



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

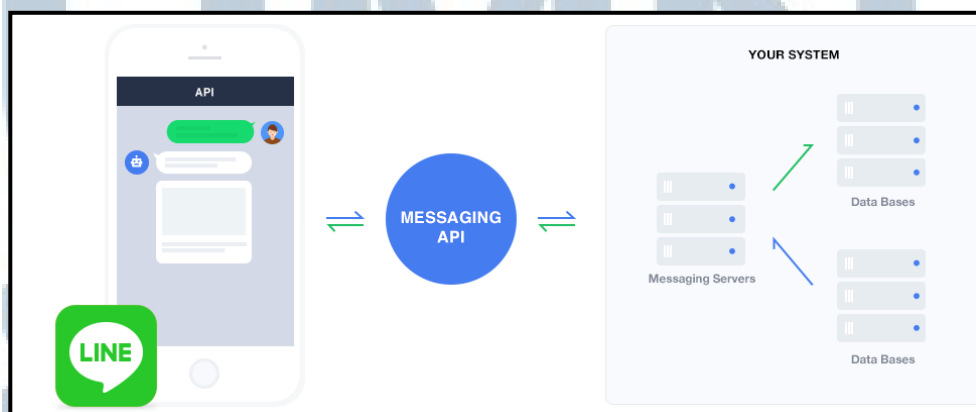
BAB II

TELAAH LITERATUR

2.1 LINE

LINE merupakan sebuah *platform* komunikasi berbasis internet yang menawarkan berbagai layanan, seperti *chatting*, *voice call*, *video call*, dan *smart portal* yang berkaitan dengan konten, hiburan, dan bisnis. LINE@ merupakan salah satu layanan dari LINE yang ditujukan bagi pelaku bisnis untuk meningkatkan serta memberikan informasi kepada konsumen. Fitur yang ditawarkan oleh LINE@ antara lain *broadcast message*, *1:1 chat*, *keyword reply*, dan *auto reply*, dimana fitur-fitur tersebut didukung dengan *Bot API*. Tahun 2016 LINE meluncurkan *messaging API* yang merupakan sebagai pengganti dari *bot API*, dimana *messaging API* memberikan kemudahan bagi pengembang untuk memberikan respon yang interaktif dalam *room chat* atau *group* dari server atau disebut *webhook* (LINE, 2017).

2.1.1 LINE Messaging API



Gambar 2.1 Cara Kerja LINE Messaging API
(Sumber : LINE Developers)

Berdasarkan Gambar 2.1 dapat dijelaskan bahwa dengan *Messaging API*, aplikasi yang dibuat dapat berkirim informasi antara server dengan aplikasi LINE pengguna melalui *platform* LINE. Ketika pengguna LINE menambahkan akun bisnis sebagai teman atau pengguna mengirimkan pesan, *platform* LINE akan mengirimkan notifikasi ke server dengan mengirimkan informasi dalam bentuk JSON melalui HTTP ke URL yang sudah didaftarkan sebagai *webhook*. Selanjutnya server akan mengirimkan respon ke pengguna berdasarkan *request* yang dikirim sebelumnya.

Dalam *LINE Messaging API* terdapat beberapa fitur yang disediakan, yaitu sebagai berikut.

a. *Webhooks*

Webhooks merupakan web server yang berisikan *source code* dari *main program* suatu aplikasi yang akan dibuat. *Webhooks* juga berfungsi untuk menangani *event* HTTP POST yang masuk dari pengguna.

b. *Reply Message*

Fitur ini digunakan untuk membalas pesan yang dikirim oleh pengguna ke akun bisnis yang dibuat. *Event* yang dilakukan oleh pengguna akan dikirimkan ke *webhook*.

c. *Template Messages*

Mengirimkan pesan dengan bentuk dan format yang beragam, seperti gambar, teks, dan tombol pilihan.

2.2 Metode AHP (Analytic Hierarchy Process)

Metode *Analytic Hierarchy Process* (AHP) dikembangkan oleh Prof. Thomas Lorie Satty seperti yang dilansir (Lengkong dkk., 2015) dalam jurnal yang berjudul “*Hybrid Multi Attribute Decision Making untuk Seleksi Penerima Bantuan Studi*” menjelaskan metode tersebut digunakan untuk menyelesaikan suatu masalah yang kompleks. Dalam penelitian yang dilakukan, metode AHP hanya sampai pada tahap pembobotan kriteria dan selanjutnya menggunakan metode VIKOR dalam hal perangkingan. Langkah-langkah dalam metode AHP (Lengkong dkk., 2015) adalah sebagai berikut.

- a. Menyusun kepentingan relatif antara dua buah kriteria ke dalam bentuk matriks perbandingan. Kriteria tersebut diukur berdasarkan skala numerik dengan nilai kepentingan 1 sampai 9. Berikut penjelasan skala kepentingan suatu kriteria.

Tabel 2.1 Skala Kepentingan Kriteria
(Sumber: Lengkong dkk, 2005)

Intentitas Kepentingan	Definisi
1	Sama penting
3	Sedikit lebih penting
5	Lebih penting
7	Sangat penting
9	Mutlak lebih penting
2,4,6,8	Nilai yang mendekati satu sama lain

Bentuk matriks perbandingan kriteria dalam penerapan metode AHP dapat dilihat pada Persamaan (2.1) dimana nilai C_1 , C_2 , dan C_m merupakan kriteria yang disusun berpasangan menjadi matriks A.

$$A = \begin{matrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{matrix} \begin{bmatrix} C_1 & C_2 & \dots & C_m \\ 1 & C_{12} & \dots & C_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \dots & 1 \end{bmatrix} \dots (2.1)$$

- b. Menormalisasikan hasil matriks yang didapat pada Persamaan (2.2) dan menghitung bobot setiap kriteria(w) pada Persamaan (2.3)

$$\sum_i a_{ij} = 1 \quad \dots(2.2)$$

$$w_i = \frac{1}{n} \sum_j a_{ij} \quad \dots(2.3)$$

- c. Menghitung perkalian matriks antara matriks (A) dengan matriks (w^T). Dimana matriks A adalah matriks perbandingan berpasangan yang didapat pada Persamaan (2.1) dan matriks (w^T) adalah nilai bobot yang didapat dari Persamaan (2.3). Selanjutnya menghitung nilai t menggunakan Persamaan (2.4).

$$t = \frac{1}{n} \sum_{i=1}^n \left(\frac{\text{elemen ke-i pada } (A)(w^T)}{\text{elemen ke-i pada } (w^T)} \right) \quad \dots(2.4)$$

Nilai t dan n dimasukkan pada Persamaan (2.5) untuk mendapatkan konsistensi (CI). Dimana nilai t didapatkan dari Persamaan (2.4) dan nilai n adalah banyaknya kriteria.

$$CI = \frac{t-n}{n-1} \quad \dots(2.5)$$

- d. Jikai nilai $CI=0$ maka matriks A konsisten, jika $CI \leq 0,1$ maka matriks A cukup konsisten, dan jika nilai $CI > 0,1$ maka matriks A sangat tidak konsisten.

2.3 Metode VIKOR

Metode VIKOR atau *Vlsekriterijumsko KOmpromisno Rangiranje* merupakan bahasa Serbia yang memiliki arti *Multicriteria Compromise Ranking*. Metode VIKOR merupakan bagian dari metode MCDM yang berfungsi dalam melakukan perangkingan dari multikriteria untuk melihat solusi atau alternatif terdekat sebagai pendekatan kepada solusi ideal dalam perangkingan (Venkata,

2007). Menurut Opricovic dan Tzeng seperti yang dilansir (Imanuwelita dkk., 2015) dalam jurnal yang berjudul “*Penentuan Kelayakan Lokasi Usaha Franchise Menggunakan Metode AHP dan VIKOR*”, tujuan utama dari metode ini adalah untuk menghasilkan solusi kompromi dengan melakukan perangkingan terhadap hasil nilai alternatif dan kriteria yang bertolak belakang. Solusi kompromi memiliki arti bahwa sebuah solusi yang layak atau yang mendekati solusi ideal, dimana kompromi berarti persetujuan yang dibuat dengan mengizinkan satu sama lain. Berikut prosedur perhitungan yang digunakan pada metode VIKOR seperti yang dilansir (Imanuwelita dkk., 2015).

- a. Menyusun kriteria dan alternatif dalam bentuk matriks F, dimana A_i adalah alternatif dari $i=1$ sampai n dan C_{xn} adalah kriteria dari $j=1$ sampai m . Penyusunan matriks A dapat dilihat pada Persamaan (2.6).

$$F = \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} \begin{bmatrix} C_{x1} & C_{x2} & \dots & C_{xn} \\ X_{11} & X_{12} & \dots & X_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{m2} \end{bmatrix} \quad \dots(2.6)$$

- b. Menghitung normalisasi matriks dengan mengikuti Persamaan (2.7).

$$R_{ij} = \frac{(f_i^*) - (f_{ij})}{(f_i^*) - (f_i^-)} \quad \dots(2.7)$$

Keterangan:

R_{ij} = nilai normalisasi matriks

f_{ij} = nilai data sampel i kriteria j

f_i^* = nilai terbaik dalam suatu kriteria

f_i^- = nilai terburuk dalam suatu kriteria

- c. Melakukan normalisasi bobot dengan mengkalikan antara nilai matriks yang sudah dinormalisasi dengan nilai bobot kriteria yang didapat dari metode AHP.

Normalisasi bobot dapat dilihat pada Persamaan (2.8).

$$\text{normalisasi bobot} = (W_j \times R_{ij}) \quad \dots(2.8)$$

Dimana R_{ij} adalah nilai normalitas matriks dan W_j adalah nilai bobot suatu kriteria yang didapat dari metode AHP.

- d. Menghitung *utility measures* dari setiap alternatif dengan menggunakan Persamaan (2.9) dan Persamaan (2.10).

$$S_i = \sum_{j=1}^n w_j \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \quad \dots(2.9)$$

$$R_i = \text{MAX}_j \left[w_j \frac{(f_j^* - f_{ij})}{(f_j^* - f_j^-)} \right] \quad \dots(2.10)$$

Berdasarkan persamaan tersebut, dapat dijelaskan bahwa S_i (*maximum group utility*) dan R_i (*minimum individual regret of the opponent*) adalah *utility measures* yang diukur dari titik terjauh dan titik terdekat dari solusi ideal. Nilai w_j adalah bobot untuk setiap kriteria yang diperoleh dari perhitungan AHP.

- e. Menghitung indeks VIKOR dari setiap alternatif i dengan menggunakan Persamaan (2.11).

$$Q_i = v \left[\frac{S_i - S^*}{S^- - S^*} \right] + 1 (1 - v) \left[\frac{R_i - R^*}{R^- - R^*} \right] \quad \dots (2.11)$$

Berdasarkan Persamaan (2.11), dapat dijelaskan bahwa nilai S^* adalah nilai terkecil dari S_i . Nilai S^- merupakan nilai terbesar dari S_i . Nilai R^* adalah nilai terkecil dari R_i , dan R^- nilai terbesar dari R_i . v adalah bobot yang berkisar antara 0 sampai 1 (umumnya bernilai 0,5). Semakin kecil nilai Q_i (VIKOR) maka semakin baik solusi alternatif yang ada.

2.4 Euclidean Distance

Euclidean distance diperkenalkan oleh Euclid yang merupakan seorang matematikawan asal Yunani. *Euclidean distance* digunakan untuk menghitung jarak antar 2 buah titik yang saling berhubungan antara sudut dan jarak, dengan menggunakan Persamaan (2.12) (Prasetiyo dan Rahmad, 2015).

$$Jarak_{km} = \sqrt{((Lat_A - Lat_B) \times 111,319)^2 + ((Long_A - Long_B) \times 111,319)^2} \quad \dots (2.12)$$

Keterangan:

Lat_A = Titik koordinat *latitude* awal

Lat_B = Titik koordinat *latitude* akhir

$Long_A$ = Titik koordinat *longitude* awal

$Long_B$ = Titik koordinat *longitude* akhir

111,319 = Hasil konversi 1 derajat bumi dalam km

2.5 System Usability Scale

System Usability Scale atau SUS merupakan metode yang digunakan untuk menilai kegunaan dari suatu sistem yang terdiri dari 10 buah pertanyaan dengan 5 pilihan jawaban antara sangat setuju sampai sangat tidak setuju. Metode ini memiliki keuntungan dimana sangat mudah untuk menentukan jumlah partisipan, dapat digunakan dalam sampel yang sedikit, dan hasil yang diberikan *valid* dengan menyimpulkan dapat digunakan atau tidaknya suatu sistem (Brooke, 2013). Berikut daftar pertanyaan yang digunakan dalam metode SUS (Pudjoatmodjo dan Wijaya, 2016).

Tabel 2.2 Daftar Pertanyaan SUS
(Sumber: Pudjoatmodjo dan Wijaya, 2016)

#	Pertanyaan	Skala
1	Saya pikir bahwa saya akan ingin lebih sering menggunakan aplikasi ini	1 s/d 5
2	Saya menemukan bahwa aplikasi ini, tidak harus dibuat serumit ini	1 s/d 5
3	Saya pikir aplikasi ini mudah untuk digunakan	1 s/d 5
4	Saya pikir bahwa saya akan membutuhkan bantuan orang teknis untuk dapat menggunakan aplikasi ini	1 s/d 5
5	Saya menemukan berbagai fungsi di aplikasi ini diintegrasikan dengan baik	1 s/d 5
6	Saya pikir ada terlalu banyak ketidaksesuaian dalam aplikasi ini	1 s/d 5
7	Saya bayangkan bahwa kebanyakan orang akan mudah untuk mempelajari aplikasi ini dengan sangat cepat	1 s/d 5
8	Saya menemukan, aplikasi ini sangat rumit untuk digunakan	1 s/d 5
9	Saya merasa sangat percaya diri untuk menggunakan aplikasi ini	1 s/d 5
10	Saya perlu belajar banyak hal sebelum saya bisa memulai menggunakan aplikasi ini	1 s/d 5

Untuk menghitung nilai SUS dapat dilakukan dengan memberikan nilai bobot dari setiap pertanyaan dimana untuk pertanyaan ganjil (1,3,5,7,9) nilai yang didapat adalah skala yang dipilih dikurangi 1, untuk pertanyaan genap (2,4,6,8,10) nilai yang didapat adalah 5 dikurangi skala. Selanjutnya menjumlahkan semua nilai yang didapat antara pertanyaan ganjil dan pertanyaan genap dan dikalikan dengan 2,5 untuk mendapatkan nilai SUS dari masing-masing responden. Nilai SUS yang didapat memiliki rentang nilai 0-100, dimana jika nilai SUS berkisar 0-50,9 termasuk dalam kategori *not acceptable*, jika nilai SUS 51-70,9 maka termasuk dalam *marginal*, dan jika nilai SUS 71-100 maka termasuk dalam kategori *acceptable*. Untuk mengetahui nilai rata-rata SUS secara keseluruhan dapat dilakukan dengan menjumlahkan nilai SUS tiap responden dan dibagi dengan banyaknya responden (Pudjoatmodjo dan Wijaya, 2016).