

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **1.1 Toko Elis Collection**

Toko Elis Collection merupakan toko yang bergerak dalam bidang penjualan barang keperluan rumah tangga seperti alat makan, blender, pisau dan lain-lain, berdiri sejak tahun 2017 Toko Elis Collection memiliki angka penjualan yang sangat tinggi. Selain itu Toko Elis Collection seiring berjalannya waktu, semakin menambah daftar barang yang dijual mulai dari perabotan rumah tangga, alat tulis, bahkan peralatan elektronik. Lokasi Toko Elis Collection sangat strategis menjadikan toko tersebut mudah dikunjungi yaitu berada di Jalan Alamanda Kelurahan Tugu Kecamatan Cimanggis Depok. Jam operasional Toko Elis Collection buka selama 12 jam yaitu mulai dari jam 08.00 – 20.00 WIB (Senin-Minggu).

Pada Toko Elis Collection juga terdapat Visi dan Misi dari Elis Collection yaitu:

Dengan semangat dan kekuatan dari Team Work menghasilkan produk berkualitas serta memberikan pelayanan yang terbaik sehingga menjadi pemimpin industri Elis Collection yang unggul dan berkualitas dengan cara:

1. Menghasilkan produk furnitur berkualitas tinggi melalui penerapan *quality assurance* (QC).
2. Mengadopsi dan mengimplementasikan perlengkapan dan teknologi produksi terkini.
3. Mengelola dan mengembangkan Sumber Daya Manusia yang Profesional.
4. Menjalankan kerjasama sebagai mitra bisnis dari para business partner untuk menghasilkan keuntungan bersama jangka panjang.
5. Berinovasi menghasilkan layanan purna jual terbaik.

Toko Elis Collection terdapat struktur organisasi yang dimana berisikan anggota yang terlibat dalam membangun Toko Elis Collection, berikut ini struktur organisasi beserta penjelasannya dapat dilihat pada Gambar 2.1



Gambar 2. 1 Struktur Organisasi Elis Collection

Keterangan dari masing-masing bagian berdasarkan Gambar 2.1 sebagai berikut

**1. Kepala Operasional**

Mengelola dan memantau operasional pelayanan sehari-hari. Melakukan pengawasan terhadap operasional pelayanan serta pemberian solusi yang terbaik sesuai dengan prosedur yang berlaku untuk menjaga kepercayaan dari konsumen

**2. Bagian Keuangan**

- Melakukan pengaturan keuangan perusahaan.
- Melakukan penginputan semua transaksi keuangan ke dalam program.
- Melakukan transaksi keuangan perusahaan.
- Melakukan pembayaran kepada supplier.
- Berhubungan dengan pihak internal maupun eksternal terkait dengan aktivitas keuangan perusahaan.

**3. Kasir**

- Melayani proses transaksi
- Memberikan informasi yang dibutuhkan pelanggan
- Melayani packaging barang pelanggan
- Memeriksa dan mencatat data penjualan
- Memeriksa daftar harga produk

- Memeriksa stok barang
- Membuat laporan penjualan
- Pelajari cara pembayaran

#### **4. Bagian Perencanaan**

Melaksanakan penyiapan koordinasi dan penyusunan rencana, dan anggaran, keuangan, perbendaharaan, verifikasi, akuntansi, evaluasi, dan pelaporan

#### **5. Bagian Penjualan & Pramuniaga**

- Melakukan penjualan dengan mencari klien atau pelanggan dengan aktif untuk mendapatkan pendapatan perusahaan.
- Melakukan analisa pelanggan untuk mencari tahu apa yang dibutuhkan mereka.
- Menjalin komunikasi kepada pelanggan untuk menjaga hubungan baik.

#### **6. Bagian Produksi & QC**

Bertanggung jawab dan mengawasi pelaksanaan proses produksi, mulai dari datangnya barang sampai barang diterima oleh Toko. Menjaga dan mengawasi agar kualitas barang dalam proses dan mutu barang jadi sesuai dengan standar yang telah ditetapkan perusahaan.

## **1.2 Aplikasi**

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi dengan tujuan untuk mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data atau diharapkan pada pemrosesan data. Aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya atau perangkat computer yang siap dipakai untuk user [7].

Aplikasi terbagi menjadi 3 jenis yakni, Aplikasi desktop, web dan *mobile*. Penjelasan dari jenis aplikasi yaitu:

### **1. Aplikasi *Desktop***

Merupakan aplikasi yang dapat dijalankan pada perangkat desktop yang perlu dilakukan install terlebih dahulu. Aplikasi *Desktop* adalah suatu aplikasi yang berjalan sendiri atau independen tanpa menggunakan browser atau koneksi internet disuatu komputer otonom. Aplikasi berbasis desktop merupakan aplikasi yang dijalankan pada masing-masing komputer

atau klien. Aplikasi berbasis desktop harus diinstall terlebih dahulu ke dalam komputer agar dapat digunakan [8].

2. Aplikasi *Web* merupakan aplikasi yang dapat dijalankan pada perangkat apapun. Untuk menjalankannya tidak perlu install dapat diakses melalui *browser* sehingga membutuhkan koneksi internet. Aplikasi web adalah suatu sistem informasi yang mendukung interaksi dengan pengguna melalui antarmuka berbasis web. Interaksi pengguna dengan web dibagi ke dalam tiga tahap, yaitu permintaan, pemrosesan, dan jawaban. Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi berupa teks, gambar, animasi, suara, ataupun gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman [9].
3. Aplikasi *Mobile* merupakan aplikasi yang dapat dijalankan pada *mobile*. Dan perlu dilakukan penginstalan terlebih dahulu pada perangkat. Aplikasi mobile adalah sebuah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau handphone. Dengan menggunakan aplikasi mobile, maka dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan Kantor, browsing dan lain sebagainya. Beberapa penelitian juga sudah banyak yang menggunakan aplikasi mobile, baik itu untuk hiburan, mempermudah dalam layanan komunikasi data, maupun sebagai pengendali alat kamera DSLR. Aplikasi mobile dibangun dengan beberapa bahasa pemrograman mobile. Adapun contoh dari mobile programming untuk ponsel diantaranya adalah Javafx mobile, J2ME, C++, C#.NET, Flash Lite, dan Kotlin [10].

### **1.3 E-Commerce**

*E-commerce* (perdagangan elektronik) adalah cara untuk menjual dan membeli barang dan jasa lewat jaringan internet. *E-commerce* bukan sebuah jasa atau sebuah barang, tetapi merupakan perpaduan antara jasa dan barang. *E-commerce* dan kegiatan yang terkait melalui internet dapat menjadi penggerak untuk memperbaiki ekonomi domestik melalui liberalisasi jasa domestik dan mempercepat integrasi dengan kegiatan produksi global[11].

Secara garis besar *e-commerce* adalah istilah yang digunakan untuk menggambarkan penjualan barang dan jasa melalui Internet. Bisnis *e-commerce* sekarang menawarkan toko *online*

di mana pelanggan dapat mengakses ribuan produk, pemesanan, pilih metode pengiriman yang diinginkan dan pembayaran dengan menggunakan atm, mobile-banking, atau kartu kredit mereka [11].

Jenis-jenis *E- Commerce* dibagi menjadi beberapa jenis sesuai dengan karakteristik yang berbeda yaitu sebagai berikut [11]:

1. *Business to Business (B2B)*

Karakteristik dari *Business to Business* yaitu *trading partners* yang sudah diketahui dan umumnya memiliki hubungan (*relationship*) yang cukup lama. Informasi hanya dipertukarkan dengan *partner* tersebut. Dikarenakan sudah mengenal lawan komunikasi, maka jenis informasi yang dikirimkan dapat disusun sesuai dengan kebutuhan dan kepercayaan (*trust*)

2. *Business to Consumer (B2C)*

Karakteristik dari *Business to Customer* yaitu terbuka untuk umum, dimana informasi disebarkan ke umum serta servis yang diberikan bersifat umum (*generic*) dengan mekanisme yang dapat digunakan oleh khalayak ramai. Sebagai contoh, karena sistem Web sudah umum digunakan maka servis diberikan dengan menggunakan basis Web.

3. Perdagangan Kolabratif (*Collaborative Commerce*)

Dalam *c-commerce*, para mitra bisnis berkolaborasi (alih-alih membeli atau menjual) secara elektronik. Kolaborasi semacam ini seringkali terjadi antara dan dalam mitra bisnis di sepanjang rantai pasokan.

4. *Comsumen to Consumen(C2C)*

Dalam C2C seseorang menjual produk atau jasa ke orang lain. Dapat juga disebut sebagai pelanggan ke palanggan yaitu orang yang menjual produk dan jasa ke satu sama lain contohnya bisnis Lelang C2C.

5. *Consumen to Business (C2B)*

Dalam C2B konsumen memeritahukan kebutuhan atas suatu produk atau jasa tertentu, dan para pemasok bersaing untuk menyediakan produk atau jasa tersebut ke konsumen. Contohnya dipriceline.com, dimana pelanggan menyebutkan produk dan harga yang diinginkan, dan *priceline* mencoba menemukan pemasok yang memenuhi kebutuhan tersebut.

6. Perdagangan Intrabisnis (*Intraorganisasional*)

Dalam situasi ini perusahaan menggunakan *e-commerce* secara internal untuk memperbaiki operasinya. Kondisi khusus dalam hal ini disebut sebagai *e-commerce* B2E (*business to its employees*).

7. Pemerintah keWarga (*Government to Citizen—G2C*)

Dalam kondisi ini sebuah entitas (*unit*) pemerintah menyediakan layanan ke para warganya melalui teknologi *Ecommerce*. Unit-unit pemerintah dapat melakukan bisnis dengan berbagai unit pemerintah lainnya serta dengan berbagai perusahaan(G2B). *Egovernment* yaitu penggunaan teknologi internet secara umum dan *e-commerce* secara khusus untuk mengirimkan informasi dan layanan publik ke warga, mitra bisnis, dan pemasok entitas pemerintah, serta mereka yang bekerja di sektor publik.

8. Perdagangan Mobile (*mobile commerce-m-commerce*)

Ketika *e-commerce* dilakukan dalam lingkungan nirkabel, seperti dengan menggunakan telepon seluler untuk mengakses internet dan berbelanja, maka hal ini disebut *m-commerce*.

#### 1.4 Rekomendasi Produk

Rekomendasi adalah istilah yang merujuk pada suatu saran atau anjuran dari seseorang terhadap orang lain. Saran atau anjuran ini bisa berupa rekomendasi untuk melakukan sesuatu ataupun rekomendasi dalam menentukan pilihan. Rekomendasi timbul karena terdapat permasalahan yang perlu ada tindakan-tindakan yang perlu dilakukan dari rencana pengambilan tindakan alternatif-alternatif yang harus dipilih. Selain itu rekomendasi di dukung dengan sumber daya yang ada dan harus dimanfaatkan serta data dan informasi yang harus diolah untuk dimanfaatkan serta yang paling penting rekomendasi itu memberikan dampak yang lebih baik [12].

Produk adalah sesuatu yang diperoleh, dibuat, dikembangkan, dijual dan dimanfaatkan atau dimiliki untuk memenuhi keinginan atau kebutuhan sehingga dipasarkan agar mencapai tujuan yang diharapkan [13]. Pada produk terbagi menjadi beberapa kelompok, diantaranya: produk berdasarkan ketahanannya dan keberwujudannya, produk konsumen, dan produk industry. Dimana untuk produk berdasarkan ketahanan dan keberwujudnya produk terdiri dari barang-barang yang tidak tahan lama, barang tahan lama, dan jasa. Sedangkan untuk produk konsumen terdiri dari barang sehari-hari, barang belaja, dan barang khusus. Dan terakhir produk industry terdiri dari barang dan suku cadang, barang modal, dan layanan bisnis dan pasokan [14].

Jadi, Rekomendasi produk adalah bagian dari strategi personalisasi pada perdagangan di mana produk secara dinamis diisi ke pengguna di halaman web, aplikasi, atau email berdasarkan data seperti atribut pelanggan, perilaku penjelajahan, atau konteks situasional yang memberikan pengalaman belanja yang dipersonalisasi. Rekomendasi produk sangat berharga bagi organisasi yang memiliki katalog produk yang sangat besar dan beragam [15].

## 1.5 Metode Apriori

Apriori adalah salah satu pendekatan yang sering digunakan pada *Frequent Itemset Mining*. Prinsip algoritma Apriori adalah jika sebuah *itemset infrequent*, maka *itemset* yang *infrequent* tidak perlu lagi dieksplor *superset* sehingga jumlah kandidat yang harus diperiksa menjadi berkurang, berikut adalah ilustrasinya. Langkah-langkah dalam algoritma apriori adalah sebagai berikut [16]:

1. Tentukan *minimum support* (nilai *minimum* dari kombinasi item).
2. Pembentukan kandidat itemset ( $C_k$ ) dibangun dengan menggabungkan  $L_{k-1}$  dengan dirinya sendiri.
3. Pemangkasan kandidat itemset dilakukan setiap  $C_k$ -*itemset* yang tidak sering muncul (*not frequent*) tidak dapat menjadi subset dari *frequent*-*itemset*.
4. Menentukan support dari kombinasi *k-itemset* sebelumnya. *Itemset* yang memenuhi *minimum support* akan dipilih sebagai pola *frequent* tinggi ( $L_k$ ) dari kandidat.
5. Tetapkan nilai *k-itemset* dari *support* yang telah memenuhi *minimum support* dari *k-itemset*.
6. Jika sudah tidak ada lagi *k-itemset* yang memenuhi *minimum support* maka proses dihentikan

Algoritma apriori adalah salah satu algoritma pengambilan data dengan aturan asosiasi (*Association rule*) untuk menentukan hubungan asosiasi suatu kombinasi item. Aturan yang menyatakan asosiasi terhadap beberapa atribut seringkali disebut *market basket analysis* atau *affinity analysis* [17].

*Market Basket Analysis* digunakan untuk mengetahui produk yang dibeli secara bersamaan oleh pelanggan agar dapat meningkatkan taktik pemasaran dan penjualan di tingkat toko. *Market Basket Analysis* termasuk katgori *accosiation rule* [16]. Penting tidaknya suatu asosiasi dapat diketahui dengan dua tolak ukur, yaitu: *support* dan *confidence*. *Support* (nilai penunjang) adalah persentasi kombinasi item tersebut dalam *database*, sedangkan *confidence* (nilai kepastian) adalah kuatnya suatu hubungan antar-item di dalam aturan asosiasi [18].

Berikut terdapat persamaan yang digunakan untuk mencari nilai dari *support* dan *confidence* dalam *association rule* dengan rumus persamaan 2.1 sampai dengan persamaan 2.3 [18].

1. *Support*, bagian transaksi yang mengandung sebuah item.

$$support(A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ transaksi} \times 100\% \quad (2.1)$$

Jika terdapat dua buah item menggunakan persamaan 2.

$$support(A) = \frac{\Sigma\ transaksi\ mengandung\ A\&B}{Total\ transaksi} \times 100\% \quad (2.2)$$

2. *Confidence*, seberapa sering item dalam B muncul di transaksi yang mengandung A.

$$confidence = \frac{\Sigma\ transaksi\ mengandung\ A\ dan\ B}{\Sigma\ Transaks\ mengandung\ A} \times 100\% \quad (2.3)$$

Untuk menentukan aturan asosiasi yang akan dipilih maka harus diurutkan berdasarkan *support x confidence*, dimana aturan diambil sebanyak n aturan yang memiliki hasil terbesar [20].

*Lift ratio* merupakan nilai yang menunjukkan kevalidan proses transaksi dan memberikan informasi apakah benar item A dibeli bersamaan item B, dengan persamaan 2.4 [16].

$$Nilai\ lift = \frac{Support(A \cap B)}{Support(A) \times Support(B)} \quad (2.4)$$

Sebuah transaksi dikatakan valid, jika mempunyai nilai *lift ratio* lebih dari 1, yang berarti bahwa dalam transaksi tersebut item A dan item B benar-benar dibeli secara bersamaan [16].

## 1.6 Metode *Content Based Filtering*

Metode *content-based filtering* merupakan metode untuk merekomendasikan item yang mirip dengan item sebelumnya yang disukai atau dipilih oleh pengguna. Kemiripan item dihitung berdasarkan pada fitur-fitur yang ada pada item yang dibandingkan. Metode ini bersifat *user independence*, tidak bergantung pada situasi apakah item tersebut merupakan item baru (yang belum pernah dipilih oleh pengguna manapun) maupun bukan item baru [19]. Langkah-langkah dalam metode *content-based filtering* adalah [4]:

1. Berlandaskan sebuah vector unsur pembuatnya, sebuah benda dipisah-pisah.
2. Berlandaskan bobot vector unsur pembuat sebuah benda, sistem nantinya menciptakan profil pemakai. Penciptaan profil pemakai dilaksanakan memanfaatkan algoritma *Term Frequency-Invers Document Frequency* (TF-IDF). Jumlah term pada sebuah naskah merupakan pengertian TF, sementara itu nilai IDF dapat dihitung memakai persamaan 2.4.



$$idf_i = \log\left(\frac{n}{df_i}\right) \quad (2.4)$$

Keterangan:

$df$  = Jumlah naskah yang mempunyai *term i*

$n$  = Jumlah seluruh naskah

3. Berlandaskan profil pemakai itu, sistem kemudian menduga evaluasi gemar atau tidak digemarinya sebuah benda berlandaskan uraian keserupaan profil pemakai dengan vector unsur pembuat benda. Seandainya sistem menduga bahwa benda itu bakal digemari oleh pemakai, lalu benda itu nantinya diusulkan ke pemakai.

Metode ini bersifat *userindependence*, tidak bergantung pada situasi apakah item tersebut merupakan item baru (yang belum pernah dipilih oleh pengguna manapun) maupun bukan item baru. Jika seorang pengguna telah memesan suatu menu hidangan pada kategori tertentu maka sistem akan mencoba merekomendasikan menu hidangan dengan kategori serupa yang juga tersedia di restoran lain yang mungkin akan disukai juga oleh pengguna tersebut [19]

Contoh kasus:

Dalam penelitian yang dilakukan oleh Wijaya dan Alfian tahun 2018, membahas terkait sistem rekomendasi laptop yang juga menggunakan *content-based filtering*. Jenis laptop yang mempunyai beragam merk dan spesifikasi yang terkadang membuat orang-orang mengalami kesusahan dan kesulitan dalam mencari, memilih ataupun membeli laptop yang tepat dan sesuai bagi kebutuhannya, tentunya orang-orang akan terbantu jika ada yang dapat merekomendasikan laptop yang sesuai bagi kebutuhannya. Untuk mendapatkan data konten dari suatu item atau pun profil pengguna bisa menggunakan algoritma TF-IDF dalam pemrosesan datanya dalam mendapatkan nilai bobot pada tiap-tiap laptop [20]. Tahapannya sebagai berikut:

1. Menentukan *query term*

Sebagai contoh misalnya user 1 ingin rekomendasi laptop dengan spesifikasi yang disukainya terlihat pada Tabel 2.1

Tabel 2. 1 Spekulasi user  
 Sumber : Wijaya & Alfian [20]

Brand	apple, asus
Series	Aspire,MacBook-Air,MacBook-Pro
OS	Linux Mac-OS-X
Tahun Rilis	2016
Ukuran layar	14 inch
Resolusi layar	1366 x 768 pixel
Processor	Intel Core i5
Kecepatan	2.2 Ghz
RAM	4 GB
Storage	512 GB
Harga	6149000

Dari Tabel 2.1 data dari spekulasi laptop yang user sukai digabungkan untuk menjadi sebuah *query*, untuk pembuatan dokumennya hanya spesifikasi laptopnya saja yang diambil dan digabungkan untuk menjadi dokumen yang terlihat pada Tabel 2.2

Tabel 2. 2 Dokumen Data Laptop  
 Sumber : Wijaya & Alfian [20]

Doc	Isi dokumen
Q	apple,asus Aspire,MacBook-Air,MacBook-Pro Linux, Mac-OS-X 2016 14 inch 1366 x 768 Intel-Core-i5 2.2 Ghz 4 gb 512 gb 6149000
1	<b>Nama : Aspire E5-475G   Core i5-7200U</b>

Doc	Isi dokumen
	Spesifikasi: acer aspire DOS 2015 14 inch 1366 x 768 pixel Intel Core i5 2.5 Ghz 4 gb 1000 gb 6149000
Doc	Isi dokumen
2	<b>Nama : Acer Aspire ES1-432- C56Y / C5GA / C52R</b> Spesifikasi: acer aspire linux 2016 14 inch 1366 x 768 pixel Intel Celeron 1.1 Ghz 2 gb 500 gb 3150000
3	<b>Nama : Apple Macbook Pro MF839 Retina</b> Spesifikasi: apple MacBook Pro Mac OS X 2014 13 inch 2560 x 1600 pixel Intel Core i5 2.7 Ghz 8 gb 128 gb 14454000
4	<b>Nama : Apple MacBook Air MMGG2</b> Spesifikasi: apple MacBook Air Mac OS X 2016 13 inch 1440 x 900 pixel Intel Core i7 2.2 Ghz 8 gb 512 gb 13954600
5	<b>Nama : Asus X441SA-BX001D / BX002D / BX003D / BX004D</b> Spesifikasi: asus x series DOS 2016 14 inch 1366 x 768 pixel Intel Celeron 2.48 Ghz 2 gb 500 gb 3185000
6	<b>Nama : Asus A455LA-WX667D / WX668D / WX669D</b> Spesifikasi: asus x series DOS 2016 14 inch 1366 x 768 pixel Intel Core i3 2 Ghz 4 gb 500 gb 4725000

## 2. Perhitungan bobot TF-IDF

Dalam menghitung bobot TF-IDF dipakai rumus sebagai berikut:

$$IDF = \frac{D}{DF}$$

$$W = TF \times (IDF + 1)$$

Keterangan:

W = bobot setiap dokumen

TF = jumlah kemunculan kata atau term dalam dokumen

D = Jumlah semua dokumen

DF = Jumlah dokumen yang mengandung kata (term)

IDF = *inverse document frequency*

Tabel 2.3 berikut adalah perhitungan untuk TF, DF, dan IDF

Tabel 2. 3 Perhitungan TF, DF, dan IDF

Sumber : Wijaya & Alfian [20]

Query	TF						DF	D/DF	IDF	IDF + 1
	D1	D2	D3	D4	D5	D6				
Apple			1	1			2	3	0.477121	1.477121
Asus					1	1	2	3	0.477121	1.477121
Aspire	1	1					2	3	0.477121	1.477121
Macbook air				1			1	6	0.778151	1.778151
Macbook pro			1				1	6	0.778151	1.778151
Linux		1					1	6	0.778151	1.778151
MAC-OS-LINUX			1	1			2	3	0.477121	1.477121
2016		1		1	1	1	4	1.5	0.176091	1.176091
Intel Core i5	1		1				2	3	0.477121	1.477121
14 inch	1	1			1	1	4	1.5	0.176091	1.176091
1366 x 766	1	1			1	1	4	1.5	0.176091	1.176091
2.2 Ghz				1			1	6	0.778151	1.778151
4 GB	1					1	2	3	0.477121	1.477121
512 GB				1			1	6	0.778151	1.778151
6149000	1						1	6	0.778151	1.778151

Selanjutnya menghitung nilai bobot TF-IDF yang terlihat pada Tabel 2.4

Tabel 2. 4 Bobot TF-IDF

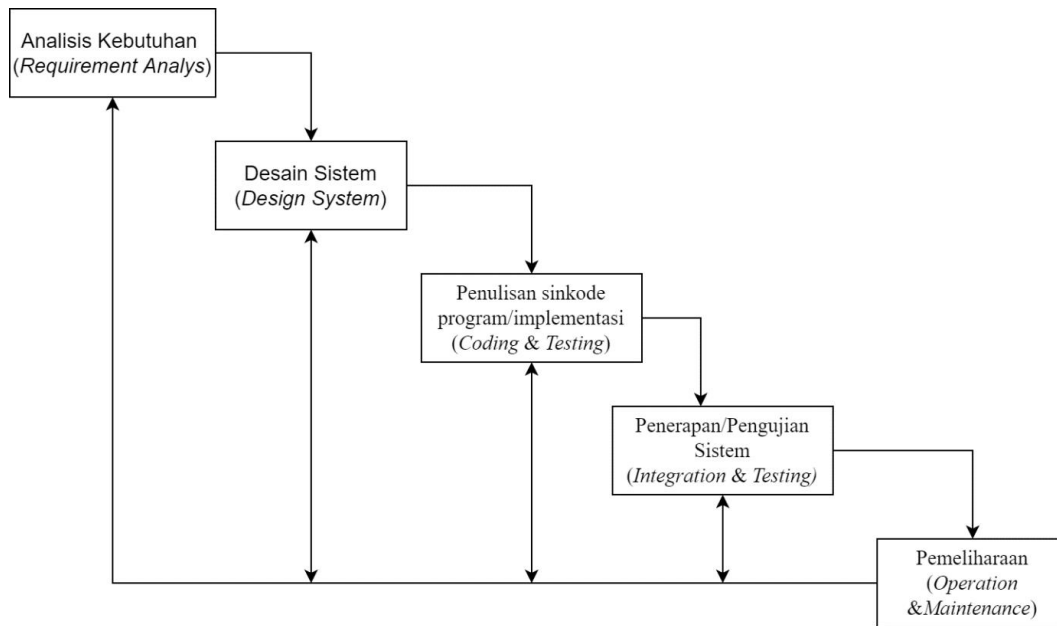
Sumber : Wijaya & Alfian [20]

<b>W = TF * IDF + 1</b>					
D1	D2	D3	D4	D5	D6
0	0	1.47712155	1.477121	0	0
0	0	0	0	1.477121	1.477121
1.47712155	1.477121	0	0	0	0
0	0	0	1.778151	0	0
0	0	1.77815125	0	0	0
0	1.778151	0	0	0	0
0	1.176091	0	0	0	0
0	1.176091	0	1.176091	1.176091	1.176091
0	0	1.477121255	1.477121	0	0
1.477121255	0	1.477121255	0	0	0
1.176091259	1.176091	0	0	1.176091	1.176091
1.176091259	1.176091	0	0	1.176091	1.176091
0	0	0	1.778151	0	0
1.477121255	0	0	0	0	1.477121
0	0	0	1.778151	0	0
1.77815125	0	0	0	0	0
<b>Jumlah bobot setiap dokumen</b>					
8.561697533	6.783546	6.209515015	9.464788	5.005395	6.482516

Hasil dari perhitungan bobot diatas diketahui bahwa bobot tertinggi terdapat pada D4 bernilai 9.464788, sehingga rekomendasi laptop untuk user 1 adalah D4 (Nama: Apple MacBook Air MMGG2 Spesifikasi: apple|MacBook Air|Mac OS X|2016|13 inch|1440 x 900 pixel|Intel Core i7|2.2 Ghz|8 gb|512 gb|13954600) [20].

### **1.7 Metode Waterfall**

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Berikut ini terdapat Gambar 2.1 yang merupakan tahapan-tahapan dalam metode *waterfall*, diantaranya yaitu [21]:



Gambar 2. 2 Metode Waterfall

1. *Requirment* (Analisis Kebutuhan)

Dalam langakah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan system analisis untuk menterjemahkan kedalam bahasa pemrograman.

2. *Design System* (Desain Sistem)

Proses *design* akan menterjemahkan syarat kebutuhan kesebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan *programmer* untuk melakukan aktivitas pembuatan sistemnya.

3. *Coding & Testing* (Penulisan sinkode program/implementasi)

*Coding* merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan meterjemahkan transaksi yang diminta oleh *user*. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap *system* tersebut dan kemudian bisa diperbaiki.

4. *Integration & Testing* (Penerapan/Pengujian Sistem)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi digunakan oleh *user*.

#### 5. *Operation & Maintenance* (Pemeliharaan)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (*peripheral* atau *system* operasi) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

### 1.8 *Website*

Website adalah sering juga disebut web, dapat diartikan suatu sekumpulan halaman yang menampilkan berbagai macam informasi baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu alur kerja yang saling berkaitan dimana masing-masing alur dihubungkan dengan jaringan halaman [22]. Website adalah sejumlah halaman website yang didalamnya terdapat topik saling terkait antara halaman yang satu dengan halaman lain, yang biasanya ditempatkan pada sebuah web-server yang dapat diakses melalui jaringan internet ataupun jaringan intranet [23].

Website merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing masing dihubungkan dengan jaringan-jaringan halaman [24].

### 1.9 *Android*

Android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux, dan menjadi salah satu sistem operasi atau *operating system* berbasis *mobile* yang sangat banyak di gunakan sekarang ini. sistem operasi ini dirancang oleh Google dengan basis *kernel* Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau *smartphone*. Jadi, android digunakan dengan sentuhan, gesekan ataupun ketukan pada layar gadget. Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan OS ini di perangkatnya tanpa lisensi atau gratis [25].

Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini.OS lainnya seperti Windows Mobile, iPhone OS, Symbian, dan masih banyak lagi.Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka [26].

Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [27].

### **1.10 PHP**

PHP merupakan singkatan dari “Hypertext Preprocessor”. PHP adalah sebuah bahasa pemrograman yang terpasang pada HTML. Bahasa pemrograman PHP ini sendiri digunakan untuk membuat tampilan web menjadi lebih dinamis, dengan bahasa pemrograman PHP bisa menjalankan beberapa file dalam satu folder dengan cara di include dan require. PHP itu sendiri sudah dapat berinteraksi dengan beberapa database walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle [28].

PHP adalah bahasa server-sidescripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari server-side scripting adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada dokumen HTML. Pembuatan web ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemrograman dan HTML sebagai pembangun halaman web. Ketika seorang pengguna internet akan membuka suatu situs yang menggunakan fasilitas server-sidescripting PHP, maka terlebih dahulu server yang bersangkutan akan memproses format HTML ke web browser pengguna internet. Dengan demikian seorang pengguna internet tidak dapat melihat kode program yang ditulis dalam PHP sehingga keamanan dari halaman web menjadi lebih terjamin [26].



Fungsi PHP adalah membuat atau mengembangkan situs web statis atau situs web dinamis. PHP juga digunakan untuk menyimpan data ke dalam database, serta membuat halaman yang dapat berubah-ubah sesuai dengan input program yang akan dibuat. Dan membuat file PHP dapat digabung menggunakan tag html, dan ketika tanpa menggunakan tag html biasanya disebut file PHP murni [29].

### **1.10.1 Framework Laravel**

Laravel adalah sebuah framework PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep *MVC (model view controller)*. Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu [30].

Pengertian framework adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, framework adalah wadah atau kerangka kerja dari sebuah website yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat website lebih singkat dan memudahkan dalam melakukan perbaikan [31].

Laravel sendiri diciptakan oleh Taylor Otwell, dia memulai proyek laravel , dia memulai proyek laravel sejak tahun 2011 dibulan april. Dimana proyek tersebut didasari karena Otwell tidak menemukan *framework* yang mempunyai kelebihan update sejalan dengan versi PHP. Oleh karena itu otwell membuat *framework* laravel dengan memberi ketentuan yaitu PHP versi 5.4 [32].

### **1.11 Javascript**

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengekseskuan perintah perintah di sisi user, yang artinya di sisi browser bukan di sisi server web. Javascript bergantung kepada browser (navigator) yang memanggil halaman web yang berisi skrip-skrip dari Javascript dan tentu saja terselip di dalam

dokumen HTML. Javascript memiliki beberapa *framework* salah satunya yaitu React yang digunakan dalam penelitian ini [33].

Adapun fungsi Java script itu antara lain: membuat website interaktif, kemudian mengembangkan aplikasi mobile, lalu menciptakan game berbasis web, serta menjalankan web server. Selain itu java script memiliki kelebihan salah satunya mudah dipelajari, interaktif dan responsif, dan sebagainya [34].

### **1.11.1 *React Native***

React Native adalah framework open source untuk membuat aplikasi multi-platform (Android dan iOS) dengan bahasa Java Script, sesuai dengan deskripsi di situs resminya “Learn once, write anywhere”. React Native merupakan framework open source besutan Facebook yang dibuat setelah Facebook sebelumnya membuat ReactJS, ReactJS sendiri merupakan sebuah library dari Facebook yang dapat digunakan untuk membangun antar muka pengguna atau user interface [26].

React Native adalah sebuah Framework dari Javascript yang banyak dikembangkan oleh perusahaan besar dalam bidang teknologi seperti Facebook. Jadi, dengan React Native kamu tidak perlu membuat aplikasi Hybrid. Selain itu, React Native mampu mengkompilasi aplikasi ke dalam sebuah Native Code pada Android maupun iOS [35].

## **1.12 *MySQL***

Definisi MySQL merupakan software RDMS (Relational Database Management System) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak pengguna dan dapat melakukan suatu proses secara sinkron atau bersamaan [30]. MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-tabel yang saling berhubungan. Keuntungan menyimpan data di database adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk tabel [22].

MySQL adalah RDBMS yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersil. MySQL sebenarnya merupakan turunan salah satu konsep utama database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah

sebuah konsep pengoperasian database terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis






### 1.13 *Unified Modeling Language (UML)*

*Unified Modeling Language (UML)* merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek [36]. UML memiliki banyak diagram diantaranya:

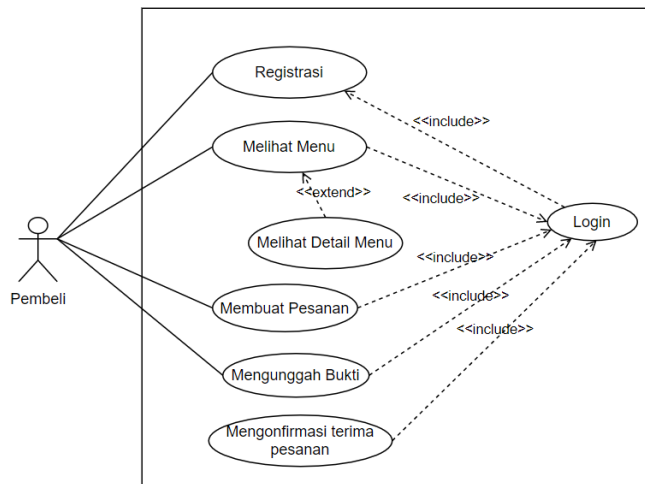
#### 1. *Use Case Diagram*

*Use Case Diagram* merupakan diagram yang harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan [36]. Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut ini terdapat simbol-simbol *use case* yang dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol-Simbol Use Case Diagram

Simbol	Nama	Keterangan
	Aktor	Merupakan Penggunaan dari sistem. Penamaan aktor menggunakan kata benda.
	<i>Use case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja.
	Assosiasi	Hubungan antara aktor dengan <i>use case</i> .
	<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
	<i>Extends</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

Gambar 2.3 merupakan contoh *Use Case Diagram*




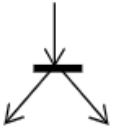





Gambar 2. 3 Contoh Use Case Diagram

## 2. Activity Diagram

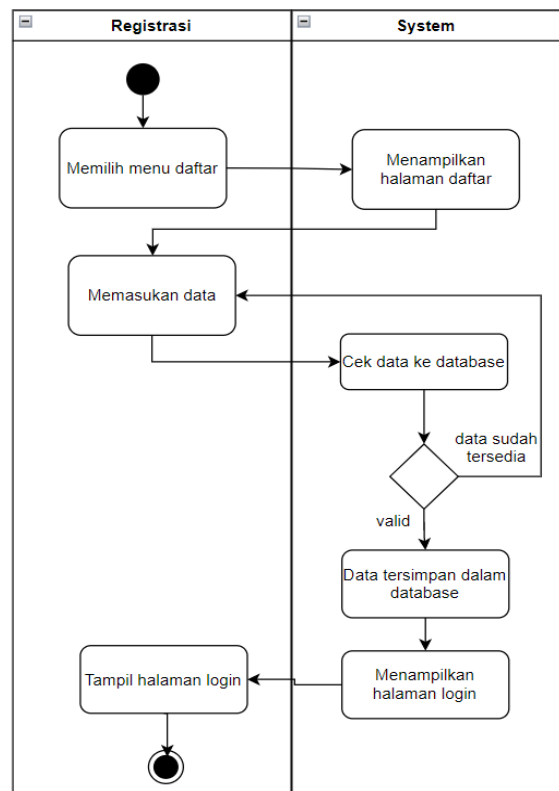
*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis [36]. Diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut ini terdapat simbol-simbol *activity diagram* yang dapat dilihat pada Tabel 2.6.

Tabel 2. 6 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Start Point</i>	Diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i>	Akhir aktivitas
	<i>Activities</i>	Menggambarkan suatu proses/ kegiatan bisnis
	<i>Fork / Percabangan</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join (Penggabungan) / Rake</i>	Digunakan untuk menunjukkan adanya dekomposisi.

Simbol	Nama	Keterangan
	<i>Decision Point</i>	Menggambarkan pilihan untuk pengambilan keputusan, <i>True</i> dan <i>False</i> .
	<i>Swimline</i>	Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

Gambar 2.4 merupakan contoh *Activity Diagram*


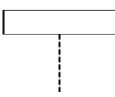

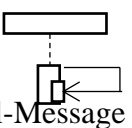


Gambar 2. 4 Contoh Activity Diagram

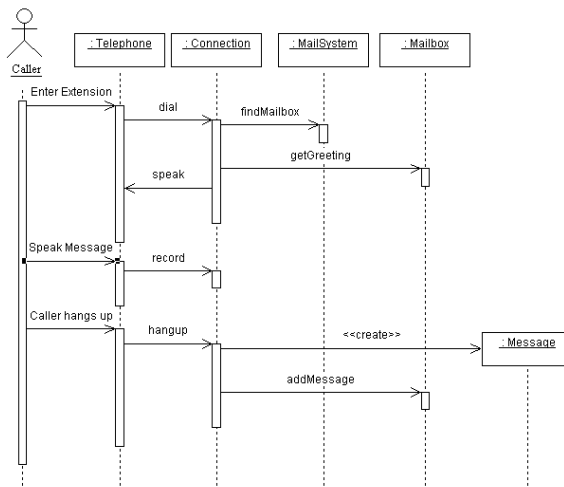
### 3. *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna dan *display*) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Setiap pesan yang dikirimkan bisa memberikan respon (return) relatif pada skenario yang dirancang di usecase diagram. Interaksi yang terjadi bisa bersifat instansiasi sebuah objek maupun static method dari sebuah class. Berikut ini adalah simbol-simbol dari *sequence diagram* dijelaskan pada tabel 2.7.

Tabel 2. 7 Simbol *Sequence Diagram*

No	Gambar	Keterangan
1.	 Aktor	Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary system</i>
2.	 Lifeline	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)
3.	 Message	Menunjukkan aliran informasi atau kendali transaksi antar elemen
4.	 Sel-Message	Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i> . Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i>

Gambar 2.5 merupakan contoh *sequence diagram*

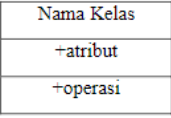








Gambar 2. 5 Contoh Sequence Diagram

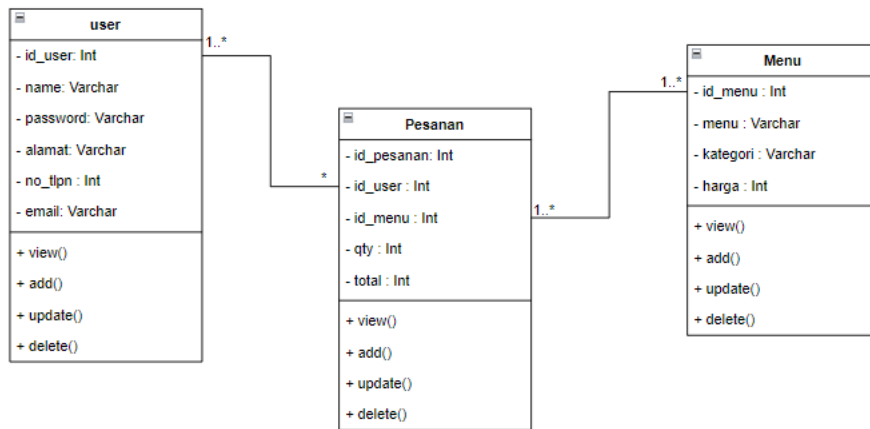
#### 4. *Class Diagram*

*Class diagram* menjelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuklah suatu sistem aplikasi [36]. Berikut ini terdapat simbol-simbol *class diagram* yang dapat dilihat pada Tabel 2.8.

Tabel 2. 8 Simbol-Simbol *Class Diagram*

Simbol	Nama	Keterangan
	Kelas	Kelas pada struktur sistem.
	Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
	Asosiasi Berarah/ <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
	Kebergantungan	Relasi antar kelas dengan makna ketergantungan antar kelas.
	Agresiasi/ <i>Aggregation</i>	Relasi antar kelas dengan makna semua-bagian.

Gambar 2.6 merupakan contoh *Class Diagram*



Gambar 2. 6 Contoh Class Diagram

### 1.14 Blackbox Testing

*Black box testing* merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan performansi, kesalahan inisialisasi dan terminasi. Pengujian Black Box bertumpu pada memastikan tiap proses sudah berfungsi sesuai dengan kebutuhan yang diharapkan. Penguji dapat menartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari sistem. Sehingga pengujian merupakan suatu cara pelaksanaan program yang bertujuan menemukan kesalahan atau error kemudian memperbaikinya sehingga sistem dapat dikatakan layak untuk digunakan [37].

Pengujian *Black Box* adalah pengujian yang memverifikasi hasil eksekusi aplikasi berdasarkan masukan yang diberikan (data uji) untuk memastikan fungsional dari aplikasi sudah sesuai dengan persyaratan (requirement). Pengujian *Black Box* ialah pengujian yang berfokus pada *interface* atau tampilan dan pengujian fungsional yang terdapat pada aplikasi, serta kesesuaian pada alur fungsi yang dibutuhkan oleh *users* [38]. Pengujian *Black Box* tidak menguji berdasarkan *source code* program. Tahapan dalam metode *blackbox testing* adalah sebagai berikut [38].

- a. Membuat *test case* untuk pengujian fungsifungsi yang terdapat di aplikasi
- b. Membuat *test case* untuk pengujian kesesuaian *flow* atau alur dari kerja suatu fungsi pada program cocok dengan apa yang dibutuhkan dan permintaan dari pengguna
- c. Mencari bugs/error berdasarkan tampilan (*interface*) pada aplikasi