

NASKAH PUBLIKASI

**SISTEM REKOMENDASI PEMILIHAN LAPTOP
MENGUNAKAN METODE PROMETHEE BERBASIS
ANDROID**



Disusun oleh:

WAHYU ADHE TRIWIBOWO

5150411362

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2020**

NASKAH PUBLIKASI

**SISTEM REKOMENDASI PEMILIHAN LAPTOP
MENGUNAKAN METODE PROMETHEE BERBASIS
ANDROID**

Disusun oleh:

WAHYU ADHE TRIWIBOWO

51504111362



Telah disetujui oleh pembimbing

Rembimbing



Dr. Erik Imran Heri U., M.Kom.

Tanggal: 21/02/2020

SISTEM REKOMENDASI PEMILIHAN LAPTOP MENGUNAKAN METODE PROMETHEE BERBASIS ANDROID

WAHYU ADHE TRIWIBOWO

*Program Studi Informatika, Fakultas Teknologi Informasi dan Elektro
Universitas Teknologi Yogyakarta*

Jl. Ringroad Utara Jombor Sleman Yogyakarta

E-mail : adhedewa1996@gmail.com / wahyu.adhe.triwibowo@gmail.com

ABSTRAK

Dewasa ini banyak laptop canggih di mana-mana, laptop yang memiliki daya tarik tersendiri dan bagus membuat orang tertarik untuk membelinya. Tetapi, kenyataannya ketika orang-orang membeli laptop yang 'bagus' mereka merasa kecewa. Hal ini dikarenakan laptop yang mereka pilih tidak sesuai dengan kebutuhan mereka, bisa jadi karena spesifikasi yang terlalu rendah sehingga mereka kurang nyaman atau justru sebaliknya spesifikasi yang diberikan terlalu tinggi sehingga mereka baru sadar bahwa mereka membuang-buang uang untuk membeli laptop yang mumpuni, padahal ia tidak membutuhkannya. Oleh karena itu, dalam penelitian ini dibuat sebuah sistem aplikasi pendukung keputusan untuk membantu dalam proses pemilihan laptop yang sesuai dengan kebutuhan menggunakan metode Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE). PROMETHEE merupakan suatu metode penentuan urutan (prioritas) dalam analisis multikriteria. Dalam kasus pemilihan laptop, metode PROMETHEE ini cocok digunakan untuk perankingan. Sistem pendukung keputusan akan dibuat menggunakan bahasa pemrograman Dart, framework Flutter dan basis data Firebase Cloud Firestore.

Kata kunci : Laptop, Sistem Pendukung Keputusan, PROMETHEE, Dart, Flutter, Firebase, Cloud Firestore.

1. PENDAHULUAN

Di zaman yang semakin modern ini banyak muncul laptop dengan beragam merek, kualitas serta variasi harga yang semakin kompetitif, hal ini mengakibatkan meningkatnya minat daya beli masyarakat. Namun seringkali masyarakat melakukan pembelian laptop hanya karena tertarik dengan design atau warna tanpa memandang spesifikasi, jeroan dari laptop itu sendiri, serta yang paling penting sesuai dengan kebutuhannya. Hal ini pada akhirnya membuat masyarakat hendaknya lebih selektif dalam memilih produk laptop. Tetapi, dalam hal memutuskan untuk memilih sebuah produk bukanlah hal yang mudah karena banyak faktor yang harus diperhatikan dan dipertimbangkan sebelum dapat memutuskan untuk membeli sebuah laptop. Pada kenyataannya tidak semua orang yang ingin membeli laptop mengetahui apa saja yang menjadi indikator dalam memilih produk laptop yang benar-benar cocok dengan selera dan sesuai dengan kriteria yang mereka butuhkan. Oleh karena itu, dibutuhkan suatu layanan sistem yang dapat mengakomodasi kebutuhan tersebut.

Dengan permasalahan yang dihadapi masyarakat sebagai klien dalam menentukan pemilihan produk laptop inilah maka diperlukan suatu sistem aplikasi pengambilan keputusan yang bisa membantu memberikan solusi berupa rekomendasi dengan pendekatan *Preference Ranking Organization Method for Enrichment Evaluation* (PROMETHEE).

2. LANDASAN TEORI

2.1. Sistem Rekomendasi

Sistem rekomendasi merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna (Ungkawa, et al., 2013). Penerapan rekomendasi didalam sebuah sistem biasanya melakukan prediksi suatu item, seperti rekomendasi film, musik, buku, berita dan lain sebagainya yang menarik user. Sistem ini berjalan dengan mengumpulkan data dari user secara langsung maupun tidak (Fadlil & Mahmudy, 2010).

2.2. Metode Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE)

Menurut Figueira dkk. (2005), PROMETHEE adalah suatu metode penentuan urutan (prioritas) dalam analisis multikriteria. Masalah pokoknya adalah kesederhanaan, kejelasan, dan kestabilan. Dugaan dari dominasi kriteria yang digunakan dalam PROMETHEE adalah penggunaan nilai dalam hubungan outranking. Semua parameter yang dinyatakan mempunyai pengaruh nyata menurut pandangan ekonomi.

Prinsip yang digunakan adalah penetapan prioritas alternatif yang telah ditetapkan berdasarkan pertimbangan $\{\forall i | f_i(.) \in \mathfrak{R}[\text{real world}]\}$ dengan kaidah dasar; $\text{Max}\{f_1(x), f_2(x), f_3(x), \dots, f_j(x), \dots, f_k(x) | x \in \mathfrak{R}\}$ dimana K adalah sejumlah kumpulan alternatif, dan f_i ($i = 1, 2, \dots, K$) merupakan nilai/ukuran relatif kriteria untuk masing-masing alternatif (Suryadi, 1998).

Dalam PROMETHEE disajikan enam bentuk fungsi preferensi kriteria. Hal ini tentu saja tidak mutlak, tetapi bentuk ini cukup baik untuk beberapa kasus. Untuk memberikan gambaran yang lebih baik terhadap area yang tidak sama, digunakan fungsi selisih nilai kriteria antara alternatif $H(d)$ dimana hal ini mempunyai hubungan langsung pada fungsi preferensi P (Suryadi, 1998):

$$\frac{\forall a, b \in A}{f(a), f(b)} \left. \begin{array}{l} f(a) > f(b) \leftrightarrow a P b \\ f(a) = f(b) \leftrightarrow a I b \end{array} \right\} \quad (2.1)$$

Dari rumus persamaan (2.1) diatas adalah untuk semua elemen kriteria a terhadap kriteria b merupakan fungsi a lebih besar dari fungsi b dan sama dengan a preferen b , atau fungsi a terhadap fungsi b adalah fungsi a sama dengan fungsi b implikasi dari fungsi a iner b .

a) Kriteria Biasa (*Usual Criterion*)

$$H(d) \begin{cases} 0 & \text{jika } d = 0 \\ 1 & \text{jika } d \neq 0 \end{cases} \quad (2.2)$$

Pada kasus ini, tidak ada beda (sama penting) antara a dan b jika dan hanya jika $f(a) = f(b)$, apabila nilai kriteria pada masing-masing alternatif memiliki nilai berbeda, pembuat keputusan membuat preferensi mutlak untuk alternatif yang memiliki nilai yang lebih baik.

b) Kriteria Quasi (*Quasi Criterion*)

Pada kasus ini, dua alternatif memiliki preferensi yang sama penting selama selisih atau nilai $H(d)$ dari masing-masing alternatif untuk kriteria tertentu tidak melebihi nilai q , dan apabila selisih hasil evaluasi untuk masing-masing

alternatif melebihi nilai q maka terjadi bentuk preferensi mutlak.

$$H(d) \begin{cases} 0 & \text{jika } -q \leq d \leq q \\ 1 & \text{jika } d < -q \text{ atau } d > q \end{cases} \quad (2.3)$$

Jika pengambil keputusan menggunakan kriteria quasi, pengambil keputusan harus menentukan nilai q , dimana nilai ini dapat menjelaskan pengaruh yang signifikan dari suatu kriteria. Dalam hal ini, preferensi yang lebih baik diperoleh apabila terjadi selisih antara dua alternatif diatas nilai q .

c) Kriteria dengan Preferensi Linier

$$H(d) \begin{cases} d/p & \text{jika } -p \leq d \leq p \\ 1 & \text{jika } d < -p \text{ atau } d > p \end{cases} \quad (2.4)$$

Selama nilai selisih memiliki nilai yang lebih rendah dari p , preferensi dari pembuat keputusan meningkat secara linier dengan nilai d . Jika nilai d lebih besar dibandingkan nilai p , maka terjadi preferensi mutlak.

d) Kriteria Level (*Level Criterion*)

$$H(d) \begin{cases} 0 & \text{jika } |d| \leq q \\ 0,5 & \text{jika } q < |d| \leq p \\ 1 & \text{jika } p < |d| \end{cases} \quad (2.5)$$

Dalam kasus ini, kecenderungan tidak berbeda q dan kecenderungan preferensi p adalah ditentukan secara simultan. Jika d berada diantara nilai q dan p , hal ini berarti situasi preferensi yang lemah ($H(d) = 0,5$).

e) Kriteria dengan Preferensi Linier dan Area yang Tidak Berbeda

$$H(d) \begin{cases} 0 & \text{jika } |d| \leq q \\ (|d| - q)/(p - q) & \text{jika } q < |d| \leq p \\ 1 & \text{jika } p < |d| \end{cases} \quad (2.6)$$

Pengambilan keputusan mempertimbangkan peningkatan preferensi secara linier tidak berbeda sehingga preferensi mutlak dalam area antara dua kecenderungan q dan p .

f) Kriteria Gaussian (*Gaussian Criterion*)

$$H(d) = 1 - \exp\{-d^2/\sigma^2\} \quad (2.7)$$

Fungsi ini bersyarat apabila telah ditentukan nilai σ , dimana dapat dibuat berdasarkan distribusi normal dalam statistik.

Tujuan pembuat keputusan adalah menetapkan fungsi preferensi π_i dan π_k untuk semua kriteria f_i ($i = 1, \dots, k$) dari masalah optimasi kriteria majemuk. Bobot (weight) π_i merupakan ukuran relatif dari kepentingan kriteria f_i ; jika semua kriteria memiliki nilai kepentingan yang sama dalam pengambilan keputusan maka semua nilai bobot adalah sama.

Perhitungan arah preferensi dipertimbangkan berdasarkan nilai indeks *leaving flow* (Φ^+), *entering flow* (Φ^-) dan *net flow*.

- a) *Leaving flow* adalah jumlah dari nilai garis lengkung yang memiliki arah menjauh dari node a dan hal ini merupakan karakter pengukuran *outranking*.
- b) *Entering flow* adalah diukur berdasarkan karakter *outranked* dari a.
- c) *Net flow* adalah selisih antara *leaving flow* dengan *entering flow*.

- PROMETHEE I

Nilai terbesar pada *leaving flow* dan nilai terkecil pada *entering flow* merupakan alternatif terbaik. *Leaving flow* dan *Entering flow* menyebabkan (Suryadi, 1998):

$$\begin{cases} a p^+ b \text{ jika } \Phi^+(a) > \Phi^+(b) \\ a I^+ b \text{ jika } \Phi^+(a) = \Phi^+(b) \end{cases} \quad (2.8)$$

$$\begin{cases} a p^- b \text{ jika } \Phi^-(a) < \Phi^-(b) \\ a I^- b \text{ jika } \Phi^-(a) = \Phi^-(b) \end{cases} \quad (2.9)$$

Dengan menggunakan metode PROMETHEE I masih menyisakan bentuk *incomparable*, atau dengan kata lain hanya memberikan solusi *partial preorder* (sebagian) (Suryadi, 1998).

$$\text{Leaving flow: } \Phi^+(a) = \frac{1}{n-1} \sum_{x \in a} \phi(x, a) \quad (2.10)$$

$$\text{Entering flow: } \Phi^-(a) = \frac{1}{n-1} \sum_{x \in a} \phi(x, a) \quad (2.11)$$

- PROMETHEE II

PROMETHEE II disajikan dalam bentuk *net flow* berdasarkan pertimbangan persamaan:

$$ap_u b \text{ jika } \Phi(a) > \Phi(b) \quad (2.12)$$

$$ap_u b \text{ jika } \Phi(a) = \Phi(b) \quad (2.13)$$

Dengan menggunakan promethee II, informasi bagi pembuat keputusan lebih komplrit dan realistik.

$$\text{Net flow: } \Phi^+(a) - \Phi^-(b) \quad (2.14)$$

Nilai dari *net flow* didapatkan dari jumlah *leaving flow* keseluruhan dikurangi dengan jumlah *entering flow* keseluruhan untuk mendapatkan nilai yang akan dijadikan acuan untuk *ranking* keseluruhan dari alternatif yang ada.

3. METODOLOGI PENELITIAN

3.1 Obyek Penelitian

Datasets yang digunakan dalam penelitian ini adalah *Datasets Laptop Prices: Characteristics and Price for 1300 laptop models* yang didapatkan dari <https://www.kaggle.com/ionaskel/laptop-prices>. Dengan kolom *company*, *product*, *typeName*, *inches*, *screenResolution*, *cpu*, *ram*, *memory*, *gpu*, *operationSystem*, *weight*, *priceEuros* dan *idr* dengan jumlah data lebih dari 1300 laptop, , tampilan *datasets* dapat dilihat pada Gambar 1.

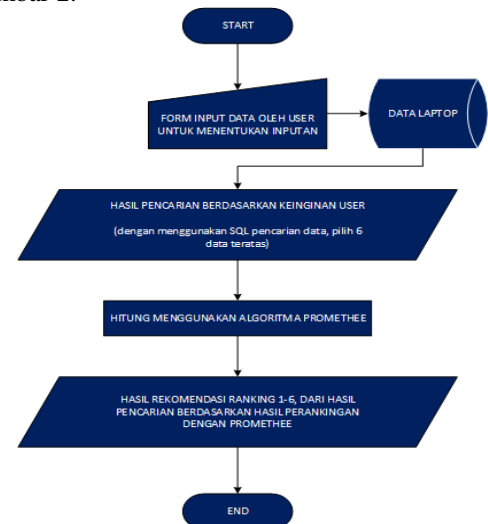
Gambar 1 Datasets Laptop Prices: Characteristics and Price for 1300 laptop models

Perankingan akan dilakukan berdasarkan kriteria yang diinginkan oleh masyarakat (klien) di antaranya, CPU, Harga, *Inches*, *Memory*, RAM dan *Weight*.

3.2 Desain dan Pembuatan Program

a) Flowchart Program

Algoritma proses kerja dari sistem rekomendasi yang akan dibangun terlihat pada Gambar 2.



Gambar 2 Flowchart Algoritma Program

b) Desain

Desain model akan digambarkan menggunakan UML (*Unified Modelling Language*). Model tersebut cocok untuk pengembangan sistem yang berorientasi objek. Bagian *interface* akan dirancangan menggunakan Flutter.

Desain sistem yang dibangun berupa tampilan widget yang berfungsi sebagai media *input* data, halaman data laptop, halaman dashboard, dan hasil rekomendasi berdasarkan perhitungan menggunakan metode PROMETHEE sebagai *output*.

c) Pembuatan Program

Sistem ini akan dibangun dalam bentuk aplikasi mobile. Bahasa pemrograman yang digunakan berupa Bahasa Pemrograman Dart, kemudian menggunakan Flutter untuk mengatur tampilan antarmuka sistem. Sistem akan melakukan penyimpanan data pada Database menggunakan Firebase.

3.3 Implementasi Sistem

Tahap awal Implementasi dilakukan menggunakan Visual Studio Code (VS Code) sebagai editor teks, Bahasa Pemrograman Dart untuk mengembangkan sistem *back-end* dan Flutter untuk membangun *front-end*. Selanjutnya implementasi yang sebenarnya akan dilakukan dengan menginstall aplikasi pada smartphone dengan OS Android.

3.4 Pengujian Sistem

Pengujian Sistem dilakukan untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan tujuan dari dibuatnya sistem. Dengan melakukan pengujian terhadap menu-menu yang terdapat pada sistem, apakah sudah berjalan sesuai dengan fungsinya. Pengujian akan menerapkan metode *black-box* yang berfokus pada fungsionalitas dari sistem yang sudah dibuat, yang nantinya hasil dari sistem akan dicek apakah sudah sesuai dengan harapan atau tidak.

3.5 Perangkat Pendukung Penelitian

Adapun perangkat pendukung berupa perangkat lunak (*Software*) dan perangkat keras (*Hardware*) dengan spesifikasi sebagai berikut:

a) Kebutuhan Perangkat Lunak (*Software*)

- Sistem Operasi : Windows 10 Home
- Pengelola Kata : Microsoft Office Word 365
- Editor Text : Visual Studio Code
- Bahasa Pemrograman : Dart
- Desain : Flutter, Microsoft Office, Visio 2016, Figma

Database : Firebase Cloud Firestore

b) Kebutuhan Perangkat Keras (*Hardware*)

- Merk Komputer : ASUS Strix 15 GL503GE EN129T
- Penyimpanan : 128GB SSD M.2 NVMe, 1T SSHD
- RAM : 8192 MB
- Processor : Intel(R) core(TM) i7-8750H CPU @ 2.20GHz (12 CPUs)

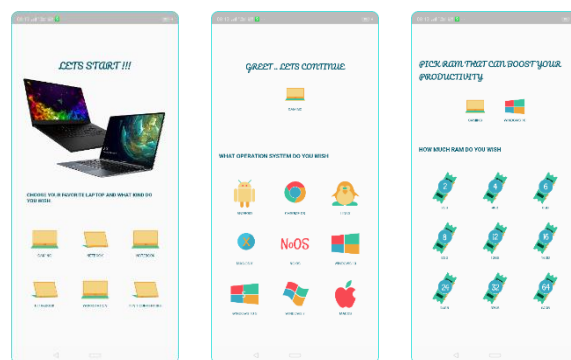
4. HASIL DAN PEMBAHASAN

4.1. HASIL

Berikut adalah proses-proses yang ada untuk mendapatkan hasil perankingan dalam sistem rekomendasi pemilihan laptop:

4.1.1 Proses Penentuan Alternatif

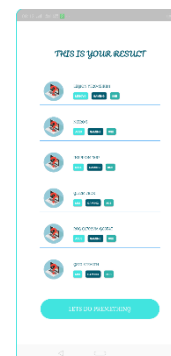
Penentuan alternatif ini akan dilakukan oleh *user* berdasarkan dengan *input-an* yang diberikan ke dalam sistem, yang mana sistem akan mencari data alternatif yang sesuai dengan *input-an user*. Setelah alternatif didapatkan maka sistem akan mengambil nilai bobot dari masing-masing alternatif yang kemudian akan di hitung dengan menggunakan metode PROMETHEE. Alur Proses penentuan alternatif terlihat pada Gambar 3.



Gambar 3 Proses Penentuan Alternatif

4.1.2 Proses Perhitungan Berpasangan

Setelah alternatif didapatkan maka yang dilakukan selanjutnya adalah dengan melakukan perhitungan berpasangan, setiap alternatif memiliki bobotnya masing-masing. Hitung berpasangan ini bertujuan untuk mendapatkan jarak dari alternatif yang dibandingkan. Selanjutnya akan dicari nilai preferensinya.



Gambar 4 Alternatif Yang Didapatkan Dari Input-an User

Dari data yang *input*-kan oleh *user* didapatkan data seperti pada Gambar 4. Data setiap alternatif tersebut memiliki bobot seperti terlihat pada tabel 5.1.

Tabel 1 Bobot Alternatif Yang Didapatkan

Kriteria	Min Maks	Bobot	Alternatif						Tipe Preferensi	Parameter		
			Legion Y520 15IKBN	NITRO 5	INSPIRON 7567	G172M 7RDX	ROG G1702VM	G570 STEALTH		q	p	s
CPU	Maks	0.2	83	83	71	83	83	78	USUAL	-	-	-
INCHES	Maks	0.2	85	85	85	95	95	95	QUASI	5	-	-
HARGA	Min	0.1	18.918.928.07	20.048.653.8	15.100.136.87	18.963.480.63	25.442.696.37	25.442.696.37	LEVEL	400.000	3.000.000	-
MEMORY	Maks	0.2	63	63	70	63	60	54	LINIER	-	5	-
RAM	Maks	0.2	55	55	55	55	55	55	L and I Area	1	5	-
WEIGHT	Min	0.1	80	80	80	80	80	80	Gaussian	-	-	1

1

4.1.3 Proses Perhitungan Preferensi

Pada proses ini sistem sudah menerapkan 6 tipe preferensi yang ada dalam metode PROMETHEE yaitu: *Usual*, *Quasi*, *Linier*, *Level*, *Linier and Indiffenrence Area* dan *Gaussian*. Hasil dari hitung berpasangan tadi akan dicari nilai preferensinya, dimana setiap kriteria memiliki preferensinya masing-masing.

4.1.4 Proses Perhitungan Index Preferensi

Setelah didapatkan nilai preferensi dari setiap kriteria maka yang selanjutnya dilakukan adalah menghitung nilai index preferensi, dimana setiap kriteria memiliki nilai bobotnya masing-masing dan total dari bobot tersebut adalah 1.

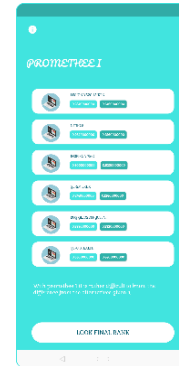
4.1.5 Proses Perhitungan Index Total Preferensi

Nilai total index preferensi ini didapatkan dari menjumlahkan nilai index preferensi berdasarkan bobot yang dibandingkan. Nilai total index preferensi ini nantinya akan digunakan dalam menghitung nilai leaving flow dan entering flow.

4.1.6 Proses Perhitungan PROMETHEE 1

Pada PROMETHEE 1 ini akan mempunyai dua nilai perankingan yaitu ranking dari *leaving flow* dan ranking dari *entering flow*, dimana nilai rankingnya didapatkan dengan cara membandingkan nilainya. Hasil

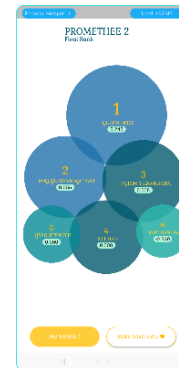
perhitungan menggunakan PROMETHEE 1 terlihat pada Gambar 5.



Gambar 5 Hasil Prangkingan dengan Promethee 1

4.1.7 Proses Perhitungan PROMETHEE 2

Karena perhitungan dengan menggunakan PROMETHEE 1 kurang maksimal maka diperlukan adanya PROMETHEE 2, dimana hasil hitungnya disebut dengan nilai *net flow*. Nilai *net flow* tersebut didapatkan dari pengurangan nilai *leaving flow* dan *entering flow* dari masing-masing alternatif. Hasil perhitungan menggunakan PROMETHEE 2 terlihat pada Gambar 6.



Gambar 6 Hasil Prangkingan dengan Promethee 2

4.1.8 Hasil Perankingan

Hasil akhir dari sistem adalah perankingan yang didapatkan dari nilai net flow PROMETHEE 2. Disini akan dilakukan pengurutan secara descending atau pengurutan dari nilai terbesar ke terkecil.

4.2. PEMBAHASAN

Berikut pembahasan dari pengujian sistem rekomendasi pemilihan laptop. Proses pengujian akan dilakukan dengan menggunakan metode black box.

4.2.1 Pengujian Black Box

Pengujian black box dalam sistem rekomendasi pemilihan laptop ini bertujuan untuk

menguji fungsionalitas yang ada dalam sistem. Dengan dilakukannya pengujian ini akan diketahui apakah sistem dapat sesuai dengan fungsionalitasnya atau tidak. Rencana pengujian untuk sistem rekomendasi pemilihan produk laptop menggunakan metode PROMETHEE adalah sebagai berikut:

Tabel 2 Pengujian Black Box

Kelas Uji	Butir Uji	Jenis Pengujian
<i>Log in user</i>	Proses <i>log in</i> ke sistem	Black box
Rekomendasi baru	Proses buat rekomendasi baru	Black box
Simpan <i>search key</i> sebagai <i>history</i>	Proses menyimpan data ke dalam <i>cloud firestore</i>	Black box
Kirim hasil rekomendasi ke <i>gmail user</i>	Proses mengirim hasil rekomendasi ke <i>email user</i>	Black box
<i>Log out user</i>	Proses <i>log out</i> dari sistem	Black box
Perhitungan dengan Metode PROMETHEE	Proses perhitungan dengan metode PROMETHEE	Black box

5. PENUTUP

5.1. Kesimpulan

Berdasarkan keseluruhan proses analisis, perancangan dan implementasi pada pembuatan sistem rekomendasi pemilihan laptop diperoleh kesimpulan sebagai berikut:

- Agar sesuai dengan keinginan pengguna peneliti merancang *input*-an untuk sistem rekomendasi dengan *input*-an yang sederhana, dimana *input*-an ini yang selalu menjadi acuan dalam memilih laptop.
- Sistem rekomendasi pemilihan laptop dapat diimplementasikan ke dalam pemrograman android dengan Bahasa pemrograman Dart sebagai *backend* dan Flutter sebagai *frontend*, selain itu peneliti juga dapat mengimplementasikan Firebase Cloud Firestore sebagai *database* yang menampung *dataset*.
- Sistem dapat menggunakan semua tipe preferensi yang ada dalam metode PROMETHEE serta keseluruhan langkah-langkah dalam metode diimplementasikan ke dalam bahasa pemrograman Dart, sistem juga dapat melakukan semua proses perhitungan yang ada dalam metode hingga mendapatkan hasil berupa *ranking* dari masing-masing alternatif. Sistem juga dapat

mengirimkan hasil perankingan tersebut ke akun gmail user.

5.2. Saran

Berdasarkan kesimpulan di atas, peneliti mencantumkan beberapa saran antara lain:

- Data laptop masih static dari dataset, mungkin sistem dapat di rework menggunakan sebuah Application Programming Interface (API) sehingga data laptop selalu up to date.
- Desain bisa dibuat lebih user friendly.

DAFTAR PUSTAKA

- Ardianty, D.A.N., dan Yunus, M., (2017), *Rancang Bangun Sistem Fuzzy untuk Rekomendasi Pemilihan Laptop*, Skripsi, S.Kom., Teknologi Informatika, STMIK Bumigora Mataram, Nusa Tenggara Barat.
- Fadlil, J. dan Mahmudy, W.F., (2007), *Pembuatan Sistem Rekomendasi Menggunakan Decision Tree dan Clustering*, *Kursor*, Vol 3 No 1, 45-66.
- Figueira, J. et al. (2005), *Multiple Criteria Decision Analysis: State of the Art Surveys*. Tersedia pada: <https://www.springer.com/gp/book/9780387230818>.
- Setiawan, H., dan Hansun, S., (2014), *Rancang Bangun Aplikasi Rekomendasi Pembelian Laptop Dengan Metode Fuzzy Database Model Tahani Berbasis Web*, *Jurnal Ilmiah Komputer dan Informatika*, Vol 3 No 2, 86-95.
- Sugiarti, Yuni, S.T., M.Kom. (2013), "Analisis dan Perancangan UML (Unified Modelling Language) Generated VB 6". Yogyakarta: Graha Ilmu.
- Suryadi, K. dan Ramadhani, M.A., (1998), "Sistem Pendukung Keputusan", PT Remaja Rosdakarya, Bandung.
- Ungkawa, U., Rosmala, D. & Aryanti, F., (2013), *Pembangunan Aplikasi Travel Recommender Dengan Metode Case Base Reasoning*, *Jurnal Informatika*, Jurusan Teknik Informatika, Institut Teknologi Nasional, Bandung.
- Wijaya, A.K., dan Alfian, D., (2018), *Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering Dan Content-Based Filtering*, *Jurnal Computech & Bisnis*, Vol 12 No 1, 11-27.