

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Rekomendasi**

Sistem rekomendasi dapat didefinisikan sebagai program yang mencoba untuk merekomendasikan *item* yang paling cocok (produk atau jasa) untuk pengguna tertentu (individu atau bisnis) dengan memprediksi minat pengguna di *item* berdasarkan informasi terkait tentang *item*, pengguna dan interaksi antara *item* dan pengguna. Tujuan dari pengembangan sistem rekomendasi adalah untuk mengurangi informasi yang berlebihan dengan mengambil informasi dan layanan yang paling relevan dari sejumlah besar data, sehingga memberikan layanan pribadi. Fitur yang paling penting dari sebuah sistem rekomendasi adalah kemampuannya untuk “menebak” preferensi dan kepentingan pengguna dengan menganalisis perilaku pengguna dan / atau perilaku pengguna lain untuk menghasilkan rekomendasi pribadi (Jie Lu, 2015).

#### **2.2 Pengertian Atlet**

Atlet adalah Individu yang memiliki keunikan dan memiliki bakat tersendiri lalu memiliki pola perilaku dan juga keperibadian tersendiri serta memiliki latar belakang kehidupan yang mempengaruhi secara spesifik pada dirinya. Rusdianto (dalam Saputro, 2014)

Individu yang terlibat dalam aktivitas olahraga dengan memiliki prestasi di bidang olahraga tersebut dapat dikatakan bahwa individu itulah yang dimaksud dengan atlet. Satiadarma (dalam Yuwanto & Sutanto, 2012)

Berdasarkan uraian diatas maka dapat disimpulkan bahwa atlet adalah

individu yang terlatih, memiliki keunikan, dan juga memiliki bakat dalam bidang olahraga yang terlatih dalam cabang olahraga.

### 2.3 Data Mining

*Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar. Dalam data mining terdapat dua pendekatan metode pelatihan, yaitu (Helmy dkk, 2013):

- a. *Unsupervised learning*, metode ini diterapkan tanpa adanya latihan (*training*) dan tanpa ada guru (*teacher*). Guru di sini adalah label dari data.
- b. *Supervised learning*, yaitu metode belajar dengan adanya latihan dan pelatih. Dalam pendekatan ini, untuk menemukan fungsi keputusan, fungsi pemisah atau fungsi regresi, digunakan beberapa contoh data yang mempunyai output atau label selama proses training.

Ada beberapa teknik yang dimiliki *data mining* berdasarkan tugas yang bisa dilakukan, setiap teknik memiliki algoritma masing-masing. Teknik dalam data mining terbagi menjadi enam kategori, yaitu:

- a. Deskripsi

Para peneliti biasanya mencoba menemukan cara untuk mendeskripsikan pola dan trend yang tersembunyi dalam data.

- b. Estimasi

Estimasi mirip dengan klasifikasi, kecuali variabel tujuan yang lebih ke arah numerik dari pada kategori.

c. Prediksi

Prediksi memiliki kemiripan dengan estimasi dan klasifikasi. Hanya saja, prediksi hasilnya menunjukkan sesuatu yang belum terjadi (mungkin terjadi dimasa depan).

d. Klasifikasi

Dalam klasifikasi variabel, tujuan bersifat kategorik. Misalnya, kita akan mengklasifikasikan pendapatan dalam tiga kelas, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

e. Klastering

*Clustering* lebih ke arah pengelompokan *record*, pengamatan, atau kasus dalam kelas yang memiliki kemiripan.

f. Asosiasi

Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu.

## 2.4 Sistem Informasi

Sistem merupakan kelompok unsur yang erat hubungannya dengan yang lain, yang berfungsi bersama untuk mencapai tujuan tertentu. Suatu sistem terdiri dari unsur, komponen, atau variable yang terorganisir, saling berinteraksi, saling bergantung satu sama lain, dan terpadu (Anggraeni, 2017).

## 2.5 WEB

*Website* adalah kumpulan dari berbagai macam halaman situs, yang terangkum didalam sebuah *domain* atau juga *subdomain*, yang tempatnya berada di dalam *World Wide Web* (WWW) di internet. Halaman *web* biasanya berupa dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang bisa diakses melalui HTTP (*Hyper Text Transfer Protocol*), HTTP yaitu protokol yang menyampaikan berbagai informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web broser* (Lengkong, et al., 2019).

## 2.6 MySQL

MySQL diciptakan di Negara Swedia oleh perusahaan MySQL AB adapun masing-masing nama yang berjasa dalam menciptakan MySQL adalah David Axmark, Allan Lason, dan Michael “*monty*” Widenius. Perangkat lunak ini tersebar luas secara garis besar karena memiliki lisensi *GNU General Public License*. Sampai sekarang, tercatat ada beberapa bahasa pemrograman yang populer yang bisa bersinkronisasi dengan MySQL, seperti C<, C++, C#, bahasa pemrograman eiffel, bahasa pemrograman *Smaltalk*, bahasa pemrograman *java*, bahasa pemrograman *lisp*, *perl*, PHP, bahasa pemrograman *phiton*, *ruby*, *REALbasic*, dan Tcl (Nst, 2012).

*MySQL* adalah *Relational Database Management (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *General Public License (GPL)*. Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. Saat ini

*MySQL* merupakan *software database* yang memegang kendali terbesar pada aplikasi *database* berbasis *web*. Arsitektur *databasenya* sangat cocok bekerja pada lingkungan tersebut, didukung kemampuannya dalam berinteraksi dengan berbagai bahasa *programmeran side-server* seperti *PHP, ASP, CGI* dan *JSP* (Prasetyo, Didik Dwi, 2003) dalam Subagio (2017).

Menurut (Buana, 2014), *MySQL* Merupakan *database server* yang paling sering digunakan dalam pemrograman *PHP*. *MySQL* digunakan untuk menyimpan data dalam *database* dan memanipulasi data-data yang diperlukan. Manipulasi data tersebut berupa menambah, mengubah, dan menghapus data yang berada dalam *database*. Berdasarkan pengertian yang ada diatas dapat penulis simpulkan bahwa pengertian *MySQL* adalah sebuah *server* basis data yang digunakan untuk membangun basis data pada aplikasi-aplikasi yang didukung oleh basis data.

## **2.7 XAMPP**

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server, MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari *X* (empat sistem operasi apapun), *Apache, MySQL, PHP* dan *Perl*. Program ini tersedia dalam *GNU (General Public License)* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis (Palit, et al., 2015).

## 2.8 TOPSIS

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) memiliki konsep dimana alternatif yang terpilih merupakan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Semakin banyaknya faktor yang harus dipertimbangkan dalam proses pengambilan keputusan, maka semakin relatif sulit untuk mengambil keputusan terhadap suatu permasalahan, apalagi jika upaya pengambilan keputusan dari suatu permasalahan tertentu, selain mempertimbangkan berbagai faktor / kriteria yang beragam juga melibatkan beberapa orang pengambil keputusan. Hal ini disebabkan konsepnya sederhana dan mudah dipahami komputasinya efisien dan memiliki kemampuan untuk mengatur kinerja relatif dari alternatif-alternatif keputusan (Munarwan dan Siddiq, 2012).

Secara umum, prosedur TOPSIS mengikuti langkah-langkah sebagai berikut :

1. Membuat matriks keputusan yang ternormalisasi.
2. Membuat matriks keputusan yang ternormalisasi terbobot.
3. Menentukan matriks solusi ideal positif dan solusi ideal negatif.
4. Menghitung perbedaan nilai untuk setiap alternatif, baik alternatif untuk sosial ideal dan negatif.
5. Menghitung kedekatan relatif terhadap solusi ideal, untuk solusi positif maupun negative.
6. Mengurutkan nilai preferensi hasil perhitungan untuk setiap kriteria.

Langkah awal menerapkan metode TOPSIS adalah menghitung rating kinerja setiap alternatif  $A_i$  pada setiap kriteria  $C_j$  yang ternormalisasi, yaitu:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

dengan  $i=1,2,\dots,m$ ; dan  $j=1,2,\dots,n$

Keterangan:

$r_{ij}$  = Matriks keputusan yang ternormalisasi

$x_{ij}$  = Elemen nilai tiap kriteria dari alternatif

Solusi ideal positif  $A^+$  dan solusi ideal negatif  $A^-$  dapat ditentukan berdasarkan rating bobot ternormalisasi ( $y_{ij}$ ) sebagai:

$y_{ij} = w_{ij} r_{ij}$  ; dengan  $i=1,2,\dots,m$ ; dan  $j=1,2,\dots,n$

$$A^+ = (y_1^+, y_2^+, y_3^+, \dots, y_n^+);$$

$$A^- = (y_1^-, y_2^-, y_3^-, \dots, y_n^-);$$

dengan

$$y_j^+ = \begin{cases} \max_i y_{ij} \\ \max_i y_{ij} \end{cases} \quad y_j^- = \begin{cases} \min_i y_{ij} \\ \min_i y_{ij} \end{cases}$$

$$i = 1,2,3, \dots, nm, \quad j = 1,2,3,\dots, n$$

Keterangan :

$Y_{ij}$  = Elemen matriks ternormalisasi [i]

Keterangan :

Y = Matriks ternormalisasi terbobot

$Y_{ij}$  = Elemen dari matriks ternormalisasi terbobot

W = Nilai bobot prefensi tiap kriteria

i = bernilai 1 sampai dengan m jumlah alternatif

j = bernilai 1 sampai dengan n jumlah kriteria

$Y_j^+$  = Max  $y_{ij}$ , jika j adalah atribut keuntungan

$Y_j^-$  = Max  $y_{ij}$ , jika j adalah atribut biaya

$Y_i^+$  = Max  $y_{ij}$ , jika j adalah atribut keuntungan

$Y_i^-$  = Max  $y_{ij}$ , jika j adalah atribut biaya

Jarak antara alternatif  $A_i$  dengan solusi ideal positif dirumuskan

sebagai :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_{ij}^+ - y_j^+)^2}$$

$i = 1, 2, \dots, m$

Keterangan :

i = bernilai 1 sampai dengan m jumlah alternatif



$j =$  bernilai 1 sampai dengan  $n$  jumlah kriteria

Jarak antara alternatif  $A_i$  dengan solusi ideal negatif dirumuskan sebagai :

$$D_i^- = \sqrt{\sum_{j=1}^n (y_j - y_{ij}^-)^2}$$

$i = 1, 2, \dots, m$

Keterangan :

$i =$  bernilai 1 sampai dengan  $m$  jumlah alternatif

$j =$  bernilai 1 sampai dengan  $n$  jumlah kriteria

Nilai preferensi untuk setiap alternatif ( $V_i$ ) diberikan sebagai:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

$i = 1, 2, \dots, m$

Keterangan :

$i =$  bernilai 1 sampai dengan  $m$  jumlah alternatif

$D =$  Jarak alternatif

Nilai  $V_i$  yang lebih besar menunjukkan bahwa alternatif  $A_i$  lebih dipilih.

## 2.9 K-Nearest Neighbor

Menurut Whidiasih, et al (2013) prinsip kerja K-Nearest Neighbor (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan  $K$  tetangga (*neighbor*) terdekatnya dalam data pelatihan. Data training diproyeksikan ke ruang berdimensi banyak, yang mana masing-masing dimensi menjelaskan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data training. Sebuah titik pada ruang ini ditandai kelas  $c$ , jika kelas  $c$  merupakan klasifikasi yang paling banyak ditemui pada  $k$  buah tetangga terdekat titik tersebut (Nanja dan Purwanto, 2015). *K-Nearest Neighbor* merupakan salah satu metode yang digunakan dalam pengklasifikasian. Prinsip kerja *K-Nearest Neighbor* (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan  $K$  tetangga (*neighbor*) terdekatnya dalam data pelatihan (Santoso, 2007).

Berikut rumus pencarian jarak menggunakan rumus *Euclidian* [3]:

Dengan:

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

Keterangan :

$x_1$  = sampel data

$x_2$  = data uji

$i$  = variabel

data

$d$  = jarak

$p$  = dimensi data

## 2.10 Unified Modeling Language (UML)

Menurut Nugroho (2015 : 87) menyatakan bahwa dengan adanya *UML*, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. *UML* diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini.

Menurut Rosa A.S dan M. Shalahuddin (2015:133) menyatakan bahwa “*UML* (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. *UML* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah *system*.”

### a. Tujuan UML

- 1) Memberikan model yang siap pakai, bahasa permodelan *visual* yang *ekspresif* untuk mengembangkan model dan dimengerti secara umum.
- 2) Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- 3) Menyatukan praktek-praktek yang terdapat dalam permodelan.

**b. Diagram-diagram dalam UML**





Ada beberapa diagram dalam UML (*Unified Modelling Language*) antara lain :

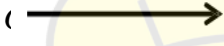

**1) Use Case Diagram**

*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, membuat atau *create* sebuah daftar belanja dan sebagainya. Seorang aktor adalah entitas manusia atau sebuah mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun simbol dari *use case* diagram antara lain:

Tabel 2.2 Simbol-simbol *Use Case Diagram*

No	Simbol	Keterangan

1	<p>Use Case</p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya di-nyatakan dengan meng-gunakan kata kerja di awal frase nama <i>use case</i>.</p>
2	<p>Aktor / <i>Actor</i></p>  <p>Nama Aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanyadinyatakan menggunakan kata benda di awal frase nama aktor.</p>
3	<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4	<p>Ekstensi / <i>Extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan</p>






		<p>prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
5	<p>Generalisasi  </p>	<p>Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6	<p>Menggunakan / <i>Include</i> /  <i>uses</i>          &lt;&lt;include&gt;&gt;  </p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan.</p>

Sumber Rosa A.S dan M.Shalahuddin (2014:90)

## 2) Activity Diagram

*Activity Diagram* menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, dan *decision* yang mungkin terjadi, dan bagaimana alur berakhir. Adapun simbol dari *Activity Diagram* antara lain:

Tabel 2.3 Simbol *Activity Diagram*

No	Gambar	Keterangan
1.	 <i>Activity</i>	Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
2.	 <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.	 Status Awal	Status awal aktiviatas sistem, sebuah diagram aktivitas memiliki status awal.
4.	 Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5.	 <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

Sumber : Rosa A.S dan M.Shalahuddin (2014:97)

## 2.11 PHP

Menurut (Betha Sidik 2014) PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu

aplikasi bukan dokumen HTML yang dibuat menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*.

Dari penjelasan di atas dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman yang digunakan untuk membuat dan mendesain pengembangan dari sebuah server web.

## 2.12 Javascript

JavaScript adalah bahasa script yang dikembangkan oleh Netscape untuk membuat dokumen yang dinamis. JavaScript merupakan bahasa script sederhana yang mempunyai kemiripan dengan bahasa pemrograman C. JavaScript juga dikenal sebagai sebuah kode pemrograman berorientasi objek (Object Oriented Programming) disingkat OOP. JavaScript memiliki keistimewaan untuk ditambahkan pada kode HTML dan membuat dokumen menjadi lebih interaktif (Sianipar, 2015).

Untuk mulai menggunakan JavaScript, ada beberapa hal yang dibutuhkan oleh seorang perancang web, yaitu :

1. Perancang harus mengetahui bagaimana menggunakan HTML dan mengedit dokumen HTML.
2. Perancang harus menggunakan browser yang mendukung pemrograman JavaScript, misalnya Netscape versi 3 atau di atasnya, Internet Explorer versi 4 atau di atasnya.
3. Meskipun penguasaan suatu bahasa pemrograman tidak menjadi hal yang utama, tetapi dengan mengetahui dan menguasai salah satu bahasa pemrograman akan sangat membantu dalam mempelajari JavaScript.



Sebagaimana dengan bahasa pemrograman, JavaScript mempunyai aturan dasar dan format sendiri. JavaScript dapat memperlakukan fungsi-fungsi secara otomatis untuk menghitung dan membuat web lebih interaktif. Agar browser mengenali statemen-statement JavaScript maka digunakan tag script (Kadir, 2014).

### **2.13 HTML**

*Hypertext Markup Language (HTML)* adalah bahasa yang digunakan untuk menulis halaman *web*. *HTML* merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)* (Kadir, 2014). *HTML* sebenarnya adalah dokumen *ASCII* atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

Mendesain *HTML* berarti melakukan suatu tindakan pemrograman. Namun *HTML* bukanlah sebuah bahasa pemrograman. Namun *HTML* hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. Menuliskan tag-tag *HTML* tidaklah sebatas hanya memasukkan perintah-perintah tertentu agar *HTML* kita dapat di akses oleh browser. Mendesain *HTML* adalah sebuah seni tersendiri. *Homepage* yang merupakan implementasi dari *HTML* adalah refleksi dari orang yang membuatnya. Untuk itu kita perlu mendesainnya dengan baik agar para pengunjung homepage yang kita buat merasa senang dan bermanfaat. Mendesain *HTML* dapat dilakukan dengan cara menggunakan *HTML Editor*, seperti notepad++, adobe dreamweaver dan lain-lain (Kadir, 2014).

### **2.14 Cascading Style Sheets (CSS)**

*Cascading Style Sheets (CSS)* adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa

markup\_(Kadir, 2014). Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan *HTML* dan *XHTML*. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk *SVG* dan *XUL*. *Spesifikasi CSS* diatur oleh World Wide Web Consortium (*W3C*). *CSS* 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 markup* lainnya) dengan presentasi dokumen (yang ditulis dengan *CSS*). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi (Kadir, 2014).

