

APLIKASI ANDROID HYBRID UNTUK PEMILIHAN LOKASI KULINER

Vidya Noer Firdausy¹⁾, Fahrul Agus²⁾, Indah Fitri Astuti³⁾

¹⁾Program Studi Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Mulawarman
Jalan Barong Tongkok Kampus Gunung Kelua Samarinda, Telp. (0541)-741118, Kalimantan Timur
E-Mail : fahrulagus@unmul.ac.id ^{1,2)}, indahfitriastuti@yahoo.com³⁾

ABSTRAK

Bisnis kuliner saat ini berkembang dengan pesat, seiring dengan perkembangan sosial media dan peningkatan jumlah permintaan konsumen. Selain itu dipicu oleh kemudahan dalam membagikan informasi yang dimanfaatkan pelaku bisnis kuliner untuk mempromosikan mereknya. Penelitian ini bertujuan untuk menerapkan metode *Simple Additive Weighting* (SAW) dan Sistem Informasi Geografis dalam sebuah aplikasi berbasis Android Hybrid. Metode SAW dipilih karena kemudahan dalam penerapan dan digunakan untuk menghasilkan rekomendasi rumah makan yang diberikan kepada pengguna sebagai referensi untuk memilih rumah makan. Hasil *output* sistem berupa lima opsi rumah makan terbaik dengan visualisasi peta dari Google Maps. Google Maps juga berperan dalam perhitungan jarak dari lokasi pengguna ke lokasi rumah makan. Sistem ini juga dikembangkan berbasis Android *hybrid* dengan menggunakan *framework* PhoneGap.

Kata Kunci : rumah makan, sistem penunjang keputusan, *simple additive weighting*, google maps, *mobile*, android, *hybrid*.

1. PENDAHULUAN

Bisnis kuliner saat ini berkembang dengan beragam menu andalan, jenis makanan, harga, dan fasilitas. Hal ini dapat menimbulkan masalah bagi konsumen yang ingin mencoba suatu rumah makan, misalnya jenis makanan khas Jepang, terdapat perbedaan harga signifikan pada rumah makan Kedai Dimsum yang lebih murah dari pada X.O Suki. Hal inilah yang menjadi salah satu faktor yang membuat sebagian masyarakat ragu untuk mencoba atau memilih tempat makan yang sesuai dengan harga, cita rasa, kapasitas dan jarak tempuh yang diinginkan.

Permasalahan ini mendorong pemanfaatan teknologi internet sebagai media untuk mendapatkan informasi rumah makan. Informasi ini memang dapat berasal dari pencarian Google ataupun media sosial seperti Path, Facebook, Twitter, dan Instagram, namun pencarian seperti ini akan sangat merepotkan.

Masyarakat memerlukan media yang lebih efisien yang dapat mengumpulkan informasi khusus rumah makan serta dapat memberikan rekomendasi rumah makan yang sesuai dengan kriteria yang diinginkan. Pemanfaatan sistem pendukung keputusan sebagai salah satu sistem cerdas terkomputerisasi dapat digunakan untuk membantu mengambil keputusan dengan cepat, tepat dan konsisten serta memberikan informasi mengenai rumah makan yang menjadi keluaran sistem.

Banyak metode yang dapat digunakan dalam mengembangkan sistem pendukung keputusan, salah satunya metode Simple Additive Weighting

(SAW) yang sesuai untuk proses pengambilan keputusan karena dapat menentukan nilai bobot untuk setiap kriteria. Sistem Pendukung Keputusan dapat diterapkan di berbagai media seperti desktop, web, dan *mobile*..

IBM pada situs resminya mengatakan bahwa 7 triliun populasi di dunia, 6 triliun memiliki *mobile phone* Hal inilah yang menggeser perkembangan teknologi dari yang sebelumnya dikembangkan berbasis web kini lebih diutamakan pengembangan aplikasi berbasis *mobile*, contohnya seperti sosial media Path dan Instagram yang dikhususkan untuk pengguna *smartphone* dan membatasi fitur penggunaan pada web. Pengembangan sistem berbasis *mobile* akan lebih memudahkan masyarakat dalam mencari informasi dan mendapatkan rekomendasi rumah makan karena mudah dibawa dan dioperasikan membuat berbagi informasi menjadi lebih nyaman. Salah satu sistem operasi *mobile* yang banyak digunakan adalah Android. Banyak vendor *smartphone* yang menggunakan sistem operasi Android seperti Samsung, Sony, LG, Lenovo, dan Xiaomi, selain itu Android merupakan sistem operasi *open source* yang menjadikannya mudah dan bebas untuk dikembangkan. Android merupakan aplikasi *native*, untuk mengembangkan sistem berbasis Android menggunakan teknologi web dapat memanfaatkan teknologi *hybrid*. *Hybrid application* merupakan teknologi menggabungkan aplikasi *native* dan teknologi web, hal ini membuat programmer dapat mengembangkan aplikasi *mobile* dengan bahasa dasar PHP, HTML, CSS, dan JavaScript.

2. TINJAUAN PUSTAKA

2.1 Sistem Pendukung Keputusan

Definisi konsep Sistem Pendukung pertama diungkapkan pada 1970 oleh Scott Morton dengan istilah "Manajemen Keputusan Sistem", berarti sistem berdasarkan komputerisasi yang dapat membantu membuat keputusan menggunakan data dan model untuk memecahkan masalah yang tidak terstruktur [6].

Sistem Pendukung Keputusan adalah sistem informasi berbasis komputerisasi, untuk menghasilkan alternatif tertentu keputusan untuk membantu sisi tertentu tentang menangani masalah data yang digunakan dan model. Sistem Pendukung Keputusan hanya menyediakan alternatif keputusan, sedangkan keputusan akhir masih ditentukan oleh pengambil keputusan. Sistem Pendukung Keputusan terintegrasi resources intelektual pribadi dan kemampuan komputer untuk meningkatkan kualitas keputusan [1].

Sistem pendukung keputusan umumnya dibangun untuk mendukung solusi dari masalah atau untuk mengevaluasi kesempatan. DSS dimaksudkan untuk mendukung manajemen dalam kerja analitis tampil dengan situasi yang tidak terstruktur dan kriteria jelas [3].

2.2 Simple Additive Weighting (SAW)

Salah satu metode penyelesaian masalah MADM adalah dengan menggunakan metode *Simple Additive Weighting* (SAW). Metode SAW sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif dari semua atribut [2]. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada [4]. Diberikan persamaan (1):

$$R_{ij} = \begin{cases} \frac{X_{ij}}{\max X_{ij}} & \text{if } j = \text{benefit} \\ \frac{X_{ij}}{\min X_{ij}} & \text{if } j = \text{cost} \end{cases} \quad (1)$$

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan persamaan (2):

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad (2)$$

Dimana W_j bobot dari setiap kriteria. Nilai V_i yang paling besar merupakan opsi terbaik untuk memilih rumah makan.

2.3 Google Maps API

Google Map adalah layanan aplikasi dan teknologi peta berbasis web yang disediakan oleh

Google secara gartis (bukan untuk kepentingan komersial), termasuk di dalamnya website Google Map (<http://maps.google.com>), Google Ride Finder, Google Transit, dan peta yang dapat disisipkan pada website lain melalui Google Maps API. Saat ini Google Map adalah layanan pemetaan berbasis web yang populer. Kita dapat menambahkan layanan Google Map ke website kita sendiri menggunakan Google Maps API. Google Maps API dapat ditambahkan ke website kita menggunakan JavaScript. API tersebut menyediakan banyak fasilitas dan utilitas untuk memanipulasi peta dan menambahkan konten ke peta melalui berbagai layanan, memungkinkan Anda untuk membuat aplikasi peta yang kuat pada website Anda.

Pengetahuan yang diperlukan untuk mengembangkan Google Maps API adalah tentang HTML dan JavaScript, sedangkan peta sudah disediakan oleh Google. Jadi kita hanya berkonsentrasi tentang data dan biarkan urusan peta ditangani oleh Google, sehingga dapat menghemat waktu. Fitur yang terdapat dalam Google Maps API, antara lain:

1. **Menampilkan Peta Sederhana**
Google Maps API memungkinkan pengguna menampilkan peta sederhana pada situs web atau aplikasi *mobile*.
2. **Marker**
Marker berguna untuk memberikan petunjuk pada lokasi tertentu. Secara *default*, *marker* menggunakan gambar standar. *Marker* dapat menampilkan *icon*. *Marker* dirancang agar dapat interaktif dengan pengguna, misalnya dengan menambahkan animasi seperti *bounce*. *Marker* secara *default* dapat ditekan dan menampilkan *infobox* yang menampilkan informasi mengenai lokasi tersebut. *Marker* juga memungkinkan untuk dipindahkan dari satu lokasi ke lokasi lain dengan cara mengubah setting *draggable* menjadi *true*.
3. **Info Windows**
Sebuah *Info windows* menampilkan konten (biasanya teks atau gambar) di jendela pop-up di atas peta, pada lokasi tertentu. *Info windows* biasanya berbentuk kotak dialog yang ujungnya berada pada titik koordinat lokasi yang ditentukan. Biasanya *info windows* ditampilkan ketika *marker* ditekan, tetapi dapat juga langsung dilampirkan pada lokasi tertentu.
4. **Direction**
Direction adalah fitur Google Maps API yang memungkinkan pengguna dapat mengetahui arah atau rute dari titik koordinat lokasi awal menuju lokasi yang dituju. Lokasi awal dan lokasi tujuan pada *direction* dapat diubah atau *draggable*, dengan mengubah *setting draggable* menjadi *true*. *Source code* JavaScript untuk menampilkan *direction* dapat dilihat pada gambar 1.

```

var directionsDisplay;
var directionsService = new
google.maps.DirectionsService();
var map;

function initialize() {
  directionsDisplay = new
google.maps.DirectionsRenderer();
  var chicago = new
google.maps.LatLng(41.850033, -87.6500523);
  var mapOptions = {
    zoom:7,
    center: chicago
  }
  map = new
google.maps.Map(document.getElementById("
map-canvas"), mapOptions);
  directionsDisplay.setMap(map);
}

function calcRoute() {
  var start =
document.getElementById("start").value;
  var end =
document.getElementById("end").value;
  var request = {
    origin:start,
    destination:end,
    travelMode:
google.maps.TravelMode.DRIVING
  };
  directionsService.route(request,
function(result, status) {
    if (status ==
google.maps.DirectionsStatus.OK) {
      directionsDisplay.setDirections(result);
    }
  });
}
}

```

Gambar 1. Source Code Menampilkan Direction

5. Geolocation

Geolocation adalah fitur Google Maps API yang memungkinkan pengguna mengetahui koordinat lokasi pengguna. *Source code geolocation* dapat dilihat pada gambar 2.

```

if(navigator.geolocation) {

navigator.geolocation.getCurrentPosition(function(position)
{
  var pos = new
google.maps.LatLng(position.coords.latitude,
position.coords.longitude);
  var infowindow = new
google.maps.InfoWindow({
    map: map, position: pos, content:
'Location found using HTML5.' });
  map.setCenter(pos);
}, function() {

```

```

  handleNoGeolocation(true);
});
} else {
  // Browser doesn't support Geolocation
  handleNoGeolocation(false);
}}

```

Gambar 2. Source Code Geolocation

6. Matriks Jarak

Matriks jarak adalah fitur Google Maps API untuk mengetahui jarak dari lokasi satu ke lokasi lainnya. *Source code* untuk matriks jarak dapat dilihat pada gambar 3.

```

function
dekat($latlng_posisi,$latlng_alternatif){
  $from = -0.470935,117.154;
  $to = -0.470945,117.145;

  $from = urlencode($from);
  $to = urlencode($to);

  $dat=
file_get_contents("http://maps.googleapis.com/
maps/api/distancematrix/json?origins=$from&d
estinations=$to&language=en-
EN&sensor=false");
  $data = json_decode($data);

  $distance = 0;

  foreach($data->rows[0]->elements as
$road) {
    $distance += $road->distance-
>value;
  }
  $km=$distance/1000;
  return $km;
}

```

Gambar 3. Source Code JavaScript Matriks Jarak

2.4 Hybrid Application

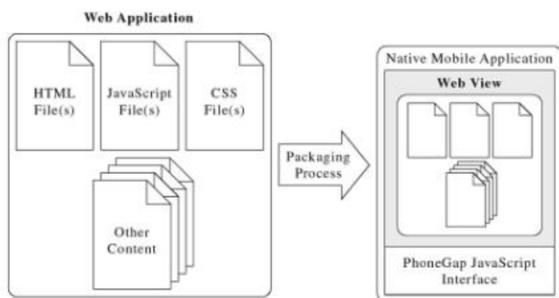
Hybrid application adalah teknologi menggabungkan dua bahasa pemrograman atau dua teknologi menjadi satu, dengan adanya teknologi ini akan memperingan pekerjaan *developer* dalam mengembangkan aplikasinya ke segala jenis sistem operasi *smartphone*.

"*Hybrid mobile app development combines both native development and web technology, but hybrid apps look and behave much more like web apps than mobile apps*" [5].

Menurut Tun, *hybrid application* menggabungkan aplikasi *native* dan teknologi web, dengan menggunakan pendekatan ini, *programmer* dapat membuat aplikasi *mobile* dengan mengandalkan kemampuan *programming* web, namun *hybrid application* akan tetap terlihat seperti aplikasi web daripada aplikasi *mobile*.

Pendekatan ini juga menghadirkan *library* terbuka siap pakai seperti PhoneGap yang menyediakan antarmuka dengan bahasa

pemrograman JavaScript yang memberikan akses untuk fitur sistem operasi yang mendasarinya. Asitektur sistem *hybrid* dapat dilihat pada gambar 4.



Gambar 4. Asitektur sistem *hybrid* [7]

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Data

Metode SAW dapat memecahkan masalah dengan multi-kriteria, yang berarti pemelihan restoran dapat diselesaikan dengan metode SAW. SAW menggunakan bobot setiap kriteria yang berasal dari input data pengguna dan nilai kriteria setiap alternatif yang digunakan untuk mendapatkan alternatif terbaik.

Kriteria yang digunakan untuk menghasilkan alternatif dalam sistem pendukung keputusan terdiri dari:

1. Harga (C1)
2. Rating (C2)
3. Kapasitas (C3)
4. Jarak (C4)

Daftar alternatif restoran dapat dilihat pada Tabel 1.

Tabel 1. Daftar Alternatif Restoran

No.	Nama Restoran	Alternatif
1.	2ndfloor Coffee Shop	A1
2.	Ariza Café	A2
3.	Anima Espresso Bar and Kitchen	A3
4.	Ayam Bakar Wong Solo	A4
5.	Angkringan Mastumin	A5
6.	Bakso Gersik	A6
7.	Balarea Kapedok	A7
8.	Barvity	A8
9.	Bebek & Ayam Goreng Sari Rasa "Pak Ndut"	A9
10.	BEMO	A10
11.	BOS Steak & Ribs	A11
12.	D'Beranda	A12
13.	D'Orange	A13
14.	Double Dippes	A14
15.	Godhong Gedhang	A15
16.	Kedai Dimsum	A16
17.	Kedai Kopi Nusantara	A17

18.	Kedai Sabindo	A18
19.	Mie Jogja Pak Karso	A19
20.	Mie Sedap Sari 2	A20
21.	Padang Upik	A21
22.	Pan & Flip	A22
23.	Solaria	A23
24.	Vin'z Ice Cream	A24
25.	Whyn*t Coffee & Pastry	A25

Kriteria yang digunakan dalam sistem pendukung keputusan memiliki unit perhitungan tidak pasti, sehingga membuat nilai yang diperoleh tidak dapat dihitung dengan perhitungan skala unit yang sama. Untuk itu diperlukan preferensi format untuk setiap kriteria yang ada, termasuk preferensi untuk kriteria bobot yang digunakan dalam proses menghasilkan alternatif rekomendasi restoran. Format preferensi pada sistem pendukung keputusan adalah sebagai berikut:

1. Kriteria harga dibagi menjadi 4 angka *fuzzy* yaitu sangat mahal, mahal, cukup murah dan murah. Angka *fuzzy* akan dikonversi ke angka *crisp* dengan rentang nilai yang dapat dilihat pada Tabel 2.

Tabel 2. Harga

Range Value	Fuzzy Number	Value
0 – 2	Sangat Mahal	0.25
3 – 5	Mahal	0.5
6 – 7	Cukup Murah	0.75
8 – 10	Murah	1

2. Kriteria *rating* dibagi menjadi 4 angka *fuzzy* yaitu sangat rendah, rendah, cukup tinggi dan tinggi. Angka *fuzzy* akan dikonversikan ke angka *crisp* dengan rentang nilai yang dapat dilihat pada Tabel 3.

Tabel 3. Rating

Range Value	Fuzzy Number	Value
0 – 2	Sangat Rendah	0.25
3 – 5	Rendah	0.5
6 – 7	Cukup Tinggi	0.75
8 – 10	Tinggi	1

3. Kriteria kapasitas dibagi menjadi 4 angka *fuzzy* yaitu kecil, sedang, cukup besar dan besar. Angka *fuzzy* akan dikonversikan ke angka *crisp* dengan rentang nilai yang dapat dilihat pada Table 4.

Tabel 4. Kapasitas

Range Value	Fuzzy Number	Value
0 – 2	Kecil	0.25
3 – 5	Sedang	0.5
6 – 7	Cukup Besar	0.75
8 – 10	Besar	1

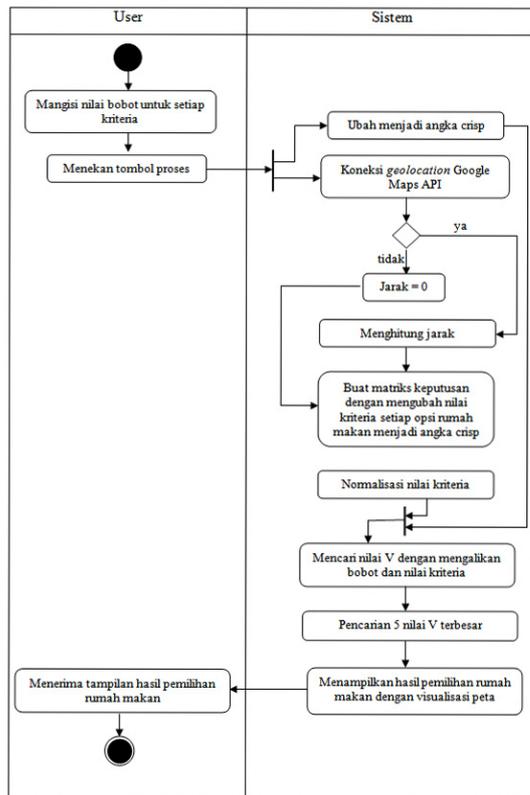
4. Kriteria jarak dibagi menjadi 3 angka *fuzzy* yaitu jauh, sedang dan dekat. Angka *fuzzy* akan

dikonversikan ke angka *crisp* dengan rentang nilai yang dapat dilihat pada Table 5.

Tabel5. Jarak

Range Value	Fuzzy Number	Value
0km => Value < 3km	Far	0.75
3km => Value < 6km	Sufficient	0.5
Value => 6km	Near	0.25

3.2 Perancangan Proses



Gambar 5. Diagram Aktifitas Proses SAW

Langkah-langkah yang dilakukan dalam menerapkan metode SAW untuk pemilihan alternatif rumah makan terbaik, yaitu :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan.
2. Menentukan *range* untuk merubah nilai bobot dan kriteria menjadi angka *crisp*.
3. Mengubah nilai bobot dan kriteria menjadi angka *crisp*.
4. Melakukan normalisasi nilai kriteria dengan membagi nilai dengan nilai kriteria terbesar.
5. Hasil akhir diperoleh dari proses perankingan yaitu hasil penjumlahan dari perkalian matriks ternormalisasi dengan bobot yang telah diubah menjadi angka *crisp*.

Diagram aktifitas proses SAW sistem dapat dilihat pada gambar 5. Contoh kasus, diambil tiga sample rumah makan yaitu kedai dimsum, kedai kopi nusantara, dan kedai sabindo. Data dapat dilihat pada tabel 4.3. Kemudian diinputkan bobot harga 5, *rating* 8, kapasitas 6, dan jarak 2. Jarak

dihitung dari titik 0 koordinat *user* yaitu Fakultas MIPA Universitas Mulawarman sampai dengan titik koordinat alternatif rumah makan.

Tabel 6 Data Rumah Makan

Alternatif	Harga	Rating	Kapasitas	Jarak
Kedai Dimsum	7	7	5	1.34 km
Kedai Kopi Nusantara	6	6	6	4.45 km
Kedai Sabindo	6	7	6	2.35 km

Tahap pertama adalah menentukan *range* atau aturan yang menjadi dasar untuk mengubah nilai kriteria dan nilai bobot menjadi angka *crisp*.

Tahap kedua mengubah bobot menjadi angka *crisp* dan membuat tabel *decision matrix* untuk alternatif rumah makan yaitu mengubah nilai kriteria menjadi angka *crisp*. Berdasarkan aturan perubahan angka *crisp*, nilai bobot harga menjadi 0,5 nilai bobot *rating* menjadi 1, nilai bobot kapasitas menjadi 0,75 dan nilai bobot jarak menjadi 0,25. Tabel *decision matrix* alternatif rumah makan dapat dilihat pada tabel 7.

Tabel 7. *Decision Matrix* Alternatif Rumah Makan

Alternatif	Harga	Rating	Kapasitas	Jarak
Kedai Dimsum	0.75	0.75	0.5	0.75
Kedai Kopi Nusantara	0.75	0.75	0.75	0.5
Kedai Sabindo	0.75	0.75	0.75	0.75

Tahap ketiga yaitu melakukan normalisasi dari *decision matrix*, penjabaran dapat dilihat pada tabel 8.

Tabel 8. Normalisasi *Matrix*

Alternatif	Harga	Rating	Kapasitas	Jarak
Kedai Dimsum	0.75/0.75 = 1	0.75/0.75 = 1	0.5/0.75 = 0.67	0.75/0.75 = 1
Kedai Kopi Nusantara	0.75/0.75 = 1	0.75/0.75 = 1	0.75/0.75 = 1	0.5/0.75 = 0.67
Kedai Sabindo	0.75/0.75 = 1	0.75/0.75 = 1	0.75/0.75 = 1	0.75/0.75 = 1

Tahap keempat dilakukan perankingan atau mencari nilai V dengan cara menjumlahkan perkalian bobot dan kriteria.

1. Kedai Dimsum
 $V = (0.5 \times 1) + (1 \times 1) + (0.75 \times 0.67) + (0.25 \times 1) = 2.25$
2. Kedai Kopi Nusantara
 $V = (0.5 \times 1) + (1 \times 1) + (0.75 \times 1) + (0.25 \times 0.67) = 2.41$
3. Kedai Sabindo

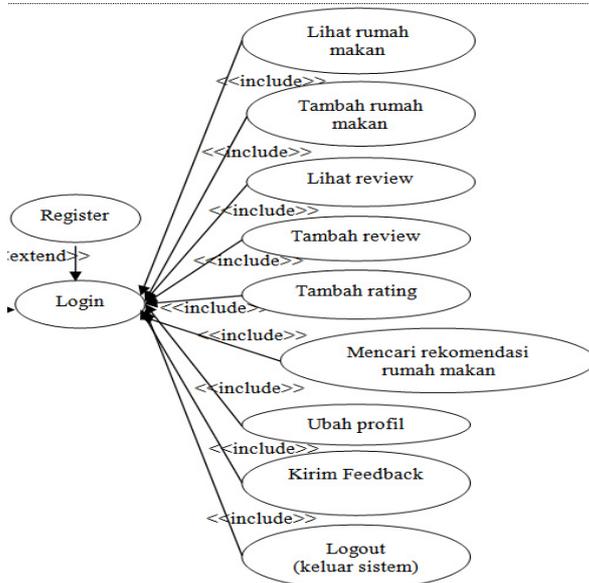
$$V = (0.5 \times 1) + (1 \times 1) + (0.75 \times 1) + (0.25 \times 1) = 2.5$$

Dari hasil tersebut dapat disimpulkan bahwa alternatif rumah makan terbaik adalah Kedai Sabindo dengan $V = 2.5$.

3.3 Perancangan Sistem

Perancangan sistem adalah rencana sistematis dalam proses pengembangan sistem setelah memiliki data yang cukup untuk menunjang kegiatan pengembangan sistem. Perancangan sistem ini menggunakan diagram UML yaitu *use case diagram*. Terdapat tiga level hak pengguna pada sistem ini yaitu admin, operator dan *user*.

Gambaran diagram *use case* sistem untuk *user* dapat dilihat pada Gambar 6.



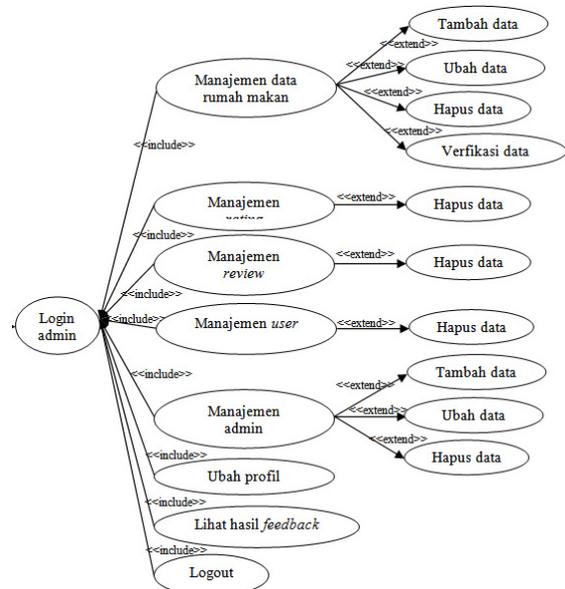
Gambar 6. Use Case Diagram Level User

User berperan dalam melakukan penambahan data rumah makan, *rating* dan *review*. Selain itu *user* dapat melihat informasi rumah makan, *rating*, maupun *review*. *User* diberi fasilitas untuk merubah profilnya, mengirim *feedback*, mencari rekomendasi rumah makan dan keluar dari sistem.

Level admin berfungsi manajemen *rating* dan *review*, manajemen *user* dan admin, melihat *feedback* dari *user*, dan manajemen data rumah makan yaitu menambah, mengedit, menghapus, dan memverifikasi rumah makan yang ditambahkan oleh *user*. Level operator memiliki fungsi yang sama dengan level admin hanya saja pada level operator tidak diberi hak untuk melakukan manajemen admin. Level *user* digunakan oleh pengunjung website yang ingin mendapatkan rekomendasi rumah makan, menambah data rumah makan, mengirim *review*, memberikan *rating* dan mengirim *feedback*. Halaman *user* dibangun berbasis Android *Hybrid* sedangkan halaman admin dan operator dibangun berbasis *website*.

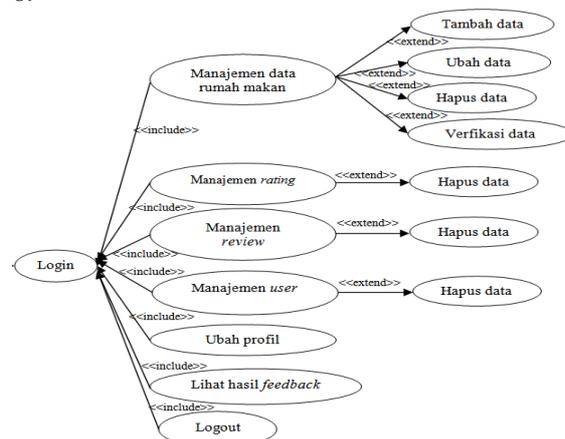
Admin berperan dalam melakukan manajemen data rumah makan yang meliputi tambah data, edit

data, dan memverifikasi data rumah makan. Admin juga berperan dalam melakukan manajemen *review* dan *rating* yaitu hapus data, manajemen *user* yaitu hapus *user*, manajemen admin yang meliputi tambah admin, edit admin, hapus admin, ubah profil, dan melihat hasil *feedback* dari *user*. Gambaran diagram *use case* sistem untuk admin dapat dilihat pada Gambar 7.



Gambar 7. Use Case Diagram Level Admin

Peran level operator sama dengan peran level admin hanya saja pada level operator tidak ada manajemen data admin. Operator hanya bisa mengubah profilnya sendiri. Gambaran diagram *use case* untuk level operator bisa dilihat pada Gambar 8.



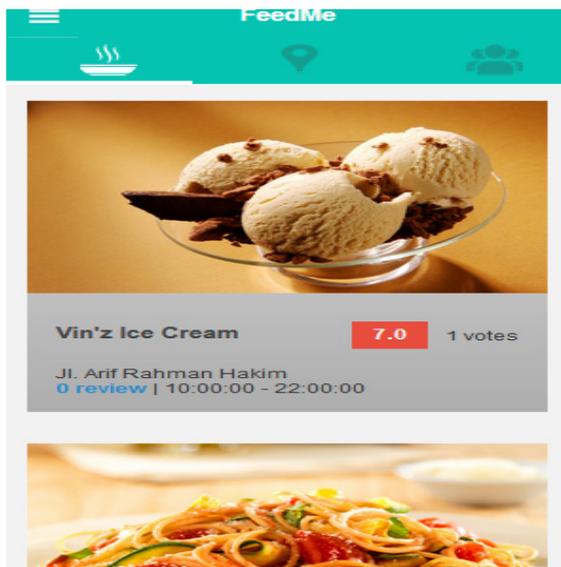
Gambar 8. Use Case Diagram Level Operator

3.4 Implementasi Sistem

Apabila sistem ini dieksekusi, maka halaman pertama yang tampil adalah halaman login yang merupakan pintu akses menuju aplikasi, pada halaman login terdapat pilihan mendaftar, menu ini disediakan bagi *user* yang belum memiliki akun. Halaman login sistem dapat dilihat pada gambar 9.

Gambar 9. Halaman Login

Setelah login, sistem akan menampilkan halaman utama yang berisi informasi rumah makan berupa foto rumah makan, nama rumah makan, *rating* beserta jumlah *user* yang memberi *rating*, alamat rumah makan, jumlah *review*, dan jam buka dan jam tutup rumah makan. Tampilan halaman utama dapat dilihat pada gambar 10.



Gambar 10. Halaman Utama

Halaman rekomendasi rumah makan adalah halaman untuk memproses pemilihan rumah makan dengan metode SAW dapat dilihat pada gambar 11.

Range	0-2	3-5	6-7	8-10
Harga	Sangat Mahal	Mahal	Cukup Murah	Murah
Rating	Sangat Rendah	Rendah	Cukup Tinggi	Tinggi
Kapasitas	Kecil	Sedang	Cukup Besar	Besar
Jarak	Sangat Jauh	Jauh	Cukup Dekat	Dekat

Gambar 11. Halaman *Form Input* Bobot

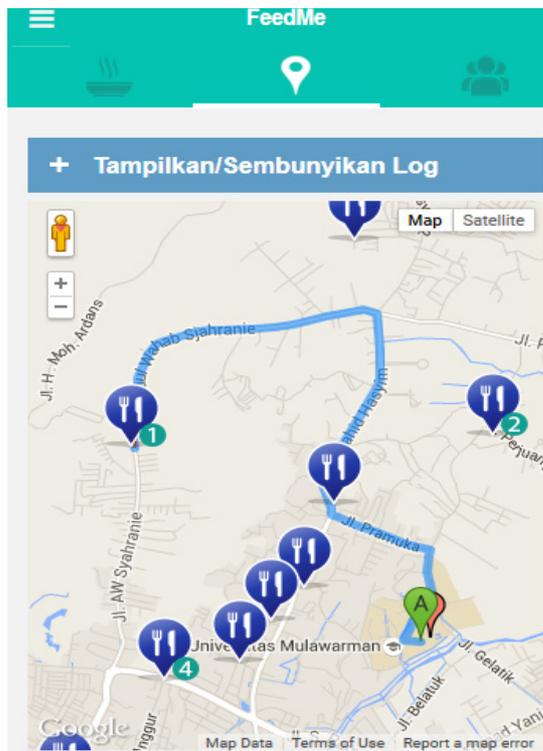
Halaman ini menampilkan keterangan untuk memasukkan nilai bobot dalam bentuk tabel agar lebih mudah dipahami oleh *user* serta *form* berupa *slider* untuk menginput nilai bobot. Sebelum masuk ke halaman hasil, *user* diharuskan menginput nilai bobot dari angka 1-10 lalu menekan *button* cari, *form input* bobot dapat dilihat pada gambar 12.

Gambar 12. Lanjutan *Form Input* Bobot

Setelah *user* mengklik *button* cari makan sistem akan melakukan proses perhitungan dengan metode SAW serta pencarian koordinat untuk menampilkan peta dan menghitung jarak.

Sistem akan menampilkan hasil dengan visualisasi. Peta dilengkapi dengan *mark* untuk setiap rumah makan, *mark direction and routing* pada rumah makan yang merupakan hasil alternatif terbaik dan *mark* berupa *badge* nomor urut untuk lima alternatif terbaik. Gambar hasil rekomendasi peta dapat dilihat pada gambar 13.

Gambar 13 menunjukkan *direction and routing* terletak pada koordinat lokasi *user* dengan *mark A* dan lokasi alternatif terbaik berada pada *mark B*. *Direction and routing* memiliki fitur *dragable* atau koordinat lokasi dapat dipindah dengan menggeser *mark direction and routing* (*mark A* dan *mark B*) ke rumah makan yang dituju ataupun ketika GPS salah membaca koordinat lokasi *user*.



Gambar 13. Halaman Hasil Rekomendasi dengan Visualisasi Peta

Button log berguna untuk menampilkan atau menyembunyikan proses pemilihan rumah makan dengan metode SAW. Form proses pemilihan rumah makan dengan metode SAW dapat dilihat pada gambar 14.



Gambar 14. Form Proses Pemilihan dengan Metode SAW

4. KESIMPULAN

Berdasarkan analisis dan pengujian yang dilakukan, maka kesimpulan yang dapat diambil adalah Penerapan metode SAW mampu memberikan opsi dalam memilih rumah makan, selain itu *hybrid application* dapat diterapkan untuk membangun aplikasi Android. Sistem ini sangat baik dikembangkan dengan teknologi API web servis dan dikembangkan ke *platform* lainnya seperti IOS.

5. DAFTAR PUSTAKA

- [1]. Daihani, D. U. 2011. *Komputerisasi Pengambilan Keputusan*. Jakarta: Elexmedia Media Komputindo.
- [2]. Fishburn, P. C. 1967. *A Problem-based Selection of Multi-Attributr Decision Making Methods*. New Jersey : Blackwell Publishing.
- [3]. Kusriani. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta : ANDI.
- [4]. Kusumadewi, S., Hartati, S., Harjoko, A., dan Wardoyo, R. 2206. *Fuzzy* .
- [5]. Tun, Phyo Min. 2014. *Choosing a Mobile Application Development Approach*. Stamford International University, Thailand.
- [6]. Turban, E., Aronson, J. E., dan Liang, T. P. 2005. *Sistem Pendukung Keputusan dan Sistem Cerdas*. Terjemahan Dwi Prabantini. Yogyakarta : Andi.
- [7]. Wargo, J. (2012). *PhoneGap Essential : Building Cross-platform Mobile Apps*. United States: Addison-Wesley Professional