



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

TINJAUAN PUSTAKA

2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang bertujuan memperkirakan informasi yang menarik bagi penggunanya dan juga membantu calon konsumen dalam memutuskan barang apa saja yang akan dibelinya (Sanjoyo, 2010). Menurut Chris Boylan, sistem rekomendasi adalah sebuah sistem informasi komputer yang membantu membuat keputusan untuk aktivitas bisnis dan organisasi (Boylan, tanpa tahun). Contoh dari penggunaan sistem rekomendasi adalah rekomendasi buku pada Amazon dan rekomendasi film pada Netflix. Perancangan dari sistem rekomendasi bergantung pada karakteristik data yang tersedia. Sebagai contohnya pada Netflix, data diberikan skala *rating* dari 1 sampai 5. Data tersebut menunjang kualitas interaksi antara pengguna dan *item*. Sebagai tambahan, sistem rekomendasi dapat mengakses data spesifik dari pengguna dan *item* seperti demografis dan deskripsi produk. Sistem rekomendasi memiliki metode yang berbeda-beda untuk memberikan rekomendasi. Berikut metode yang umum digunakan dalam sistem rekomendasi (Melville, 2010).

1. Collaborative Filtering

Sistem *Collaborative Filtering* bekerja dengan cara mengumpulkan timbal balik dalam bentuk *rating* dari item pada cakupan tertentu dan mencari kemiripan pada tingkah laku *rating* beberapa pengguna untuk menentukan rekomendasi terhadap suatu *item*.

2. Content-based

Content-based memberikan rekomendasi dengan cara membandingkan nilai representasi dari data *item* yang ada dengan ketertarikan dari pengguna.

3. Knowledge-based

Knowledge-based bekerja berdasarkan kemiripan pemenuhan kebutuhan pengguna dari sebuah *item*.

4. Hybrid-based

Pada tahun 2010, Melville mengajukan sebuah *framework* rekomendasi dengan menggabungkan *Content-based* dan *Collaborative Filtering*. Cara *Content-based* digunakan untuk mengubah matriks rating pengguna menjadi matriks rating penuh, kemudian *Collaborative Filtering* digunakan untuk menyediakan rekomendasi.

2.2 Multiple Criteria Decision Making (MCDM)

Multi-Criteria Decision Making dibagi menjadi dua bagian, yaitu *Multi Objective Decision Making* (MODM) dan *Multi Attribute Decision Making* (MADM) (Triantaphyllou, 1998).

MODM biasanya digunakan untuk pemecahan masalah-masalah pada ruang kontinu, seperti permasalahan pemrograman matematis (Zimmerman, 1991). Sementara itu, MADM adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari MADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses pemeringkatan yang akan menyeleksi alternatif yang sudah diberikan (Wibowo, 2009).

Menurut Kusumadewi (2006), terdapat beberapa metode yang dapat digunakan untuk menyelesaikan masalah MADM, yaitu sebagai berikut

1. *Simple Additive Weighting Method* (SAW)
2. *Weighted Product* (WP)
3. ELECTRE
4. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
5. *Analytic Hierarchy Process* (AHP)

2.3 Simple Additive Weighting

Simple Additive Weighting (SAW) atau yang sering disebut juga dengan metode penjumlahan terbobot adalah salah satu metode yang paling sering digunakan untuk *Multi Attribute Decision Making* (Gayatri, 2013).

Konsep dasar dari metode ini adalah mencari nilai rata-rata yang didapat dari perhitungan untuk setiap alternatif dengan mengalikan nilai skala yang diberikan untuk atribut dari sebuah alternatif dengan bobot pentingnya atribut tersebut yang diberikan oleh penentu keputusan yang kemudian dijumlahkan nilai objek untuk semua kriteria (Azar, 2000).

Keuntungan dari metode SAW ini adalah ketepatan penilaian karena berdasarkan nilai bobot yang sudah ditentukan untuk setiap atribut yang kemudian akan dilanjutkan dengan proses perbandingan yang sesuai dengan alternatif terbaik yang didapatkan dari sejumlah alternatif lainnya (Rusnadi, 2014).

Menurut penelitian yang dilakukan Usito pada tahun 2013, terdapat delapan tahapan dari implementasi metode *Simple Additive Weighting*.

1. Menentukan alternatif-alternatif yang ada, yaitu A_i .
2. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_j .
3. Memberikan nilai rating kecocokan setiap alternatif kepada setiap kriteria.
4. Menentukan bobot preferensi atau tingkat kepentingan (W) setiap kriteria.

$$W = [W_1, W_2, W_3, \dots, W_j]$$

... (2.1)

5. Membuat tabel rating kecocokan dari setiap alternatif terhadap setiap kriteria yang ada.
6. Membuat matriks keputusan (R) yang dibentuk dari tabel rating kecocokan dari setiap alternatif pada setiap kriteria. Nilai R pada setiap alternatif (A_i) pada setiap kriteria (C_j) yang telah ditentukan, di mana, $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

$$R = \begin{bmatrix} r_{11} & \dots & r_{1j} \\ \vdots & \ddots & \vdots \\ r_{i1} & \dots & r_{ij} \end{bmatrix}$$

... (2.2)

7. Kemudian menghitung matriks keputusan ternormalisasi dengan menghitung nilai rating kinerja ternormalisasi (n_{ij}) dari alternatif A_i pada kriteria C_j .

Untuk kriteria positif:

$$n_{ij} = \frac{r_{ij}}{r_j^{max}} \quad i = 1, \dots, m; j = 1, \dots, n$$

... (2.3)

Untuk kriteria negatif:

$$n_{ij} = \frac{r_j^{min}}{r_{ij}} \quad i = 1, \dots, m; j = 1, \dots, n$$

... (2.4)

Keterangan:

- a. Kriteria keuntungan apabila nilai memberikan keuntungan bagi pengambilan keputusan, sebaliknya kriteria biaya apabila menimbulkan biaya bagi pengambilan keputusan.
 - b. Apabila berupa kriteria keuntungan, maka nilai dibagi dengan nilai dari setiap kolom. Sedangkan untuk kriteria biaya, nilai dari setiap kolom dibagi dengan nilai r_{ij} .
8. Hasil akhir nilai preferensi (V_i) diperoleh dari penjumlahan dari perkalian elemen baris matriks ternormalisasi (n) dengan bobot preferensi (W) yang sesuai dengan elemen kolom matriks (W).

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

... (2.5)

2.4 Zomato API

Zomato merupakan *startup* direktori restoran asal India yang diluncurkan pada tahun 2008. Dengan investasi sebesar 3 triliun Rupiah yang berhasil mereka raih dari berbagai investor, Zomato sudah beroperasi di 23 negara, termasuk Indonesia. Di Indonesia sendiri, sudah ada 30.000 restoran yang terdaftar dalam Zomato (Pratama, 2017).

Di tahun 2011, Zomato merilis API *public*, yang mengizinkan developer untuk mengintegrasikan aplikasi yang dibuat dengan konten yang disediakan oleh

Zomato. Dengan limit 1000 *call* per 24 jam dan maksimal 100 hasil per *call*, developer bisa mendapatkan informasi dari semua restoran yang terdaftar di Zomato. Zomato API juga menyediakan filter untuk menyaring hasil pencarian restoran berdasarkan tipe restoran dan jenis masakan yang disediakan oleh restoran tersebut (Gorev, 2011).

```

restaurant
├── R
│   ├── apikey :
│   ├── id : "7414319"
│   ├── name : "Rainbow Kitchen"
│   └── url : "https://www.zomato.com/jakarta/rainbow-kitchen-pinang?utm_source=api_basic_user&utm_medium=api&utm_campaign=v2.1*"
├── location
│   ├── address : "Mall @ Alam Sutera, Lantai 2, Jl. Jalur Sutera Barat, Pinang, Tangerang"
│   ├── locality : "Mall @ Alam Sutera, Pinang"
│   ├── city : "Tangerang"
│   ├── city_id : 74
│   ├── latitude : "-6.2227194444"
│   ├── longitude : "106.6532972222"
│   ├── zipcode : ""
│   ├── country_id : 94
│   ├── locality_verbose : "Mall @ Alam Sutera, Pinang, Tangerang"
│   ├── switch_to_order_menu : 0
│   ├── cuisines : "Chinese, Dim Sum"
│   ├── average_cost_for_two : 300000
│   ├── price_range : 3
│   ├── currency : "IDR"
│   └── offers
│       ├── thumb : "https://b.zmtcdn.com/data/res_imagery/7414319_RESTAURANT_7500e0f252ddc3f6d6f58865fcd38eb.jpg?fit=around%7C200%3A150&crop=200%3A150%3B%2A%2C%2A"
├── user_rating
│   ├── aggregate_rating : "2.8"
│   ├── rating_text : "Average"
│   ├── rating_color : "FFBA00"
│   └── votes : "7"
├── photos_url : "https://www.zomato.com/jakarta/rainbow-kitchen-pinang/photos?utm_source=api_basic_user&utm_medium=api&utm_campaign=v2.1#tabtop"
├── menu_url : "https://www.zomato.com/jakarta/rainbow-kitchen-pinang/menu?utm_source=api_basic_user&utm_medium=api&utm_campaign=v2.1&openSwipeBox=menu&showMinimal=1#tabtop"
├── featured_image : "https://b.zmtcdn.com/data/res_imagery/7414319_RESTAURANT_7500e0f252ddc3f6d6f58865fcd38eb.jpg"
├── has_online_delivery : 0
├── is_delivering_now : 0
├── deeplink : "zomato://restaurant/7414319"
├── has_table_booking : 0
├── events_url : "https://www.zomato.com/jakarta/rainbow-kitchen-pinang/events#tabtop?utm_source=api_basic_user&utm_medium=api&utm_campaign=v2.1"
├── establishment_types
├── establishment_type
│   └── id : "16"

```

Gambar 2.1 Data Restoran Zomato API

Gambar 2.1 merupakan data restoran yang diberikan oleh Zomato API Service. Beberapa informasi yang diberikan oleh Zomato API Service adalah nama restoran, alamat restoran, jenis makanan yang ditawarkan, estimasi biaya, jumlah *upvote* untuk restoran tersebut, dan *rating* dari restoran.

2.5 Computer Usability Satisfaction Questionnaires

Computer Usability Satisfaction Questionnaires (CUSQ) adalah kuesioner yang dibuat oleh karyawan IBM, James R. Lewis, untuk mengukur kepuasan *user*

dalam menggunakan sebuah sistem atau aplikasi komputer (Lewis, 1995). Lewis mengembangkan kuesioner ini untuk mencari tahu tingkat kepuasan *user* dalam situasi *non-laboratory*.

Kuesioner ini memiliki 19 pertanyaan dan dibagi menjadi empat bagian, setiap bagian menilai aplikasi dari aspek yang berbeda-beda. Pertanyaan nomor 1 sampai 8 mengukur kegunaan aplikasi, pertanyaan nomor 9 sampai 15 mengukur kualitas informasi yang dihasilkan oleh aplikasi, pertanyaan nomor 16 dan 17 mengukur kualitas dari antarmuka aplikasi, dan pertanyaan nomor 18 sampai 19 mengukur kepuasan *user* terhadap aplikasi secara umum. Menurut Diehl (1992), diperlukan paling sedikit 30 *sample* untuk menggunakan aplikasi dan mengisi kuesioner ini. Daftar pertanyaan dari CUSQ dapat dilihat di Tabel 2.1.

Tabel 2.1 Tabel Pertanyaan Kuesioner CUSQ

No	Pertanyaan	Aspek
1	Overall, I am satisfied with how easy it is to use this system.	System Usability & Overall Satisfaction
2	It is simple to use this system.	
3	I can effectively complete my work using this system.	System Usability & Overall Satisfaction
4	I am able to complete my work quickly using this system.	
5	I am able to efficiently complete my work using this system.	
6	I feel comfortable using this system.	
7	It was easy to learn to use this system.	
8	I believe I became proficient in using this system quickly.	Information Quality & Overall Satisfaction
9	The system gives error messages that clearly tell me how to fix problems.	
10	Whenever I make a mistake using the system, I recover easily and quickly.	
11	The information provided with this system is clear.	
12	It was easy to find the information I needed.	
13	The information provided with the system is easy to understand.	

Tabel 2.1 Tabel Pertanyaan Kuesioner CUSQ (Lanjutan)

No	Pertanyaan	Aspek
14	The information is effective in helping me complete my work.	Information Quality & Overall Satisfaction
15	The organization of information on the system screens is clear.	
16	The interface of this system is pleasant.	Interface Quality & Overall Satisfaction
17	I like using the interface of this system.	
18	This system has all the functions and capabilities I expect it to have.	Overall Satisfaction
19	Overall, I am satisfied with this system.	

2.6 Skala Likert

Skala Likert adalah skala pengukuran yang dikembangkan oleh Rensis Likert, sosiolog dari Universitas Michigan. Menurut Uebersax (2006), Skala Likert adalah skala pengukuran untuk objek yang memiliki lebih dari satu atribut untuk dinilai.

Skala Likert adalah salah satu metode yang dapat memetakan data kualitatif menjadi data kuantitatif, sehingga sebuah penelitian kualitatif dapat lebih mudah diolah dan dipetakan ke dalam suatu kesimpulan (Trochim, 2006).

Tabel interpretasi skor Skala Likert dapat dilihat di Tabel 2.2.

Tabel 2.2 Tabel Interpretasi Skor Skala Likert

Interval Nilai	Interpretasi Nilai
Skor < 20%	Sangat Lemah
Skor \geq 20% dan < 40%	Lemah
Skor \geq 40% dan < 60%	Cukup
Skor \geq 60% dan < 80%	Kuat
Skor \geq 80%	Sangat Kuat

2.7 Rumus Haversine

Rumus Haversine adalah rumus untuk menentukan jarak antara dua titik di sebuah permukaan bola dengan memperhitungkan nilai dari bujur dan lintang dari kedua titik tersebut (Baker, 1995). Rumus perhitungannya dapat dilihat di Rumus

2.6.

$$d = r \sqrt{\sin\left(\frac{\Delta\varphi}{2}\right)^2 + \cos\varphi_1 \cos\varphi_2 \sin\left(\frac{\Delta\lambda}{2}\right)^2}$$

... (2.6)

Dengan keterangan notasi sebagai berikut.

d adalah jarak antara kedua titik.

r adalah radius dari bola.

$\Delta\varphi$ adalah selisih dari nilai garis lintang kedua dikurang nilai garis lintang pertama dalam satuan radian.

$\Delta\lambda$ adalah selisih dari nilai garis bujur kedua dikurang nilai garis bujur pertama dalam satuan radian.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA