

Sistem Penunjang Belanja Pedagang Keliling Di Lokasi Sekitar Menggunakan Haversine Berbasis Android

Richard Gozali, Liliana, Yulia

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jln. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

gozalirichard@gmail.com, lilian@petra.ac.id, yulia@petra.ac.id

ABSTRAK

Pedagang keliling merupakan usaha yang dilakukan oleh seseorang untuk menjual barang/jasa seperti sayur, buah, dan lain-lain dengan cara berkeliling ke tempat-tempat dengan rute tertentu. Pada masa pandemi ini sebagian orang membeli produk secara *online* dikarenakan orang takut untuk pergi ke tempat yang memiliki banyak kerumunan. Lonjakan permintaan ini membuat banyak orang mencari penjual sayur-sayuran dan buah-buahan. Dengan adanya penelitian ini dapat membantu pelanggan untuk menentukan lokasi keberadaan pedagang keliling yang berada di sekitar.

Dengan penggunaan rumus *haversine* dapat mengetahui perkiraan jarak antara pelanggan dan pedagang keliling. Pada masa pandemi serta kebiasaan tinggal di rumah saja, membuat –banyak orang lebih memilih memasak sendiri dibandingkan dengan membeli. Kesusahan untuk menentukan masakan yang ingin dimasak. Dengan penggunaan *beautifulsoup* dapat mengambil menu resep makanan nya ingin dibuat.

Dari hasil dari *survey* yang diberikan kepada pelanggan dan pedagang keliling telah dihasilkan bahwa aplikasi yang dibuat mudah untuk digunakan dan hasil *survey* memberikan nilai yang cukup memuaskan. Berdasarkan dari pengujian jarak antara *haversine* dengan *google maps*. Dihasilkan rumus *haversine* cukup akurat saat dibandingkan dengan *google maps*. Hasil perbandingan jarak rata-rata yang dihasilkan berupa 0.315 Km.

Kata Kunci: *Web Scarping, BeautifulSoup, Kotlin, Android, Haversine*

ABSTRACT

A traveling merchant is an effort made by a person to sell goods/services such as vegetables, fruit, and others by traveling to places with certain routes. During this pandemic, some people buy products online because people are afraid to go to places with large crowds. This surge in demand has made many people look for sellers of vegetables and fruits. With this research, it can help customers to determine the location of the existence of traveling merchants in the vicinity.

By using the haversine formula, you can find out the approximate distance between the customer and the traveling merchant. During the pandemic and the habit of staying at home, many people prefer to cook for themselves compared to buying. It's hard to decide which dish to cook. With the use of beautifulsoup, you can take the recipe menu of the food you want to make.

From the results of the survey given to customers and traveling merchants, it has been shown that the application is made easy to

use and the survey results provide a satisfactory value. Based on testing the distance between haversine and google maps. The resulting haversine formula is quite accurate when compared to google maps. The results of the comparison of the average distance produced are 0.315 Km.

Keywords: *Web Scarping, BeautifulSoup, Kotlin, Android, Haversine*

1. PENDAHULUAN

Pedagang keliling merupakan usaha yang dilakukan oleh seseorang untuk menjual barang/jasa seperti sayur, buah, dan lain-lain dengan cara berkeliling ke tempat-tempat dengan rute tertentu. Kebutuhan sayur mayur, buah serta bahan baku lauk pauk lainnya dibutuhkan untuk kebutuhan sehari-hari. Banyak masyarakat lebih memilih tukang sayur keliling daripada pergi ke pasar atau berbelanja di tempat keramaian [7]. Pandemi *Covid-19* serta kebiasaan tinggal di rumah saja, membuat banyak orang lebih memilih memasak sendiri dibandingkan dengan membeli [9].

Salah satu bisnis yang tumbuh di masa pandemi *Covid-19* adalah retail, terutama yang melayani kebutuhan sehari-hari seperti sayur-sayuran dan buah-buahan. Lonjakan permintaan ini membuat banyak orang mencari penjual sayur-sayuran dan buah-buahan. Dikarenakan pada masa pandemi ini sebagian orang membeli produk melalui *online* [13]. Kesusahannya untuk menemukan para pedagang keliling disekitar membuat para ibu – ibu kesusah dalam mencari bahan untuk memasak [2]. Pada penelitian sebelumnya yaitu “*Implementation Of Haversine Formula And Best First Search Method In Searching Of Tsunami Evacuation Route*” memiliki perkiraan tempat terdekat untuk pergi ketempat yang dituju.

Di penelitian pada skripsi ini, saya ditambahkan fitur perkiraan jarak. Melalui fitur ini, para pelanggan dapat mengetahui jarak sampai dari pedagang keliling.

Masalah yang diangkat dari penelitian ini Apakah rumus *haversine* dapat membantu aplikasi ini untuk mendapatkan ketepatan jarak yang sesuai. Apakah aplikasi dapat dipakai dengan mudah oleh pelanggan dan pedagang keliling. aplikasi ini untuk membeli produk dengan lebih mudah dan cepat. Dengan menggunakan rumus *haversine* diharapkan dapat membantu para pelanggan untuk mendapatkan lokasi terdekat dari pedagang keliling dan membuat pengantar produk yang dibeli lebih cepat dan lebih efisien dibandingkan dengan cara yang biasa saja. Adanya aplikasi untuk tukang sayur keliling ini dapat membantu banyak orang dalam menemukan tukang sayur keliling pada area lokasi sekitarnya.

2. LANDASAN TEORI

2.1 Haversine Formula

Haversine Formula merupakan sebuah algoritma yang dibuat untuk mengatur jarak terdekat menggunakan letak lokasi *latitude* dan *longitude* [1].

$$d = 2 \times R \times \arcsin \left(\sqrt{\sin^2 \left(\frac{\text{lat}_x - \text{lat}_y}{2} \right) + \left(\cos(\text{lat}_x) \times \cos(\text{lat}_y) \times \sin^2 \left(\frac{\text{long}_x - \text{long}_y}{2} \right) \right)} \right) \quad (1)$$

Keterangan :

- lat x = Latitude dari koordinat pertama
- lat y = Latitude dari koordinat kedua
- long x = Longitude dari koordinat pertama
- long y = Longitude dari koordinat kedua
- R = Radius Bumi 6.371 Km
- d = jarak yang dihasilkan

2.2 Web Scraping

Website adalah sekumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan dapat diakses secara langsung melalui halaman depan (*home page*) menggunakan sebuah *browser* dengan menggunakan *URL* [14]. *Website* pertama kali dibuat oleh Tim Berners-Lee pada akhir 1980 dan baru resmi online pada tahun 1991. Tujuan awal *website* adalah memudahkan peneliti untuk bertukar informasi atau melakukan perubahan informasi [4].

Web scraping adalah teknik untuk melakukan ekstraksi data dan informasi dari suatu *website* kemudian menyimpannya dalam format tertentu. Biasanya *scraping* ini bisa dilakukan salah satunya untuk mendapatkan produk atau barang yang berada di situs tertentu. Teknik *web scraping* bisa dilakukan dengan manual atau otomatis menggunakan *tools* [11]. *Parsing HTML* adalah teknik yang menggunakan JavaScript untuk menargetkan *linear HTML* dan *nested HTML*. Teknik *parsing* ini bisa dengan lebih cepat melakukan identifikasi *script HTML* BeautifulSoup merupakan library dari *Python* yang diperuntukan untuk *web scraping* yang menggunakan package *bs4*. Digunakan untuk ekstraksi tipe file *XML* atau *HTML*.

2.3 Text Mining

Text Mining adalah mengambil data berupa teks atau kata - kata dimana sumber data didapatkan dari dokumen, Bertujuan untuk mendapatkan kata - kata yang diinginkan. Proses ekstraksi terhadap pola berupa informasi yang berguna dari sejumlah besar sumber data teks. *Text Mining* dianggap proses dua tahap diawali dengan penerapan struktur terhadap sumber data dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan penambangan data. Tujuan dari *text mining* adalah mendapatkan informasi yang berguna dari sekumpulan dokumen. Sumber data yang digunakan adalah kumpulan teks yang memiliki format yang tidak terstruktur. Tugas khusus yang dimiliki antara lain yaitu *text categorization* dan *text clustering*. *Text mining* juga menerapkan konsep dan teknik *data mining* untuk mencari pola teks [10].

2.4 Grabmart dan Gomart

Merupakan aplikasi *ondemand* yang menyediakan produk seperti belanjaan, makanan kemasan, produk perawatan kesehatan, produk kecantikan dan hadiah. *Gomart* merupakan layanan pesan antar dimana pelanggan memesan produk belanjaan sesuai

aplikasi lalu, *Mitra Driver* akan mengambil pesanan yang dipesan di outlet yang dituju [6]. *Grabmart* merupakan salah satu fungsi aplikasi yang berguna untuk membeli produk yang sesuai dengan kebutuhan pelanggan.

2.5 Tinjauan Studi

2.5.1 Implementation Of Haversine Formula And Best First Search Method In Searching Of Tsunami Evacuation Route

Pada paper "*Implementation Of Haversine Formula And Best First Search Method In Searching Of Tsunami Evacuation Route*". Menggunakan rumus *haversine* sebagai penentuan jarak yang tsunami dari lokasi yang didapat dan diberikan daerah yang aman untuk tsunami. Di dalam *paper* ini juga memberikan fitur untuk mengetahui jarak yang di dapat oleh pengguna kemudian akan diberikan perkiraan waktu tsunami yang akan datang. Pada skripsi yang dibuat juga menggunakan perhitungan *haversine* untuk menentukan jarak - jarak dari pedagang keliling dan juga menentukan batasan jarak dari pedagang keliling [3].

2.5.2 Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform

Pada paper "*Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform*" membuat sistem darurat bagi pengguna di sekitar. Menggunakan metode *haversine* untuk menentukan lokasi darurat yang berada di sekitar. Penggunaan metode *haversine* sesuai dengan kebutuhan aplikasi yang diperuntukan untuk mencari jarak dengan kordinat lokasi *GPS*. Penggunaan rumus *haversine* hampir menyerupai dengan *paper* ini. Menggunakan *haversine* sebagai penentuan range untuk menampilkan jarak untuk *emergency office*. Hampir serupa dengan hasil yang diinginkan skripsi yaitu menampilkan *range* pedagang keliling [5].

2.5.3 Determine the Shortest Path Problem Using Haversine Algorithm, A Case Study of SMA Zoning in Depok

Pada paper "*Determine the Shortest Path Problem Using Haversine Algorithm, A Case Study of SMA Zoning in Depok*" membuat sistem untuk mengetahui letak sekolah terdekat bagi pengguna di sekitar. Menggunakan metode *haversine* untuk menentukan lokasi sekolah yang berada di sekitar. Penggunaan metode *haversine* sesuai dengan kebutuhan aplikasi yang diperuntukan untuk mencari jarak dengan kordinat lokasi *GPS*. Penggunaan rumus *haversine* hampir menyerupai dengan *paper* ini. Menggunakan *haversine* sebagai penentuan *range* untuk menampilkan jarak untuk lokasi sekolah. Hampir serupa dengan hasil yang diinginkan skripsi yaitu menampilkan *range* pedagang keliling [12].

3. ANALISIS DAN DESAIN SISTEM

3.1 Analisis Data

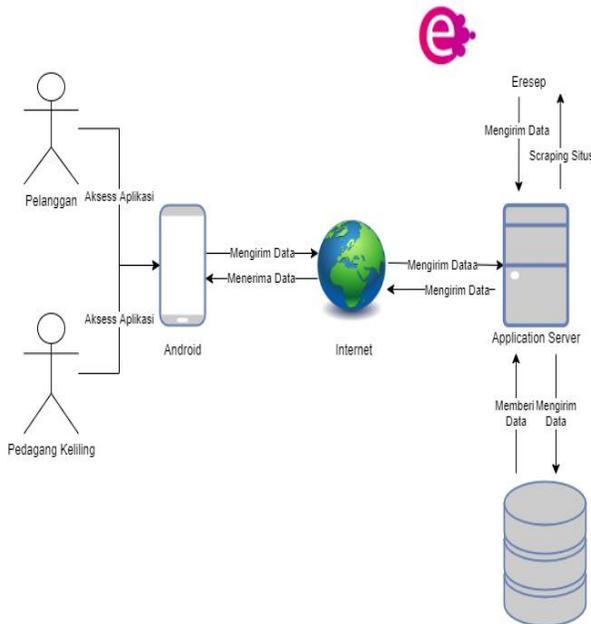
Selama pembuatan dilakukan analisis data terhadap *gomart* dan *grabmart* sebagai studi perbandingan. Data yang digunakan dalam pembuatan sistem ini pertama adalah lokasi atau koordinat *latitude* dan *longitude* dari pengguna pedagang keliling dan pengguna. Data kedua yang diberikan adalah file gambar berupa *file jpg/jpeg* untuk produk dari pedagang keliling. Yang digunakan untuk mengupload gambar produk ke *server*. Data yang selanjutnya yang dibutuhkan adalah data pribadi dari

pedagang keliling yang berupa *region/daerah* yang dibutuhkan untuk menghasilkan *query* bagi pelanggan. Data selanjutnya *real time location* pelanggan yang digunakan untuk mendapat list dari pedagang keliling terdekat diareanya

3.2 Desain Arsitektur

Untuk mengetahui bagaimana suatu sistem bekerja, dibutuhkan sebuah arsitektur sistem. Dengan desain arsitektur bertujuan agar pengguna bisa memahami proses yang terjadi.

Gambar 1. Arsitektur Sistem dan Aplikasi keseluruhan aplikasi. Pelanggan dan pedagang keliling mengirim data berupa lokasi *latitude* dan *longitude* ke server melalui internet. Server mengambil data menu masakan yang berada di dalam *eresep.com*. Yang kemudian disimpan kedalam *database*. Pelanggan mengirim produk –produk apa saja yang akan dibeli kemudian akan tersimpan ke dalam *database*. Mengirim *list* pedagang keliling yang berada disekitar ke dalam aplikasi pelanggan.



Gambar 1. Arsitektur Sistem dan Aplikasi

3.3 Analisis Sistem

Pada penulisan skripsi ini dilakukan analisis terhadap aplikasi sistem aplikasi *gojek* dan *gomart* untuk melihat – lihat fitur yang telah ada. Dan mencari tahu fitur yang dapat ditambah dan digunakan dalam pembuatan sistem penunjang belanja pedagang keliling di lokasi sekitar.

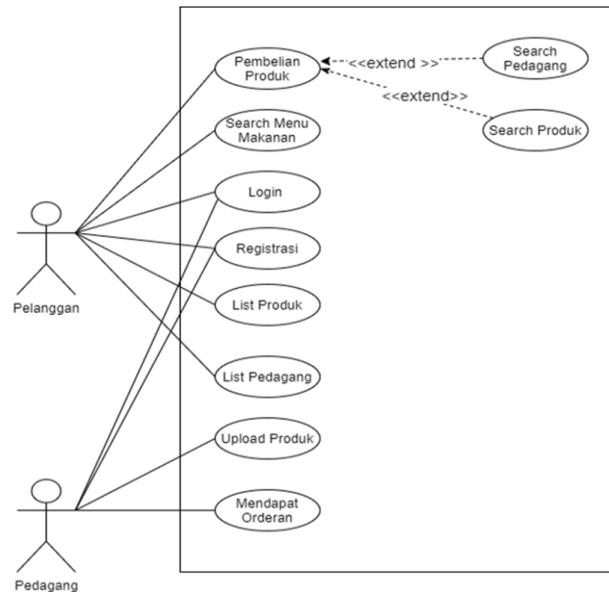
3.4 Analisis Permasalahan

Kesusahannya pedagang keliling untuk menjual barangnya secara manual dan kesusahan pada pedagang keliling untuk mengetahui siapa saja yang ingin membeli barang dagangannya sehingga pedagang keliling tidak usah untuk putar –putar untuk mendapatkan pelanggannya. Diharapkan juga membantu para pelanggan pedagang keliling untuk mengetahui lokasi pedagang keliling atau penjual yang dibutuhkan. Membantu pelanggan dalam pembelian hari – H dan memberikan catatan untuk pembelian di esok paginya. Kesusahannya para pelanggan untuk menemukan pedagang keliling yang berada di sekitar dan mengetahui produk-produk apa saja yang dibawa oleh pedagang

keliling. Dengan adanya fitur pembelian membantu pedagang keliling untuk dapat menghafal pembelian apa saja yang di dapat oleh pelanggan. Pelanggan dapat melakukan pemesanan secara mudah dan dapat melakukan pemesanan dengan memilih hari yang diinginkan. Para pelanggan kesusahan juga dalam mengetahui posisi *real time* pedagang keliling.

3.5 Use Case Diagram

Use case diagram merupakan *diagram* yang menggambarkan sebuah aktor dengan sistem. *Use case diagram* dapat mendeskripsikan satu *actor* atau lebih dari sebuah sistem. Di dalam *use case diagram* dalam penelitian ini terdapat dua aktor. Seperti pada Gambar 2. Use Case Diagram. Pelanggan yang memiliki hak kegunaan *login*, registrasi, *search* produk, *search* pedagang keliling, melihat *list* pedagang keliling dan melakukan *order* produk. Pedagang keliling memiliki *hak akses login*, *registrasi*, *upload* produk, mendapat *order*an.



Gambar 2. Use Case Diagram

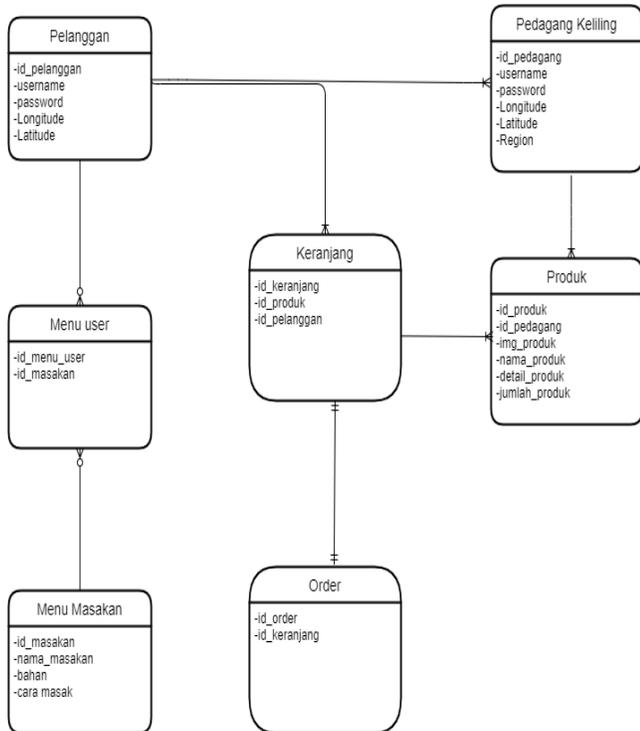
3.5.1 Entity Relationship Diagram

Pada penelitian ini memiliki 5 entity, seperti pada Gambar 3. Entity Relationship Diagram *Entity* Pelanggan yang memiliki attribute *id_pelanggan*, *username*, *password*, *longitude* dan *latitude*. *Entity* Pedagang Keliling memiliki attribute *id_pedagang*, *username*, *password*, *longitude*, *foreign key id_pedagang*, *img_produk*, *nama_produk*, *jumlah_produk* dan *detail_produk*. *foreign key id_pedagang*, *img_produk*, *nama_produk*, *jumlah_produk* dan *detail_produk*.

Entity keranjang memiliki *attribute id_keranjang*, *foreign key id_produk* dan *foreign key id_pelanggan*. *Entity* Order memiliki attribute *id_order* dan *id_keranjang*. Pelanggan memiliki *relation 1 to many* ke pedagang keliling. Pedagang keliling memiliki relasi *1 to many* ke produk. Pelanggan memiliki relasi *1 to many* ke keranjang. Produk memiliki relasi *1 to many* ke keranjang. Keranjang memiliki relasi *1 to 1* ke order.

Entity menu masakan memiliki *attribute id_masakan*, *nama_masakan*, *bahan_masakan*, dan *cara_memasak*. Memiliki relasi antara *1 to many* ke *menu_user*. Dimana *menu_user* memiliki *id_menu_user* dan *id_masakan*. Setelah menu makan di *scraping*. Menu masakan akan tersimpan terlebih dahulu di menu

masakan kemudian menu_masakan user kemudian akan menyesuaikan data yang diambil ke dalam database tersebut.

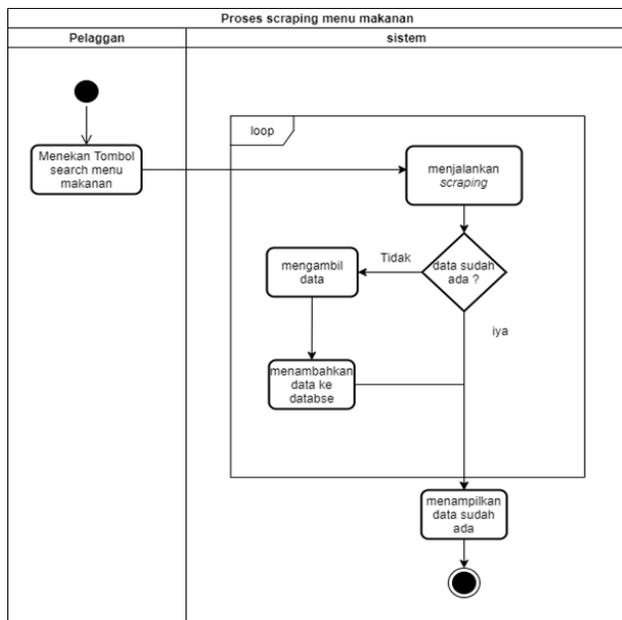


Gambar 3. Entity Relationship Diagram

3.6 Activity Diagram

3.6.1 Proses Scraping

Proses *scraping* merupakan salah satu proses dalam skripsi ini, data yang didapatkan dari proses *scraping* ini merupakan menu makanan, bahan makanan dan cara memasak. Perlunya memahami struktur *HTML* dari situs agar data yang diperlukan bisa diambil dengan lancar dan tanpa ada kesalahan. Seperti pada Gambar 4. Activity Diagram Proses Scraping Menu Makanan



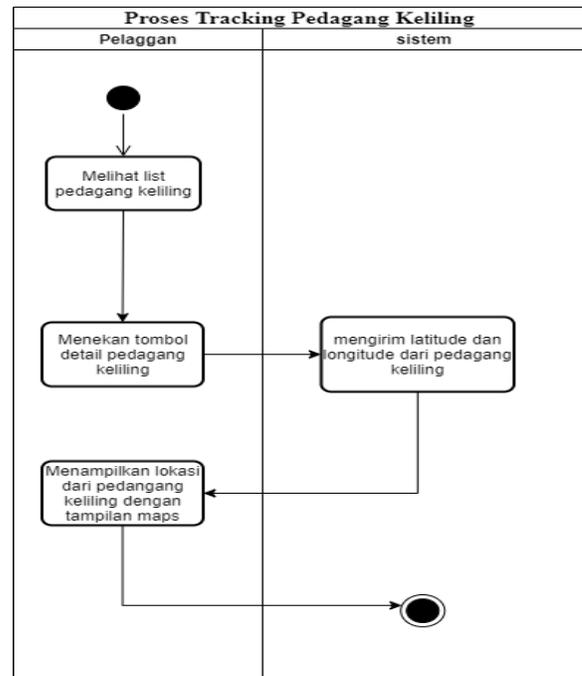
Gambar 4. Activity Diagram Proses Scraping Menu Makanan

Menggunakan *library python BeautifulSoup* untuk melakukan pengambilan data dari situs *eresep.com*. *BeautifulSoup* digunakan untuk mengambil data yang ada di situs tersebut [8]. *BeautifulSoup* menggunakan *link* yang dikirim ke dalam parameter *python* yang kemudian digunakan untuk mengakses halaman situs tersebut. Data yang di ambil berupa *file JSON* yang didapat didalam situs tersebut [15]

Activity diagram digunakan untuk melihat proses *scraping*, baik di situs *eresep* atau situs resep lainnya. Pengguna akan menuliskan menu makan yang akan dipilih untuk melakukan *scraping* resep masakan dari situs yang dituju, sehingga sistem akan menjalankan proses *scraping*. Untuk setiap resep masakan yang terdapat akan diperiksa terlebih dahulu apakah data yang akan diambil sudah terdapat di *database*. Hal itu dilakukan dengan cara melakukan *query* dengan judul masakan dan *author* resep masakan. Apabila data belum ada di dalam *database*, maka proses *scraping* untuk data tersebut diselesaikan dan proses dimasukan ke dalam *database*.

3.6.2 Proses Tracking Pedagang Keliling

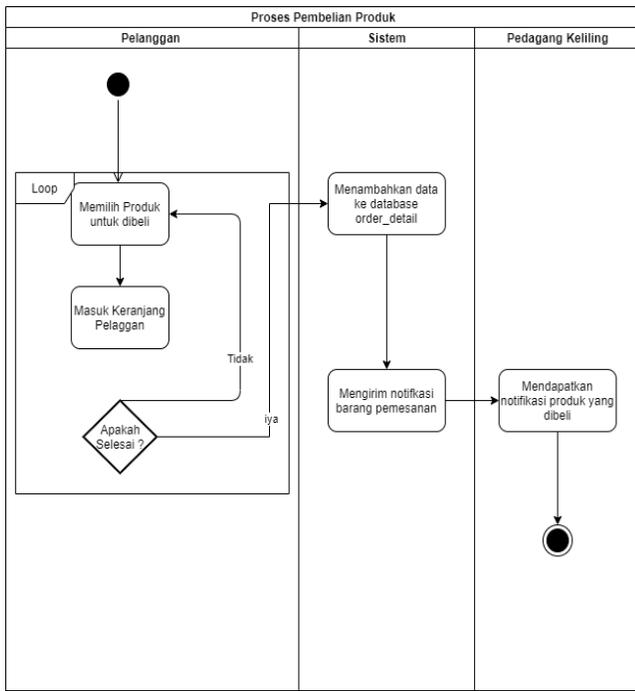
Pada Gambar 5. Pedagang Keliling merupakan activity diagram untuk melakukan pencarian barang atau pedagang keliling. Apabila pengguna masuk ke halaman aplikasi untuk melihat *list* pedagang keliling. Setelah melihat *list* pedagang keliling maka akan menekan tombol lihat lokasi. Akan masuk ke dalam aktivitas detail lokasi pedagang keliling. Di detail lokasi pedagang akan terlihat lokasi dari pedagang keliling yang ditampilkan di *google maps*.



Gambar 5. Activity Diagram Tracking Pedagang Keliling

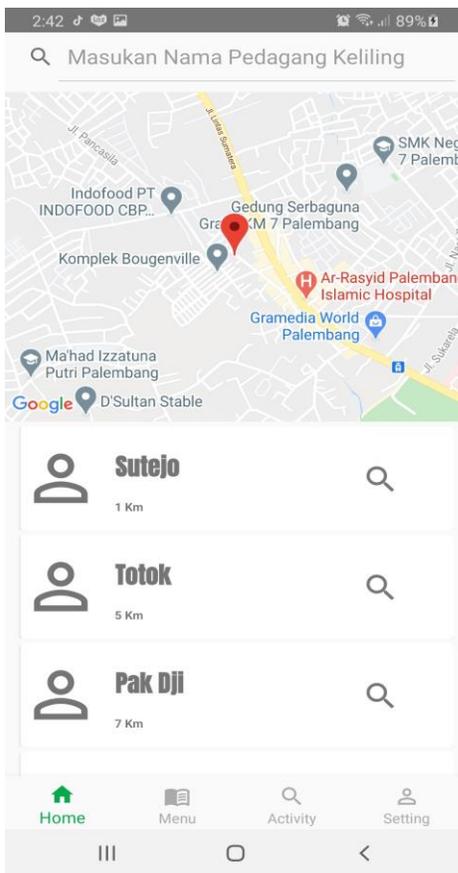
3.6.3 Proses Pembelian Barang

Pada proses Gambar 6. merupakan fitur pelanggan untuk melakukan pembelian barang. Pertama pelanggan akan menginput barang – barang kedalam keranjang belanjaan dan lalu barang akan diantar oleh pedagang keliling. Pelanggan akan melakukan tambah keranjang disetiap produk yang ingin dibeli. Kemudian produk akan disimpan kedalam database setiap keranjang *user*.



Gambar 6. Pembelian Produk

3.7 Tampilan Aplikasi



Gambar 7. Tampilan Pelanggan



Gambar 8. Tampilan Pedagang Keliling

Pada Gambar 7. dan pada Gambar 8 saat melakukan perancangan tampilan aplikasi. Tampilan pelanggan menampilkan pedagang keliling apa saja yang terlihat di lokasi sekitar. Tampilan pedagang menampilkan apa saja produk –produk yang telah dibawak oleh pedagang keliling.

4. PENGUJIAN SISTEM

4.1 Perbandingan Jarak Antara Google Maps dengan Haversine

Pada Tabel 1 melakukan pengujian sebanyak sepuluh kali. Telah dihasilkan bahwa jarak haversine cukup akurat dalam menentukan jarak, tetapi google maps lebih akurat dikarenakan dengan jalan-jalan apa saja yang boleh dilewati. Pada percobaan pertama dilakukan di *Pakuwon Mall* dengan tujuan *Bebek Sinjay HR Muhamad* hasil yang didapat dari hasil *google maps* berupa jarak 2.5 Km dan jarak untuk penggunaan rumus *haversine* adalah 2.5 Km.

Pada percobaan kedua menggunakan lokasi awal yang berada di *Pakuwon Mall* dan tujuan berada di *Graha Famili Blok S*. Dimana hasil jarak yang dihasilkan oleh *google maps* berupa 3.3 km dan jarak yang dihasilkan oleh aplikasi adalah 2.2 Km. Jarak yang dihasilkan berbeda jauh dikarenakan tidak ada jalan yang dapat dilewati jika ditarik garis lurus ke koordinat yang dituju.

Tetapi *google maps* lebih akurat dikarenakan dengan jalan-jalan apa saja yang boleh dilewati. Dengan menggunakan *google maps* didapatkan hasil yang lebih optimal dari pada menggunakan dengan haversine. Penggunaan *haversine* hanya melakukan tarik garis lurus dari koordinat awal dengan koordinat tujuan. Oleh karena itu jarak yang dihasilkan kurang maksimal. Telah dilakukan perhitungan rata –rata antara jarak perbandingan haversine dengan google maps sebesar 0.315 Km. Haversine memiliki kelebihan tidak perlu request ke API google maps untuk setiap data yang dihasilkan.

Tabel 1. Perbandingan Jarak Haversine dan Google maps

No	Lokasi Awal		Lokasi Tujuan		Haversine	Google Maps
	Latitude	Longitude	Latitude	Longitude		
1	-7.28914	112.675721	-7.278842	112.683846	2,5 Km	2,3 Km
2	-7.28914	112.675721	-7.292604	112.69581333	2,2 Km	3,3 Km
3	-7.28914	112.675721	-7.295381	112.676227	0,7 Km	0,9 Km
4	-7.28914	112.675721	-7.281424	112.684188	1,3 km	1,5 km
5	-7.28914	112.675721	-7.283906	112.667220	1,1 Km	1,2 Km
6	-7.293437	112.685555	-7.293869	112.690465	0,6 Km	0,65 Km
7	-7.293437	112.685555	-7.301813	112.690619	1,1 Km	1,6 Km
8	-7.293437	112.685555	-7.301212	112.681319	1.0 Km	1,5 km
9	-7.293437	112.685555	-7.302983	112.687297	1,1 Km	1,1 Km
10	-7.293437	112.685555	-7.28914	112.675721	1,2 Km	1,5 Km

4.2 Kuesioner Pelanggan

Untuk mengetahui penilaian pengguna pelanggan tentang program ini, dilakukan penelitian dan pengujian terhadap pelanggan. Dari hasil kuisisioner yang telah dikumpulkan, didapatkan data seperti pada yang berisi tentang hasil kuisisioner aplikasi tersebut. Pada Tabel 2. merupakan hasil kuisisioner.

Tabel 2. Hasil Kuisisioner Pelanggan

Nama	Pertanyaan						
	#1	#2	#3	#4	#5	#6	#7
Corry	5	5	5	5	5	5	5
Arin	5	4	5	4	4	5	5
Erny	4	5	5	5	4	5	5
Erna	4	4	4	5	5	4	5
Lina	5	4	5	5	4	5	4

Dari presentasi diatas ditarik kesimpulan bahwa :

- Tampilan aplikasi dinilai dengan 92%
- Lokasi yang ditampilkan untuk mengetahui pedagang keliling dinilai dengan 92%
- Fitur mencari menu makanan memiliki nilai 96%.
- Fitur untuk mencari pedagang keliling mudah untuk digunakan dengan nilai 96%
- Fitur pembelian produk dengan nilai 88%
- Fitur tracking pedagang keliling dengan nilai 96%

- Aplikasi untuk pelanggan mudah untuk digunakan dinilai dengan 96%

4.3 KuesionerPedagang Keliling

Pada Tabel 3. merupakan hasil dari kuisisioner pelanggan.

Tabel 1. Hasil Kuesioner Pedagang Keliling

Nama	Pertanyaan				
	#1	#2	#3	#4	#5
Tri	3	5	4	5	4
Syafi	4	4	5	3	5
Slamet	4	5	5	4	5
Arjuna	5	4	5	4	5
Paulus	5	5	4	5	5

Dari presentasi diatas ditarik kesimpulan bahwa :

- Tampilan aplikasi dinilai dengan 84%
- Data produk keliling telah mencukupi dengan 92%
- Fitur lokasi pelanggan dinilai dengan 92%
- Fitur *chatting* memudahkan untuk berkomunikasi dinilai dengan 84%
- Aplikasi untuk pedagang keliling mudah untuk digunakan dinilai dengan 96%

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian sistem yang telah dilakukan, dapat diambil beberapa kesimpulan, diantar lain :

- Penggunaan *Haversine* dalam penentuan jarak dianggap akurat. Dihilangkan jarak rata-rata antara *haversine* dengan *google maps* sebesar 0.315 Km.
- Berdasarkan hasil kuesioner dari pelanggan didapatkan jawaban dari pelanggan memuaskan dengan nilai rata-rata 93.71%. Pelanggan menilai bahwa aplikasi mudah untuk digunakan.
- Berdasarkan hasil kuesioner pedagang keliling didapatkan jawaban dari pedagang keliling cukup memuaskan dengan nilai rata-rata 89.6%. Pedagang keliling menilai aplikasi cukup mudah untuk digunakan.

5.2 Saran

Saran yang dapat diberikan untuk melakukan penyempurnaan dan pengembangan program lebih lanjut, antara lain :

- Menambahkan fitur untuk mencatat apa saja bahan yang terdapat di dalam suatu menu masakan.
- Memperbaiki beberapa tampilan yang masih kurang sempurna pada aplikasi.
- Menambahkan situs penyedia menu masakan yang lain untuk di-scraping
- Data produk pedagang keliling yang dikumpul berasal dari dua pedagang keliling, kemungkinan ada produk yang tidak terdata dan tidak termasuk.

6. REFERENSI

- [1] Agarwal, P. 2021. Haversine formula to find distance between two points on a sphere. *Geekforgeeks*. URI=<https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere>
- [2] Andes, R. J., & Sunaryanto, L. T. 2020. Peran Promosi Penjualan Melalui Instagram Terhadap Penjualan Sayur Organik Di Po. Sayur Organik Merbabu (SOM). *Jurnal Sosial Ekonomi Pertanian*, 16(1), 27. <https://doi.org/10.20956/jsep.v16i1.10244>
- [3] Anisya, Swara, G. Y. 2017. Implementation Of Haversine Formula And Best First Search Method In Searching Of Tsunami Evacuation Route. *IOP Conference Series: Earth and Environmental Science*, 97, 012004. <https://doi.org/10.1088/1755-1315/97/1/012004>.
- [4] Anniss, M. 2014. What is a website and how do I use it ?. *Britannica Educational Publishing in association with Rosen Educational Publishing*.
- [5] Basyir, M., Nasir, M., Suryati, S., & Mellyssa, W. 2018. Determination of Nearest Emergency Service Office using Haversine Formula Based on Android Platform. *EMITTER International Journal of Engineering Technology*, 5(2), 270–278. <https://doi.org/10.24003/emitter.v5i2.220>
- [6] Chan, A., Maharani, M., & Tresna, P. W. 2017. Perbandingan Pengalaman Pengguna Pada Aplikasi Mobile GO-JEK Dan Grab (Studi Pada Konsumen PT GO-JEK Dan PT GRAB Indonesia Di DKI Jakarta). *AdBispreneur*, 2(2). <https://doi.org/10.24198/adbispreneur.v2i2.13183>
- [7] Deslatama, Y. 2020. Berkah di Balik Pandemi Covid-19. *Liputan 6*. URI=<https://www.liputan6.com/regional/read/4238240/berkah-di-balik-pandemi-covid-19-tukang-sayur-keliling-kewalahan-layani-pembeli>
- [8] Hajba, G. L. 2018. Handling JavaScript. Website Scraping with Python 169–192. https://doi.org/10.1007/978-1-4842-3925-4_5
- [9] Halidi. R. 2021. Survei Sebut Pandemi Covid-19 Bikin Menu Makanan Rumah Makin Hits. *Suara.com*, URI=<https://www.suara.com/lifestyle/2020/07/21/173838/survei-sebut-pandemi-covid-19-bikin-menu-makanan-rumah-makin-hits?page=all>
- [10] Heryanto, A. 2017. Analisis Opini Pada Media Sosial Facebook Dengan Menggunakan Text Mining Terhadap Produk Hijab (Studi Kasus : Zoya Dan El Zatta). *TEMATIK*, 4(1), 86–95.
- [11] Hillen, J. 2019. Web Scraping for Food Price Research. *British Food Journal*, 121(12), 3350–3361. <https://doi.org/10.1108/bfj-02-2019-0081>
- [12] Ikasari, D., Widiastuti, Andika, R. 2021. Determine the Shortest Path Problem Using Haversine Algorithm, A Case Study of SMA Zoning in Depok. *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. <https://doi.org/10.1109/hora52670.2021.9461185>, 7(1)
- [13] Kannan, P., & Li, H. “. 2017. Digital marketing: A framework, review and research agenda. *International Journal of Research in Marketing*, 34(1), 22-45. doi:10.1016/j.ijresmar.2016.11.006
- [14] Mozilla. 2021. HTML basics - Learn web development: MDN. *Learn web development*. URI=https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basic
- [15] Squarespace. 2018. Build a Website. *Squarespace Developers*. URI=<https://developers.squarespace.com/what-is-jsom>