

Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: <https://jurnal.unimed.ac.id/2012/index.php/cess>



Penentuan Peta Lokasi Bengkel Terdekat Menggunakan Algoritma Bubble Sort

Determining the Nearest Workshop Location Map Using the Bubble Sort Algorithm

Indra Chaidir^{1*}, Ibnu Akil²

^{1,2} Program Studi Sistem Informasi Universitas Bina Sarana Informatika
Jl. Kramat Raya No. 98, Senen, Jakarta Pusat, Indonesia.
email: ¹indra@bsi.ac.id, ²ibnu.ial@bsi.ac.id

Diterima: 07 Juli 2021 | Diterima setelah perbaikan: 12 Oktober 2021 | Disetujui: 27 Desember 2021

ABSTRAK

Terbatasnya informasi tentang lokasi bengkel bagi pengendara bermotor yang menyebabkan kesulitan bagi seseorang yang sedang mengalami kendala pada kendaraannya di lokasi tertentu. Aplikasi pemetaan lokasi pada Android menggunakan Google Maps yang dirancang dapat memberikan informasi pada pengguna mengenai lokasi bengkel terdekat menggunakan algoritma bubble sort. Metode ini adalah salah satu metode untuk mencari jarak terdekat pada pemetaan. Tujuan dari penelitian ini adalah untuk memberi informasi, penunjuk arah lokasi-lokasi bengkel yang ada disekitar pengguna dan diurutkan berdasarkan jarak menggunakan algoritma bubble sort.

Kata Kunci: *Android, GPS, Aplikasi, Pencarian, Bubble Sort.*

ABSTRACT

Limited information about the location of the workshop for motorists that causes difficulties for someone who is having problems with his vehicle in certain locations. The location mapping application on Android using Google Maps is designed to provide information to users about the location of the nearest workshop using the bubble sort algorithm. This method is one of the methods to find the closest distance in the mapping. The purpose of this research is to provide information, directions to workshop locations around the user and sorted by distance using the bubble sort algorithm.

Keywords: *Android, GPS, Application, Search, Bubble Sort.*

*Penulis Korespondensi:
email: indra@bsi.ac.id

1. PENDAHULUAN

"Salah satu kelebihan teknologi mobile saat ini adalah Geographic Information System (GIS) yang telah terpasang nya teknologi Location Based Service (LBS) yang merupakan salah satu dari implementasi mobile GIS yang menampilkan direktori kota, navigasi dan sebagainya." [1]

Cukup banyak terjadi beragam kendala yang menimpa pengguna sepeda motor, seperti kendaraan mogok, ban bocor atau kendala lainnya yang terjadi saat mereka melakukan perjalanan ke suatu tempat. Para pengendara ini dapat memanfaatkan smartphone untuk mencari lokasi bengkel terdekat.

Sering terjadi kendala lokasi yang tidak diketahui beberapa orang yang sedang melakukan perjalanan yang membutuhkan informasi lokasi bengkel ataupun tambal ban. [2] "Hal ini menjadi salah satu alasan mengapa seseorang memerlukan suatu informasi tentang lokasi – lokasi penyedia layanan untuk kendaraan bermotor yang sangat diperlukan dalam mengunjungi suatu daerah yang baru dikunjunginya."

Memanfaatlan Algoritma Bubble Sort yang diterapkan pada aplikasi pencarian lokasi bengkel terdekat. Merupakan salah satu metode untuk membandingkan posisi pengendara dengan beberapa lokasi bengkel yang ada di sekitarnya.

2. METODOLOGI

Metode penelitian ini, penulis melakukan pengumpulan bahan dan melaksanakan metode deskriptif. Metode ini merupakan teknik menggambarkan suatu keadaan yang terjadi berdasarkan fakta.

2.1. Teknik Pengumpulan Data

- a. Observasi
Penulis mengamati dari media sosial elektronik dan juga secara tidak langsung untuk memperoleh data-data mengenai aplikasi-aplikasi berbasis android.
- b. Wawancara
Mengumpulkan data dengan tanya jawab kepada pemilik bengkel, tambal ban dan juga pengendara motor dilapangan untuk mendapatkan data perihal tambal ban dan bengkel.
- c. Studi Pustaka
Mengumpulkan data-data yang diperlukan dengan mengambil materi-materi melalui buku-buku, jurnal, perpustakaan dan file-file yang berhubungan dengan topik yang diperlukan untuk dijadikan sebagai referensi dalam penulisan penelitian ini, seperti mempelajari implementasi algoritma bubble sort pada bahasa java mobile.

2.2. Metode Pengembangan Sistem

- a. Analisis
Mengidentifikasi masalah yang terjadi pada pengguna sepeda motor dijalan, menganalisa sistem yang berjalan yaitu android pada smartphonee. Menggunakan Algoritma Bubble Sort pada pengurutan data terdekat
- b. Perancangan
Membuat rancangan yang diperlukan untuk pengembangan aplikasi yang akan dibuat.
- c. Impelementasi
Tahapan dimana aplikasi sudah diuji dan dapat digunakan oleh user

3. HASIL DAN PEMBAHASAN

3.1. Algoritma Bubble Sort

Algoritma Bubble Sort merupakan teknik mengurutkan item dengan menukarkannya ke posisi yang ada disebelahnya secara terus menerus hingga mengurutkan data dari yang terbesar ke yang terkecil (ascending) atau sebaliknya (descending). Pendapat lain pengurutan yang menggunakan algoritma Bubble Sort dilakukan dengan cara membandingkan elemen yang diseleksi dengan elemen yang berikutnya [3]. Algoritma pengurutan memiliki kelebihan dan kelemahan masing-masing, tidak semua algoritma dapat digunakan untuk melakukan pengurutan data [4]. Metode bubble sort terlebih dahulu menyimpan semua data lokasi dalam bentuk array dan digunakan dalam pengurutan berdasarkan jarak antar lokasi. Dalam pembahasan ini dijelaskan bahwa algoritma Bubble Sort memiliki kelebihan sederhana dan mudah diimplementasikan, dan algoritma ini memiliki kompleksitas waktu Average Case dan Worst Case yaitu $O(n^2)$ [5].

Berikut merupakan contoh dari algoritma Bubble Sort dengan array [3 0 4 2 7].

Proses ke-1

[3 0 4 2 7] → [0 3 4 2 7]

[0 3 4 2 7] → [0 3 4 2 7]

[0 3 4 2 7] → [0 3 2 4 7]

[0 3 2 4 7] → [0 3 2 4 7]

Proses ke-2

[0 3 2 4 7] → [0 3 2 4 7]

[0 3 2 4 7] → [0 2 3 4 7]

[0 2 3 4 7] → [0 2 3 4 7]

[0 2 3 4 7] → [0 2 3 4 7]

Proses ke-3

[0 2 3 4 7] → [0 2 3 4 7]

[0 2 3 4 7] → [0 2 3 4 7]

[0 2 3 4 7] → [0 2 3 4 7]

[0 2 3 4 7] → [0 2 3 4 7].

Algoritma Bubble Sort dalam bahasa pemrograman:

```
procedure bubbleSort ( A : list of sortable items )
defined as:
do
swapped := false
  for each i in 0 to length(A) - 2
    inclusive do:
      if A[i] > A[i+1] then
        swap( A[i], A[i+1] )
        swapped := true
```

```

        end if
    end for
while swapped
end procedure
    
```

[6] Kunci pengurutan secara *ascending* terletak pada perintah IF $A[i] > A[i+1]$, sementara untuk pengurutan secara *descending*, perintahnya menjadi IF $A[i] < A[i+1]$. Untuk penelitian ini digunakan metode ascending karena akan mengurutkan jarak terkecil hingga jarak terjauh pada list lokasi.

Table 1. Pseudocode dari Algoritma

Urut	Pseudocode
a.	“objek” meminta data pada url (database : id, nama, lati, longi,detail)
b.	data array [tambanlist] yang di ambil dari “objek”
c.	for (i = 0 ; < data[tambanlist] ; i++)
d.	“objek baru” = data[tambanlist] (i)
e.	“lokasi user” mengambil koordinat latitude dan longtitude saat ini.
f.	“lokasi tambal ban dan bengkel” yang mengambil koordinat latitude dan longtitude dari “objek baru”
g.	jarak lokasi” = “lokasi user’ menuju “lokasi tambal ban dan bengkel” dengan satuan kilometre
h.	“jarak” = mengambil nilai dari “jarak lokasi”
i.	Mendeskripsikan “variabel” yang diambil dari “objek baru”
j.	Mendeskripsikan “variabel” “hasil jarak” yang diambil dari nilai “jarak” dengan satuan kilometer
k.	Mendeskripsikan “kontak” menggunakan hashmap (key,value)
l.	kontak (mengambil pendeskripsian dari “objek baru” dan “variabel”)
m.	kontak (“jarak”, “hasil jarak”)
n.	memasukan “kontak” kedalam array “kontak lis
o.	sort (membandingkan nilai – nilai dari “kontak list”)
p.	membandingkan nilai 1, nilai 2,..., nilai n (hingga banyak nya nilai)
q.	menjadi hasil akhir nilai $1 \leq \text{nilai } 2, \dots, \leq \text{nilai } n$ sehingga nilai akan terurut dari yang terkecil hingga terbesar.

```
JSONObject jsonObj = new JSONObject(jsonStr);
// Getting JSON Array node
JSONArray contacts = jsonObj.getJSONArray("data");
// looping through All Contacts
for (int i = 0; i < contacts.length(); i++) {
    JSONObject c = contacts.getJSONObject(i);
    Location myloc= new Location("lokasi user");
    myloc.setLatitude(latitude);
    myloc.setLongitude(longitude);

    Location lokasitmbn = new Location("lokasi tmbn");
    lokasitmbn.setLatitude(Double.parseDouble(c.getString("lati")));
    lokasitmbn.setLongitude(Double.parseDouble(c.getString("longi")));
    float jaraklok = myloc.distanceTo(lokasitmbn)/1000;
    String jarak = String.format("%.2f", jaraklok);

    String idtmbn = c.getString("id");
    String namatmbn = c.getString("nama");
    String latitude = c.getString("lati");
    String longitude = c.getString("longi");
    String alamat = c.getString("detail");
    String rsltjarak = jarak+ " KM";

    // tmp hash map for single contact
    HashMap<String, String> contact = new HashMap<>();

    // adding each child node to HashMap key => value
    contact.put("id", idtmbn);
    contact.put("nama", namatmbn);
    contact.put("lati", latitude);
    contact.put("longi", longitude);
    contact.put("detail", alamat);
    contact.put("jarak", rsltjarak);
    contact.put("distance", String.valueOf(jaraklok));

    // adding contact to contact list
    contactList.add(contact);

    Collections.sort(contactList, new Comparator<HashMap<String,
String>>() {
        @Override
        public int compare(HashMap<String, String> o1, HashMap<String,
String> o2) {
            return Double.compare(Double.parseDouble(o1.get("distance")),
Double.parseDouble(o2.get("distance")));
        }
    });
}
```

Gambar 1. Implementasi Source Code Java Mobile

Pada bahasa pemrograman diatas terdapat sintak sort:

```
Collection.sort(contactList, new Comparator<HashMap<String,String>>(){
    public int compare()
    {
        Return Double.compare()
    }
})
```

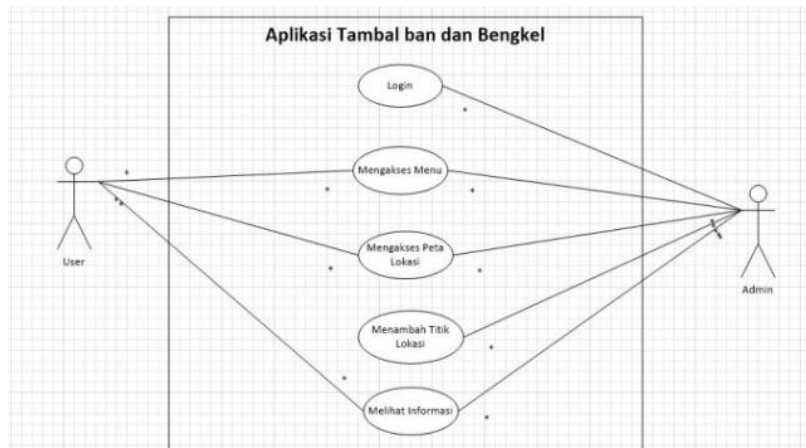
Data-data bengkel yang tersimpan didalam variabel contactList, akan dibandingkan (compare) dan diurutkan. Jenis dari algoritma ini termasuk kedalam jenis algoritma comparison sort, karena melakukan perbandingan dalam operasi diantara elemen-elemen array yang disediakan [7].

3.2. Implementasi Pada Infield Modeling Langauge (UML)

Menggunakan pemodelan UML sebagai gambaran implementasi dari aplikasi yang dibuat, [8] "Unifield Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam blueprint dimana di dalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik."

1. Diagram Use Case

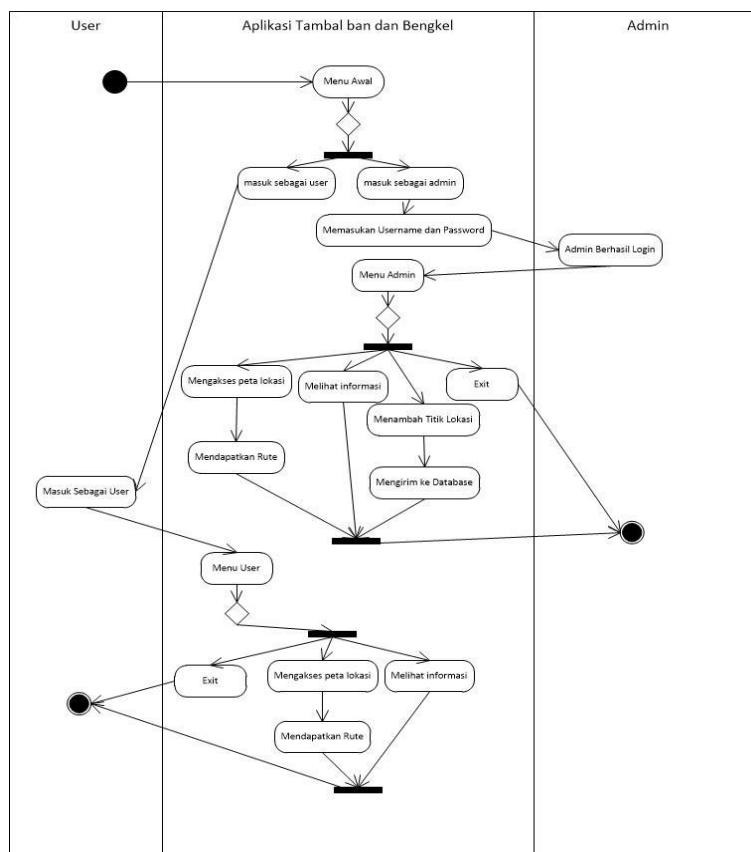
Dalam aplikasi *use case* di deskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat, *use case* digunakan untuk mengetahui fungsi apa saja yang berada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



Gambar 2. Use Case Diagram Pada Aplikasi

2. Diagram Activity

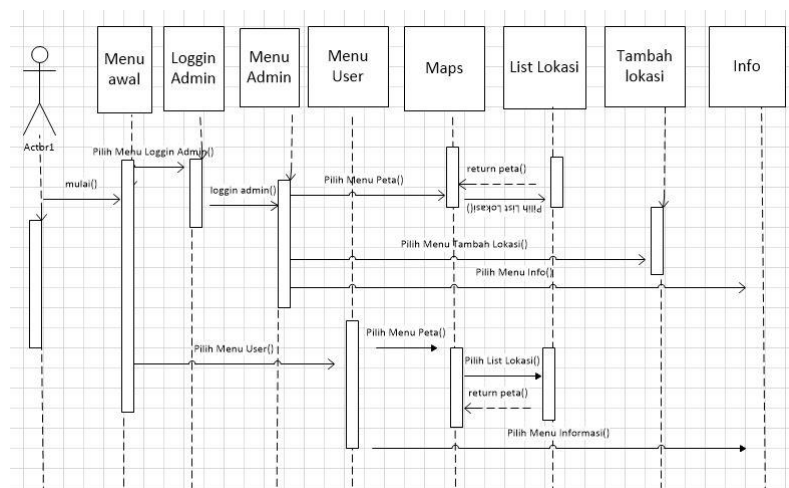
Diagram yang kedua adalah sistem yang dilakukan pengguna saat menggunakan aplikasi yang digambarkan



Gambar 3. Diagram Activity Aplikasi Pencarian Tambal Ban dan Bengkel

3. Diagram Sequence

Pada diagram sequence diatas pengguna akan membuka Menu Awal dan memilih untuk login sebagai admin atau user, jika login sebagai admin, setelah itu akan tampil menu admin, admin memilih maps, maka akan tampil menu maps yang menampilkan pemetaan titik lokasi tambal ban dan bengkel disekitar, lalu dapat langsung memilih titik yang ada pada maps dan akan mendapatkan rute yang ditampilkan oleh google maps, atau admin memilih list lokasi yang akan menampilkan list lokasi tambal ban dan bengkel yang terurut berdasarkan jarak terdekat dengan satuan KM, jika admin memilih tambah lokasi pada menu utama admin, yang akan menampilkan form tambah lokasi dengan menginputkan id dan nama, lalu memilih tambah lokasi untuk mendapatkan titik koordinat (latitude dan longitude), setelah itu admin mengirimkan data tambah lokasi ke database dan titik akan tampil pada menu maps, dan untuk user tidak berbeda jauh dengan admin, hanya saja user tidak dapat menambahkan titik lokasi tambal ban dan bengkel.



Gambar 4. Diagram Sequence Aplikasi

3.3. Analisa Kebutuhan Software

1. Aplikasi yang akan dibuat yaitu menggunakan system operasi basis android, "Android merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet [2]. Android menyediakan perlengkapan dan application Programming Interface (API) untuk pengembangan dalam Bahasa Java. Didalam android terdapat sistem operasi, middleware, dan aplikasi kunci, Android Software Development Kit (SDK)."
2. Dalam aplikasi ini juga memanfaatkan teknologi gps, [2] dalam Dwiyantini menjelaskan bahwa: "GPS merupakan sistem koordinat global yang dapat menentukan koordinat posisi benda dimana saja dibumi baik koordinat lintang, bujur, maupun ketinggian." Hal ini memudahkan pengembang memanfaatkan nilai-nilai geografis dari teknologi GPS yang berupa koordinat untuk mengembangkan aplikasi-aplikasi yang berbasis LBS (Location Based Service). Sehingga dapat memberikan informasi dimana posisi pengguna berada, apa saja yang dekat dengan pengguna, arah rute menuju suatu letak atau lokasi, dan lain-lain.

3. Dalam membuat aplikasi ini dibutuhkan software yang dapat menunjang kebutuhan compabiliti aplikasi basis android dan juga support untuk teknologi gps, penulis menggunakan android studio sebagai software perancangan aplikasi, [9] dalam (Satyaputra & Aritonang, 2016) "Android Studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat open source." Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.
4. Adapun algoritma yang digunakan dalam aplikasi ini yaitu menggunakan algoritma Bubble Sort , [10] dalam Yahya (2014:136) menjelaskan "bubble sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya, jika elemen sekarang > elemen berikutnya maka posisinya ditukar, kalau tidak, tidak perlu ditukar, misalnya untuk $n = 7$ maka akan dilakukan $(n - 1) = 6$ tahap (mulai dari 0 sampai dengan $n - 2$) . Algoritma bubble sort ini melakukan perbandingan antara setiap elemen, kemudian melakukan penukaran jika terdapat elemen yang tidak sesuai urutannya atau salah. Perbandingan akan terus dilakukan sehingga tidak ada lagi pertukaran data."

3.4. Implementasi

Setelah proses perancangan dan pembuatan, tahap selanjutnya yaitu proses implementasi kedalam *smartphone android*. Pertama install aplikasi kedalam *smartphone* android terlebih dahulu, kemudian masuk kedalam aplikasi pencarian tambal ban dan bengkel berbasis gps dan android.

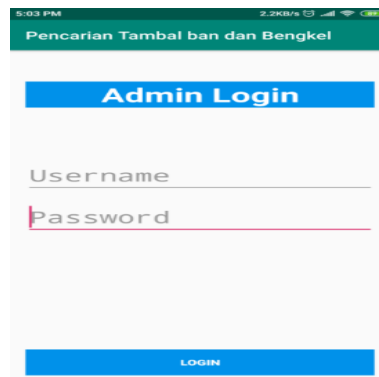
1. Halaman Menu Awal



Gambar 5. Halaman Menu Awal

Gambar 5 adalah halaman tampilan menu awal, ketika pertama kali aplikasi dijalankan. Dimenu ini pengguna akan memilih masuk sebagai *admin* atau sebagai *user*.

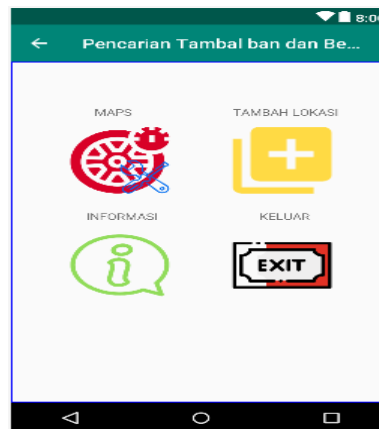
2. Halaman Login Admin



Gambar 6. Halaman Login Admin

Pada halaman login admin ini menampilkan form login dengan menginputkan username dan password, setelah berhasil login selanjutnya akan masuk ke halaman utama admin.

3. Halaman Menu Utama Admin



Gambar 7. Halaman Menu Utama Admin

Pada gambar 7 adalah tampilan halaman utama admin yang terdiri dari menu maps, menu tambah lokasi, menu informasi dan keluar.

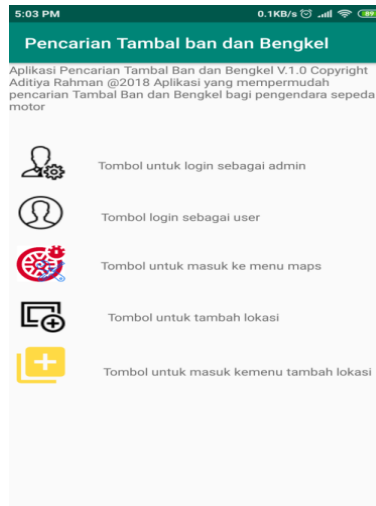
4. Halaman Menu Utama User



Gambar 8. Halaman Menu Utama User

Pada gambar di atas adalah menu utama *user* yang menyerupai menu utama admin, namun pada menu utama *user* tidak terdapat menu tambah lokasi.

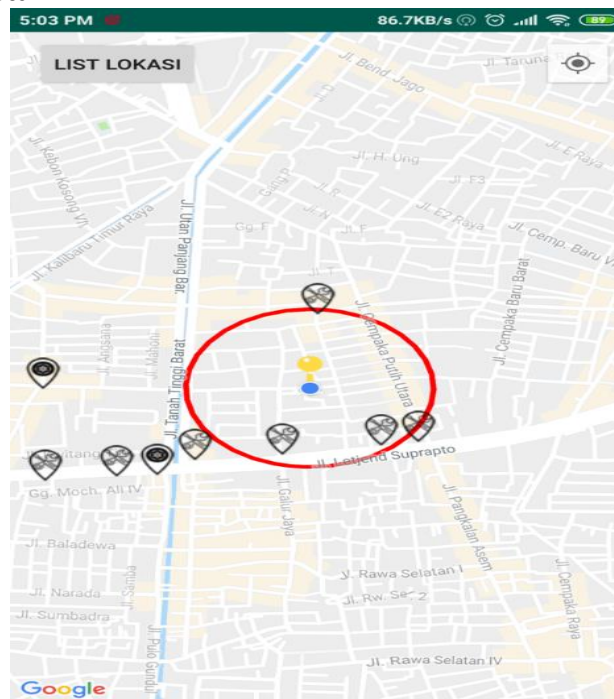
5. Halaman Menu Informasi



Gambar 9. Halaman Menu Informasi

Pada gambar 9 adalah tampilan halaman menu informasi, yang menampilkan informasi seputar aplikasi.

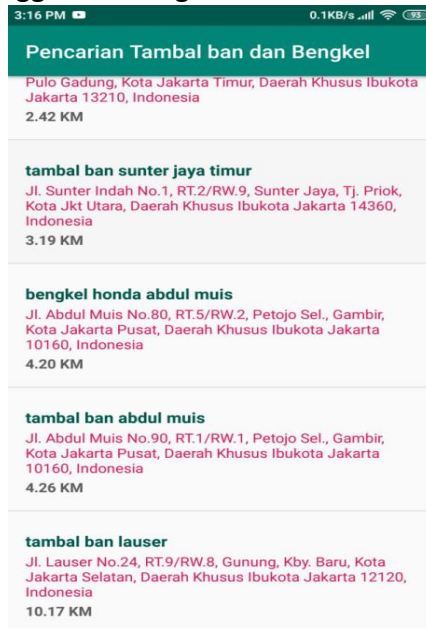
6. Halaman Menu Peta



Gambar 10. Halaman Menu Peta

Pada gambar di atas merupakan menu utama peta yang menunjukkan titik dimana kita berada dan pengguna dapat langsung melihat area sekitar, apakah terdapat titik tambal ban atau bengkel yang terdekat dari titik pengguna aplikasi, atau pengguna dapat memilih menu list lokasi untuk melihat list lokasi tambal ban atau bengkel terdekat.

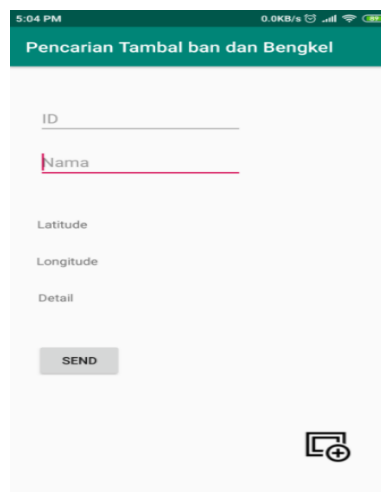
7. Halaman List Lokasi Menggunakan Algoritma Bubble Sort



Gambar 11. List Lokasi Terurut *Ascending* ke *Descending*

Pada gambar di atas merupakan tampilan menu list lokasi yang menampilkan list lokasi tambalban atau bengkel terdekat yang sudah terurut dengan satuan KM, menu ini dapat di akses dengan menekan tombol (*Button*) "List Lokasi" pada menu utama peta. Halaman ini hasil dari Algoritma Bubble Sort. Terlihat pada list diurutkan secara kecil ke besar atau ascending.

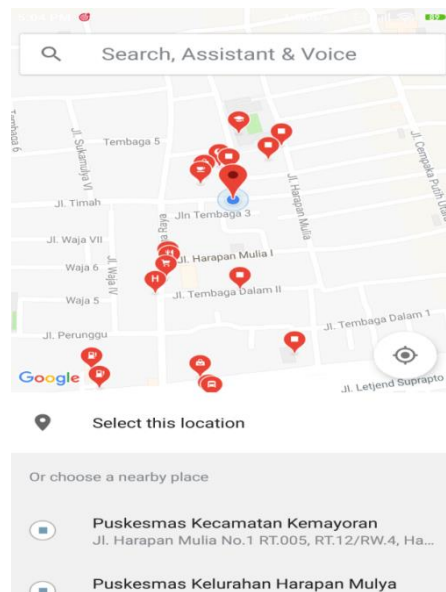
8. Halaman Tambah Lokasi



Gambar 12. Halaman Tambah Lokasi

Pada gambar 12 merupakan menu tambah lokasi yang dapat di akses dengan *login* sebagai *admin*, untuk id di *input* secara manual dengan inisial "bk" untuk bengkel dan "tb" untuk tambal ban, lalu dapat mengisi nama lokasi tambal ban atau bengkel yang sesuai, lalu mengambil titik untuk mendapatkan koordinat *latitude* dan *longitude* serta alamat, lalu tekan tombol *send* untuk mengirimnya ke *database*.

9. Halaman Peta Tambah Lokasi



Gambar 13. Halaman Peta Tambah Lokasi

Pada gambar di atas merupakan halaman peta tambah lokasi yang dapat di akses dengan menekan tombol (*Button*) pada menu tambah lokasi, setelah menentukan titik koordinat nya, lalu tekan “*select this location*” untuk mendapatkan, alamat dan titik koordinat *latitude* dan *longtitude* yang akan otomatis terinput ke form menu tambah lokasi.

4. KESIMPULAN

Dari hasil pembuatan aplikasi serta pembahasan pada point - point sebelumnya, penulis mencoba menyimpulkan dari seluruh pokok bahasan mengenai aplikasi pencarian tambal ban dan bengkel berbasis gps dan android. Adapun kesimpulannya sebagai berikut:

1. Dengan memanfaatkan Algoritma ini yang diimplementasikan pada Google Map berbasis (Geographic Information System) dapat menjadi salah satu metode pencarian lokasi terdekat.
2. Dalam aplikasi ini penulis menggunakan Google API khususnya Maps API untuk android, yang mendukung dari teknologi mobile saat ini yaitu GIS (Geographic Information System) yang juga terintegritas teknologi LBS (Location Based Service) yang dapat menampilkan direktori kota, titik koordinat, rute, navigasi dan sebagai nya pada aplikasi pencarian tambal ban dan bengkel berbasis gps dan android ini.

REFERENSI

- [1] M. Rizal and F. Latifah, “Perancangan Aplikasi Lokasi Wisata Kota Jakarta Menggunakan Algoritma Sequential Search Berbasis,” *J. PILAR Nusa Mandiri*, vol. 13, no. 2, pp. 227–232, 2017.
- [2] F. Sandy and I. Chaidir, “Pemanfaatan GPS (Global Positioning System) Menentukan Posisi Autocare Terdekat Menggunakan Metode Sequential,” vol. 8, no. 2, pp. 131–137, 2017.

- [3] S. Roma Rio, M. Yusman, and F. Febi Eka, "Algoritma Bubble Sort Dan Selection Sort," *J. Komputasi*, vol. 5, no. 1, pp. 81–87, 2017.
- [4] N. Rachmat, "Perbandingan Bubble Sort, Shell Sort Dan Kombinasi Bubble Sort Dengan Shell Sort," *J. Sist. Komput. Musirawas*, vol. 3, no. 1, p. 59, 2018, doi: 10.32767/jusikom.v3i1.308.
- [5] E. Sunandar and I. Indrianto, "Implementasi Algoritma Bubble Sort Terhadap 2 Buah Model Varian Pengurutan Data Menggunakan Bahasa Program Java," *Petir*, vol. 13, no. 2, pp. 255–265, 2020, doi: 10.33322/petir.v13i2.1008.
- [6] A. Kadir, *Konsep dan Implementasi Struktur Data dalam Pemrograman Delphi*, 1st ed. Yogyakarta: Andi, 2011.
- [7] U. Penentuan and N. Prestasi, "Gunawan, Penggunaan Algoritma Sorting Bubble Sort Untuk Penentuan Nilai Prestasi Siswa 296," vol. 8, pp. 296–304, 2019.
- [8] M. T. Prihandoyo, "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web," *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 126–129, Jan. 2018, doi: 10.30591/JPIT.V3I1.765.
- [9] Martini, E. Mufida, and J. Meitry, "Aplikasi Parenting Untuk Makanan Sehat Bayi Berbasis Android," *J. Ilmu Pengetah. Dan Teknol. Komput.*, vol. 3, no. 2, pp. 239–244, 2018.
- [10] P. A. Rahayuningsih, "Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting)," *EVOLUSI - J. Sains dan Manaj.*, vol. 4, no. 2, Feb. 2016.