

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pariwisata**

Istilah pariwisata (tourism) baru muncul di masyarakat di masyarakat kira-kira pada abad ke-18, khususnya sesudah Revolusi Industri di Inggris. Istilah pariwisata berasal dari dilaksanakannya kegiatan wisata (tour), yaitu suatu aktivitas perubahan tempat tinggal sementara dari seseorang, di luar tempat tinggal sehari-hari dengan suatu alasan apa pun selain melakukan kegiatan yang bisa menghasilkan upah atau gaji (Muljadi, 2012).

Pariwisata adalah perjalanan dari satu tempat ketempat lain bersifat sementara, dilakukan perorangan atau kelompok, sebagai usaha mencari keseimbangan atau keserasian dan kebahagiaan dengan lingkungan dalam dimensi sosial budaya, alam, dan ilmu (Kodhyat dalam Kurniansah, 2014).

#### **2.2 Metode Algoritma K-Nearest Neighbor (K-NN)**

Menurut (Antonio Mucherino 2009) dalam bukunya yang berjudul "*k-Nearest Neighbor Classification*". *K-Nearest Neighbor* (K-NN) adalah suatu metode yang menggunakan algoritma *supervised* dimana hasil dari query instance yang baru diklasifikasikan berdasarkan mayoritas dari label class pada K-NN. Tujuan dari algoritma K-NN adalah mengklasifikasikan objek baru berdasarkan atribut dan training data.

Algoritma K-NN bekerja berdasarkan jarak terpendek dari query instance ke training data untuk menentukan K-NN. Salah satu cara untuk menghitung jarak dekat atau jauhnya tetangga menggunakan metode *Euclidian Distance*, sering digunakan untuk menghitung jarak sehingga berfungsi menguji

ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua obyek. *Euclidian Distance* dirumuskan dalam persamaan (2.1).

$$= \left( \sum_{k=1}^m (x_{ik} - x_{jk})^2 \right)^{1/2} \quad (2.1)$$

Dimana:

$X_{ik}$  = nilai X pada training data

$X_{jk}$  = nilai X pada testing data

m = batas jumlah banyaknya data

Jika hasil nilai dari rumus di atas besar maka akan semakin jauh tingkat keserupaan antara kedua objek dan sebaliknya jika hasil nilainya semakin kecil maka akan semakin dekat tingkat keserupaan antar objek tersebut. Objek yang dimaksud adalah training data dan testing data.

Dalam algoritma ini, nilai k yang terbaik itu tergantung pada jumlah data. Ukuran nilai k yang besar belum tentu menjadi nilai k yang terbaik begitupun juga sebaliknya.

Langkah-langkah untuk menghitung algoritma k-NN:

1. Menentukan nilai k.
2. Menghitung kuadrat jarak euclid (*query instance*) masing-masing objek terhadap training data yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak euclid terkecil.
4. Mengumpulkan label class Y (klasifikasi *Nearest Neighbor*).
5. Dengan menggunakan kategori *Nearest Neighbor* yang paling mayoritas maka dapat diprediksikan nilai query instance yang telah dihitung.

### 2.3 Metode Algoritma Fuzzy C-Means

Ada beberapa algoritma clustering data, salah satu diantaranya adalah Fuzzy C-Means (FCM). Fuzzy C-Means (FCM) adalah salah satu teknik pengclusteran data yang mana keberadaan tiap-tiap titik data dalam suatu cluster ditentukan oleh derajat keanggotaan. Teknik ini pertama kali diperkenalkan oleh Jim Bezdek pada tahun 1981. (Kusumadewi, 2010).

Konsep dasar FCM, pertama kali adalah menentukan pusat cluster, yang akan menandai lokasi rata-rata untuk tiap-tiap cluster. Pada kondisi awal, pusat cluster ini masih belum akurat. Tiap-tiap titik data memiliki derajat keanggotaan untuk tiap-tiap cluster. Cara memperbaiki pusat cluster dan derajat keanggotaan tiap-tiap titik data secara berulang, maka akan dapat dilihat bahwa pusat cluster akan bergerak menuju lokasi yang tepat. Perulangan ini didasarkan pada minimisasi fungsi objektif yang menggambarkan jarak dari titik data yang diberikan ke pusat cluster yang terbobot oleh derajat keanggotaan titik data tersebut. Output dari FCM bukan merupakan fuzzy inference system, namun merupakan deretan pusat cluster dan beberapa derajat keanggotaan untuk tiap-tiap titik data. Informasi ini dapat digunakan untuk membangun suatu fuzzy inference system. (Kusumadewi, 2010).

Algoritma Fuzzy C-Means (FCM) adalah sebagai berikut:

1. Input data yang akan di cluster  $X$ , berupa matriks berukuran  $n \times m$  ( $n$ = jumlah sampel data,  $m$  = atribut setiap data).  $X_{ij}$  = data sampel ke- $i$  ( $i=1,2,\dots,n$ ), atribut ke- $j$  ( $j=1,2,\dots,m$ ).

2. Langkah selanjutnya ialah menentukan beberapa input yang dibutuhkan dalam perhitungan fuzzy c-means, yaitu:
  - a. Jumlah cluster (c) ialah banyaknya cluster yang akan dibentuk sesuai dengan kebutuhan pengclusteran.
  - b. Pangkat (w) ialah nilai eksponen. o Maksimum iterasi (MaxIter) merupakan batas pengulangan atau looping. Looping akan berhenti jika nilai maksimal iterasi sudah tercapai.
  - c. Error terkecil ( $\xi$ ) berupa batasan nilai yang membuat perulangan akan berakhir setelah didapatkan nilai error yang diharapkan. o Fungsi objektif awal ( $P_0 = 0$ ) ialah suatu fungsi yang akan dioptimumkan (maksimum atau minimum), nilai 0 berarti untuk mendapatkan nilai minimum. o Iterasi awal ( $t = 1$ ), iterasi adalah adalah sifat tertentu dari algoritma atau program komputer di mana suatu urutan atau lebih dari langkah algoritmik dilakukan secara berulang. Iterasi awal ialah pada perulangan keberapakan program akan dimulai.
3. Membangkitkan bilangan random  $\mu_{ik}$ ,  $i=1,2,\dots,n$ ;  $k=1,2,\dots,c$ ; sebagai elemen-elemen matrik partisi awal U. Hitung jumlah setiap kolom:
 
$$Q_i = \sum_{k=1}^c \mu_{ik}$$
 $Q_i$  ialah jumlah setiap kolom dari nilai random sebuah matirk, jumlah Q tergantung dari berapa jumlah kriteria penilaian.
4. Hitung pusat cluster ke-k:  $V_{kj}$ , dengan  $k=1,2,\dots,c$ ; dan  $j=1,2,\dots,m$

$$V_{kj} = \frac{\sum_{i=1}^n ((\mu_{jk})^w * X_{ij})}{\sum_{i=1}^n ((\mu_{jk})^w)}$$

$V_{kj}$  ialah titik pusat tiap cluster, jumlah  $V_{kj}$  tergantung dari berapa cluster yang akan dibentuk dan n ialah jumlah proposal.

5. Menghitung fungsi objektif pada iterasi ke-t,  $P_t$

$$V_{(t)} = \sum_{i=1}^n \sum_{k=1}^c \left( \left[ \sum_{j=1}^m (X_{ik} - V_{kj})^2 \right] (\mu_{ik})^w \right)$$

t merupakan iterasi yang dihitung, jika iterasi dimulai dari 1 maka pada awal perhitungan nilai t ialah 1. Iterasi akan berulang sesuai dengan ketentuan iterasi yang sedang berjalan. Hitung perubahan matrik partisi.

$$\mu_{ik(t)} = \frac{\left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right] \frac{-1}{w}}{\sum_{k=1}^c \left[ \sum_{j=1}^m (X_{ij} - V_{kj})^2 \right] \frac{-1}{w}}$$

Iterasi akan tetap berulang jika nilai atau kondisi-kondisi tertentu belum tercapai, adapun kondisi tersebut ialah jika:  $(|P_t - P_{t-1}| < \xi)$  atau  $(t > \text{MaxIter})$  maka berhenti yang mana  $P_t$  ialah pusat cluster iterasi ke t kurang dari nilai error yang diharapkan atau jika t (jumlah iterasi) sudah lebih besar daripada iterasi maksimum. Namun jika iterasi akan diulang lagi dengan t + 1 akan mengulang proses yang ke-4 atau menghitung pusat cluster lagi. (Kusumadewi, 2010).

## 2.4 Website

*Web* merupakan suatu ruang informasi di mana sumber-sumber daya yang berguna diidentifikasi oleh pengenal global yang disebut *Uniform User Identifier* (URL). URL dapat diibaratkan suatu alamat, dimana alamat tersebut terdiri atas:

1. Protokol yang digunakan oleh suatu *browser* untuk mengambil informasi.
2. Nama computer (*server*) dimana informasi tersebut berada.
3. Jalur atau *path* serta nama *file* dari suatu informasi.

Format umum dari URL adalah sebagai berikut :

Protokol\_transfer://nama\_host/path/nama\_file.

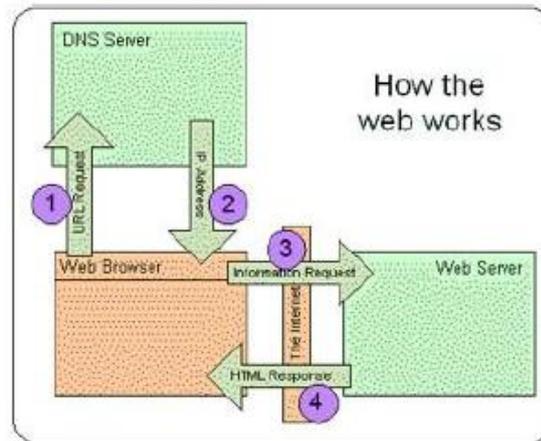
Contoh: <http://www.mine.com/e-journal/index.html>

Dari contoh tersebut dapat disimpulkan bahwa:

1. http adalah protocol yang digunakan.
2. www.mine.com adalah nama *host* atau *server* komputer dimana informasi yang dicari berada.
3. *e-journal* adalah jalur atau *path* dari informasi yang dicari.
4. Index.html adalah nama *file* dimana informasi tersebut berada.

Sebuah halaman *web* diakses dengan menggunakan *web browser* dengan menuliskan URL nya atau mengikuti *link* yang menuju kepadanya. *Uniform Resource Locator* (URL) akan menunjukkan lokasi dokumen yang dikelola oleh sebuah *web server*. URL diubah menjadi alamat IP *server web* yang bersangkutan. *Browser* kemudian mengirimkan request *Hypertext Transfer Protocol* (HTTP) ke *web server* dan *web server* akan menjawab dokumen yang diminta dalam format *Hypertext Markup Language* (HTML). HTTP adalah suatu protocol yang menentukan aturan yang perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan oleh *web server* dalam menyediakan dokumen yang diminta *web browser*. Protokol ini merupakan protocol standar yang digunakan untuk mengakses dokumen HTML. Dalam komunikasi jaringan computer diatur dengan protocol yang memungkinkan beragam jaringan computer untuk berkomunikasi. Protokol ini secara resmi dikenal dengan *Transmission Control Protocol* (TCP/IP) yang merupakan cara untuk mempacketkan sinyal

elektronik sehingga data tersebut dapat dikirim ke computer lain. Gambar di bawah adalah menggambarkan cara kerja *web* yang diakses.



**Gambar 2. 1 Cara Kerja Web Yang di Akses**

*Web* dapat dikategorikan menjadi dua yaitu *web* statis dan *web* dinamis atau interaktif. *Web* statis adalah *web* yang menampilkan informasi-informasi yang sifatnya statis atau tetap, sedangkan *web* dinamis adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan user yang sifatnya dinamis. Untuk membuat *web* dinamis dibutuhkan pemrograman *web* yang mempunyai dua kategori, yaitu :

1. *Server-side Programming*

Perintah-perintah program atau *script* dijalankan di *web server*, kemudian hasilnya dikirimkan ke *browser* dalam bentuk HTML.

2. *Client-side Programming*

Perintah program dijalankan di *web browser* sehingga ketika *client* meminta dokumen yang mengandung *script*, maka *script* tersebut akan *download* dari *server*nya kemudian dijalankan di *browser* yang bersangkutan.

## 2.5 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, 2015). *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.

## 2.6 PHP

Menurut Kadir (2015) “*PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. *PHP* disebut juga pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. *PHP* adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan *open source* yaitu pengguna data mengembangkan kode-kode fungsi

sesuai kebutuhannya.

Menurut Shalahuddin (2015:22) “*PHP (Perl Hypertext Preprocessor)* adalah bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis”. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Kelebihan-kelebihan dari *PHP* yaitu:

1. *PHP* merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. *PHP* dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti *IIS* atau *PWS* juga pada *apache* yang bersifat *open source*.
3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada *PHP* lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembangannya.
4. Jika dilihat dari segi pemahaman, *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.

*PHP* dapat berjalan pada 3 operating sistem, yaitu: *Linux*, *unix*, dan *windows*, dan juga dapat dijalankan secara *runtime* pada suatu *console*.

## **2.7 Unified Modeling Language**

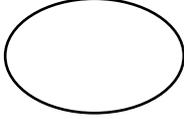
*UML (Unified Modeling Language)* merupakan metodologi kolaborasi antara metoda – metoda *booch*, *OMT (Object Modeling Technique)* serta *OOSE (Object Oriented Engineering)* dan beberapa metoda lainnya merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa “pemograman berorientasi objek” (OOP). ( Adi Nograho; 2015 : 4).

a. *Usecase Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. Adapun notasi *use case diagram* dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Tipe Relasi pada *Usecase Diagram*

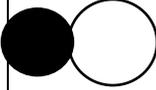
Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>usecase</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>usecase</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>usecase</i> sumber pada suatu titik

		yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Usecase</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Activity Diagram*

*Activity diagram* atau Diagram Aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Sebuah *activity diagram* memiliki :

**Tabel 2.2** Tipe Relasi pada *Activity Diagram*

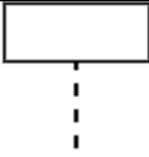
Simbol	Penjelasan
	<i>Initial State</i> Mempresentasikan dimulainya alur kerja suatu sistem dalam <i>activity diagram</i> .
	<i>Fork</i> Adanya percabangan paralel dari aktivitas
	<i>Final State</i> Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam <i>activity diagram</i> .

c. *Sequence Diagram*

Merupakan diagram yang menunjukkan aliran fungsionalitas dalam *usecase*. *Sequence* diagram adalah satu dari dua interaksi diagram yang mengilustrasikan objek-objek yang berhubungan dengan *usecase* dan *message* atau pesan-pesannya. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

**Tabel 2.3** Komponen *Sequence Diagram*

Simbol	Nama Elemen	Keterangan
--------	-------------	------------

	Objek	Komponen yang menjadikan sebuah objek dalam membuat diagram
	<i>Stimulus</i>	Untuk menandakan hubungan komunikasi antar objek
	<i>Self Stimulus</i>	Fungsi sama dengan <i>Stimulus</i> , tetapi pesan yang disampaikan dikirimkan untuk objek itu sendiri
	<i>Focus Control</i>	Sebagai tempat untuk hasil input atau output dari sebuah proses yang dilakukan oleh objek ataupun aktor yang ada dalam sistem

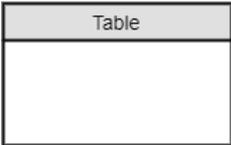
d. *Class Diagram*

Diagram kelas atau *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama. Disamping itu diagram kelas bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasinya satu dengan lainnya. Itulah sebabnya diagram kelas menjadi diagram yang paling populer di *UML*. Komponen diagram kelas adalah :

1. Asosiasi adalah *class-class* yang berhubungan satu sama lain secara konseptual. yaitu menghubungkan dua kelas menjadi satu asosiasi.
2. Atribut adalah properti dari sebuah kelas. Atribut ini menjelaskan batas nilai yang mungkin ada pada obyek dari kelas. Sebuah kelas mungkin mempunyai nol atau lebih atribut. Contoh dari atribut :

3. Operasi adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang anda (atau *class* yang lain) dapat lakukan untuk sebuah *class*. Sama halnya dengan atribut, kita bisa juga memberikan tambahan informasi untuk operasi dengan menambahkan parameter yang akan dilakukan oleh operasi dengan tanda kurung.

**Tabel 2.4** Komponen *Class Diagram*

Simbol	Nama Elemen	Keterangan
	Kelas	Simbol untuk membangun sebuah pemrograman dengan objek Terdiri 3 bagian, bagian <b>atas</b> adalah nama kelas, bagian <b>tengah</b> adalah atribut, dan bagian <b>bawah</b> adalah metode dari kelas tersebut
	Garis Asosiasi	Simbol yang menggambarkan adanya hubungan antara satu kelas dengan kelas lainnya
	Generalisasi	Simbol yang menandakan adanya generalisasi dari kelas input untuk menghasilkan data yang dibutuhkan

## 2.8 Internet

Jaringan antar komputer yang saling berkaitan secara terus menerus baik melalui *email*, *transmisi file*, dan komunikasi dua arah antar individu atau kelompok. *Internet* atau *international network* merupakan rangkaian jaringan terbesar didunia dimana semua jaringan yang berada pada semua organisasi dihubungkan dengan suatu jaringan terbesar sehingga dapat saling berkomunikasi. (Mulyanto, 2009)

Menurut (Sibero, 2013), *Internet (interconnected network)* adalah jaringan komputer yang menghubungkan antar jaringan secara *global*, *internet* dapat juga disebut jaringan dalam suatu jaringan yang luas.

Sedangkan Menurut (Afrianto, 2011), *Internet* merupakan kependekan dari kata "*Internetwork*", yang berarti rangkaian komputer yang terhubung menjadi beberapa rangkaian jaringan. Berdasarkan pendapat yang dikemukakan dapat disimpulkan bahwa, *Internet* adalah suatu rangkaian atau jaringan komputer yang menghubungkan jaringan komputer dalam suatu jaringan yang luas.