



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Teori Pengumpulan Data**

##### **2.1.1 Kuesioner**

Kuesioner adalah suatu teknik pengumpulan informasi yang memungkinkan analis mempelajari sikap-sikap, keyakinan, perilaku, dan karakteristik beberapa orang yang bias terpengaruh oleh sistem yang diajukan atau sistem yang sudah ada.

Kuesioner adalah alat populer dan fundamental untuk memperoleh informasi tentang pengetahuan dan persepsi publik. Kuesioner bias memberikan nilai tambah terhadap informasi yang dibutuhkan oleh instansi atau perusahaan yang membutuhkan. Kuesioner sering digunakan penelitian pemasaran kuantitatif dan penelitian sosial. Sebuah kuesioner adalah serangkaian pertanyaan yang diajukan kepada individu untuk memperoleh informasi yang berguna secara statistik mengenai topik yang diberikan. Bila dibuat dengan benar dan bertanggung jawab, kuesioner menjadi instrumen penting yang dapat digunakan untuk membuat pernyataan tentang kelompok atau orang tertentu atau seluruh populasi (Roopa, et al., 2012)

**N U S A N I A R A**

### 2.1.2 Data Mining

Data *Mining* adalah ekstraksi pola yang menarik dari data dalam jumlah besar. Suatu pola dikatakan menarik apabila pola tersebut tidak sepele, implisit, tidak diketahui sebelumnya, dan berguna. Pola yang disajikan harus mudah dipahami, berlaku untuk data yang akan diprediksi dengan derajat kepastian tertentu, berguna, dan baru. Penggalan data memiliki beberapa nama alternatif, meskipun definisi eksaknya berbeda, seperti KDD (knowledge discovery in database), analisis pola, arkeologi data, pemanenan informasi, dan intelegensia bisnis. Penggalan data diperlukan saat data yang tersedia terlalu banyak (misalnya data yang diperoleh dari sistem basis data perusahaan, e-commerce, data saham, dan data bioinformatika), tetapi tidak tahu pola apa yang bisa didapatkan.

Data *Mining* muncul pada pertengahan tahun 1990-an dan menjadi alat ampuh yang sesuai untuk mengambil pola yang sebelumnya tidak diketahui dan informasi bermanfaat dari kumpulan data yang besar. Berbagai penelitian menyoroti bahwa teknik *Data Mining* membantu pemegang data untuk menganalisis dan menemukan hubungan yang tidak terduga antara data mereka yang pada gilirannya membantu pengambilan keputusan. (Tomar, et al., 2013).

### 2.1.3 Skala Likert

Skala likert menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan,

sangat setuju, setuju, tidak memutuskan, tidak setuju, dan sangat tidak setuju (Likert1932) dan dapat digunakan untuk menghitung nilai akhir pada UAT suatu sistem. Kemudahan penggunaan skala likert menyebabkan skala ini lebih banyak digunakan oleh peneliti. Tipe/format data likert adalah satu pertanyaan yang berdiri sendiri yang menggunakan format jawaban seperti pada skala likert. Jika seorang peneliti menggunakan butir pertanyaan yang banyak tetapi berdiri sendiri (bukan merupakan suatu indikator dari sifat tertentu misalkan sikap, kebiasaan atau pengetahuan), maka peneliti tersebut tidak dapat menggabungkan nilai dari jawaban tiap pertanyaan. Tabel 2 memperlihatkan contoh butir-butir pertanyaan yang berdiri sendiri, sehingga penjumlahan dari jawaban tidak dapat dilakukan. Pada format data likert, penggunaan analisis data parametrik tidak disarankan karena format data likert mempunyai skala kelas ordinal. Analisis data yang dapat digunakan misalnya median, modus, dan frekuensi. (Budiaji, 2013)

## 2.1 Teori Pengolahan Data

### 2.1.1 TOPSIS

*TOPSIS* adalah istilah populer dari Technique for Order Preference by Similarity to Ideal Solution adalah metode yang dikenalkan pertama kali oleh Yoon dan Hwang, dimana alternatif yang dipilih memiliki jarak terdekat dengan solusi ideal positif dan dan memiliki jarak terjauh dari solusi ideal negatif. (Purnomo, et al., 2013). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan jarak

terpanjang (terjauh) dari solusi ideal relatif dari sudut dari sudut pandang geometris dengan menggunakan jarak Euclidean (jarak antara dua titik) untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut sedangkan solusi relatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut. TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal relatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. (Pardede, 2013)

### 2.1.2 Kelebihan Topsis

Kelebihan dari metode TOPSIS adalah :

1. mudahnya dalam memahami teori TOPSIS dilihat dari kesederhanaan alur proses metode TOPSIS yang tidak begitu rumit. Karena menggunakan relatif kriteria dan variabel alternatif sebagai pembantu untuk menentukan keputusan.
2. Perhitungan komputasi yang efisien dan cepat
3. Mampu dijadikan sebagai pengukur kinerja alternatif dan juga alternatif keputusan dalam sebuah bentuk output komputasi yang sederhana
4. Dapat digunakan sebagai metode pengambilan keputusan yang lebih cepat

### 2.1.3 Kekurangan Topsis

Kekurangan dari metode TOPSIS adalah :

1. Belum adanya penentuan bobot prioritas yang menjadi prioritas hitungan terhadap kriteria yang berguna untuk meningkatkan validitas nilai bobot perhitungan kriteria.
2. Belum adanya bentuk relatif untuk penilaian alternatif terhadap nilai kriteria, biasanya bentuk relatif ini diinterpretasikan dalam sebuah bilangan fuzzy.
3. Belum adanya mediator seperti hierarki jika diproses secara mandiri maka dalam ketepatan pengambilan keputusan cenderung belum menghasilkan keputusan yang sempurna.

## **2.3 Teori Aplikasi Website**

### **2.3.1 Web Application**

*Web Application* merupakan sebuah aplikasi yang menggunakan teknologi browser untuk menjalankan aplikasi dan diakses melalui jaringan relatif (Remick, 2011). Sedangkan menurut (Rouse, 2011) *Web Application* adalah sebuah program yang disimpan di Server dan dikirim melalui internet dan diakses melalui antarmuka browser.

Salah satu pendekatan untuk mengembangkan aplikasi Web berorientasi layanan adalah mengubah tingkat tinggi model bisnis dengan bahasa komposisi yang mengimplementasikan proses bisnis dengan web servis. Analisis dan desain berorientasi objek dan diagram berbasis UML biasanya digunakan dalam proses pengembangan perangkat lunak. (Al-Fedaghi, 2011)

### 2.3.2 Database

*Database* adalah kumpulan data yang disimpan secara sistematis di dalam *relative* yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Dengan kata lain, semua akses ke *database* akan ditangani oleh DBMS (*Database Management System*).

Perancangan *database* merupakan bagian dari proses pengembangan *database* yang melibatkan analisis definisi masalah (spesifikasi dan persyaratan) dan menyediakan semua temuan yang diperlukan untuk membangun struktur data yang logis. Definisi masalah menentukan secara keseluruhan atau kurang tujuan, kebutuhan, persyaratan dan batasan data yang diharapkan dapat mendukung beberapa organisasi operasi. Struktur data logis awalnya dapat diekspresikan dalam bahasa sederhana. (Letkowski, 2011)

Peneliti memilih MySQL yang merupakan salah satu perangkat lunak *system manajemen database SQL (DBMS)* yang *multithread*, dan *multi-user* sebagai penunjang perancangan dan pengembangan aplikasi pendukung keputusan pemilihan sepeda motor bertransmisi otomatis karena memiliki beberapa keunggulan dalam hal kompatibilitas, yang berarti bahwa MySQL dapat berjalan pada berbagai *platform* sistem operasi.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

## 2.4 Metode Perancangan Sistem

### 2.4.1 *Waterfall*

Metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan sebagai berikut :

#### 1. *Requirements Analysis*

Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau Telati langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

#### 2. *System Design*

Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain Sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

### 3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut *unit*, yang terintegrasi dalam tahap selanjutnya. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

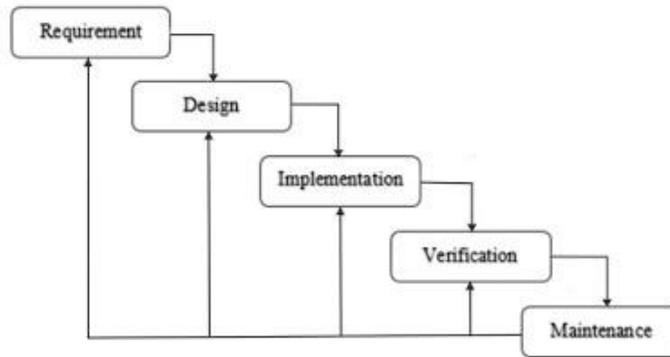
### 4. *Verification*

Seluruh *unit* yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing *unit*. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan.

### 5. *Maintenance*

Merupakan tahap akhir dalam model *waterfall*. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi *unit* sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

(Balaji, et al., 2012). Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini.



Gambar 2.1 Tahapan Metode *Waterfall*

Dilihat dari gambar 2.1 metode *waterfall* yang diusulkan oleh *Royce* pada tahun 1970 merupakan perangkat lunak sekuensial linier *Software Development Life Cycle (SDLC)*. Modelnya diberi nama *waterfall* karena merepresentasikan bentuk diagram seperti bertingkat dari atas kebawah. *Waterfall* juga dikelas sebagai *classical lifecycle model*, didalamnya termasuk *system requirements, initial and detailed design, implementation, testing, operations* serta *maintenance*. Keunggulan dari penggunaan *waterfall* antara lain:

1. Setiap tahap telah didefinisikan dengan baik dan jelas
2. Mudah digunakan dan dimengerti
3. Setiap tahap selesai sesuai dengan periode yang telah ditentukan agar periode tahap selanjutnya berjalan sesuai jadwal
4. Sebagai model linear, *waterfall* mudah untuk diimplementasi
5. Hanya membutuhkan sumber daya yang minim untuk menjalankan *waterfall*.

Sedangkan terdapat pula kekurangan dari penggunaan *waterfall*, antara lain:

1. Apabila terjadi kesalahan pada langkah sebelumnya, akan sangat sulit untuk kembali ke langkah sebelumnya tersebut untuk memperbaiki kesalahan. Apabila fase desain telah salah, akan sangat rumit untuk berlanjut ke tahap implementasi.
2. Seringkali klien tidak terlalu mengerti akan penggunaan *software* sehingga menimbulkan banyak kebingungan
3. Perubahan kecil atau eror yang muncul pada *software* yang telah selesai dibuat dapat menyebabkan banyak permasalahan kedepannya.

(Saxena, et al., 2016)

#### **1.4.2 Unified Modelling Language (UML Diagram)**

UML merupakan struktur dan teknik permodelan desain program berorientasi objek serta aplikasinya atau satu set pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sistem perangkat lunak dalam hal objek. (Rumbaugh , et al., 2014). Berikut adalah beberapa tools UML yang akan digunakan dalam pengembangan penelitian kali ini:

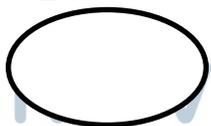
##### **1. Use Case Diagram**

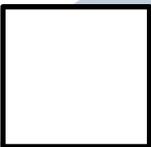
*Use case* diagram adalah gambaran secara ringkas pengguna sistem dan apa saja yang dapat dilakukan oleh tiap penggunanya tergantung peran masing-masing. *Use case* diagram mendeskripsikan sebuah interaksi antara satu aktor atau lebih dengan sistem yang akan dibuat.

Tampilan *use case* menangkap perilaku suatu sistem, sub sistem atau kelas saat muncul ke pengguna luar. Hal ini membagi fungsi sistem menjadi transaksi yang berarti bagi para aktor. Sebuah *use case* menggambarkan interaksi dengan aktor sebagai konsekuensi dari pesan antara sistem dan satu atau lebih aktor.

Tabel 2.1 adalah simbol-simbol yang digunakan dalam *use case* diagram :

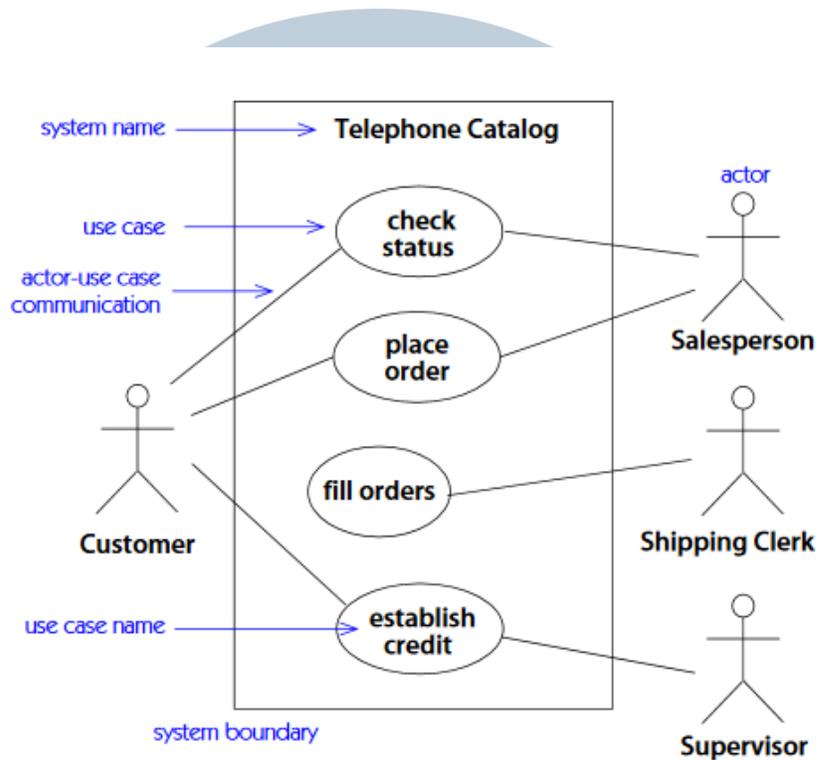
Tabel 2.1 Simbol dan Keterangan *Use Case* Diagram  
 Sumber : (Rumbaugh , et al., 2014)

Simbol	Nama Simbol	Keterangan
 Actor	<i>Actor</i>	Mewakikan himpunan peran baik orang, sistem ataupun alat ketika berinteraksi dengan <i>use case</i>
	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada diatas objek induk (ancestor)
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

	<i>Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
	<i>Dependency</i>	Menggambarkan suatu hubungan yang ketergantungan pada suatu elemen yang mempengaruhi elemen lain
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan

# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.2 Contoh *Use Case* Diagram  
 Sumber : (Rumbaugh , et al., 2014)

Gambar 2.2 menggambarkan contoh dari *use case* diagram beserta dengan keterangan dari setiap fungsi gambar yang tertera pada use case tersebut.

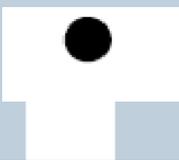
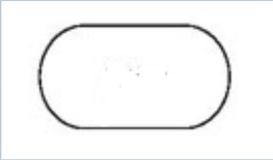
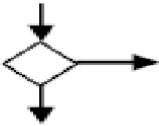
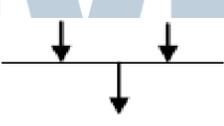
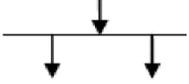
## 2. Activity Diagram

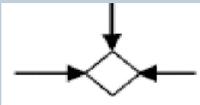
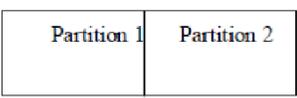
*Activity* diagram menggambarkan *workflow* (alur kerja) dari sebuah sistem atau proses bisnis. Namun yang perlu diperhatikan adalah bahwa *activity* diagram fokus terhadap aktivitas sistem. *Activity* diagram memiliki peran seperti *flowchart* namun memiliki perbedaan dimana *activity* diagram dapat mendukung perilaku paralel tetapi *flowchart* tidak bisa.

Tabel 2.3 berikut adalah simbol-simbol yang biasa digunakan dalam *activity* diagram :

Tabel 2.2 Simbol dan Keterangan *Activity Diagram*

Sumber : (Rumbaugh , et al., 2014)

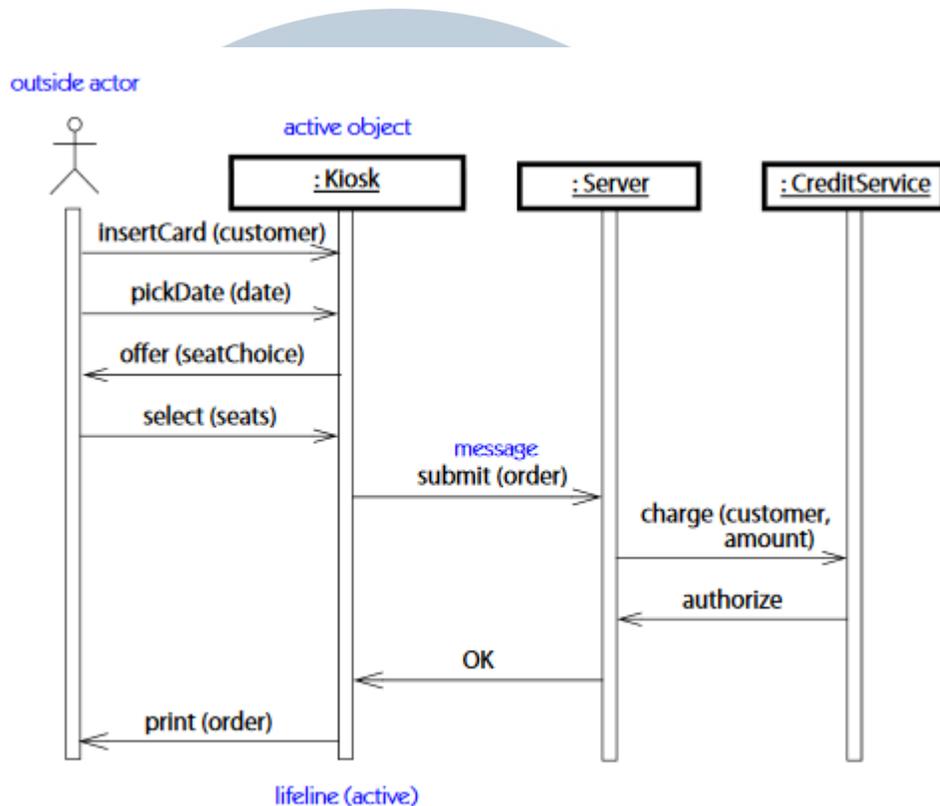
Nama Simbol	Simbol	Keterangan
<i>Initial Node</i>		<i>Initial Node</i> diwakili oleh lingkaran yang terisi
<i>Activity</i>		<i>Activity</i> diwakili oleh persegi panjang dengan siku tumpul
<i>Edge</i>		Transisi digambar sebagai panah yang diarahkan untuk menunjukkan aliran kontrol diantara aktivitas
<i>Decision Node</i>		<i>Decision Node</i> diwakili oleh berlian yang biasanya memiliki aliran kedalam dan dua aliran keluar
<i>Join</i>		<i>Join</i> diwakili dengan garis biasa disebut dengan synchronization bar menerima beberapa kedalam dan satu aliran keluar
<i>Fork</i>		<i>Fork</i> diwakili dengan garis biasa disebut dengan synchronization bar yang menerima satu aliran keluar kedalam dan kedua

<i>Merge</i>		<i>Merge</i> dilambangkan dengan kotak bentuk berlian dengan beberapa arus kedalam
<i>Swim Lanes</i>		Menggambarkan arus aktivitas yang bersamaan dan oleh siapa ini dilakukan
<i>Final Node</i>		<i>Final Node</i> diwakili oleh lingkaran yang terisi dan dikelilingi oleh garis lingkaran

### 3. *Sequence* Diagram

*Sequence* diagram digunakan untuk merepresentasikan atau memodelkan aliran pesan, peristiwa dan tindakan antara objek atau komponen dari suatu sistem. Urutan diagram digunakan terutama untuk merancang, mendokumentasikan dan memvalidasi arsitektur, antarmuka dan logika sistem dengan menggambarkan urutan tindakan yang perlu dilakukan untuk menyelesaikan tugas atau skenario. Mereka adalah alat desain yang berguna karena mereka memberikan pandangan dinamis dari perilaku sistem yang dapat sulit untuk mengekstrak dari diagram spesifikasi statis. dapat dilihat pada gambar 2.3.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.3 Contoh *Sequence Diagram*

Sumber : (Rumbaugh , et al., 2014)

Pada Gambar 2.3 merupakan contoh implementasi sequence diagram mengikuti teori-teori yang benar berdasarkan referensi buku referensi manual *unified modeling language*.

#### 4. *Class Diagram*

*Class diagram* adalah presentasi grafis dari tampilan statis yang menunjukkan koleksi elemen model deklaratif (statis), seperti kelas, jenis, dan kontennya serta hubungan. Diagram kelas dapat menampilkan tampilan paket dan mungkin berisi simbol untuk sekumpulan paket

*Class diagram* menunjukkan presentasi grafis dari tampilan statis. Biasanya beberapa *class diagram* diperlukan untuk menampilkan seluruh tampilan statis. *Class diagram* individu tidak perlu menunjukkan pembagian dalam model yang mendasarinya, meskipun di-visi logis, seperti paket, adalah batas alami untuk membentuk diagram. (Rumbaugh , et al., 2014)

## 2.5 Penelitian Sebelumnya

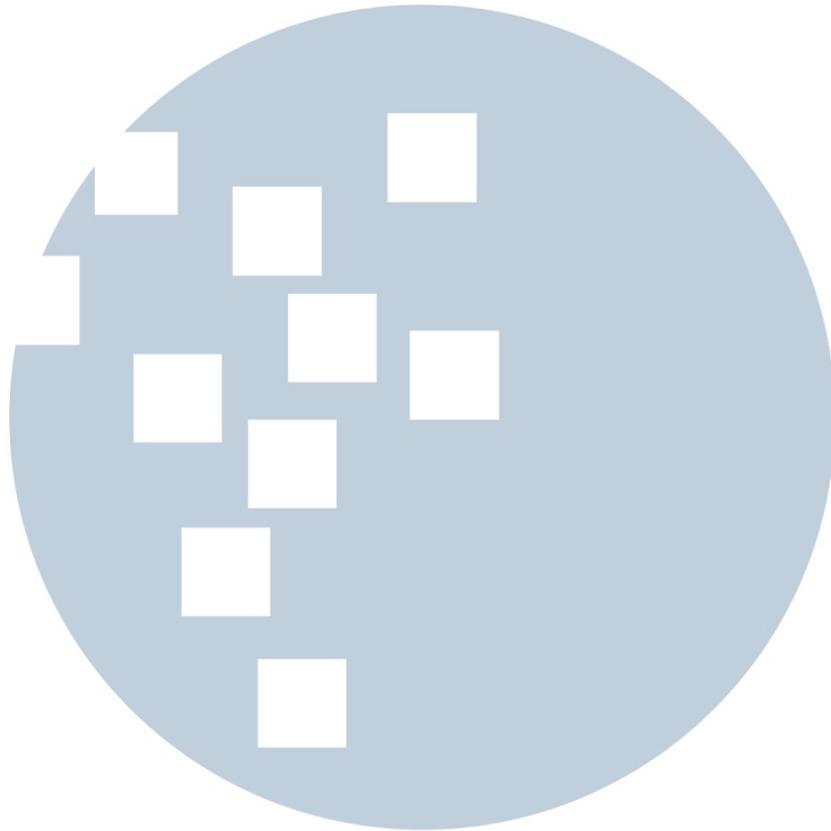
Tabel 2.3 Penelitian Sebelumnya

No.	Nama	Judul	Metode	Nama Jurnal	Hasil Penelitian	Kesimpulan
1.	Pardede, 2013	Sistem Pendukung Keputusan Pemilihan Kendaraan Bermotor Dengan Menggunakan Metode TOPSIS	TOPSIS	Pelita Informatika Budi Darma, Volume : IV, Nomor : 1, Agustus 2013	TOPSIS mampu membantu pemilihan sepeda motor berdasarkan kriteria yang dipilih pengguna	<i>Output</i> berupa data visualisasi
2.	Hastuti, 2013	Sistem Pendukung Keputusan Dalam Menentukan Pembelian Jenis Mobil Dengan Menggunakan Metode TOPSIS	TOPSIS	Pelita Informatika Budi Darma, Volume : IV, Nomor : 3, Agustus 2013	Memudahkan dalam mencari dan memecahkan masalah dalam memilih pembelian mobil dengan pengambilan keputusan multikriteria.	<i>Output</i> berupa data visualisasi
3.	Ariyono, et al, 2014	Analisis Faktor-Faktor Yang Mempengaruhi	AHP	Jurnal Pendidikan Ekonomi	Mengetahui faktor-faktor yang mempengaruhi	<i>Output</i> berupa data visualisasi

		i Keputusan Pembelian Sepeda Motor Matic Honda Pada Dealer Astra Motor Denpasar		Universitas Pendidikan Ganesha Vol. 4 No. 1	hi pembelian sepeda motor matic Honda di dealer Astra Motor Denpasar	
4.	Putra, et al, 2017	Sistem Pendukung Keputusan Pemilihan Sepeda Motor Jenis Sport 150cc Berbasis Web Menggunakan Metode Analytic Hierarchy Proses (AHP)	AHP	Jurnal TEKNOIF Vol. 5 No. 2	Menghasilkan aplikasi pendukung keputusan pemilihan sepeda motor sport 150cc	<i>Output</i> berupa aplikasi pendukung keputusan pemilihan sepeda motor sport 150cc berbasis AHP

Pada tabel 4.3 merupakan penelitian sebelumnya yang memiliki kemiripan dan korelasi dengan penelitian yang kini masih dalam tahap pengembangan dan dapat menjadi referensi dan pengembangan sistem lanjutan agar lebih baik dan lebih kompleks.





UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA