relasional. *Yogyakarta: Andi*.
- Komputer, W. (2010). *Panduan Praktis Menguasai Pemrograman Web dengan JavaScript 2009*. Penerbit Andi.
- Kotler, P., & Armstrong, G. (2012). Principles of marketing 11th ed.
- Kroenke, D. M., & Gray, C. D. (2006). Toward a Next Generation Data Modeling Facility: Neither the Entity-Relationship Model nor UML Meet the Need. *Journal of Information Systems Education*, 17(1).
- Lathifah, L. (2020). Penerapan Enterprise Architecture pada Penerimaan Mahasiswa Baru menggunakan TOGAF di Universitas X Palembang. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 7(3), 647-655.
- Lathifah, L., Suaidah, S., Anam, M. K., & Suandi, F. (2021). Pemodelan Enterprise Architecture Menggunakan Togaf Pada Universitas X Palembang. *Jurnal Teknoinfo*, 15(1), 7-12.
- Laudon, K. C., & Laudon, J. P. (2004). *Management information systems: Managing the digital firm*. Pearson Educación.
- Lubis, A. (2016). *Basis data dasar*. Deepublish.
- Marakas, G. M., & O'Brien, J. A. (2013). *Introduction to information systems* (p. 624). New York: McGraw-Hill/Irwin.
- Muhammad Fakhri Husein; Amin Wibowo. (2006.). Sistem informasi manajemen. Yogyakarta :: UPP STIM YKPN.

- Mumtahana, H. A., Nita, S., & Tito, A. W. (2017). Pemanfaatan Web E-Commerce untuk meningkatkan strategi pemasaran. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, 3(1), 6-15.
- Pressman, R. S. (2002). Rekayasa perangkat lunak pendekatan praktisi (buku satu). *Yogyakarta: Andi*.
- Pressman, R. S., & Maxim, B. R. (2010). Software Engineering: A practitioners Ap-proach. McGrawHill. *New York*, 68.
- Sommerville, Ian (2011). "Software Engineering 9th Edition". Addison-Wesley.
- Sukanto, dan Shalahuddin, M. (2018). Rekayasa Perangkat Lunak (Edisi Revi). Bandung: Informatika Bandung.
- Sutabri, T. (2012). *Konsep sistem informasi*. Penerbit Andi.
- Whitten, Jeffrey L. and Lonnie D. Bentley (2007). "Systems Analysis and Design Methods 7th Edition". New York: McGraw-Hill Companies, Inc.
- Winarno, Sugeng. (2006). Jaringan Komputer Dengan TCP/IP, Membahas Konsep dan Teknik Implementasi TCP/IP dalam Jaringan Komputer. *Penerbit Modula, Bandung*, 310.
- Wong, J. (2010). Internet Marketing for Beginners, Elex Media Komputindo. *Jakarta: Elex Media Komputindo*.
- Zulkarnain, A. (2019). Penerapan Mobile-First Design pada Antarmuka Website Profil Sekolah Menggunakan Metode Human-Centred Design (Studi Kasus: SMPN 21 Malang. *Jurnal Ilmiah Teknologi Informasi Asia*, 13(2), 125-136.