

BAB II

LANDASAN TEORI

2.1 Dasar Teori

2.1.1 Sistem Informasi Pemetaan

Sistem merupakan bagian-bagian yang saling berkaitan dan beroperasi bersama untuk mencapai beberapa sasaran atau tujuan (Nugraha, 2012). Sedangkan informasi merupakan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Dari kedua pengertian di atas, dapat disimpulkan bahwa sistem informasi merupakan komponen yang beroperasi secara bersama-sama untuk mengolah data untuk menjadi sebuah informasi yang dapat diterima oleh user. Sistem informasi sendiri memiliki banyak ragam, seperti Sistem Informasi Manajemen (SIM), Sistem Informasi Akademik (SIK), Sistem Informasi Desa (SID), Sistem Informasi Pemetaan, dan masih banyak lagi.

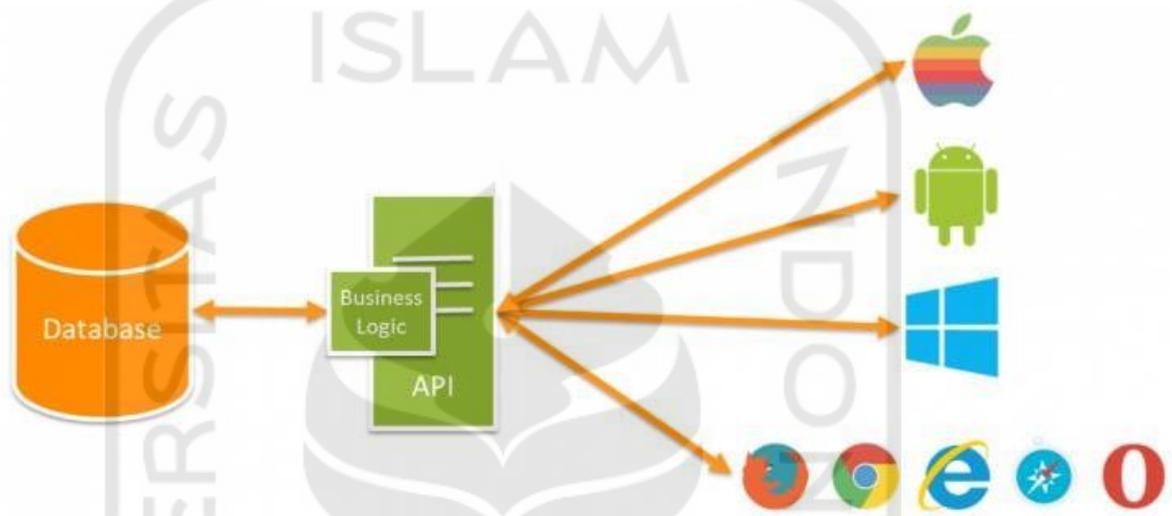
Sistem informasi dapat diimplementasikan dan dimanfaatkan oleh berbagai bidang keilmuan dan institusi, seperti bidang pendidikan, bidang pemerintahan, bidang kedokteran, hingga bidang pemetaan wilayah. Sistem informasi pemetaan sendiri identik dengan sebuah sistem yang biasa disebut Sistem Informasi Geografis (SIG). Hal tersebut dikarenakan kedua sistem informasi tersebut menggunakan data wilayah dan geografis untuk memberikan bentuk digital dan analisis. Sistem informasi pemetaan juga sering digunakan untuk memberikan hasil analisis terhadap bencana yang terjadi di sebuah wilayah.

Sistem informasi pemetaan telah memberikan banyak manfaat terhadap masyarakat dan pemerintahan, seperti implementasi aplikasi sistem informasi pemetaan geografis tata ruang di DKI Jakarta. Berdasarkan informasi yang diterbitkan pada portal berita online beritajakarta.id, bahwa Kepala Dinas Cipta Karya, Tata Ruang, dan Pertanahan (CKTRP) DKI Jakarta, Benny Agus Chandra mengatakan, bahwa sistem informasi tersebut digunakan sebagai media bagi publik yang membutuhkan data dan informasi ketataruangan, pertanahan, dan pembangunan gedung di Ibukota (DKI Jakarta) (Surapati & Tristanto, 2017).

2.1.2 Twitter API (Application Programming Interface)

API (*Application Programming Interface*) merupakan sebuah *interface* yang dapat diimplementasikan dengan menggunakan perangkat lunak (*software*) sehingga perangkat

lunak tersebut dapat berinteraksi dengan perangkat lunak lainnya, seperti halnya tampilan interface user yang memungkinkan *user* untuk berinteraksi dengan komputer (Prasetyadi, 2011). Dengan memanfaatkan API, *developer* dapat memanfaatkan beberapa perangkat lunak untuk melakukan suatu proses. Selain itu, tujuan dari useran API adalah mempercepat proses pengembangan sebuah sistem atau aplikasi dengan menggunakan fungsi-fungsi secara terpisah, sehingga *developer* tidak perlu membuat fungsi atau fitur yang serupa. Cara kerja API dapat dilihat pada gambar 2.1.



Gambar 2.1 Cara kerja API

API bekerja dengan cara membantu aplikasi berinteraksi dengan *library* dengan mengikuti serangkaian aturan yang ditentukan sebelumnya oleh API itu sendiri. Pendekatan ini memudahkan *developer* untuk membuat aplikasi yang berkomunikasi dengan berbagai *library* tanpa harus memikirkan kembali strategi yang digunakan selama semua *library* mengikut API yang sama. Kelebihan lain dari metode ini menunjukkan betapa mudahnya menggunakan *library* yang sama dengan bahasa pemrograman yang berbeda. Salah satu contoh pemanfaatan API adalah REST (*Representational State Transfer*) API dari Twitter, di mana API tersebut menyediakan akses *read* dan *write* data dengan mengintegrasikan Twitter ke dalam aplikasi kita sendiri.

Twitter API juga dapat memiliki 3 bagian, yakni *Search API* yang dapat memudahkan *user* dalam mengelola *query search* pada konten twitter. *Keyword* dapat dicari dengan berdasar pada kata khusus atau mencari tweet yang lebih spesifik berdasarkan *username* Twitter. Selanjutnya, *Representational State Transfer (REST) API* yang digunakan untuk membangun sebuah aplikasi Twitter yang lebih kompleks dan memerlukan inti dari Twitter.

Dan terakhir adalah *Streaming API* yang digunakan *developer* untuk kebutuhan yang lebih intensif. Selain itu, *streaming API* dapat menghasilkan aplikasi yang mampu mengetahui tweet dan pengikut dari sebuah akun.

Menurut artikel yang diterbitkan oleh Anugrah Sandi pada *portal codepolitan*, dibutuhkan beberapa hal untuk membuat API (Sandi, 2017), di antaranya:

1. Mendukung fungsi CRUD yang bekerja melalui HTTP *protocol* dengan *method* GET, POST, PUT dan DELETE.
2. Memiliki *response Accept Header* dan HTTP status code.
3. *Response* dengan format JSON, XML atau format apapun yang diinginkan. Akan tetapi ke banyak digunakan ke dalam format JSON.
4. Mendukung fitur MVC (*Model View Controller*) seperti *routing*, *controllers*, *action results*, *filter*, model, IOC *container*, dan lain-lain.
5. Web API dapat berjalan di *Apache* atau *web server* lainnya yang didukung sesuai bahasa pemrograman yang digunakan.

2.1.3 Google Maps API

Google Maps merupakan sebuah layanan peta digital gratis yang disediakan oleh Google dan bersifat *open-source*. Karena bersifat *open-source*, *Google Maps* dapat dikembangkan sesuai dengan keinginan *developer* dengan menggunakan *library* yang telah disediakan oleh Google. Selain itu, *Google Maps* juga dapat diintegrasikan ke dalam aplikasi-aplikasi baik berbasis *mobile* maupun berbasis website dengan menggunakan *Google Maps API*.

Google Maps API merupakan sebuah layanan yang diberikan oleh Google kepada para pengguna untuk memanfaatkan *Google Maps* dalam mengembangkan aplikasi (Masykur, 2014). Pada *Google Maps* terdapat 4 jenis model peta (Masykur, 2014), yakni:

1. ROADMAP, jenis model peta untuk menampilkan peta 2 dimensi.
2. SATELLITE, jenis model peta untuk menampilkan peta dari hasil foto satelit.
3. TERRAIN, jenis model peta untuk menunjukkan relief fisik permukaan bumi. Contohnya menunjukkan bentuk fisik pegunungan atau sungai.
4. HYBRID, jenis model peta yang akan menunjukkan peta dari hasil foto satelit dan terdapat nama kota beserta alamat di dalamnya.

Untuk menggunakan *Google Maps API*, diperlukan token (*API Keys*) yang nantinya akan disisipkan pada code aplikasi atau sistem yang dibangun. Token atau *API Keys*

merupakan kode unik yang telah digenerasikan oleh Google untuk memberikan akses terhadap *Google Maps* API. Token tersebut didapatkan dari halaman resmi *Google Maps*. *Google Maps* API sendiri memungkinkan user atau developer untuk menampilkan peta suatu wilayah, memberikan arahan rute tercepat kepada user, menampilkan alamat dan posisi latitude longitude dari sebuah tempat, dan dapat memberikan perhitungan waktu dari satu rute ke rute lainnya.

2.1.4 Text Mining

Text Mining merupakan proses data mining untuk mencari pola dalam teks. Proses tersebut berguna untuk mencari informasi yang bermanfaat untuk tujuan tertentu. *Text Mining* bertujuan untuk mencari kata-kata dan mendapatkan informasi yang berguna. Di mana informasi tersebut mampu mewakili isi dari dokumen yang berkaitan sehingga dapat dilakukan analisis yang berhubungan antar dokumen.

Sumber dokumen yang digunakan merupakan kumpulan teks dengan format yang tidak terstruktur atau kurang terstruktur. Proses *text mining* memerlukan beberapa tahap awal, yakni mempersiapkan supaya teks dapat diubah menjadi lebih terstruktur.

2.1.5 Klasifikasi

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 basis data yang besar (Ridwan, Suyono, & Sarosa, 2013). Data mining memiliki tujuan untuk menemukan, menggali, dan mengolah data atau informasi yang dimiliki. Data Mining juga memiliki berbagai macam teknik, di antaranya K-Means dan NBC (*Naïve Bayes Classifier*).

Menurut Prasetyo (2012), klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu pembangunan model sebagai *prototype* untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

Destuardi dan Surya Sumpeno menjelaskan pada papernya tahun 2009, bahwa klasifikasi merupakan proses pengelompokkan dokumen ke dalam kelas berbeda. Di dalam prosesnya, tiap dokumen menunjuk pada satu kelas tertentu, maka dibutuhkan proses untuk

menggali informasi dari dokumen tersebut sehingga dokumen tersebut mampu mempresentasikan dari kelasnya agar setiap kata yang muncul di dalam dokumen dapat mempunyai nilai. Secara sederhananya, klasifikasi memiliki tugas untuk memilih label kelas yang benar untuk *input* yang diberikan.

Dalam tugas–tugas klasifikasi dasar, setiap input dianggap terpisah dari semua input lainnya dan himpunan label ditentukan sebelumnya. Klasifikasi memiliki 2 metode, yakni *Hand-coded rules* dan *Supervised machine learning*. *Hand-coded rules* merupakan metode yang tidak membutuhkan data *training*. Metode ini memiliki hasil akurasi tinggi namun membutuhkan waktu yang sangat lama. Sedangkan *supervised machine learning* melakukan labeling secara otomatis dengan metode oleh mesin dan menggunakan data *training* yang sudah diberikan label.

Pekerjaan utama dari klasifikasi yakni, pembangunan model sebagai *prototype* untuk disimpan sebagai memori dan penggunaan model tersebut untuk melakukan pengenalan atau prediksi pada suatu objek data lainnya agar diketahui di kelas mana objek data tersebut dalam model yang sudah disimpannya.

2.1.6 Naïve Bayes Classification (NBC)

Naïve Bayes Classification (NBC) merupakan teknik klasifikasi berdasarkan *Teorema Bayes* dengan asumsi independensi di antara para prediktor. *Naïve Bayes Classifier* memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes*. Dalam istilah sederhana, penggolongan *Naive Bayes* menganggap bahwa kehadiran fitur tertentu di kelas tidak terkait dengan kehadiran fitur lainnya (Hidayatullah, 2014). Kelebihan yang dimiliki oleh NBC yakni dalam kesederhanaan komputasi, hanya membutuhkan jumlah data pelatihan (*training data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Selain itu, kecil kemungkinan terjadi *over-fitting* atau kondisi yang memungkinkan terjadi proses generalisasi. Namun *Naïve Bayes* memiliki kelemahan dalam asumsi yang sulit dipenuhi, yaitu independensi *feature* kata.

Metode ini dapat digunakan untuk berbagai macam kebutuhan, seperti klasifikasi dokumen, deteksi maupun filtering spam, serta berbagai klasifikasi lainnya. Klasifikasi *Naïve Bayes* mendasar pada teori yang memandang semua fitur dari data sebagai bukti dalam suatu probabilitas. Kaitan *Naïve Bayes* dengan klasifikasi, korelasi, serta bukti klasifikasi adalah hipotesis dalam *Teorema Bayes* merupakan label kelas yang nantinya menjadi tujuan dari

pemetaan dalam klasifikasi. Bukti–bukti merupakan fitur yang nantinya akan menjadi masukan dalam model klasifikasi.

Ciri utama *Naïve Bayes* memiliki asumsi yang sangat kuat (naif) dari independensi masing–masing kejadian atau kondisi. Rumus Bayes dalam (Muslehatin dkk, 2017) secara umum dapat diberikan sebagai berikut:

$$P(H|X) = \frac{P(H|X).P(H)}{P(X)}$$

Keterangan:

X	= Data dengan class yang belum diketahui
H	= Hipotesis data X merupakan suatu class spesifik
P(H X)	= Probabilitas hipotesis H berdasarkan kondisi x (posteriori prob.)
P(H)	= Probabilitas hipotesis H (prior prob.)
P(X H)	= Probabilitas X berdasarkan kondisi tersebut
P(X)	= Probabilitas dari X

Setelah melakukan klasifikasi, selanjutnya akan dilakukan pengukuran evaluasi performa atau yang sering disebut *Performance Evaluation Measure* (PEM). Sesuai dengan namanya, *Performance Evaluation Measure* (PEM) merupakan tahapan yang digunakan untuk mengukur performa dari suatu sistem. *Performance Evaluation Measure* (PEM) cukup sering digunakan dalam *training* data yang bertujuan untuk mengevaluasi model yang telah dibuat. Perhitungan untuk mendapatkan *Performance Evaluation Measure* (PEM) biasanya diterapkan secara kombinasi ataupun secara parsial. Beberapa perhitungan *Performance Evaluation Measure* (PEM) di antaranya:

1. *Accuration*

Accuration merupakan perbandingan antara informasi yang dijawab oleh sistem dengan benar dan keseluruhan informasi. Atau secara gampangnya, *accuration* merupakan pengecekan ketepatan.

Accuration didapat dari rumus:

$$\frac{TP + TN}{total}$$

2. Precision

Precision memiliki fungsi untuk mengetahui seberapa besar akurasi nilai yang diprediksi dengan benar berdasarkan seluruh hasil prediksi yang dianggap benar.

Precision didapat dari rumus:

$$\frac{TP}{TP + FP}$$

3. Recall

Recall memprediksi nilai yang benar diprediksi sebagai *TRUE*, dengan jumlah nilai yang harusnya diprediksi sebaga *TRUE*.

Recall didapat dari rumus:

$$\frac{TP}{TP + FN}$$

4. F1 score

F1 score merupakan perhitungan rata-rata dari perolehan *precision* dengan *recall*. *F1 score* tidak melihat nilai prediksi yang salah (*false positif* dan *false negative*). *F1 score*

Ini cocok digunakan untuk menemukan nilai actual negative yang besar.

Perhitungan *F1 score* didapatkan dari rumus:

$$2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Keterangan:

TP = True Positive

TN = True Negative

FP = False Positie

FN = False Negative

2.2 Penelitian Sebelumnya

Naive Bayes adalah sebuah metode yang berdasarkan pada suatu asumsi penyederhanaan di mana nilai atribut secara kondisional saling bebas apabila diberikan nilai output (Santosa, 2007). *Naive Bayes Classifier* merupakan sebuah metode klasifikasi yang berakar pada teorema Bayes. Ciri utama *Naive Bayes Classifier* adalah asumsi yang sangat kuat (naif) dari independensi masing-masing kondisi atau kejadian (Natalius, 2010). Secara sederhana, NBC mengasumsikan bahwa ada atau tidaknya fitur tertentu dari suatu kelas tidak memiliki keterkaitan dengan keberadaan fitur lainnya. Keuntungan penggunaan NBC adalah hanya diperlukan sejumlah kecil data pelatihan untuk mengestimasi parameter yaitu rata-rata dan varian dari variabel yang diperlukan untuk klasifikasi (Saraswati, 2012). Metode ini merupakan metode probabilistik sederhana yang menerapkan Teorema Bayes.

Penelitian yang dilakukan Faya Mahdia dan Fiftin Noviyanto sebelumnya adalah membangun sebuah sistem informasi manajemen logistik dan memanfaatkan *Google Maps API* dan GPS (*Global Positioning System*) di dalamnya. Sistem tersebut memudahkan pengguna untuk mengetahui posko terdekat. Dalam hal itu, peran GPS sangat diperhatikan. Sehingga, pengguna yang tidak mengizinkan penggunaan GPS akan sulit mendapatkan informasi. Kekurangan dari penelitian ini yakni hanya terbatas di daerah Yogyakarta, sehingga pengguna dari daerah lain tidak dapat menggunakannya. Namun kelebihan yang dapat ditingkatkan dari penelitian ini adalah pemanfaatan *Google Maps API*. Perbandingan penelitian dapat dilihat pada tabel 2.1.

Tabel 2.1 Perbandingan Penelitian

No	Penulis	Objek	Fokus Penelitian	Metode
1.	Ahmad Fathan Hidayatullah (2014)	Twitter	Analisis Sentimen dan Kategori Terhadap Tokoh Publik Pada Data Twitter Menggunakan <i>Naive Bayes Classifier</i> .	Metode <i>Naive Bayes Classifier</i> .

2.	Faya Mahdia dan Fiftin Noviyanto (2013)	Google Maps API	Pemanfaatan <i>Google Maps</i> API Untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis <i>Mobile Web</i> (Studi Kasus : Badan Penanggulangan Bencana Daerah Kota Yogyakarta)	
3	Fauzan Masykur (2014)	Google Maps API	Implementasi Sistem Informasi Geografis Menggunakan <i>Google Maps</i> API dalam Pemetaan Asal Mahasiswa	
4	Agus Qomaruddin Munir 2014	Google Maps	Sistem Informasi Geografi Pemetaan Bencana Alam Menggunakan <i>Google Maps</i>	
5	Sandi Fajar Rodiyansyah dan Edi Winarko (2012)	Twitter	Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan <i>Naive Bayesian Classification</i>	Metode <i>Naive Bayes Classification</i>
6	Amir Hamzah (2012)		Klasifikasi Teks Dengan <i>Naive Bayes Classifier</i> Untuk Pengelompokkan Teks Berita dan <i>Abstract</i> Akademis.	Metode <i>Naive Bayes Classifier</i>