

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

2.1.1 Pengertian Sistem

Menurut Jogianto (2005:2) pada buku Analisis dan Desain Sistem Informasi mendefinisikan sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi.tujuan tertentu.

2.1.2 Pengertian Informasi

Menurut Sutarman (2012:14), “Informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima”.

2.2 Rancang Bangun

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Pressman, 2002). Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik (Ladjamudin, 2005).

2.3 Konsep Dasar Web

Secara umum, website (web) dipahami sebagai sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk digital baik itu *teks*, gambar, animasi yang disediakan melalui jalur internet sehingga dapat diakses dari seluruh dunia yang memiliki koneksi internet.

2.4 Website

World Wide Web (WWW) disingkat menjadi web, yang menurut Yuhefizar (2008:159) “Web adalah suatu metode untuk menampilkan informasi di internet, baik berupa *teks*, gambar, suara maupun *video* yang interaktif dan mempunyai kelebihan untuk menghubungkan (*link*) satu dokumen dengan dokumen lainnya (*hypertext*) yang dapat diakses melalui *browser*”. *Browser* adalah perangkat lunak untuk mengakses halaman-halaman web, seperti *Internet Explorer*, *Mozilla Firefox*, *Opera*, *Safari*, dan lain-lain.

2.4.1 Web Browser Menurut Limantra(2009:1)

“Web *Browser* adalah aplikasi perangkat lunak yang memungkinkan penggunanya untuk berinteraksi dengan *teks*, *image*, *video*, *games*, dan informasi lainnya yang berlokasi pada halaman web pada *World Wide Web*(WWW) atau *Local Area Network* (LAN)”.

2.4.2. Web Server Menurut Wahyono (2007:155)

“Web *Server* merupakan komputer yang digunakan yang digunakan sebagai *host* berbagai aplikasi web, baik dalam lingkungan internet maupun intranet.

2.5 Perangkat Lunak Yang Digunakan

2.5.1 XAMPP

Menurut Dadan (2015:28) XAMPP adalah salah satu aplikasi web server apache yang terintegrasi dengan mysql dan phpmyadmin. XAMPP adalah singkatan dari *X*, *Apache Server*, *MYSQL*, *PHPMyadmin*, dan *Phyton*. Huruf X di depan menandakan XAMPP bisa diinstal di berbagai operating system. XAMPP dapat diinstal pada *Windows*, *Linux*, *MacOS*, dan *Solaris*. Sampai saat ini, XAMPP masih diberikan secara gratis, bebas di-download dan digunakan tanpa harus membayar.

2.5.2 MySQL

Menurut Miftahul Huda (2010:181) MYSQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS(*database management system*), *database* ini multithread, multi-user. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi *GNU General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus yang bersifat khusus.

2.5.3 Sublime Text

Sublime text salah satu kode editor yang biasa digunakan oleh para programmer untuk membuat suatu program.

Menurut Supono dan Putratama (2016:14) “Sublime text merupakan perangkat lunak text editor yang digunakan untuk membuat atau meng-edit suatu aplikasi. Sublime text mempunyai fitur plugin tambahan yang memudahkan “programmer”.

2.6 Bahasa Pemrograman

2.6.1 PHP

Menurut Anhar (2010:3) PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan.

2.6.2 PHP Native

PHP native merupakan pemrograman web perpaduan bahasa pemrograman yang didasari dengan bahasa pemrograman PHP yang mana bisa disisipi oleh text Javascript, CSS, Bootstrap dan lain-lain. Native sendiri artinya asli, yakni 9 pemrograman php yang murni disusun dan di coding/ dibangun oleh para programmer sendiri tanpa ada istilah tambahan buat settingan/ konfigurasi lainnya.

2.7 Penjualan

Menurut Mulyadi (1997:204), Menyatakan bahwa, dalam transaksi penjualan kredit, jika order dari pelanggan telah dipenuhi dengan pengiriman barang atau penyerahan jasa, untuk jangka waktu tertentu perusahaan memiliki piutang kepada pelanggannya.

2.8 Pengertian Collaborative Filtering

Collaborative Filtering merupakan proses penyaringan atau pengevaluasian item menggunakan opini orang lain (Schafer dkk, 2007). Pada prosesnya metode *Collaborative Filtering* melakukan penyaringan data berdasarkan tingkah laku karakteristik pengguna sehingga dapat memberikan informasi yang baru kepada pengguna lainnya karena sistem memberikan informasi berdasarkan pola satu kelompok pengguna yang hampir sama.

Sistem rekomendasi berbasis kolaboratif (*collaborative-based*) dibuat untuk mengatasi kelemahan dari sistem rekomendasi berbasis konten (*content-based*) (Adhitya pratama et.al.,2013) yaitu:

- a. Pendekatan *collaborative* dapat bekerja dalam domain dimana terdapat sedikit konten yang berasosiasi dengan item atau ditempat dimana konten sulit dianalisis menggunakan komputer seperti ide, masukan, atau opini sehingga menjadi *reliable*.
- b. Pendekatan *collaborative* mempunyai kemampuan untuk menyediakan rekomendasi yang tidak terduga atau tidak disengaja. Misalnya dapat merekomendasikan *item* yang relevan kepada pengguna sekaligus tidak mengandung konten dari profil pengguna tersebut.

Walaupun dalam beberapa penelitian rekomendasi berbasis kolaboratif dapat menutupi kelemahan dari rekomendasi berbasis konten, rekomendasi berbasis kolaboratif, memiliki kekurangan, antara lain (Adomavicius dan Tuzhilin,200) :

1. *Cold-start problem*

Cold-start problem atau *new item problem* disebabkan karena *collaborative filtering* menggunakan rating atau preferensi pengguna untuk merekomendasikan sesuatu kepada pengguna lain. Ketika rating yang dibutuhkan tidak tersedia atau hingga *item* baru dinilai oleh sejumlah besar pengguna, maka sistem rekomendasi tidak akan dapat merekomendasikannya.

2. *Sparsity Problem*

Pada data yang berukuran besar, jika banyak item baru yang sedikit di-rating oleh pengguna, maka item tersebut memiliki prediksi yang buruk dan menghasilkan rekomendasi yang buruk.

Terdapat dua metode atau pendekatan yang digunakan dalam *collaborative filtering* yaitu *user-based collaborative filtering* dan *item-based collaborative filtering*.

2.9 Algoritma Collaborative Filtering

Menurut teori dan penggunaannya Schafer membagi algoritma *collaborative filtering* kedalam dua kelas yang berbeda yaitu dengan menggunakan algoritma probablistik dan non probablistik. Suatu algoritma dianggap probablistik bila algoritma tersebut berdasarkan model probablistik. Algoritma tersebut menggunakan opini orang lain (Schafer dkk, 2007). Model peratingan dalam sistem rekomendasi berbasis *collaborative filtering* ini memiliki dua bentuk yaitu *User based collaborative filtering* dan *item based collaborative filtering*.

a. *User – Based Collaborative Filtering*

Model *user-based collaborative filtering* mengasumsikan bahwa cara yang baik untuk menemukan *item* yang menarik bagi *user* tertentu adalah dengan mencari *user* lain yang memiliki minat yang sama. Jadi, pada awalnya *user-based CF* mencoba untuk menemukan *user neighbor* berdasarkan pada *user similarity* dan kemudian setiap nilai rating dari *user neighbor* akan dijadikan bahan rekomendasi bagi user aktif. Algoritma ini

bekerja berdasarkan asumsi bahwa setiap pengguna merupakan bagian dari kelompok yang memiliki kesamaan dengan pengguna lainnya. Dasar dari rekomendasi dengan algoritma ini adalah bahwa rekomendasi yang dihasilkan disusun berdasarkan *item* yang disukai oleh setiap *user*. *Item* yang direkomendasikan merupakan hasil rekomendasi menurut apa yang disukai para *user* lainnya. Berikut adalah langkah-langkah dari algoritma *user-based collaborative filtering*:

- Tentukan matrik rating *user item*

Hitung nilai similituditas antar *user* terhadap pemilihan *item* tertentu dengan persamaan:

1. *Cosine-based similarity*

Metode ini biasa digunakan untuk menghitung kesamaan rating yang diberikan *user* dengan *user* lainnya. Persamaannya sebagai berikut:

$$sim(u, v) = \frac{\sum_{i=1}^m r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i=1}^m (r_{u,i})^2} \sqrt{\sum_{i=1}^m (r_{v,i})^2}}$$

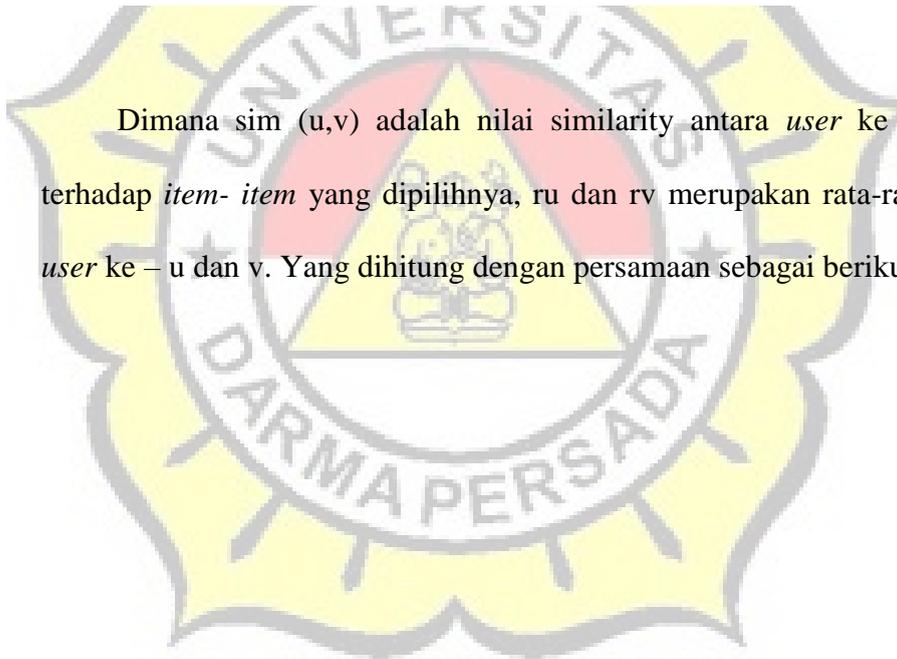
Dimana $sim(u,v)$ adalah nilai similarity antara *user* ke u dan v terhadap *item-item* yang dipilihnya, $r_{u,i}$ adalah nilai rating *user* ke u terhadap *item* ke i dan $r_{v,i}$ adalah nilai rating *user* ke v terhadap *item* ke i .

2. *Correlation-based similarity*

Metode ini digunakan untuk mengukur seberapa dekat hubungan antar 2 variabel. Dasar dari metode ini adalah model regresi linier sehingga hasil perhitungan *similarity* menyatakan hubungan yang linier diantara kedua variabel tersebut. Berikut adalah persamaan *correlation-based similarity* :

$$\text{sim}(u, v) = \frac{\sum_{i=1}^m (r_{u,i} - \bar{r}_u) (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i=1}^m (r_{v,i} - \bar{r}_v)^2}}$$

Dimana $\text{sim}(u, v)$ adalah nilai *similarity* antara *user* ke u dan v terhadap *item-item* yang dipilihnya, r_u dan r_v merupakan rata-rata rating *user* ke u dan v . Yang dihitung dengan persamaan sebagai berikut :



$$\bar{r}_u = \frac{1}{m} \sum_{i=1}^m r_{ui}$$

Dan

$$\bar{r}_v = \frac{1}{m} \sum_{i=1}^m r_{vi}$$

Sedangkan r_{ui} merupakan rating *user* ke u terhadap *item* ke i , dan r_{vi} merupakan rating *user* ke v terhadap *item* ke i . Hitung nilai prediksi *user* tertentu terhadap suatu *item* dengan persamaan :

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) x P_{a,u}}{\sum_{u=1}^n P_{a,i}}$$

Dimana $P_{a,i}$ nilai prediksi dari user a terhadap item ke i ; $P_{a,u}$ merupakan nilai similarity antara user a dan u ; sedangkan n merupakan jumlah user yang memiliki jarak terdekat dengan $P_{a,u}$

2.10 Peralatan Pendukung (Tool System)

2.10.1 UML (*Unified Modeling Language*)

Menurut Nugroho (2010:6), "UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek)." Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Berdasarkan pendapat yang dikemukakan Nugroho tersebut dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*).

2.10.2 Model UML

1. *Use Case* Diagram

Menurut Munawar (2005:63) *Use Case* adalah deskripsi fungsi dari perspektif pengguna. *Use Case* Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah system yang ditekankan adalah "apa" yang diperbuat sistem bukan "bagaimana". Sebuah *Use Case* mempresentasikan sebuah interaksi antar aktor dengan system. Seorang *actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem agar dapat melakukan sesuatu hal yang

telah ditentukan. *Use Case* dapat membantu mempresentasikan sebuah rancangan kepada klien.

2. *Scenario*

Menurut Munawar (2005), skenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. *Form* skenario merupakan penjelasan penulisan *use case* dari sudut pandang *actor*.

1. Nama *use case* yaitu penamaan *use case* yang menggunakan kata kerja
 2. Deskripsi yaitu penjelasan mengenai tujuan *use case* dan nilai yang akan didapatkan oleh aktor
 3. Kondisi sebelum (*pre-condition*) yaitu kondisi-kondisi yang perlu ada sebelum *use case* dilakukan.
 4. Kondisi sesudah (*post-condition*) yaitu kondisi-kondisi yang sudah dipenuhi ketika *use case* sudah dilaksanakan.
 5. Alur dasar (*basic flow*) yaitu alur yang menceritakan jika semua aksi yang dilakukan adalah benar atau proses yang harusnya terjadi
 6. Alur alternatif (*alternatif flow*) yaitu alur yang menceritakan aksi alternatif, yang berbeda dari alur dasar.
- ## 3. *Activity Diagram*

Rosa dan Shalahuddin (2013:161), menjelaskan tentang *activity diagram* sebagai berikut: *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas

menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

2.11 *System Development Life Cycle (SDLC)*

Menurut Sukamto (2015: 26) *Software Development Life Cycle* atau SDLC adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem perangkat lunak sebelumnya.

SDLC memiliki beberapa model yang sering digunakan dalam pengembangan sistem diantaranya adalah sebagai berikut:

1. *Waterfall*

Model *waterfall* merupakan paradigma model pengembangan perangkat lunak paling tua dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian, dan pemeliharaan.

2. *Prototype*

Model *prototype* merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama

yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau *waterfall development system*.

Dalam model ini, *prototype* dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya sesuai dengan keinginan dan kebutuhan pelanggan.

3. *Rapid Application Development (RAD)*

Model *Rapid Application Development (RAD)* adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.