

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Terhadap Penelitian Yang Terkait

Penerapan Sistem Penjualan Online Dengan *Collaborative Filtering* Pada Kuker Hesti.

Menghadirkan sebuah website yang menarik dan tidak terikat dengan teknologi tertentu sudah menjadi suatu kebutuhan yang harus terpenuhi dalam sebuah website. Trend internet untuk menjual cukup efektif dalam pasar yang berkembang dan telah dapat menciptakan suatu ide yang unik yang dapat diterapkan secara efektif. Internet dapat menjadi sarana bertukar informasi dengan siapa saja, kapanpun dan dimanapun berada, maka dari itu sangat di perlukan perdagangan elektronik yang aktivitasnya berkaitan dengan pembelian, penjualan, pemasaran barang ataupun jasa dengan memanfaatkan elektronik seperti internet ataupun jaringan komputer.

Semenjak terjadinya perkembangan tersebut, akhirnya membawa manusia pada fenomena penerimaan informasi yang lebih, atau bisa dikenal dengan *information overload*. Masalah ini tercapai ketika tercapai titik dimana informasi yang didapatkan oleh seseorang melebihi dari apa yang dibutuhkan untuk membuat suatu keputusan yaitu *recommender system*. *Recommender system* merupakan sebuah sistem yang dapat menyediakan rekomendasi, teknik yang digunakan dalam memberikan informasi dimana melibatkan kolaborasi beberapa pengguna dan beberapa sudut pandang adalah teknik *Collaborative Filtering*.

*Collaborative Filtering* memberikan rekomendasi kepada pengguna berdasarkan kemiripan selera dengan pengguna lain. Dengan *Collaborative Filtering* pelanggan bisa terbantu untuk memilih sebuah produk yang mereka inginkan. Sebuah perusahaan yang bisa bertahan lama tidak hanya mengandalkan kekuatan produk mereka saja, tetapi dengan adanya kerjasama tim manajemen yang handal, pengiriman barang yang tepat waktu, memberikan pelayanan yang bagus kepada calon konsumen, struktur organisasi bisnis yang baik, serta desain situs web yang baik dan mudah dioperasikan calon pembeli yakni seperti menyediakan jasa pembelian yang cepat, ramah kemudian informasi barang atau jasa yang lengkap dan jelas, memberikan perhatian khusus seperti rekomendasi pembelian dan menyediakan fitur untuk berdiskusi seperti masukan dari pelanggan, dan ini sangat mempermudah transaksi jual-beli secara online.

Tujuan dalam penelitiannya adalah *Collaborative filtering* merupakan salah satu algoritma yang digunakan untuk menyusun *recommender system* dan telah terbukti memberikan hasil yang sangat baik. Rating produk merupakan elemen terpenting dari algoritma ini, rating diperoleh dari sebagian besar customer di mana customer secara explicit memberikan penilaiannya terhadap produk. Kesimpulannya ialah system memberikan imbal balik kepada customer dengan mengolah data-data tersebut. (Retno dan Wahyu, 2019).

## **2.2. Sekilas Tentang Pemesanan**

Desain dan Implementasi Sistem Pemesanan Makanan dan Program Promosi Penjualan pada Mie Pangsit Ayam Sudi.

Pemesanan adalah aktivitas yang selalu dilakukan konsumen sebelum proses membeli. Agar dapat memberikan kepuasan kepada konsumen maka dari itu, tentunya perusahaan harus mempunyai sistem pemesanan yang baik, (Susanti dan Prabowo, 2014), Pemesanan juga merupakan proses, pembuatan, cara pemesanan kepada orang lain. Pemesanan dapat juga dikatakan sebagai memesan, pesanan, maupun permintaan dengan pembelian jasa ataupun barang kepada penjual. Hal ini biasanya dilakukan pada saat transaksi jual beli. Langkah-langkah pemesanan yakni melakukan kontak secara langsung dengan penjual dan konsumen akan memesan barang yang ingin dibeli. Setelah barang yang dipesan ada maka barulah konsumen melakukan pembayaran. (Enje Aprilla, 2020).

### **2.3. Sekilas Tentang Collaborative Filtering**

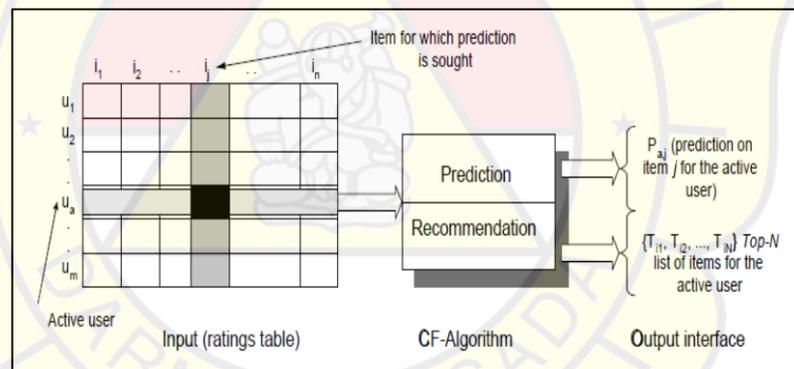
Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan Metode *User Based Collaborative Filtering*.

#### **2.3.1. Collaborative Filtering Sebagai Sistem Rekomendasi**

Sistem rekomendasi merupakan suatu sistem yang dapat memberikan informasi dan rekomendasi yang membantu *user* dalam membuat keputusan berdasarkan data yang telah ada sebelumnya. Salah satu metode yang dipergunakan dalam pembuatan suatu sistem rekomendasi yang terkenal hingga saat ini adalah *Collaborative Filtering*. Pada implementasi *Collaborative Filtering* dalam sistem rekomendasi mata kuliah pilihan, data yang dibandingkan akan disimpan dalam sebuah matriks dua dimensi dimana dimensi dari y atau baris akan menyatakan mahasiswa dan dimensi x atau kolom akan menyatakan mata kuliah.

### 2.3.2. Proses Collaborative Filtering

Ide dasar dari menghitung similarity antara mahasiswa  $U_1$  dengan  $U_m$  adalah dengan membandingkan data semua nilai mata kuliah yang dari mahasiswa  $U_1$  dengan  $U_m$  untuk menghasilkan nilai prediksi pada mata kuliah yang belum pernah diambil oleh mahasiswa  $U_m$ . Penerapan metode ini dilakukan dengan mencari semua user aktif yang memiliki nilai *Top-K similarity* paling tinggi dengan mahasiswa  $U_m$ . Setelah ditemukan *Top-K* mahasiswa yang memiliki nilai similarity tertinggi dengan mahasiswa  $U_m$ , maka akan dilakukan perhitungan nilai prediksi pada mata kuliah yang belum pernah diambil oleh user  $U_m$ . Berdasarkan pada hasil nilai prediksi ini, maka akan ditentukan mata kuliah terbaik yang akan direkomendasikan kepada mahasiswa.



**Gambar 1. 1** Collaborative Filtering Process

### **2.3.2.1. Item Based Collaborative Filtering**

*Item Based Collaborative Filtering* merupakan suatu metode yang memberikan rekomendasi item dengan membandingkan semua item berdasarkan *rating* user dengan user tertentu untuk mencari *Top-K similarity user* aktif dengan user tertentu.

Kelebihan *Item Based Collaborative Filtering* :

1. Algoritma dapat digunakan pada banyak kondisi.
2. Sangat mudah dalam melakukan update pada database dikarenakan pada algoritma ini prediksi selalu dilakukan berdasarkan pada semua user pada database.
3. Lebih cocok dipergunakan pada karakteristik data dimana perbedaan jumlah item yang dimiliki oleh user yang satu dengan lainnya kecil atau bersifat merata.

Kelemahan *Item Based Collaborative Filtering* :

1. Membutukan waktu yang lama. Hal ini dikarenakan setiap terjadi penambahan data user baru, akan dilakukan update pada nilai prediksi.

*Item Based Collaborative Filtering* dipilih berdasarkan karakteristik data dari mahasiswa beserta dengan mata kuliah yang diambil dimana terdapat jumlah sks mata kuliah yang hampir sama yang dimiliki oleh setiap mahasiswa dalam satu angkatan beserta dengan jumlah sks mata kuliah yang sama bagi setiap mahasiswa disetiap angkatan untuk mencapai kelulusan yaitu 144 sks.

### 2.3.3. Algoritma Pada *Collaborative Filtering*

Dalam penggunaan dari metode pada *Collaborative Filtering*, terdapat algoritma yang dipergunakan sebagai basis dalam perhitungan pada metode *Collaborative Filtering* yang telah dipilih. Dua macam algoritma yang dapat digunakan dalam menentukan nilai *similarity* diantaranya adalah:

#### 2.3.3.1. Cosine Similarity

*Cosine Similarity* atau *Vector Based Similarity* merupakan algoritma dimana nilai *similarity* antara  $i$  dan  $j$  digambarkan sebagai suatu sudut yang terbentuk diantara 2 buah [5]. Pada algoritma ini apabila terdapat mahasiswa yang belum mengambil mata kuliah maka nilai pada mata kuliah tersebut akan dianggap sebagai nilai terendah yaitu 0.

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} \quad (1)$$

#### 2.3.3.2. Pearson Correlation

Pendekatan *Pearson Correlation* adalah sebuah metode yang dikembangkan oleh Karl Pearson. Korelasi (*correlation*) adalah sebuah teknik pengukuran yang menentukan seberapa dekat relasi antar dua himpunan bilangan yang berbeda. Dengan syarat himpunan bilangan tersebut harus memiliki urutan yang tetap dan berpasangan satu dengan lainnya antar kedua himpunan. Hasil pengukuran dapat berupa relasi positif ataupun relasi negative. Relasi positif menunjukkan bahwa kedua himpunan memiliki kecenderungan kenaikan atau penambahan nilai yang sejajar. Sedangkan relasi negative menunjukkan kedua himpunan memiliki kecenderungan penurunan atau

pengurangan nilai yang sejajar. Sejajar dalam konteks ini berarti penurunan atau kenaikan nilai yang saling mengikuti antar kedua variable tersebut. Salah satu teknik pengukuran korelasi adalah *Pearson Correlation Moment Correlatio* atau biasa disingkat menjadi *Pearson Correlation*. Berikut ini adalah persamaan yang digunakan untuk menghitung *similarity* antar *item* yang di-*rating* oleh *user*.

$$PC(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}} \quad (2)$$

Dimana:

$i$  merupakan mahasiswa yang nilainya akan diprediksi.

$j$  merupakan mahasiswa yang datanya digunakan untuk memprediksi.

$(i, j)$  merupakan nilai *similarity* antara  $i$  dan  $j$ .

$R_{u,i}$ , merupakan nilai mata kuliah  $u$  dari mahasiswa  $i$ .

$R_{u,j}$ , merupakan nilai mata kuliah  $u$  dari mahasiswa  $j$ .

$\bar{R}_j$  merupakan nilai rata-rata dari semua mata kuliah dari mahasiswa  $j$ .

$\bar{R}_i$  merupakan nilai rata-rata dari semua mata kuliah dari mahasiswa  $i$ .

### 2.3.3.3. Nilai Prediksi

Setelah dilakukan perhitungan *similarity* untuk mencari Top-K mahasiswa yang memiliki *similarity* yang tertinggi dengan mahasiswa  $U_m$  maka langkah yang dilakukan adalah menghitung nilai prediksi pada mata kuliah yang belum diambil oleh mahasiswa  $U_m$  berdasarkan pada data nilai mata kuliah dari semua Top-K mahasiswa [2]. Rumus dari Nilai Prediksi dapat dinyatakan sebagai berikut[6]:

$$P_{u,i} = \bar{R}_a + \frac{\sum_{i \in U} (R_{u,i} - \bar{R}_i) * S_{i,j}}{\sum_{i \in U} |S_{i,j}|} \quad (3)$$

Dimana:

$i$  merupakan mahasiswa yang nilainya akan diprediksi.

$j$  merupakan mahasiswa yang datanya digunakan untuk memprediksi.

$P_{u,i}$ , adalah nilai prediksi mata kuliah yang dari mahasiswa  $i$  pada mata kuliah  $u$ .

$i \in U$  merupakan *Top-K* mahasiswa yang memiliki similarity paling tinggi dengan mahasiswa.

$\bar{R}_a$  merupakan rata-rata dari semua mata kuliah mahasiswa yang dicari.

$R_{u,i}$ , merupakan nilai mata kuliah mahasiswa  $i$  terhadap mata kuliah  $u$ .

$\bar{R}_i$  merupakan nilai rata-rata dari semua mata kuliah dari mahasiswa  $u$ .

$S_i$ , adalah nilai similarity.

#### 2.3.3.4. MAE (Mean Absolute Error)

MAE digunakan untuk menghitung rata-rata perbedaan secara mutlak antara nilai mata kuliah prediksi dengan nilai mata kuliah sebenarnya [3]. Jika  $P_{u,i}$ , adalah nilai prediksi mata kuliah dari mahasiswa  $u$  untuk mata kuliah  $i$ , maka  $R_{u,i}$ , adalah nilai mata kuliah yang sebenarnya. Persamaan MAE dirumuskan sebagai berikut :

$$MAE = \frac{\sum_{(u,i)} |P_{u,i} - R_{u,i}|}{N} \quad (4)$$

Dimana :

$i$  merupakan mahasiswa yang nilainya akan diprediksi.

$P_{u,i}$ , adalah nilai prediksi mata kuliah yang dari mahasiswa  $i$  pada mata kuliah  $u$ .

$R_{u,i}$ , adalah nilai mata kuliah yang sebenarnya dari mahasiswa  $i$ .

N adalah jumlah data.

## **2.4. Pemrograman Aplikasi**

### **2.4.1. Internet**

Internet kependekan dari (*Interconnected Network*) merupakan sistem global dari seluruh jaringan komputer yang saling terhubung menggunakan standar Internet Protocol Suite (TCP/IP) untuk melayani miliaran pengguna diseluruh dunia. Dengan begitu maraknya informasi dan kegiatan di internet, menjadikan internet seakan – akan sebagai dunia tersendiri yang tanpa batas. Dunia dalam internet disebut juga dunia maya (*cyberspace*). (Abdul Rozaq, Khairunnisa Fitri Lestari, dan Sindi Handayani, 2015).

### **2.4.2. Web**

*World wide web* atau sering di kenal sebagai web adalah suatu layanan sajian informasi yang menggunakan konsep hyperlink (tautan), yang memudahkan surfer (sebutan para pemakai komputer yang melakukan browsing atau penelusuran informasi melalui internet). Keistimewaan inilah yang telah menjadikan web sebagai service yang paling cepat pertumbuhannya. Web mengizinkan pemberian *highlight* (penyorotan atau penggaris bawahan) pada katakata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, *frase*, *movie clip*, atau file suara. Web dapat menghubungkan dari sembarang tempat dalam sebuah dokumen atau gambar ke sembarang tempat di dokumen lain. Dengan sebuah browser yang memiliki *Graphical User Interface* (GUI), link-link dapat di hubungkan ke tujuannya

20 dengan menunjuk link tersebut dengan mouse dan menekannya. (Randi V. Palit, Yaulie D.Y. Rindengan, ST.,MM.,MSc., Arie S.M. Lumenta, ST., MT, 2015).

### **2.4.3. HTML**

Menurut Hidayatullah dan Kawistara (2015) *Hypertext Merkup Languange* (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman web.

Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dari halaman web dan isinya.
2. Membuat tabel dalam halaman web.
3. Mempublikasikan halam web secara online.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via web.

### **2.4.4. PHP**

PHP adalah singkatan rekursif dari PHP *Hypertext Preprocessor* yang merupakan scripting yang tempat pengolahannya berada di server. Hal ini berarti PHP dapat bekerja dalam dokumen HTML yang menempatkan hasil permintaan scripting PHP. Dengan PHP, situs yang dibuat tidak berupa kumpulan halaman statik yang informasinya tidak sering diperbaharui. Kesederhanaan dari bahasa PHP dan mesin *scripting*. Siklus pengembangan yang lebih pendek karena pemudahan dalam pembuatan modul dan komponenkomponen yang dapat digunakan pengembangan berikutnya. Memiliki konektivitas ke server basis data. Bersifat *open source* dan tidak bergantung pada platform manapun. (Ahmad Yani, Beni Saputra. Jurnal Petir, 2018).

#### **2.4.5. Cascading Style Sheets (CSS)**

*Cascading Style Sheets* ( CSS ) adalah skrip yang digunakan dalam kode HTML untuk menciptakan kumpulan style yang digunakan untuk memperindah tampilan halaman website. CSS biasanya digunakan karena unsur kepraktisannya.

Dengan menggunakan CSS, pemrogram hanya perlu mendefinisikan style satu kali. CSS biasanya digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markah lainnya) dengan presentasi dokumen ( yang di tulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi. (Muhamad Egi, Achmad Udin Zailani, 2018).

#### **2.4.6. JavaScript**

JavaScript merupakan bahasa scripting yang pada awalnya dikembangkan oleh Netscape. Dengan menggunakan *javascript*, dapat dibuat tampilan web yang lebih interaktif lagi. Agar user dapat menjalankan javascript, alat yang dibutuhkan hanyalah *browser* yang kemampuan *javascript*-nya telah diaktifkan. Penulisan *javascript* dapat dipadukan pada halaman HTML, dan dari sini dapat dilihat bahwa *javascript* merupakan bahasa pemrograman yang berjalan di sisi *client*. Karena proses dari *javascript* tergantung proses dari browser yang digunakan oleh user. Contoh penggunaan *JavaScript* adalah sebagai berikut:

```
<html>
```

```
<body>

<script language="javascript">

document.write("Ini dihasilkan oleh javascript");

</script>

</body>

</html>
```

#### **2.4.7. Bootstrap**

Menurut Mohamad Rizki Adhiasta (2016) dalam jurnal “Perancangan Informasi Batik Solo Berbasis *Bootstrap*”. *Bootstrap* adalah sebuah *framework* yang dikembangkan pengembang Twitter pada pertengahan tahun 2010. Sebelum menjadi kerangka kerja *open-source*, *Bootstrap* dikenal sebagai *Twitter Blueprint*. Dan terus berkembang sampai saat ini dan *Bootstrap* telah menjadi salah satu front-end framework yang paling populer dan merupakan proyek open source di dunia. Platform ini awalnya dikembangkan pada ajang Hackweek, sebuah perhelatan developer yang diadakan Twitter. *Bootstrap* digambarkan sebagai CSS sederhana namun dibangun dengan *pre-processor* yang menyediakan lebih banyak daya dan fleksibilitas ketimbang CSS standar. Awalnya dirilis pada Jumat, 19 Agustus, 2011, dan sudah memiliki lebih dari 20 produk release termasuk yang terbesar adalah versi 2 dan versi 3. Hadirnya *Bootstrap 2*, maka fungsionalitas responsif terhadap seluruh kerangka sebagai stylesheet opsional telah di tambahkan dan pada *Bootstrap 3*, maka responsif secara default telah ditambahkan untuk pendekatan pertamanya terhadap ponsel.

Framework Bootstrap terdiri dari bahasa html dan css yang juga menyediakan efek *javascript* yang dibangun menggunakan *jquery*.

#### **2.4.8. Database Server MySQL**

MySQL adalah Sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah MySQL *FreeSoftware* yang berada dibawah Lisensi GNU/GPL ( *General Public License* ).

MySQL Merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius . Selain database server, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *Server*, yang berarti program kita berposisi sebagai *Client*.

Jadi MySQL adalah sebuah database yang dapat digunakan sebagai *Client* maupun *server*. *Database* MySQL merupakan suatu perangkat lunak database yang berbentuk *database* relasional atau disebut *Relational Database Management System* ( *RDBMS* ) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language* ).

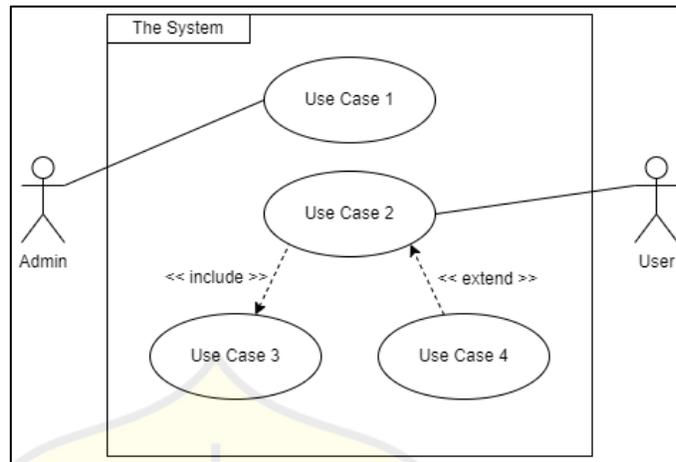
## 2.5. Pemodelan Sistem Dengan UML

*Unified Modelling Language (UML)* adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan *UML* kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena *UML* juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti *C++*, *Java*, *C#* atau *VB.NET*. Walaupun demikian, *UML* tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan syntax/semantik. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari 3 notasi yang telah ada sebelumnya: *Grady Booch OOD (Object Oriented Design)*, *Jim Rumbaugh OMT (Object Modeling Technique)*, dan *Ivar Jacobson OOSE (Object-Oriented Software Engineering)*. Sejarah *UML* sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch*, *metodologi coad*, *metodologi OOSE*, *metodologi OMT*, *metodologi shlaer-mellor*, *metodologi wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-

sendiri, yang 22 mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

### **2.5.1. Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah Use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua feature yang ada pada sistem. Sebuah *Use case* dapat meng-include fungsionalitas *Use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa Use case yang di-include akan dipanggil setiap kali *Use case* yang meng-include dieksekusi secara normal. Sebuah *Use case* dapat di-include oleh lebih dari satu *Use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *Use case* juga dapat meng-extend *Use case* lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar *Use case* menunjukkan bahwa *Use case* yang satu merupakan spesialisasi dari yang lain.



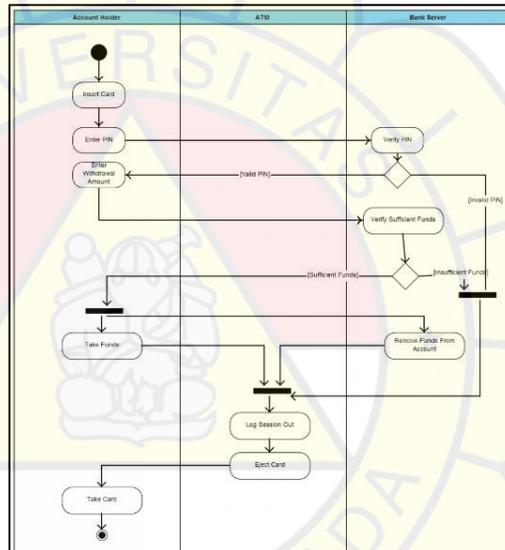
**Gambar 2. 1** Contoh *Use Case Diagram*

### 2.5.2. Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *Activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar *UML* menggunakan segiempat dengan sudut

membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa object *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

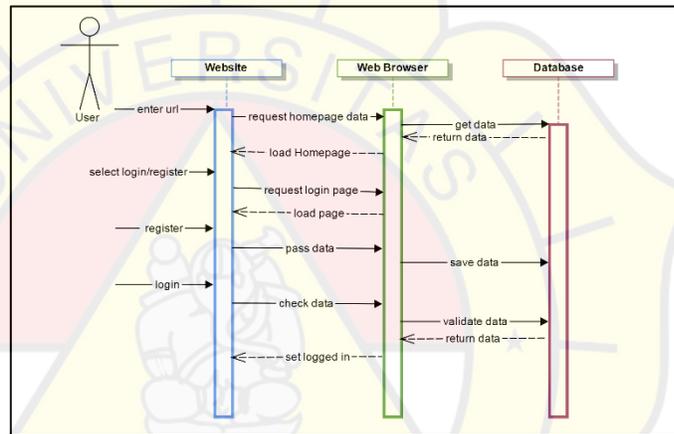


**Gambar 2. 2** Contoh *Activity Diagram*

### 2.5.3. Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan

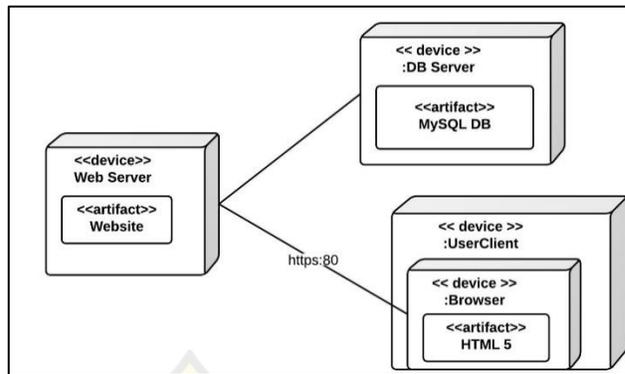
sebagai respon dari sebuah event untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.



**Gambar 2.3** Contoh *Sequence Diagram*

#### 2.5.4. Deployment Diagram

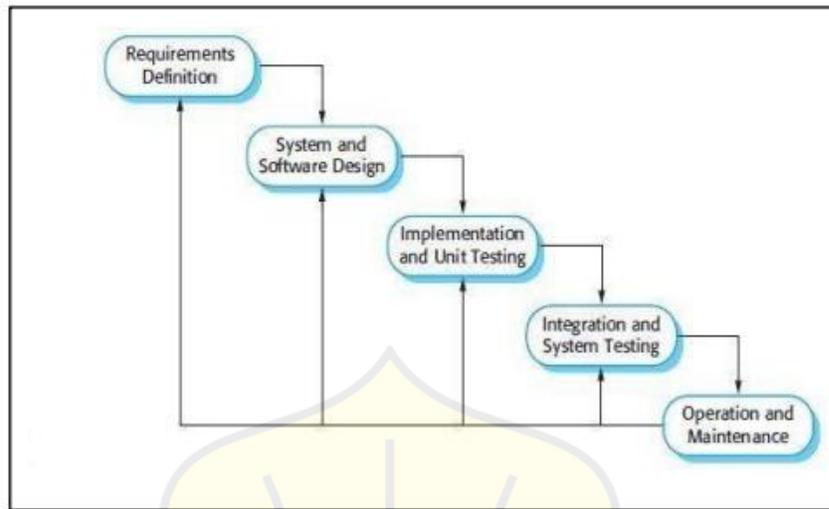
*Deployment/physical diagram* menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



**Gambar 2. 4** Contoh *Deployment Diagram*

## 2.6. Metodologi Pengembangan Sistem

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. Dalam pengembangannya metode waterfall memiliki beberapa tahapan yang runtut: requirement (analisa kebutuhan), desain sistem (system design), Coding & Testing, Penerapan Program, pemeliharaan. (Chrisantus Trisianto. 2018. Jurnal Teknologi Informasi : Penggunaan Metode Waterfall Untuk Pengembangan Sistem Monitoring Dan Evaluasi Pembangunan Pedesaan).



**Gambar 2. 5** Metodologi *Waterfall* (Rosa A. S dan M. Shalahuddin, 2016)

Adapun penjelasan urutan dari tahapan-tahapan yang dimiliki metodologi waterfall adalah sebagai berikut:

### **2.6.1. Requirement (Analisa Kebutuhan)**

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau study literatur. Seseorang sistem analisa akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. 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 analisa untuk menterjemahkan kedalam bahasa pemrograman.

### **2.6.2. Design System (Design Sistem)**

Proses design akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, *representasi interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

### **2.6.3. Coding & Testing (Penulisan Sinkode Program/Implementation)**

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. 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.

### **2.6.4. Penerapan/Pengujian Program (Integration & Testing)**

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

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

Perangkat lunak yang susah 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) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

